

**T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ
ANABİLİM DALI**

**KAMU KURUM VE KURULUŞLARI İNTERNET SİTELERİNİN
2007/4 SAYILI GENELGE KAPSAMINDA DEĞERLENDİRİLMESİNE
YÖNELİK BİR ARAÇ GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Erkan DURAN

Balıkesir, Ağustos – 2008

**T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ
ANABİLİM DALI**

**KAMU KURUM VE KURULUŞLARI İNTERNET SİTELERİNİN
2007/4 SAYILI GENELGE KAPSAMINDA DEĞERLENDİRİLMESİNE
YÖNELİK BİR ARAÇ GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Erkan DURAN

Balıkesir, Ağustos – 2008

T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ
ANABİLİM DALI

KAMU KURUM VE KURULUŞLARI İNTERNET SİTELERİNİN
2007/4 SAYILI GENELGE KAPSAMINDA DEĞERLENDİRİLMESİNE
YÖNELİK BİR ARAÇ GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Erkan DURAN

Tez Danışmanı: Yrd. Doç. Dr. Taner TANRISEVER

Sınav Tarihi : 20.08.2008

Jüri Üyeleri : Yrd. Doç. Dr. Taner TANRISEVER (Danışman, BaÜ) 

Doç. Dr. Mustafa GÖKTEPE (BaÜ) 

Yrd. Doç. Dr. Ayşen KARAMETE (BaÜ) 

Balıkesir, Ağustos – 2008

ÖZET

KAMU KURUM VE KURULUŞLARI İNTERNET SİTELERİNİN 2007/4 SAYILI GENELGE KAPSAMINDA DEĞERLENDİRİLMESİNE YÖNELİK BİR ARAÇ GELİŞTİRİLMESİ

Erkan DURAN

Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü,
Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

(Yüksek Lisans Tezi / Tez Danışmanı: Yrd. Doç. Dr. Taner TANRISEVER)

Balıkesir, 2008

Bu çalışmanın amacı, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu - Marmara Araştırma Merkezi tarafından hazırlanan ve 2007/4 sayılı Başbakanlık genelgesi ile yayımlanan “Kamu Kurumları İnternet Sitesi Kılavuzu” kapsamında kamu kurum ve kuruluşları internet sitelerini değerlendirmeye yönelik bir araç geliştirilmesidir.

Çalışmada öncelikle söz konusu kılavuz kapsamında belirlenen kriterlere göre değerlendirme işlemlerini yapabilen bir PHP class’ı (sınıf’ı) yazılmış, sonra da bu class’ı kullanarak çalışan bir araç geliştirilmiştir.

Class’ın ve aracın geliştirilmesinin ardından, 35 kamu kurum ve kuruluşu internet sitesi ana sayfası, “<http://webtest.balikesir.edu.tr>” adresinde yer alan geliştirilmiş bu araç ile “İçerik (Asgari)”, “Tasarım (Erişilebilirlik)”, “Tasarım (Teknik Özellikler)” ve “Genel Görünüm” olmak üzere dört ana başlıktaki kriterlere göre değerlendirilmiştir. Değerlendirmeler sonucunda, bu web sayfalarının; asgari içerik ile ilgili kriterleri büyük ölçüde sağladıkları ancak diğer üç ana başlıkta yer alan kriterlere göre hem standartlar yönünden, hem de erişilebilirlik yönünden eksiklerinin olduğu ortaya çıkmıştır.

Anahtar Sözcükler: Web Sayfası Değerlendirme Aracı, İnternet Siteleri Kılavuzu, W3C, Web Standartları, Erişilebilirlik, e-Devlet

ABSTRACT

THE DEVELOPMENT OF A TOOL AIMED AT THE EVALUATION OF THE PUBLIC INSTITUTIONS WEBSITES UNDER THE CIRCULAR 2007/4

Erkan DURAN

**Balıkesir University, Institute of Science
Department of Computer Education and Instructional Technology**

(M. Sc. Thesis / Supervisor: Assist. Prof. Dr. Taner TANRISEVER)

Balıkesir, 2008

The aim of this study is developing a tool aimed at evaluation of public institutions websites according to “Public Institutions Websites Guide” which is prepared by The Scientific and Technological Research Council of Turkey – Marmara Research Center and published with circular 2007/4 of The Office of The Prime Minister.

Firstly in this study, a PHP class which is able to evaluate according to criterions which determined under the guide was written. Afterwards, a tool works by using this PHP class was developed.

After developing PHP class and the tool, 35 public institutions website main pages were evaluated with this developed tool which is available at “<http://webtest.balikesir.edu.tr>” according to criterions in four main categories as being “Content (Minimum)” , “Design (Accessibility)” , “Design (Technical Properties)” and “General View ”. At the result of evaluations, it was come out that these web pages provided almost all of the criterions related with minimum content. But, it was also come out that these web pages have deficiencies for both standards and accessibility according to criterions in other three main categories.

Keywords: Webpage Evaluation Tool, Websites Guide, W3C, Web Standards, Accessibility, e-Government

İÇİNDEKİLER

ÖZET	ii
ABSTRACT	iii
İÇİNDEKİLER	iv
SİMGELER ve KISALTMALAR	vi
ŞEKİL LİSTESİ	vii
TABLO LİSTESİ	viii
ÖNSÖZ	ix
1. GİRİŞ	1
1.1 2007/4 Sayılı Başbakanlık Genelgesi Hakkında	4
1.2 Erişilebilirlik (Accessibility)	5
1.2.1 Web Standartları	9
1.2.2 W3C/WAI WCAG	10
1.3 e-Devlet (Elektronik Devlet)	11
1.4 Araştırmanın Amacı ve Önemi	13
1.5 Araştırma Problemi	14
2. LİTERATÜR TARAMASI	15
2.1 Kamu Kurum ve Kuruluşları Siteleri ile İlgili Çalışmalar	15
2.1.1 Türkiye’deki Çalışmalar	15
2.1.2 Dünyadaki Çalışmalar	21
2.2 Web Standartları ve Erişilebilirlik ile İlgili Araçlar	24
3. YÖNTEM	27
3.1 Değerlendirme İçin Belirlenen Kriterler	30
3.2 Web Sayfası Değerlendirme Class’ı (Sınıf’ı)	33
3.2.1 Class’ın Sınırlılıkları	34
3.2.2 Class’taki Fonksiyonlar ve Görevleri	34
3.2.2.1 Constructor (Kurucu) Fonksiyon	35
3.2.2.2 Private (Özel) Fonksiyonlar	37
3.2.2.3 Public (Genel) Fonksiyonlar	45
3.2.2.4 Dönen Değerler ve Sonuç Fonksiyonları	52
3.2.3 Class’ın Kullanımı	57
3.3 Web Sayfası Değerlendirme Aracı	58
4. BULGULAR VE YORUMLAR	60

4.1 İçerik (Asgari).....	60
4.2 Tasarım (Erişilebilirlik).....	61
4.3 Tasarım (Teknik Özellikler).....	63
4.4 Genel Görünüm.....	68
5. SONUÇ VE ÖNERİLER.....	75
5.1 Class ve Araç ile İlgili Sonuç ve Öneriler.....	75
5.2 Değerlendirilen Web Sayfaları ile İlgili Sonuç ve Öneriler.....	79
EKLER.....	81
EK A KAMU KURUMLARI İNTERNET SİTESİ KILAVUZU.....	81
EK B WEB SAYFASI DEĞERLENDİRME CLASS'I.....	112
EK B.1 Kaynak Kodu.....	112
EK B.2 Sonuç Fonksiyonları.....	143
EK B.3 İçerik Testleri Dil Dosyaları.....	144
EK B.3.1 Türkçe Dil Dosyası.....	144
EK B.3.2 İngilizce Dil Dosyası.....	144
EK C WEB SAYFASI DEĞERLENDİRME ARACI.....	146
EK C.1 Kaynak Kodu.....	146
EK C.2 Veritabanı Tablolarını Oluşturan SQL Kodları.....	167
EK C.2.1 Info Tablosu.....	167
EK C.2.2 Webpage Tablosu.....	168
EK C.3 Kullanılan Veritabanı Class'ı.....	170
EK C.4 Arayüz Dil Dosyaları.....	176
EK C.4.1 Türkçe Dil Dosyası.....	176
EK C.4.2 İngilizce Dil Dosyası.....	179
EK C.5 Kriterlerle İlgili Dil Dosyaları.....	183
EK C.5.1 Türkçe Dil Dosyası.....	183
EK C.5.2 İngilizce Dil Dosyası.....	185
EK C.6 Stil Sayfası Kodu.....	187
EK C.7 Ekran Görüntüleri ve Özellikleri.....	193
EK C.7.1 Giriş Ekranı.....	193
EK C.7.2 Değerlendirme Sonuçları Ekranı.....	194
EK C.7.3 Yönetici Modu Ekranı.....	196
EK D ARACIN DİZİN VE DOSYA YAPISI.....	198
EK E ARACIN AKIŞ DİYAGRAMI.....	199
EK F DEĞERLENDİRİLEN WEB SAYFALARI.....	200
KAYNAKLAR.....	201

SİMGELER ve KISALTMALAR

eDTr	e-Dönüşüm Türkiye
WAI	Web Accessibility Initiative
TÜBİTAK	Türkiye Bilimsel ve Teknik Araştırma Kurumu
TÜBİTAK-MAM	TÜBİTAK - Marmara Araştırma Merkezi
W3C	The World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines
G2C	Government to Citizen
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
PHP	Personal Home Page
NYP	Nesneye Yönelik Programlama

ŞEKİL LİSTESİ

<u>Sekil No</u>	<u>Adı</u>	<u>Sayfa</u>
Şekil 1.1	Bir Web Sayfasına Erişim	5
Şekil 1.2	Bir Web Sayfasının Üç Farklı Tarayıcıda Görüntüsü. (A) Internet Explorer 6.0 (B) Mozilla Firefox 2.0 (C) Lynx 2.8	7
Şekil 2.1	AIS Web Erişilebilirliği Araç Çubuğu	25
Şekil 2.2	Web Developer Araç Çubuğu	26
Şekil C.1	Web Sayfası Değerlendirme Aracı Giriş Ekranı	194
Şekil C.2	Değerlendirme Sonuçları Örnek Ekran Görüntüsü	195
Şekil C.3	Değerlendirme Sonuçları Simgeleri	196
Şekil C.4	Yönetici Modu Ekran Görüntüsü	196
Şekil D.1	Aracın Dizin ve Dosya Yapısı	198
Şekil E.1	Aracın Akış Diyagramı	199

TABLO LİSTESİ

<u>Tablo No</u>	<u>Adı</u>	<u>Sayfa</u>
Tablo 1.1	Kamu Kurum ve Kuruluşları Alan Adları İle İlgili İstatistikler.....	3
Tablo 4.1	İçerik / Genel Değerlendirme Sonuçları	60
Tablo 4.2	Kuruma Erişim Bilgileri Değerlendirme Sonuçları	61
Tablo 4.3	Tarayıcı Tipi Değerlendirme Sonuçları	62
Tablo 4.4	Ekran Çözünürlüğü Değerlendirme Sonuçları.....	62
Tablo 4.5	Tasarım/Genel Değerlendirme Sonuçları.....	63
Tablo 4.6	Sayfa Boyutu Değerlendirme Sonuçları.....	65
Tablo 4.7	Kodlama Değerlendirme Sonuçları.....	65
Tablo 4.8	Genel Görünüm/Genel Değerlendirme Sonuçları.....	68
Tablo 4.9	Logolar Değerlendirme Sonuçları.....	68
Tablo 4.10	Sayfada Gezinme Değerlendirme Sonuçları	69
Tablo 4.11	Metinler Değerlendirme Sonuçları.....	71
Tablo 4.12	Resimler Değerlendirme Sonuçları	72
Tablo 4.13	216 Renk Paletinin Kullanımı Değerlendirme Sonuçları.....	73
Tablo 4.14	Tablolar Değerlendirme Sonuçları.....	73
Tablo F.1	Değerlendirilen Web Sayfaları.....	200

ÖNSÖZ

Öncelikle böyle bir çalışmayı bana öneren ve çalışmalarımın her aşamasında zaman ve mekân gözetmeksizin değerli vaktini bana ayıran, karşılaştığım tüm sorunlarda büyük bir özveriyle bana yardımcı olan çok değerli hocam ve aynı zamanda tez danışmanım Sayın Yrd. Doç. Dr. Taner TANRISEVER'e sonsuz teşekkürlerimi bildirmek isterim.

Yüksek lisans çalışmalarımın başladığı ilk günden beri beni destekleyen ve yardımlarını esirgemeyen, hiç çekinmeden ve her zaman kendilerine danışabildiğim değerli hocalarım Sayın Yrd. Doç. Dr. Ayşen KARAMETE'ye, Sayın Yrd. Doç. Dr. Sabri KOCAKÜLAH'a, Sayın Yrd. Doç. Dr. Suat IŞILDAK'a ve Sayın Yrd. Doç. Dr. Ömer GEMİCİ'ye teşekkür etmeyi bir borç bilirim.

Özellikle tez aşamasında verdikleri destekten dolayı Kepsut İlçe Milli Eğitim Müdürü Sayın Ramazan AYDOĞDU ve Kepsut Hacer Özer İ.Ö.O. Müdür Yardımcısı Sayın Recep DEDE'ye çok teşekkür ederim.

Öte yandan çalışmalarım sırasında beni hiç yalnız bırakmayan ve gece gündüz demeden hep yardımcı olmaya çalışan sevgili dostlarım Eray YILMAZ'a ve Tuncay ÇELİK'e ne kadar teşekkür etsem azdır.

Yüksek lisans eğitimimin başladığı ilk günden bu çalışma tamamlanıncaya kadar geçen süre içerisinde kendileriyle yeterince ilgilenemediğim, hatta zaman zaman istemeden kırdığım başta ailem ve İlgül TEPEDELEN olmak üzere tüm arkadaşlarıma anlayışlarından dolayı teşekkür ederim.

Balıkesir, 2008

Erkan DURAN

1. GİRİŞ

Geçmişten günümüze insanlar yaşamlarını kolaylaştırmak için çok çeşitli araçlar geliştirmişlerdir. Hiç kuşkusuz bu araçların en hızlı gelişenlerinden biri bilgisayardır. Özellikle son yirmi yılda bilgi ve iletişim teknolojilerindeki gelişmeler sonucunda bilgisayar ve internet gündelik hayatın bir parçası olmuştur.

Çağımızda haberleşme ve bilginin hızlı bir şekilde herkese ulaşması büyük önem taşımaktadır. İnternet ortamında herkesin bilgiye ulaşabilmesi, bilginin büyük bir hızla artarak gelişmesinde en önemli faktördür. Bugün dünyanın herhangi bir noktasında üretilen bilgiye anında ulaşım internet ile mümkün olabilmektedir [1].

İnsanlar arasındaki hem yeni hem de hızlı ve sınır tanımayan bir haberleşme sistemi olarak internet, son yılların en önemli sosyolojik olaylarından biri haline gelmiştir. e-Ticaret, e-Devlet, e-Kütüphane gibi onlarca uygulamasıyla internet; ekonomiden spora, askeriye akademik hayata toplum kültürünü bir bütün olarak değiştirmektedir [2].

Bilgi ve iletişim teknolojilerindeki gelişmelerin toplumsal yaşama yansımaları; eğitim, sağlık, tarım ve sanayi başta olmak üzere bütün toplumsal alanları, örgütlenme ve yaşam tarzını, kamu yönetimi anlayışını da önemli ölçüde değiştirmiştir [3]. Nitekim çağdaş bireyler, kendilerine hizmet veren kurumların daha hızlı, daha açık ve daha az maliyetle çalışmalarını istemektedirler. Bu nedenle, toplumun ihtiyaçlarına cevap vermek ve üstlendiği görevleri etkili bir şekilde yerine getirmek esasına yönelik olan kamu yönetiminde; karar alma sürecinin basitleştirilmesi, hizmetteki kalitenin artırılması ve zamanın daha verimli bir şekilde kullanılması modern toplumsal yaşamın bir gereği olmuştur [4].

Bu bağlamda bilgi ve iletişim teknolojilerinin sunduğu olanaklar, kamu kurumlarının hizmet sunumunda da kullanılmaya başlanmış, dünyadaki gelişmelerle

birlikte ülkemizde de birçok kamu kurumunda bilgi işlem merkezleri ve internet siteleri oluşturulmuş, bilgisayar ortamında bilgi saklanmaya başlanmıştır. Bu gelişmelere paralel olarak 2003 yılında ise birçok kamu hizmetinin elektronik ortamda sunumunu öngören e-Dönüşüm Türkiye (eDTr) Projesi başlatılmıştır [3]. Bu proje ile;

- Bilgi ve iletişim teknolojileri politikaları ve mevzuatının, öncelikle Avrupa Birliği müktesebatı çerçevesinde gözden geçirilerek yeniden düzenlenmesi, bu konuda e-Avrupa ve kapsamında aday ülkeler için öngörülen eylem planının ülkemize uyarlanması
- Vatandaşın, bilgi ve iletişim teknolojileri yardımıyla, kamusal alandaki karar alma süreçlerine katılımını sağlayacak mekanizmaların geliştirilmesi
- Kamu idaresinin, şeffaf ve hesap verebilir hale getirilmesine katkıda bulunulması
- Kamu hizmetlerinin sunumunda, bilgi ve iletişim teknolojilerinden azami ölçüde yararlanılarak iyi yönetim ilkelerinin hayata geçirilmesine katkıda bulunulması
- Bilgi ve iletişim teknolojilerinin kullanımının yaygınlaştırılması
- Bilgi ve iletişim teknolojisi alanında kaynak israfını azaltmak amacıyla, kamunun mükerrerlik arz eden veya örtüşen ilgili yatırım projelerinin bütünleştirilmesi, izlenmesi, değerlendirilmesi ve yatırımcı kamu kuruluşları arasında gerekli koordinasyonun sağlanması
- Sektördeki, özel sektör faaliyetlerine yukarıdaki ilkeler ışığında yol gösterilmesi

amaçlanmıştır [5].

Kamu internet sitelerinin devlet ile vatandaş arasındaki dolaysız iletişim ve etkileşimini sağlayan pencereler olarak önemi arttıran eDTr Projesi [3] ve Bilgi Edinme Kanunu'nun da etkisiyle kamu hizmeti sunan internet sitelerinin sayılarında önemli artışlar olmuştur.

Kamu kurum ve kuruluşlarına tahsil edilmiş bazı alan adları ile ilgili son altı yıla ait istatistikler şu şekildedir [6]:

Tablo 1.1 Kamu Kurum ve Kuruluşları Alan Adları İle İlgili İstatistikler

	gov.tr	k12.tr	bel.tr	edu.tr	pol.tr	mil.tr	org.tr
2003	1647	800	138	170	22	8	2310
2004	2315	1449	741	180	105	8	2596
2005	3310	3352	1121	189	138	8	2992
2006	5595	9381	2222	298	228	8	4682
2007	6992	9619	1976	338	275	8	5345
2008*	7413	9426	2039	351	305	8	5732

* 2008 Mayıs ayı

Tablo 1.1’den de sitelerin sayılarında önemli artışlar olduğu görülmektedir. İnternetin toplum tarafından da yaygın olarak kullanılması; e-Devletin dış yüzü olan kamu kurumları internet sitelerinin önemini artırmıştır. e-Devletin; vatandaşlara şeffaf ve etkin bir hizmet sunmak, vatandaşlara bilgi sağlamak, vatandaş-devlet etkileşimini arttırmak vb amaçlarının gerçekleştirilmesinde kamu internet sitelerinin önemi tartışılmazdır. Bununla birlikte, sitelerin kurum yöneticilerinin inisiyatifinde, kurumsal düzeyde ve standartlara uygunluğu gözetilmeden oluşturulması zafiyet yaratmıştır [3]. e-Devlet olarak adlandırılan süreçte ortaya çıkan kamu kurumları internet sitelerinin uluslararası standartlara uygunluk yönünden ciddi eksiklikleri bulunmakta ve e-Devlet sürecinde, kamuda ortak stratejilerin gelişmediği belirtilmektedir [7]. Oysa standartlara uyum; erişilebilirlik, kolay kullanım, kolay dolaşım, sitenin hızlı yüklenmesi, maliyet ve zaman tasarrufu, sitelerin bakım ve güncelleme kolaylığı gibi birçok bakımdan önem arz etmektedir [3].

eDTr Projesi eylem planlarında, konunun önemi dikkate alınarak, kamu internet sitelerinin uluslararası standartlara uygunluğunu sağlamak üzere, WAI (WEB Accessibility Initiative) de dikkate alınarak, “*kamu kurumları internet sitesi kılavuzunun hazırlanması*” eylemine yer verilmiştir [3, 8].

“Kamu Kurumları İnternet Sitesi Kılavuzu Hazırlanması”, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu - Marmara Araştırma Merkezi (TÜBİTAK-MAM) sorumluluğunda yürütülen çalışmalar neticesinde tamamlanmış ve 2007/4 sayılı genelge ile yayımlanmıştır [9].

1.1 2007/4 Sayılı Başbakanlık Genelgesi Hakkında

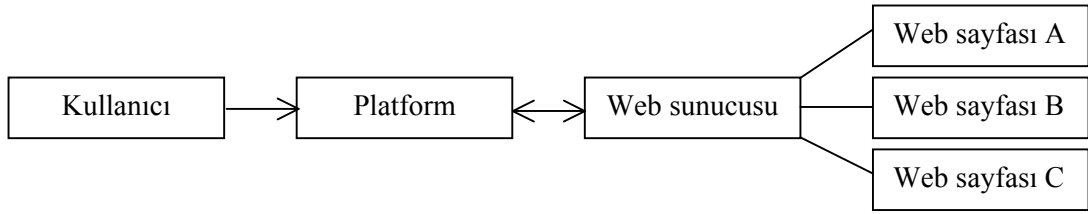
İlgili genelgede, eDTr Projesi çalışmaları kapsamında; 2005 Eylem Planı’nda, kamu kurum ve kuruluşlarının internet sitelerinde asgari düzeyde içerik ve tasarım uyumunun sağlanmasına yönelik olarak “Kamu Kurumları İnternet Sitesi Kılavuzunun Hazırlanması” eylemine yer verildiği, TÜBİTAK-MAM sorumluluğunda yürütülen çalışmalar neticesinde “Kamu Kurumları İnternet Sitesi Kılavuzu”nun (EK A) hazırlandığı belirtilmiştir. Ayrıca merkezi ve yerel düzeydeki tüm kamu kurum ve kuruluşlarınca yeni açılacak internet sitelerinde, söz konusu kılavuzda yer verilen usul ve esaslara uyulmasının zorunlu olduğu, kullanımda olan kamu internet sitelerinin de; kılavuzda belirtilen standartlara uyumlu olmayan unsurları, bütçe imkânları ve öncelikler çerçevesinde en kısa zamanda bu esaslara uyumlu hale getirmesi gerektiği vurgulanmıştır [9].

Kılavuzda, kılavuzun temel amacının, kamu kurumları internet sitelerinin sağlaması gereken asgari içeriği tanımlamak ve bu sitelerin en geniş kitlelerce erişilebilir olması amacıyla site tasarımına dönük öneriler içermek olduğu belirtilmiştir [10].

Bu bağlamda, kılavuz kapsamındaki erişilebilirlik ile e-Devlet kavramları ve bu kavramların önemi aşağıda ayrıntılı olarak açıklanmaya çalışılmıştır.

1.2 Erişilebilirlik (Accessibility)

Literatürde erişilebilirlikle ilgili olarak yapılmış birçok tanım bulunmaktadır. Ancak, “engelli kullanıcıların web sayfalarına erişebilmesi” gibi erişilebilirliğin eksik yapılmış tanımlarına rastlamak oldukça mümkündür. Oysa erişilebilirlik, aygıt bağımsızlığı da dahil olmak üzere, bireylerin (kullanıcıların) engelli olup olmadığına, kullanılan platforma (aygıt, tarayıcı, işletim sistemi vb) bakılmaksızın web sayfalarına ve bu sayfaların öğelerine en kolay ve düzgün bir şekilde erişebilmek gibi geniş bir kavramı ifade etmektedir [3, 11-14]. Tanımdan yola çıkarak erişilebilirliği “kullanıcı” ve “platform” olarak iki boyuta ayırmak mümkündür. Basitçe bir web sayfasına erişim ise şu şekilde gösterilebilir:



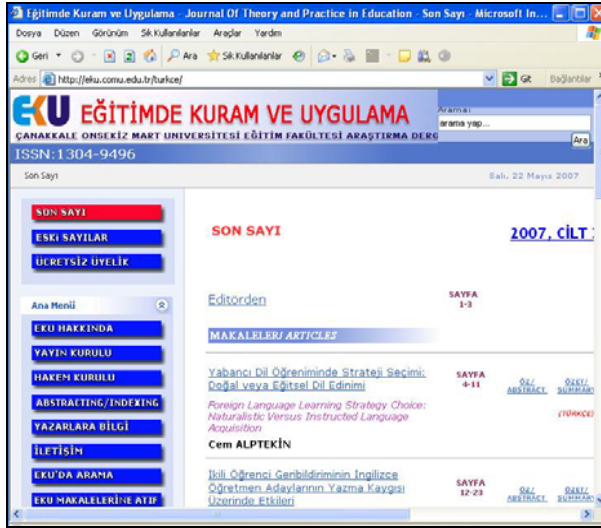
Şekil 1.1 Bir Web Sayfasına Erişim

Şekil 1.1’den görülebileceği gibi kullanıcılar web sayfalarına kullandıkları platform aracılığı ile erişmektedirler. Web tarayıcıları ise internet üzerindeki tüm bilgilere bakabilmek ve bu bilgilerle etkileşim halinde olabilme olanağı veren; kullanılan bilgisayar, cep telefonu vb aygıtlardaki yüklü işletim sistemlerinde çalışan uygulama programlarıdır ve birçok firma, çok değişik işletim sistemi üzerinde çalışabilen web tarayıcıları geliştirmektedir [15].

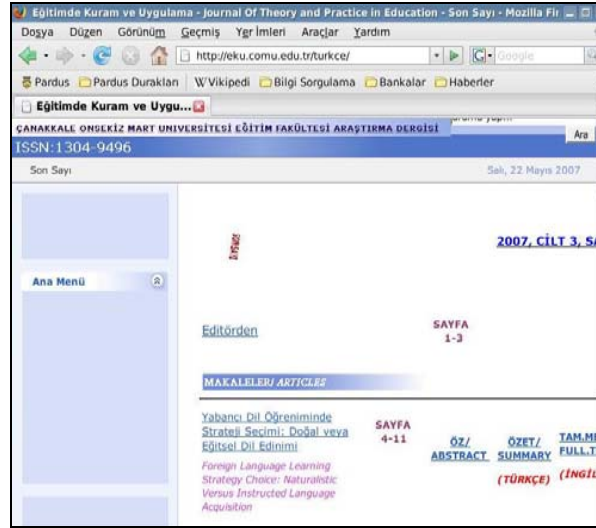
Bu noktada web tarayıcıları oldukça önemlidir. Çünkü kullanıcılar çeşitli tarayıcılarla ve işletim sistemleriyle web sayfalarını ziyaret etmektedirler [16-19]. Ancak, gerek web tarayıcılarının The World Wide Web Consortium (W3C) tarafından standartları belirlenmiş kuralları (web standartları) kendilerine göre yorumlaması ve bu tarayıcıların ön tanımlı değerlerinin farklı olması, gerekse web tasarımcılarının W3C’nin standartlarını uygulamaması ve hazırlanan sayfaları farklı

web tarayıcılarında, çeşitli işletim sistemleri ile değişik çözünürlüklerde sınınamaması, web sayfalarının farklı platformlarda erişilebilirlik ile ilgili çeşitli problemler çıkarmasına neden olabilmektedir [19-23].

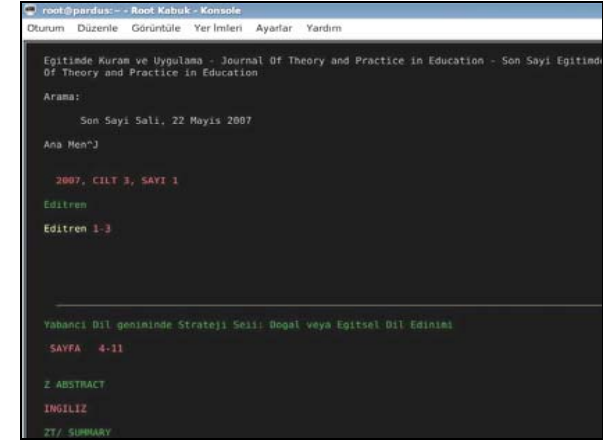
Örnek teşkil etmesi açısından bir web sayfasının üç farklı web tarayıcısındaki ekran görüntüsü Şekil 1.2’de verilmiştir. Şekil 1.2 incelendiğinde, ilgili web sayfasında bulunan ana menünün; üç farklı web tarayıcısında aynı içeriği verebilecek şekilde görüntülenemediği ve erişilebilirliğin sadece engelli kişiler tarafından bir web sayfasına erişim şeklinde olmadığı görülmektedir.



(A)



(B)



(C)

Şekil 1.2 Bir Web Sayfasının Üç Farklı Tarayıcıda Görüntüsü. (A) Internet Explorer 6.0 (B) Mozilla Firefox 2.0 (C) Lynx 2.8

Öte yandan erişilebilirliğin “kullanıcı” olarak adlandırılan boyutuna bakıldığında, web sayfaları; kullandığı aygıt, işletim sistemi, web tarayıcısı tercihi farklı olan kullanıcılarla birlikte [16-18] çok çeşitli kullanıcı grupları tarafından ziyaret edilmektedir [23]:

- JAWS, ekran okuyucu vb özel program kullanan görme engelli kullanıcılar
- Monitörden okudukları bilgiyi anlayamayan ama kendilerine bu bilgi okuduğunda anlayabilen öğrenme engelli (learning disability) kullanıcılar
- Arka plan ile yazı arasında yüksek kontrasyonu olan renkleri tercih eden, sayfanın yazı tipini istediği boyuta getirmek isteyen, “büyüteç” adı verilen yazılımları kullanan görme bozukluğuna sahip kullanıcılar
- Bilgisayarın faresini kullanamayan, klavyeye bağımlı olan bedensel engelli kullanıcılar
- Kırmızı ile yeşil renklerini birbirinden ayırt edemeyen renk körlüğü olan kullanıcılar
- Düşük modem hızına sahip olup web sayfasındaki resimleri kapatarak sayfaları görüntülemek isteyen kullanıcılar
- Arama motorları robotları/örümcekleri

Görüldüğü gibi web sayfaları çok çeşitli kullanıcılar tarafından ziyaret edilmektedir. Ancak dünyada interneti kullanırken olumsuz olarak etkilenen ve engelli olan milyonlarca insan bulunmaktadır. Mevcut birçok web sitesi içerdikleri erişilebilirlik engellerinden dolayı engelli kişilerin ulaşımını çok zor veya olanaksız kılmaktadır [24]. Birçok site tasarımcısı, sayfalarını kendi ekranı, çözünürlüğü, tarayıcısı, renk seçimi vb tercihleri doğrultusunda hazırlamakta ve engelli kişilerin siteyi kullanabilmeleri problemi çok ihmal edilen bir konu olmaktadır [25]. Bununla birlikte dünyada yaklaşık 650 milyon kişinin engelli olduğu tahmin edilmekte [26], ülkemiz nüfusunun % 12.29’unu engelliler oluşturmakta [27] ve ülkemizde internet kullananların da yaklaşık % 9’unu engelli kullanıcılar oluşturmaktadır [28].

Bilindiğinin aksine engelli kullanıcılar web sayfalarını JAWS, ekran okuyucu gibi cihazların yanı sıra görsel tarayıcılar ile de ziyaret etmekte ve özellikle bu cihazların (JAWS vb) fiyatları nedeniyle birçok engelli kullanıcı, diğer ucuz alternatiflere yönelmektedir. Ayrıca engelli kullanıcılar toplum ve çevre zorlukları nedeniyle, evden dışarı fazla çıkmamayı tercih etmekte ve günlük işlerini (alış-veriş, banka işlemleri vb) internetten yapmayı tercih etmektedir [29]. Halbuki web siteleri veya yazılımları erişilebilir yapılsa, engelli kişiler de web sayfalarını etkili bir şekilde kullanabilecektir [24].

Erişilebilirlik konusunda “Kamu Kurumları İnternet Sitesi Kılavuzu” kapsamında yer alan W3C’nin web standartları ve WAI erişilebilirlik rehberi WCAG (Web Content Accessibility Guidelines) son derece önemlidir.

1.2.1 Web Standartları

Günümüzde hemen her alanda bir takım standartların bulunduğu düşünüldüğünde, bu durumun milyonlarca kullanıcısı olan web’de de geçerli olması kaçınılmaz olmaktadır.

Web standartları, internet ile ilgili tüm alanlarda araştırmaları yöneten ve standartları geliştiren; üniversiteler, Microsoft, Netscape, Macromedia gibi firmalar ve internete ilişkin çeşitli alanlarda çalışan uzmanların bir araya gelmesiyle oluşan uluslararası konsorsiyum (W3C) tarafından geliştirilmiştir [29, 30].

Değişik web cihazlarının, web içinde sunulan bilgileri aynı şekilde kullanıcıya göstermesine yardım eden bir takım teknolojilerin toplamı olan web standartları sayesinde sunulan bilgi, herkes tarafından erişebilir olmakta ve kullanıcıların internete nasıl, ne ile bağlandığı değil; neye, hangi bilgiye ulaşmak istediği önemli hale gelmektedir [3, 30].

Web standartları, web tasarımcılarına *bir sayfanın nasıl kodlanacağını göstermesinin yanı sıra web tarayıcılarında da bir sayfanın nasıl gösterileceğine*

ilişkin kurallar barındırmakta; böylece web tasarımcıları, her türlü web tarayıcıları, internet kullanıcıları, web tasarım şirketleri, web sitesi isteyen müşteriler vb aynı dili konuşabilmektedir [3, 30].

Web sayfalarında sunum ve bilginin birbirinden ayrılması gerektiğini önemle vurgulayan ve teşvik eden web standartlarının [30] bazı faydaları şu şekildedir [31]:

- Erişilebilirlik
- Yeni teknolojilere uyum
- Sayfaların hızlı yüklenmesi/düşük bant genişliği
- Arama motorlarında yüksek sıralama
- Sitenin kolay bakımı
- Düşük maliyet, para ve zaman tasarrufu
- Hukuksal sorumluluğun azalması

“Kamu Kurumları İnternet Sitesi Kılavuzu” incelendiğinde, kılavuzun web standartları dikkate alınarak hazırlandığı görülmektedir. Web standartları ile ilgili ayrıntılı bilgi <http://www.w3.org> sitesinden elde edilebilir.

1.2.2 W3C/WAI WCAG

WAI, engelli kullanıcıların internet erişilebilirliğinin sağlanmasına dönük çalışmalar yapan; kılavuzlar, destek materyalleri hazırlayan uluslararası geniş katılım ile kurulmuş olan bir organizasyondur. WAI'nin hazırladığı rehberlerden biri olan WCAG, engelli kullanıcılar tarafından erişilebilir internet sayfaları hazırlanmasına yol göstermektedir. Bu rehber, internet sayfaları ya da uygulamalarında yer alan tüm içerik için; metinler, resimler, formlar, sesler ve diğer öğeler için erişilebilirliği artırıcı önerilerde bulunmaktadır [10].

“Kamu Kurumları İnternet Sitesi Kılavuzu” WAI/WCAG erişilebilirlik rehberi dikkate alınarak hazırlanmıştır [8, 10]. Rehber ile ilgili ayrıntılı bilgi

<http://www.w3.org/WAI> adresinden veya “Kamu Kurumları İnternet Sitesi Kılavuzu”ndan edinilebilir.

1.3 e-Devlet (Elektronik Devlet)

Genel olarak e-Devlet; kamu kurum ve kuruluşları, vatandaşlar ve ticari kurumlar arasındaki bilgi, hizmet ve mal alışverişlerinde bilgi ve iletişim teknolojilerini kullanarak performans ve verimlilik artışını hedefleyen bir devlet modeli olarak tanımlanmaktadır [32].

Temelde kamunun hizmet verdiği alanlarda bilgi ve iletişim teknolojilerinin kullanılması yoluyla şeffaf, vatandaşa yakın, daha ucuz ve verimli çalışan bir idari yapı olan e-Devlet’in, Devletten – Vatandaşlara (G2C: Government-to-Citizen), Devletten – Özel Sektöre (G2B: Government-to-Business), Devletten – Çalışanlarına (G2E: Government-to-Employee) ve Devletten – Devlete (G2G: Government-to-Government) olmak üzere dört yönünün olduğu söylenebilir [4]. Bunların arasında en hızlı gelişeni ve en çok bilineni ise Devletten-Vatandaşlara olan G2C türüdür. Yaşam sürecinin doğum, evlilik, ölüm, çalışma, emeklilik, seyahat, eğitim vb her noktasının bir şekilde devlet ile bağlantısı olduğu düşünüldüğünde bu kaçınılmaz hale gelmektedir. Bu bağlamda e-Devlet, devlete e-posta yoluyla talep veya şikâyet iletilmesinden internet üzerinden ödeme yapılmasına kadar, günlük hayatımızın birer parçası olan birçok işlemin elektronik ortamda çözümlenmesini kapsamaktadır [33].

e-Devlet ile amaçlananlar ve bu amaçların gerçekleşmesi durumunda sağlanacak yararlar şu şekildedir [34]:

Amaçlananlar:

- Devletin şeffaflaşması
- Devletin hızlı ve etkin bir şekilde işleyişinin sağlanması
- Her düzeyde vatandaşın yönetime katılımının sağlanması
- Kurumlar arası bilgi alışverişinin sağlanarak iş ve veri yinelenmesinin önlenmesi

- Vatandaşların yaşamlarının kolaylaştırılması
- Karar vericilerin bilgiye dayalı karar verme süreçlerinin geliştirilmesi ve hızlandırılması

Yararlar:

- Zamansal kazanç
- Maliyetlerin düşmesi ve verimliliğin artması
- Memnuniyetin artması
- Ekonomik gelişim
- Hayat kalitesinin artması
- Bireysel katılımın artması
- Kâğıt bağımlılığının ve kullanımının azalması
- Kamu ile olan işlemler için vatandaşın talep ettiği bilgilere bir noktadan ve doğru şekilde ulaşılarak insan yanlışlarının en aza indirgenmesi
- Verilecek olan hizmetin daha hızlı sunulması, işletme giderlerinin azalması, doğru bilgiye ulaşım
- Kamu ve vatandaş için karar almada kolaylık ve hızın artması
- Vatandaşların taleplerinin ön plana çıkması
- Vatandaş ile devlet arasındaki ilişkilerin gelişmesi, güven ortamının kuvvetlenmesi
- Kamuya güvenin artması

Konunun önemi açısından eDTr Projesi kapsamındaki 2005 Eylem Planı'nda da:

- Bilgi toplumu stratejisi: 1,
- Teknik altyapı ve bilgi güvenliği: 6,
- Eğitim ve insan: 10,
- Hukuki altyapı: 3,
- e-Devlet: 19,
- e-Sağlık: 5,

- e-Ticaret: 6

eylem olmak üzere 7 başlık altında toplam 50 eyleme yer verildiği, ancak aralarında “Kamu Kurumları İnternet Sitesi Kılavuzunun Hazırlanması” eyleminin de yer aldığı “e-Devlet” başlığında 19 eylemin olduğu görülmektedir [8].

e-Devlet hizmetlerinin vatandaşların kullanımına uygun olup olmadığı ve kurumların elektronik ortamda hizmet sunma beceri ve deneyimlerinin bir göstergesi olan kamu kurumları internet siteleri aracılığıyla, gerek vatandaşlar gerekse özel sektör kuruluşları, kamu hizmetlerinden çevrimiçi yararlanabilmektedir. Hizmetlerin kamu internet sitelerinden hızlı, etkileşimli, kolay kullanılabilir ve erişilebilir bir şekilde sunulması; sitelerin tasarım, dolaşım, içerik, erişilebilirlik açısından standartlara uygun kurulmasına bağlıdır [3].

Bu bağlamda e-Devletin dışa yansıyan yüzünü oluşturan kamu kurumları internet sitelerinin “Kamu Kurumları İnternet Sitesi Kılavuzu”nda yer alan önerileri dikkate almaları e-Devlet açısından son derece önemlidir.

1.4 Araştırmanın Amacı ve Önemi

Araştırmada 2007/4 sayılı genelge ile yayımlanan söz konusu kılavuz (EK A) kapsamında, kamu kurum ve kuruluşları internet sitelerinin değerlendirilmesine yönelik bir araç geliştirilmesi amaçlanmıştır.

Söz konusu kılavuz incelendiğinde; kılavuzun; erişilebilirlik, web standartları, e-Devlet gibi çok önemli kavramların dikkate alınarak hazırlandığı görülmektedir. Kamu kurum ve kuruluşları internet sitelerinin bu kılavuzda belirtilen usul ve esaslara uygunluklarının değerlendirilebilmesi, bu yönde eksikliklerin giderilebilmesi e-Devlet sürecinde; uluslararası standartların yakalanması, erişilebilirliği en yüksek düzeyde tutarak vatandaşların e-Devlet hizmetlerinden mümkün olduğunca yararlanması bakımından çok önemlidir. Bununla birlikte söz konusu kılavuzun web standartları ve erişilebilirlik göz önünde

bulundurularak hazırlanması nedeniyle geliştirilmesi hedeflenen araçtan sadece kamu kurum ve kuruluşları değil diğer internet siteleri de yararlanabilecektir.

Öte yandan Sayıştay'ın "E-Devlete Geçişte Kamu Kurumları İnternet Siteleri Performans Denetim Raporu"nda, TÜBİTAK-MAM tarafından hazırlanan bu kılavuzdaki standartlarla ilgili olarak;

"...kamu internet sitelerinin bu kılavuzdaki standartlara uygunluğunu kontrol edecek ve zamanında düzeltici önlemler alınmasını sağlayacak bir mekanizma da öngörülmemiştir. Bu nedenle ülkemizde kamu internet siteleri, uluslararası standartlara uygunluk gözetilmeden ve ulusal bir kamu internet yayın politikası olmaksızın, sadece kurumların inisiyatifinde yayın yapmaktadır. Dolayısıyla, yapılan yatırımların topluma fayda olarak geri dönüşümü risk altındadır. Söz konusu riskin azaltılması için anılan kılavuzun tüm kamu kurumlarınca uygulanması ve bu sitelerin kılavuza uygunluğunun, oluşturulacak bir kontrol mekanizması tarafından sürekli izlenmesi gerekmektedir. Böyle bir mekanizma, kamu internet sunum çalışmalarının yürütülmesi; uluslar arası standartlara uyumun ve siteler arasında içerik ve tasarım tutarlılığının sağlanması ile sitelere tüm toplum kesimlerinin erişebilirliğinin kolaylaştırılması adına son derece önemlidir. Bu nedenle; kurumların internet sitelerinin standartlara uygun hale getirilmesi ve teknolojik bağımlılığı önlemek ve teknolojilerin temininde tasarruf sağlamak için; alınması gereken tedbirlerin neler olduğunu belirleyecek, standartları güncelleyecek, kurumların ihtiyacı olan teknik desteği sağlayacak ve nihai olarak hizmet sunum kalitesinin artırılmasına katkı sağlayacak bir yapı oluşturulmalıdır. " [3, s. 109]

ifadeleri, geliştirilmesi hedeflenen araç ile bu noktada ortaya çıkan boşluğun bir nebze de olsun doldurulması açısından çalışmanın oldukça önemli olduğunu göstermektedir.

1.5 Araştırma Problemi

Araştırmada, eDTr Projesi/2005 Eylem Planı çerçevesinde; TÜBİTAK-MAM tarafından hazırlanan "Kamu Kurumları İnternet Sitesi Kılavuzu" doğrultusunda "Kamu kurum ve kuruluşları internet sitelerinin değerlendirilmesine yönelik bir araç geliştirilebilir mi?" problemine cevap aranmıştır.

2. LİTERATÜR TARAMASI

Yapılan literatür taramaları sonucunda TÜBİTAK-MAM tarafından hazırlanan söz konusu kılavuz kapsamında; kamu kurum ve kuruluşları internet sitelerini değerlendirmeye yönelik araç geliştirilmesiyle ilgili bir çalışmaya rastlanamamıştır. Ancak, günümüzde özellikle e-Devlet çalışmalarının ve internet kullanımının da yaygınlaşmasından dolayı e-Devlet ve web siteleriyle ilgili yapılmış çalışmalara sıklıkla rastlanmıştır.

İki kısma ayrılan bu bölümde; ilk önce e-Devlet ve internet siteleri ile ilgili çeşitli değerlendirmelerin de bulunduğu “Kamu Kurum ve Kuruluşları Siteleri ile İlgili Çalışmalar” incelenmiş, daha sonra “Web Standartları ve Erişilebilirlik ile İlgili Araçlar” a kısaca değinilmiştir.

2.1 Kamu Kurum ve Kuruluşları Siteleri ile İlgili Çalışmalar

2.1.1 Türkiye’deki Çalışmalar

Literatür taramalarında, kamu kurum ve kuruluşları internet sitelerinin değerlendirilmesine yönelik çalışmaların içerisinde konuya en yakın iki çalışmaya rastlanmıştır. Bunlardan en kapsamlı olanının, T.C. Sayıştay Başkanlığı’nın “e-Devlete Geçişte Kamu Kurumları İnternet Siteleri” başlıklı performans denetim raporunun [3] olduğu görülmüştür. Bu çalışmada kurum internet siteleri; içerik, tasarım, erişilebilirlik, güvenlik gibi kullanım özellikleri açısından değerlendirilmiştir. Ayrıca kurumların bilişim teknolojileri alt yapısını öğrenmek ve internet sitelerinin genel durumunu değerlendirmek için çalışmada kurumlarla anket çalışması yapılmıştır. Yapılan anket 36 kamu kurumundan gelen cevaplar doğrultusunda değerlendirilmiş, kurum internet siteleri ise iyi derecede bilgisayar

kullanabilen 8 kullanıcı tarafından; “tasarım”, “dolaşım”, “içerik” ve “erişilebilirlik” başlıkları altındaki 35 kriter doğrultusunda test edilmiştir. Bu kriterler [3]:

Tasarım

- Ana sayfa tasarımı (sade ve basit)
- Uygun simge/simgeler (resim)
- Kontrolsüz kayan metin kullanılmaması
- Gereksiz web aracı kullanılmaması (gereksiz araç, ses, görüntü vb)
- Sayfalarda standart yerleşim düzenleri
- Uygun sayfa başlığı kullanımı
- Bilgi formu açıklamaları (formlardaki açıklamalar)
- Metinlerin okunabilir olması
- Uygun arka plan rengi
- Kolay site ismi
- Açık ve özlü ifadelerin kullanımı

Dolaşım

- Ana sayfa bağlantısı bulunması (her sayfada)
- Uygun menü kullanımı (dolaşımı kolaylaştıran)
- Yön okları kullanımı (sayfalarda gezinme)
- Çalışan linkler
- Uygun link kullanımı
- İlgili ve erişilebilir (kırık olmayan) linkler

İçerik

- Telefonla site sorun giderme (kullanıcılara yardım)
- e-mail ile site sorun giderme (kullanıcılara yardım)
- Kurum iletişim bilgileri (e-posta, adres, telefon vb)
- Kurumsal bilgiler (teşkilat yapısı, mevzuat, faaliyetler vb)
- Kurum logosu kullanımı (açık ve görülebilir/her sayfada)
- Yabancı dilde site yayını
- Yazdırılabilir sayfa dönüşümü

- Alternatif doküman programları (PDF, DOC, XLS vb türler)
- Üye kaydı yapılabilmesi
- Üye kaydının basit olması
- Değişikliklerin üyeye gönderilmesi (güncelleme vb)
- Site içi arama motoru
- Arama motorunda gelişmiş arama seçeneği
- Güncelleme tarihinin belirtilmesi

Erişilebilirlik

- Görme engellilerin erişilebilirliği
- Duyma engellilerin erişilebilirliği
- Farklı tarayıcı kullananların erişilebilirliği (IE, Mozilla, Netscape, Opera)

şeklindedir. Siteleri test ederken, siteleri test eden her kullanıcıdan kriterin özelliğine göre ilgili kritere 0-2 veya 0-4 arası puan vermesi istenmiştir. Kullanıcıların her kriter için verdikleri puanlar toplanarak sitenin ilgili kriterden aldığı toplam puan, başlıkların altındaki kriterlerin toplam puanlarının toplanmasıyla da sitelerin başlıklardan aldıkları puanlar hesaplanmıştır. Böylece her kriter ve her ana başlık için alınabilecek tam puanlar çerçevesinde sitenin 100 puan üzerinden değerinin hesaplandığı ifade edilmiştir. Sitelerin test edilmesinde bu kriterler ile e-hizmetlerin kullanılabilirliği yerine sitelerin standartlara uygunluk düzeyinin değerlendirilmesinin amaçlandığı, bu nedenle ilgili çalışmada karma bir kullanıcı grubunun kullanılmasına gerek görülmediği belirtilmiştir [3].

Windows XP-Internet Explorer yüklü bilgisayarlarda, 32 kamu internet sitesinin test edildiği bu çalışmada, sitelerin standartlara uygunluk oranlarının “tasarım”da % 54, “dolaşım”da % 55, “içerik”de % 40, “erişilebilirlik”de ise % 57 olduğu ifade edilmiştir. “İçerik” ve “erişilebilirlik” başlıklarında kamu internet sitelerinin çok iyi sonuçlar alamadığı, özellikle “erişilebilirlik” bölümünde; görme engellilerin kamu internet sitelerini kullanmakta problem yaşadıkları, bunun yanı sıra farklı tarayıcılarda bazı sitelerin görüntülenmesinde çeşitli sorunlarla karşılaşıldığı belirtilmiştir. Ayrıca çalışmadaki ankette, kurumlara; internet sitelerini hazırlamada

uluslararası standartları dikkate alıp almadıkları sorulmuş, ancak kurumların sadece % 17'sinin bu soruya olumlu cevap verebildikleri belirtilmiştir [3].

Konuya yakın olan diğer çalışmadaysa 24 kamu kurumu internet sitesinin ana sayfası, kriterlerinin WAI erişilebilirlik rehberi ile “Kamu Kurumları İnternet Sitesi Kılavuzu”nun dikkate alınarak belirlendiğinin ifade edildiği; 8 kategori altındaki şu 30 kriter açısından değerlendirilmiştir [35]:

Çevrimiçi Bilgi

- Logo ve açık ad
- Başlık
- Telefon-iletişim
- Adres bilgisi
- Bağlantılar
- Sitenin amacı
- Yayınlar
- Veri tabanları
- Yardım
- Sık sorulan sorular
- Ses dosyaları
- Video dosyaları

Elektronik Hizmetler

- Kuruma gitmeden alınabilecek hizmetler
- Çevrimiçi yardım (sohbet odası, interaktif)

Gizlilik ve Güvenlik

- Ziyaretçi bilgilerinin kullanılmayacağına dair bilgi
- Gizlilik ve güvenlik sözleşmesi

Engelli Erişimi

- Engelli erişim seçenekleri

Yabancı Dil Erişimi

- Farklı dillerdeki erişim

Kamu Erişimi

- Personel e-posta
- Arama
- Mesaj tahtası
- Güncelleme bilgisi
- e-bülten ve hatırlatma panosu

Üst Veri

- Sorumluluk-yazar (site kurucusu)
- Tanımlama (siteyle ilgili genel bilgi için açıklayıcı bilgiler)
- Anahtar kelimeler (aranan hizmetlere kolayca ulaşmak için)

Teknik Özellikler

- Kırık bağlantılar
- Dosya boyutu
- Yüklenme süresi
- Güvenlik düzeyi

Kamu web sayfalarının genel ve teknik özellikler açısından değerlendirilmesinin amaçlandığının ifade edildiği bu çalışmada, “üst veri” ve “teknik özellikler” kategorisindeki kriterler için WebXact/Bobby programının kullanıldığı, araştırma bulgularına göre ise kamu web sayfalarının genel özellikler açısından çoğunlukla belirlenen kriterleri karşıladığı, hizmet sunumu, teknik özellikler ve etkileşimli hizmetler konusunda ise kayda değer oranda eksikliklerin görüldüğü belirtilmiştir [35].

Literatür taramalarında karşılaşılan, konuya yakın olarak yapılmış bu iki çalışmada da, kriterler “Kamu Kurumları İnternet Sitesi Kılavuzu”ndaki kriterlerle

genel olarak örtüşmektedir. Ancak söz konusu kılavuz, web standartlarının da göz önünde bulundurulduğu, teknik detayları da içeren daha fazla öneri içermektedir.

Örnekleminde ağırlıklı olarak kamu kurumları internet sitelerinin yer aldığı, Türkiye’deki internet sitelerinin standartlara uyumluluğunun değerlendirildiği başka bir çalışmada ise şu dört kriter açısından değerlendirme yapılmıştır [7]:

- W3C HTML (Hyper Text Markup Language) hata sayısı
- Sitelerin barındırıldığı sunucu tipi
- Ana sayfanın byte cinsinden boyutu
- Kötü-çoklu kodlama (karakter kodlaması)

241 internet sitesinin değerlendirildiği ilgili çalışmadaki bulgular şu şekildedir [7]:

- W3C HTML geçerlilik testi ortalama hata sayısı 127, en az hata 2, en fazla hata 1186’dır.
- Hiçbir sitede doküman tip bildirimini kullanılmamıştır.
- Sitelerin barındırıldığı sunucuların oranına göre (Microsoft-IIS % 58, Apache % 37, diğerleri % 5) açık kaynak Apache sunucusu tercihi yaklaşık dünya ortalamasının yarısı kadardır.
- Hemen her sitede uluslararası standartlarda olmayan karakter kodlaması yapılmış, bazı sitelerde birden fazla karakter kodlaması yapılmış ve bazı sitelerde ise hiçbir karakter kodlaması yapılmamıştır.
- Sayfa boyutu açısından kamu internet sitelerinde aşırı yüksek uçta değerler bulunmaktadır.

Bu bulgulara göre çalışmada, e-Devlet sürecinde kamu sitelerinin hem uluslararası standartlara uygunluk yönünden ciddi eksikliklerinin bulunduğu, hem de bu süreçte kamuda ortak stratejilerin gelişmediği, daha yüksek standart uyumluluğu gösteren ve tedarikçiye bağımlılık riski taşımayan açık kaynak kodlu yazılımların özellikle kamu hizmetlerinde kullanılmasının daha uygun olacağı belirtilmiştir [7].

Yapılan başka bir çalışmada ise 6 e-Devlet sitesi kullanılabilirlik açısından incelenmiştir. Çalışmada katılımcılardan, incelenen sitelere has görevler verilerek bu siteleri kullanmaları istenmiştir. Yapılan gözlem ve ses kaydı verilerinin analizi sonucunda, en sıklıkla karşılaşılan sorunların menü tasarımı, bağlantı başlıkları, ana sayfada içerik tasarımı ile çalışmayan arama kutuları ve aktif olmayan bağlantıların olduğu ifade edilmiştir [36].

2.1.2 Dünyadaki Çalışmalar

Kamu kurumları internet siteleri için hazırlanmış olan rehberlere çeşitli ülkelerde de rastlanmaktadır. Yeni Zelanda Hükümeti, kamu sektörü internet sitelerinin yönetimi ve tasarımı için bir rehber hazırlamış ve 2004 yılında güncellediği bu rehberde belirledikleri standartları; “adalet”, “bütünlük”, “güven” ve “ekonomi” olmak üzere dört temel değer üzerine yerleştirdiğini belirtmektedir [37]. Rehber incelendiğinde, rehberin web standartları ve WAI erişilebilirlik rehberi de dikkate alınarak hazırlandığı, rehberde tasarımcılara; hem içerik ile ilgili hem de teknik özelliklerle ilgili ayrıntılı açıklamalar yapıldığı görülmektedir. Ayrıca rehberde, hazırlanan sitelerin çok çeşitli web tarayıcılarında ve değişik işletim sistemlerinde çalışması gerektiği ve sitelerin teknik, erişilebilirlik ve kullanılabilirlik testlerinin mutlaka yapılmasının gerekliliği vurgulanmaktadır [37].

Rehber konusunda benzer bir çalışma Avustralya’da da bulunmaktadır. Avustralya Hükümeti Bilgi Yönetimi Bölümü sitesinde bulunan rehber, çok ayrıntılı olmasa da erişilebilirlik, içerik ve teknik yöntemler hakkında bilgiler içermektedir [38].

Bu konuda en kapsamlı ve ayrıntılı çalışmanın ise Birleşik Krallık’ta olduğu görülmektedir. Birleşik Krallık’ın, devlet web siteleri için rehber niteliğinde hazırladığı ve 2007 yılında güncellediği el kitabı; altı bölümden oluşmaktadır [39]:

- *Web sitesi yönetimi:* web sitesinin amacı, web sitesinin yönetimi, reklamlar ve sponsorlar, değerlendirme ve kontrol listeleri, alan adı kaydı vb.
- *Web sitelerinin içeriği:* asgari site gereksinimleri, diğer dillerin kullanımı, web grafikleri, tarayıcı uyumlulukları, genel erişilebilirlik vb.
- *Dosya depolama ve yapı:* web sunucuları, dokümanların yapısı, dosyaları isimlendirme, dosyaların boyutları vb.
- *HTML dili, diğer formatlar ve betikler:* HTML sayfaları, HTML olmayan dosya formatları, taşınabilir dosya formatı, betikler ve programlama, çerezler, *meta* etikleri, web tarayıcısı özellikleri vb.
- *Geliştirme:* Yayımlama, televizyonda web sitesi, diğer dijital ortamlar (cep bilgisayar, PDA, oyun konsolları...) vb.
- *Teknik detaylar:* HTML Geçerliliği, XHTML, stil sayfaları, HTML tabloları, çerçeveler, web sayfasında dolaşma vb.

Yeni Zelenda, Avustralya ve Birleşik Krallık'ın rehberleri ile TÜBİTAK-MAM tarafından ülkemiz için hazırlanan internet sitesi kılavuzu karşılaştırıldığında, kılavuzların birçok temel noktada birbiriyle kesiştikleri görülmektedir.

Bununla birlikte Amerika Hukuk Kütüphanesi Derneği'ne bağlı Yasal Elektronik Bilgi Erişimi Komisyonu'nun web sayfasında; yasal bilgi sağlayan bütün web siteleri için genel değerlendirme kriterleri “İçerik”, “Organizasyon”, “Dolaşım ve Kullanılabilirlik” ile “Erişilebilirlik” olmak üzere dört ana bölümde toplanmaktadır [40]. “İçerik” bölümünde yasal belgeler ve bu belgelerin sunum formatları, kapsamı, bağlamı, güncelleme tarihi vb ile ilgili genel kriterler yer almakta, “organizasyon” bölümünde ise kullanıcıların aradıkları belgeleri kolayca sitede bulabilmeleri veya bu belgelere kolayca ulaşabilmeleri için gerekli organizasyonun yapılmasının gerekliliği belirtilmektedir. “Dolaşım ve Kullanılabilirlik” bölümünde ise yasal belgelerin içeriğinde başka yasal belgelerden söz edilmesi durumunda bu belgelere link verilmesi, web sitelerinde yardım/sıkça sorulan sorular sayfalarının ve site haritasının bulundurulması gibi konular açıklanmaktadır. “Erişilebilirlik” ile ilgili kısımda ise WAI erişilebilirlik rehberine

uyulması, kuruma erişim bilgilerinin sayfada yer alması, sitenin farklı web tarayıcılarında aynı içeriği verebilmesi vb belirtilmektedir [40].

Web sayfası değerlendirmesi ile ilgili olarak yapılan bir çalışmada, Avustralya hükümetinin değişik kademelerinde çalışan portal geliştiricileriyle görüşmeler yapılarak ve önceki araştırma sonuçlarından başarılı web sitelerinin anlamlı bileşenleri hakkında elde edilen verilerden yola çıkılarak e-Devlet web sitelerinin değerlendirilmesiyle ilgili 106 sorudan oluşan bir aracın geliştirildiği ifade edilmektedir [41]. Excel çalışma sayfası formatında olduğu ifade edilen bu araçta yer alan soruların; “Güvenlik/Gizlilik”, “Kullanılabilirlik”, “İçerik”, “Hizmetler”, “Vatandaş Katılımı” ve “Özellikler” olmak üzere 6 kategoride toplandığı belirtilmiş, ancak söz edilen Excel belgesine çalışmada yer verilmemiştir [41].

Öte yandan 2002 yılında bir şirket tarafından yapılan e-Devlet ile ilgili araştırmada, en iyi örnek e-Devlet web sitesi olarak belirtilen beş ülkeye ait web sitesi (Singapur, Kanada, Finlandiya, Hong Kong ve Avustralya), W3C HTML geçerlilik testi, WebXact/Bobby, Netmechanic gibi araçlarla değerlendirilmiştir. Araştırmanın bulgularından bazıları şu şekildedir [42]:

- WebXact/Bobby erişilebilirlik testi sonuçlarına göre bu sitelerde erişilebilirlikle ilgili bazı hatalar bulunmaktadır. Önceliklere göre en az hata bulunan veya hiç hata olmayan Kanada ve Hong Kong siteleridir.
- Singapur ve Hong Kong sitelerinde stil sayfaları kullanılmamıştır.
- Kanada ve Avustralya sitelerinde yüksek oranda *meta* etiketleri kullanılmıştır.
- En hızlı 2 sn ile Hong Kong sitesi, en yavaş ise 14 sn ile Kanada ve Singapur siteleri yüklenmektedir.
- W3C HTML geçerlilik testinden sadece Kanada sitesi başarılı olmuştur. Singapur ve Hong Kong sitelerinde ise doküman tip bildirimi yapılmamıştır.

- Netmechanic sonuçlarına göre tüm sitelerde tarayıcılarla ilgili problemlere rastlanmıştır. Ayrıca sadece Kanada'nın sitesinde kırık olmayan linkler tespit edilebilmiştir.

Çalışmada, bu bulgulara göre, 2002'deki bu araştırmada düşük sıralarda yer almış diğer e-Devlet sitelerinde ise daha fazla hatalara rastlanabileceği belirtilmektedir [42].

Yapılan diğer bir araştırmada ise 50 Amerika Birleşik Devletleri hükümet sitesi, engelli kişilerin erişilebilirliklerini değerlendirmek için Webxact/Bobby aracı ile test edilmiştir [43]. Çalışmada, birçok federal kanunda devlet web sitelerine engelli kişilerin erişebilmesinin şart koşulduğu, belirtilen araç ile yapılan erişilebilirlik testi sonuçlarına göre sadece % 22'lik bir oranın yakalanabildiği ve bulgulara göre ise ABD hükümetinin beklediği erişilebilirlik hedeflerine ulaşmadığının söylenebileceği belirtilmektedir [43].

2.2 Web Standartları ve Erişilebilirlik ile İlgili Araçlar

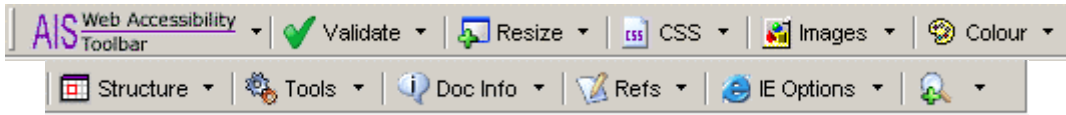
Web standartları ve erişilebilirlik konusunda gerek çevrimiçi ve ücretsiz, gerekse ticari amaçlı birçok araç bulunmaktadır [44]. Özellikle W3C'nin internet sitesinde yer alan W3C geçerlilik servisleri (<http://validator.w3.org> ve <http://jigsaw.w3.org/css-validator/>) "Kamu Kurumları İnternet Sitesi Kılavuzu"nda da önerilmektedir. Bu servisler ile HTML, XHTML vb web dokümanları ile CSS (Cascading Style Sheets) dosyalarının geçerlilikleri kontrol edilebilmektedir [45, 46].

Bunlarla birlikte literatürde erişilebilirlik konusunda hazırlanmış çeşitli araç çubuklarına da rastlanmaktadır. Avustralya'daki bir organizasyon sitesinde yayımlanan erişilebilirlik ile ilgili araç çubuğundaki bazı temel özellikler şu şekildedir [47]:

- W3C HTML ve CSS geçerlilik testlerini yapabilme
- Pencereyi yeniden boyutlandırarak sayfayı görüntüleyebilme

- Stil sayfalarını kapatarak sayfayı görüntüleyebilme
- Resimleri kapatarak sayfayı görüntüleyebilme
- Sayfayı ve sayfadaki resimleri siyah/beyaz olarak gösterebilme
- Kullanılan *accesskey*, *tabindex* niteliklerini tespit edebilme
- Kodlamada kullanılan çeşitli etiketler ile ilgili bilgiler
- Sayfayı yakınlaştırma/uzaklaştırma
- Sayfa boyutu/indirme hızı ile ilgili bilgiler
- Bazı erişilebilirlik araçları ile erişilebilirlik testleri vb

Windows&Internet Explorer’da çalıştığı belirtilen araç çubuğunun görüntüsü aşağıdaki gibidir:

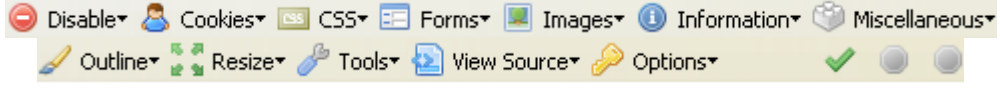


Şekil 2.1 AIS Web Erişilebilirliği Araç Çubuğu

Firefox, Flock ve Seamonkey web tarayıcıları için tasarlandığı belirtilen başka bir araç çubuğu olan “Web Developer”da da benzer özellikler bulunmaktadır [48]:

- Cache, Java, Javascript vb kapatma
- Çerezleri kapatma/silme
- Stil sayfalarını kapatma/görüntüleme
- Resimleri kapatma
- Formlar ve form elemanları ile ilgili bilgiler
- *accesskey*, *tabindex* niteliklerini görüntüleme
- Kodlamada kullanılan çeşitli etiketler ile ilgili bilgiler
- Pencereyi yeniden boyutlandırma
- Bazı erişilebilirlik araçları ile erişilebilirlik testleri
- W3C HTML/CSS geçerlilik testleri vb

Sözü edilen araç çubuğunun görüntüsü Şekil 2.2’de görüldüğü gibidir.



Şekil 2.2 Web Developer Araç Çubuğu

Erişilebilirlikle ilgili ücretsiz veya ticari olan araçlardan bazıları ise şunlardır [44]:

- CSE HTML Validator (<http://www.htmlvalidator.com/>)
- JAWS®for Windows
(http://www.freedomscientific.com/fs_products/software_jaws.asp)
- Lynx Viewer (<http://www.delorie.com/web/lynxview.html>)
- The WAVE Accessibility Checker
(<http://www.wave.webaim.org/index.jsp>)
- WebXact/Bobby (<http://webxact.watchfire.com/>)
- InFocus™ (<http://www.ssbtechnologies.com/products/infocus/>)
- Lift
(http://www.usablenet.com/products_services/products_services.html)

Bu araçlar genel olarak W3C/WAI erişilebilirlik rehberi ile ABD Federal Hükümeti Bölüm 508 erişilebilirlik kılavuzuna göre web sayfalarını test etmektedir [44]. Bununla birlikte otomatik araçların; erişilebilirlik ile ilgili genel bir fikir verdiği, daha çok sorunlu olan bölümleri gösterdiği, bazen bu araçların bulamayacağı problemlerin olabildiği veya sorun olmayacak bir şeyi sorunlu olarak gösterebildiği belirtilmektedir. Ayrıca bir sitenin tüm erişilebilirlik araçlarında başarılı olmasının sitenin tamamen erişilebilir olduğunu göstermediği ve el yordamı ile müdahalenin gerektiği vurgulanmaktadır [49].

3. YÖNTEM

Bu çalışmada kamu kurum ve kuruluşları internet sitelerinin 2007/4 sayılı genelge kapsamında değerlendirilmesine yönelik bir araç geliştirilmesi hedeflendiği için, öncelikle bu genelge ile birlikte yayımlanan TÜBİTAK-MAM tarafından hazırlanmış olan söz konusu “Kamu Kurumları İnternet Sitesi Kılavuzu” (EK A) ayrıntılı bir şekilde incelenmiştir. Ardından bu kılavuzda yer alan kriterlerden hangilerinin değerlendirme işleminde kullanılacağına karar verilmiştir. Bu nedenle,

- El yordamına gerek duyulmadan mümkün olduğu kadar bilgisayar tarafından otomatik olarak değerlendirilebilen kriterler
- İçerik ile ilgili kriterlerden; kamu kurum ve kuruluşları web sayfalarının geneli için geçerli olan temel kriterler

değerlendirme işlemlerinde kullanılmıştır. Bununla birlikte mümkün olduğunca çok sayıda kriter seçilmesine özen gösterilmiş ve ayrıca kılavuzda belirtilmemiş ancak literatür taramalarında sıklıkla karşılaşılan erişilebilirlik ve standartlar ile ilgili bazı kriterlerin de değerlendirme işlemine alınmasına karar verilmiştir.

Geliştirilmesi hedeflenen aracın açık kaynak kodlu olması düşünülmüş ve aracın yazılmasında açık kaynak dillerden biri olan PHP (Personel Home Page) tercih edilmiştir. Öte yandan bu aracın ileride başka kişiler/programcılar tarafından geliştirilebilmesine olanak sağlamak için direk olarak aracın yazılmasına geçilmemiş, öncelikle *değerlendirme işlemlerini yapabilen bir PHP class'ı (sınıf'ı)* yazılmaya çalışılmıştır. Nesneye Yönelik Programlama (NYP) olarak da adlandırılan bu yöntemin avantajları şu şekildedir [50]:

- Programcının kendisine yüklemiş olduğu yüklerin (değişkenler, durumlar, fonksiyonlar, bunların aralarındaki ilişki vb); nesnelere

metotlar ve özellikler içermesinden dolayı programcıdan nesneye yüklenebilmesi

- Nesnelerin gerçek yaşam problemlerini programlamaya aktarmada kolaylık sağlaması
- NYP'nin kalıtım özelliği sayesinde, yazılan kodun başka amaçlar için ek özellikler eklenerek tekrar kullanılabilmesi
- Yazılan kodun daha okunabilir ve anlaşılabilir olması
- Programcıların kolektif çalışmalarına olanak sunması
- Zor problemleri sunduğu yaklaşım ile daha kolay ve hızlı çözülebilir hale getirebilmesi
- Sonradan kullanılabilir ve geliştirilebilir olması

“*Web Sayfası Değerlendirme Class*”ı olarak adlandırılan, EK B’de kaynak kodu sunulan ve bu çalışmada geliştirilmesi hedeflenen aracın *çekirdeğini* oluşturan bu class, “Kamu Kurumları İnternet Sitesi Kılavuzu” (EK A) kapsamında belirlenen kriterler dikkate alınarak yazılmıştır. Class’ın yazılmasının ardından; bu class’ı, bu class’ın girdilerini ve çıktılarını kullanarak çalışan, kaynak kodu EK C’de sunulan, bu çalışmada sözü edilen araç (bir web sayfası) geliştirilmiştir. Bu bağlamda ve NYP mantığı çerçevesinde araç içinde kullanılmış olan class öne çıkmakta ve önem kazanmaktadır. Çünkü asıl değerlendirme işlemi yapan class ve bu class’taki fonksiyonlar olmakta, değerlendirme işlemi yaptıran ise araç olmaktadır. Class’ın yazılmasında şu noktalara dikkat edilmiştir:

- Class’ın yazımında, yukarıda da değinildiği gibi söz konusu kılavuz kapsamında ve literatür taramaları sonucunda belirlenen kriterlere göre kodlama yapılmıştır.
- Class’ın açık kaynak kodlu olması düşünüldüğünden class içinde kullanılan açıklamalar İngilizce olarak yazılmıştır.
- Class, içerik ile ilgili kriterlerde web sayfalarını hem Türkçe hem de İngilizce olarak değerlendirilebilecek şekilde kodlanmıştır.
- Class, değerlendirilen sayfanın ilgili kriterle ilgili başarılı, başarısız vb durum değerlendirmesinin yanı sıra, bu kriterlerle ilgili basit

istatistikler (yüzde vb) verecek şekilde yazılmıştır. Böylece bu verilerin istatistiksel olarak kullanılabilmesi amaçlanmıştır.

Class'ı kullanarak geliştirilen araçta (web sayfasında) ise şunlar göz önünde bulundurulmuştur:

- Değerlendirilen web sayfası ile ilgili elde edilen verilerin bir veritabanına kaydedilmesi
- Değerlendirme sonucu class'tan gelen veriler doğrultusunda ve/veya el yordamı gerektiren kriterlerin (örneğin içerikle ilgili olanların), bu yöntemle değerlendirilmesinden sonra belirlenen sonucun bu veritabanına girilebilmesi/güncelleştirilebilmesi (yönetici modu)
- Değerlendirilen web sayfası sonuç değerleri ile değerlendirilmiş bütün web sayfalarından elde edilmiş verilere göre hesaplanan ortalama değerlerin karşılaştırılabilmesi
- Geliştirilen araçta ortalamaların alan adı ve ülke kodu türüne göre filtrelenebilmesi (gov.tr, edu.tr vb genel ortalamaları gibi)
- Class, içerik ile ilgili kriterlerde İngilizce olarak da değerlendirme yapabilecek şekilde tasarlandığı için bu durumun araca da yansıtılması
- Değerlendirme kriterlerinin kısa açıklamalarının araçta bulunması

Class'ın ve aracın geliştirilmesinin ardından bu araç ile gov.tr, edu.tr, mil.tr, k12.tr, org.tr, pol.tr ve bel.tr uzantılı 5'er site olmak üzere toplam 35 kamu kurum ve kuruluşu internet sitesinin ana sayfası değerlendirilmiştir. Değerlendirme işlemine alınan siteler rasgele seçilmiştir. Ancak sitelerin seçilmesinde;

- Sağlıklı sonuçlar için sitenin (i)frame (çerçeve) içermemesi
- Sitenin W3C CSS geçerlilik testinin yapılabilmesi

durumu göz önünde bulundurulmuştur.

Bu bölümde öncelikle değerlendirme işlemlerinde kullanılmak üzere belirlenen kriterler, yazılan class, class'ta kullanılan fonksiyonlar ve class'ın

kullanımı açıklanmış, sonra da class'ı kullanarak geliştirilmiş web sayfası değerlendirme aracına değinilmiştir. Değerlendirilen siteler ile ilgili sonuçlar ise bulgular ve yorumlar bölümünde sunulmuştur.

3.1 Değerlendirme İçin Belirlenen Kriterler

“Kamu Kurumları İnternet Sitesi Kılavuzu”nun incelenmesi sonucunda değerlendirme işlemi için belirlenen kriterler “İçerik (Asgari)”, “Tasarım (Erişilebilirlik)”, “Tasarım (Teknik Özellikler)” ve “Genel Görünüm” olmak üzere kılavuzdaki başlıklara da benzer olarak dört ana başlıkta toplanmıştır. Belirlenen kriterler ise şu şekildedir:

- *İçerik (Asgari)*
 - *Genel*
 - Kurum logosu
 - Kurumun açık adı
 - Sitenin amacı
 - Site içi bağlantılar
 - Duyurular/haberler
 - Arama mekanizması
 - Yardım sayfalarına bağlantı
 - Sıkça sorulan sorular sayfasına bağlantı
 - Şikayet/geri bildirim sayfalarına bağlantı
 - *Kuruma Erişim Bilgileri*
 - Adres
 - Telefon
 - E-posta
 - E-posta uygunluğu
- *Tasarım (Erişilebilirlik)*
 - *Tarayıcı Tipi*

- Internet Explorer
- Mozilla Firefox
- Konqueror
- Lynx
- *Ekran Çözünürlüğü*
 - 1280 x 1024
 - 1024 x 768
 - 800 x 600
- *Tasarım (Teknik Özellikler)*
 - *Genel*
 - W3C HTML Geçerlilik Testi
 - W3C CSS Geçerlilik Testi
 - TIDY Erişilebilirlik Testi (Seviye 1-2-3)
 - Javascript'ler
 - Javaapplet'ler
 - *meta* etiketleri
 - Çerezler
 - *Sayfa Boyutu*
 - Resimsiz
 - Resimlerle birlikte
 - *Kodlama*
 - W3C doküman tip bildirim
 - Yorumların uygunluğu
 - Sayfa başlığı (uzunluk, uygunluk)
 - Doğru etiketleme (TIDY hata ve uyarıları)
 - Doğru etiketleme (*body* etiketi ve nitelikleri)
 - HTML form kullanımı, formlarda *label* ve *tabindex* nitelikleri kullanımı

- *Genel Görünüm*
 - *Genel*
 - Ana menü kullanılması
 - *Logolar*
 - Metinlerin okunabilirliği
 - GIF formatı
 - Statik logolar
 - *Sayfada Gezinme*
 - Resim içeren bağlantılarda resimlerdeki *alt* nitelikleri
 - HTML menü kullanımı
 - *tabindex* niteliklerinin kullanımı
 - *accesskey* niteliklerinin kullanımı
 - Resim haritaları kullanım uygunluğu, harita alanlarında *alt* niteliklerinin kullanımı
 - *Metinler*
 - *font* etiketlerinin kullanılmaması
 - Statik yazılar
 - Yazı tipi büyüklükleri (harici ve dahili CSS'ler ile satırda kullanılan stiller)
 - *Resimler*
 - GIF/JPEG formatı
 - Resim boyutu
 - *width* niteliklerinin kullanımı
 - *height* niteliklerinin kullanımı
 - *alt* niteliklerinin kullanımı (uzunluk, son karakterlerle birlikte)
 - *216 Renk Paletinin Kullanımı*
 - Harici ve dahili CSS kodlarında
 - Satırda (inline) kullanılan stillerde

- HTML niteliklerinde
- *Tablolar*
 - Tablo sonu etiketlerinin kullanımı
 - Tablo arka plan rengi kullanılmaması
 - Tablo arka plan resmi kullanılmaması

Kriterlerin belirlenmesi işleminin ardından “Web Sayfası Değerlendirme Class”ının yazılmasına geçilmiştir.

3.2 Web Sayfası Değerlendirme Class’ı (Sınıf’ı)

Bir web sayfasının oluşturulmasında çeşitli HTML düzenleyiciler, çeşitli içerik yönetim sistemleri veya PHP, ASP vb diller tercih edilebilmektedir [51]. Öte yandan bir web sayfası hangi teknoloji ile hazırlanırsa hazırlansın sonuçta bir HTML kodundan oluşmaktadır [10, 51]. Bu bağlamda sözü edilen class’ın yazılmasında öncelikle değerlendirilecek web sayfasının HTML kodunun parse (ayıklanma-ayrıştırılma) işlemine gereksinim duyulmaktadır. Bu noktada hem açık kaynak kodlu olması, hem söz konusu kılavuzda kod kalitesi kontrolü için önerilmesi, hem de PHP eklentilerinde yer alması nedeniyle HTML kodunun parse edilmesinde HTML Tidy tercih edilmiştir. Tidy, HTML kodlarının hatalarını düzenleme-temizleme vb işlemlerini yapabilmekte, aynı zamanda HTML belgesini parse edip dallanmalı bir yapıda gösterebilmektedir [52]. Tidy ile ilgili ayrıntılı bilgi <http://tr2.php.net/tidy> sitesinden elde edilebileceği gibi <http://tidy.sourceforge.net/> veya <http://www.w3.org/People/Raggett/tidy/> sitelerinden de edinilebilir.

Tidy ile parse edilmiş HTML kodunda veri çekebilmenin güç olduğu az sayıda durum için ise literatür taramalarında karşılaşılan bir HTML Parser [53], class’ta kullanılmak üzere yeniden düzenlenip class formatına uygun hale getirilmiştir (EK B.1, “parse_html” fonksiyonu).

Bununla birlikte web sayfalarının görünümünü deęiřtirmede, sitelerin g¼ncellenme ve bakım iřlemlerini kolaylařtıran stil sayfalarının (CSS) [10] kodlarının parse edilmesi iin ise CSSTidy kullanılmıřtır. CSSTidy, stil sayfası kodlarını hem optimize edebilmekte, hem de parse edebilmektedir [54]. Bu alıřmada CSS kodlarını parse etmek amacıyla yararlanılan CSSTidy ile ilgili bilgi, <http://csstidy.sourceforge.net/> sitesinden elde edilebilir.

3.2.1 Class'ın Sınırlılıkları

Bu alıřmada geliřtirilmesi hedeflenen arata kullanılmak üzere yazılan class řunlarla sınırlıdır:

- HTML 4.01
- CSS 2.1
- HTML Tidy 2.0
- CSS Tidy 1.3
- PHP 5

Ařaęıda, yazılan class'ta yer alan fonksiyonlar ve bu fonksiyonların alıřma prensipleri aıklanmıřtır.

3.2.2 Class'taki Fonksiyonlar ve G¼revleri

Class'ın yazılmasında y¼ntem kısmında da belirtilen noktalar dikkate alınarak kodlama iřlemi yapılmıřtır. Class'ta yazılan fonksiyonlar *constructor (kurucu)*, *private (zel)* ve *public (genel)* fonksiyonlar olmak üzere 3 t¼rl¼d¼r. Constructor fonksiyonlar nesne oluřturulduęunda (veya class alıřtırıldıęında) otomatik olarak alıřmakta, private fonksiyonlar sadece class iinden aęırıldıęında alıřmakta ve kalıtım koluyula aktarılamamakta, public fonksiyonlar ise class iinden veya dıřından aęırıldıklarında alıřabilmektedir [55]. te yandan bu alıřmada, class iinde private olarak tanımlanan fonksiyonların kalıtım yoluyla aktarılabilen protected

(korunmalı) [55] fonksiyonlara çevrilmeleri, class'ın çalışmasına etki etmeyecektir. Bu nedenle istenilirse bu fonksiyonlar protected türüne de çevrilebilirler.

Bununla birlikte değerlendirme kriterlerinin başarılı, başarısız vb sonuçlarını elde etmek için bir PHP betiği (test_functions.php) yazılmıştır (EK B.2). Bu betikteki fonksiyonlar, web sayfasının ilgili kriterden aldığı değere ilişkin başarılı, başarısız vb değerlendirme sonucunu geri döndürmek için public fonksiyonların içinde sıklıkla kullanılmıştır. Bu nedenle public fonksiyonlardan dönen değerlere ilişkin açıklamaların, bu PHP betiği anlatılırken verilmesinin daha uygun olacağı düşünülmüştür.

Bu bağlamda çalışmadaki bu bölüm; “constructor (kurucu) fonksiyon”, “private (özel) fonksiyonlar”, “public (genel) fonksiyonlar” ile “dönen değerler ve sonuç fonksiyonları” olmak üzere dört kısımda ele alınmıştır.

3.2.2.1 Constructor (Kurucu) Fonksiyon (EK B.1)

Constructor fonksiyonlar nesne oluşturulduğunda otomatik çağrıldıkları için nesnenin ilk özellik ve işlemlerini yaptırmak için oldukça uygundur ve dışarıdan parametre kabul etmektedirler [50]. Yazılan class'ta da bu avantajdan yararlanılmış ve dışarıdan dört parametre alabilecek şekilde kodlama yapılmıştır. Bu parametreler sırasıyla:

- Değerlendirilecek web sayfasının adresi
- İçerik ile ilgili değerlendirme dili (Türkçe/İngilizce)
- W3C HTML geçerlilik testinin gerçekleştirilip gerçekleştirilmemesi
- W3C CSS geçerlilik testinin gerçekleştirilip gerçekleştirilmemesi

şeklindedir. Constructor fonksiyon dışarıdan tetiklendiğinde öncelikle seçilen dile göre içerik ile ilgili değerlendirmeler için gereken dil dosyası (EK B.3) ile değerlendirilecek kriterlerin sonucunu (başarılı, başarısız vb) belirlemede kullanılan test_functions.php betiğini belleğe yüklemektedir. Bu işlemin ardından belleğe

yüklenen dil dosyasında tanımlanan değerler ile W3C testlerinin yapılıp yapılmaması isteği, değerlendirme işlemlerinde kullanılmak üzere ayarlarla ilgili dizi değişkenine (\$settings) aktarılmaktadır.

Constructor fonksiyon bu işlemlerden sonra sırasıyla şu işlemleri gerçekleştirmektedir:

- Değerlendirilecek web sayfası adresinin geçerli bir adres olup olmadığı kontrol eder ve adresten domain (alan adı) ve path (yol) bilgilerini elde eder.
- “create_folders” fonksiyonunu ile değerlendirilecek web sayfasına ait HTML kodunu, CSS kodlarını vb saklamak için gereken klasörleri oluşturur.
- “get_url_content” fonksiyonu ile değerlendirilecek web sayfasına ait HTML kodunu ilgili klasöre kaydeder.
- Değerlendirilecek web sayfasının HTML kodunun parse edilmesini HTML Tidy ile gerçekleştirir.
- “get_css_content” fonksiyonu ile değerlendirilecek web sayfasında kullanılan CSS dosyalarını/kodlarını elde eder ve ilgili klasöre kaydeder.
- “parse_css” fonksiyonu ile değerlendirilecek web sayfasında bulunan CSS kodlarının parse edilmesini CSSTidy ile gerçekleştirir.
- “get_images ” fonksiyonu ile değerlendirilecek web sayfasında kullanılan her bir resmi ilgili klasöre kaydeder.
- W3C HTML geçerlilik testinin yapılması istenmişse, “get_w3c_html_content” fonksiyonu ile bu testi W3C’nin sitesinde gerçekleştirir ve testin sonuç dosyasını ilgili klasöre kaydeder.
- W3C CSS geçerlilik testinin yapılması istenmişse, “get_w3c_css_content” fonksiyonu ile bu testi W3C’nin sitesinde gerçekleştirir ve testin sonuç dosyasını ilgili klasöre kaydeder.

3.2.2.2 Private (Özel) Fonksiyonlar

Class'ta, class içinden çağırılmak üzere yazılmış 20 adet private fonksiyon yer almaktadır. Bu fonksiyonlar ve görevleri aşağıda açıklanmıştır.

- *“get_nodes2” fonksiyonu (EK B.1)*

İlk ikisi giriş, diğer üçü çıkış olmak üzere toplam beş parametreyle çalışan bu fonksiyon, değerlendirilecek web sayfasının Tidy ile parse edilmesi sonucu ortaya çıkan dallanmalı yapıdan (`$this->tidyhtml`), istenilen HTML etiketlerini çekmekte, bundan dolayı class içinde kritik bir öneme sahip olmaktadır. İlk parametre Tidy ile parse edilmiş HTML kodunun dallanmalı yapısı (`$this->tidyhtml`), ikinci parametre ise bu yapıdan elde edilmek istenilen etiketin Tidy sabit adı veya kodudur (örneğin `TIDY_TAG_IMG`). Beşinci parametre ise bu yapıdan elde edilen etiketlerin, bu etiketlerin niteliklerinin vb bilgilerinin yer aldığı, fonksiyon tarafından döndürülecek diziyi (array) yakalayan çıkış parametresidir. Örneğin, değerlendirilen web sayfasından *img* etiketlerini çekmek ve dönen diziyi `$imgs` değişkenine aktarmak için class içinde şu kod kullanılmıştır:

```
$this->get_nodes2($this->tidyhtml, TIDY_TAG_IMG, " ", $imgs);
```

Fonksiyondaki üçüncü ve dördüncü parametreler sırasıyla ilgili etiketlerin parent (ebeveyn) ve sibling (kardeş) etiketlerini elde etmektedir. Ancak, fonksiyon içinde kodlama yapılmasına rağmen, class içinde bunlar kullanılmamıştır.

- *“parse_html ” fonksiyonu (EK B.1)*

Üç giriş parametresiyle çalışan bu fonksiyon, çeşitli HTML kodlarını parse etme amacıyla kullanılmıştır. Class'ta HTML kodunun parse işlemi için genel olarak Tidy kullanılmasına rağmen, Tidy ile parse işlemi yapılmış web sayfasının dallanmalı yapısından istenilen veriyi çekebilmenin güç olduğu durumlarda bu fonksiyon tercih edilmiştir.

Düzenli ifadeler (regular expressions) kullanılarak yazılmış bu fonksiyondaki ilk parametre HTML kodu, ikinci parametre etiket, son parametre ise ne tür veri alınmak istendiğini belirten tip parametresidir. Verilen HTML kodu ve etikete göre fonksiyona dört ayrı tip belirtilerek dört ayrı sonuç alınabilir. Tipler ve alınabilecek sonuçlar şunlardır:

- “b” : Eşleşen ilk elementin içeriğini döndürür.
- “ba”: Eşleşen tüm elementlerin içeriklerini döndürür.
- “a”: Eşleşen tüm etiketleri nitelikleri ile birlikte (başlangıç etiketlerini) döndürür.
- “nitelik”: Belirtilen etikette belirtilen nitelik varsa, bu niteliklere atanmış değerleri döndürür.

Örneğin, değerlendirilecek bir web sayfasının sayfa başlığının elde edilmesinde class içinde şu kod kullanılmıştır:

```
$title = $this->parse_html($titles[0]['value'], "title", "b");
```

- “*enclose*” fonksiyonu (EK B.1)

“`parse_html`” fonksiyonu içinden kullanılmak üzere yazılmış bu fonksiyon başlangıç ve bitiş etiketleri arasındaki veriyi elde eden düzenli ifade deseni döndürmektedir.

- “*parse_css*” fonksiyonu (EK B.1)

Bu fonksiyon kendisine parametre olarak verilen CSS kodunu, CSSTidy ile parse etmektedir.

- “*set_name*” fonksiyonu (EK B.1)

“`create_folders`” fonksiyonu ile değerlendirilecek web sayfasına ait verilerin tutulacağı klasör oluşturulurken ve web sayfasında kullanılan resim dosyaları

kaydedilirken, bu klasör ve dosya isimlerinde -özellikle Windows işletim sisteminde geçerli olan- dosya ve klasör adlandırma kurallarına uygun olmayan karakterler (?, *, : vb) bu fonksiyonda “_” karakteri ile değiştirilmektedir.

- “*check_url*” fonksiyonu (EK B.1)

Bu fonksiyon ile değerlendirilecek web sayfasında kullanılan; harici CSS dosyaları, import (ithal) edilen CSS dosyaları ve resim dosyalarının indirilmesinde, bu dosyaların gerçek veya tam adresleri elde edilmektedir. Örneğin, bir HTML kodunda yer alan 1.gif dosyası indirilmek istenildiğinde bu dosyanın yolu ve adı bu fonksiyon ile <http://www.abc.edu.tr/1.gif> şekline çevrilmektedir.

- “*check_import*” fonksiyonu (EK B.1)

Bu fonksiyon, “*get_css_content*” fonksiyonu tarafından gönderilen stil sayfaları kodunda, *screen* veya *all* medya tipine sahip import edilen bir CSS dosyasının olup olmadığını yinelemeli olarak (recursively) kontrol etmektedir. Import edilen her CSS dosyası, gönderilen stil sayfası kodunun ardına eklenmektedir. Böylece gönderilen stil sayfasının (harici veya dahili) tam CSS kodu elde edilmekte ve bu CSS kodu, yineleme işlemlerinin bitiminde fonksiyonun ikinci parametresiyle “*get_css_content*” fonksiyonuna geri döndürülmektedir.

Bununla birlikte fonksiyon; her yinelendiğinde (import edilen bir CSS dosyasına rastlanıldığında) ve ilk çalıştırıldığında, “*css_write*” fonksiyonu yardımı ile gönderilen stil sayfalarını ve import edilen CSS dosyalarını ilgili klasöre kaydetmektedir.

Öte yandan import işlemlerinin CSS kodlarının en üstünde yapılması gerekmektedir [56]. Bu fonksiyonun kodlamasında, sayfalarda yer alan import tanımlamalarının CSS kodlarının başında yapıldığı kabul edilmiş, sadece “@import ...” biçiminde kodlamalar için düzenli ifade yazılmış ve “@media ...” biçiminde import işlemleri dikkate alınmamıştır.

- “*css_write*” fonksiyonu (EK B.1)

Bu fonksiyon kendisine verilen stil sayfası dosyalarını/kodlarını ilgili klasöre kaydetmektedir.

- “*scan_directory_recursively*” fonksiyonu (EK B.1)

Bu fonksiyon, kendisine parametre olarak verilen klasörde (ve bu klasörün alt klasörlerinde) yer alan dosyaların bir listesini dizi şeklinde döndürür. Üç parametresi olan fonksiyon yinelemeli (recursively) olarak çalışmaktadır. İlk parametrede klasör adı, ikinci parametrede istenilirse filtreleme için dosya uzantısı verilmektedir. Son parametre ise her dosyanın yol (path), ad, uzantı ve boyut bilgilerini içeren diziyi elde etmek için kullanılmaktadır.

- “*create_folders*” fonksiyonu (EK B.1)

Bu fonksiyon, değerlendirilecek web sayfasının verilerini (CSS, resim, W3C testi sonuç dosyaları vb) saklamak için çeşitli klasörler oluşturmaktadır. Öncelikle değerlendirilecek olan bütün web sayfalarının saklandığı “www” adlı klasör oluşturulmakta, ardından değerlendirilmek istenilen web sayfasına ait bir ana klasör oluşturulmakta ve diğer klasörler de bu klasörün altında oluşturulmaktadır. Klasör isimleri \$settings değişkeninden alınmaktadır.

- “*get_url_content*” fonksiyonu (EK B.1)

Bu fonksiyon, değerlendirilmek istenilen web sayfasını; “create_folders” fonksiyonu ile oluşturulan ilgili klasöre kaydetmekte, bu web sayfasının HTML kodunu ise \$this->fcont değişkenine aktarmaktadır. Bu değişken, değerlendirilmek istenilen web sayfasının Tidy ile parse işlemleri için kullanılmaktadır.

Bu işlemin ardından “parse_html ” fonksiyonunda kullanılmak üzere, web sayfasındaki yorumların (comments), betiklerin (*scripts*) ve dahili CSS kodlarının temizlenmesiyle web sayfasının yeni bir içeriği oluşturulmakta ve elde edilen bu yeni

içerik, başka bir değişken olan `$this->fcont2`'ye aktarılmaktadır. “`parse_html`” fonksiyonu, yorum kodlarının arasında veya betik (*script*) kodlarının arasında yer alan HTML kodlarıyla da eşleşebileceğinden; böyle bir yöntemle hatalı sonuçların ortaya çıkması engellenmeye çalışılmıştır. Bu nedenle “`parse_html`” ile ilgili işlem yapıldığında değerlendirilmek istenilen web sayfasının gerçek kodu (`$this->fcont`) yerine bu değişkendeki (`$this->fcont2`) içerik kullanılmıştır.

Öte yandan içerik ile ilgili değerlendirmelerde kullanılmak üzere web sayfasındaki tüm etiketlerin temizlenmesiyle yeni bir içerik elde edilmiş ve `$this->fcont3` değişkenine aktarılmıştır. Bu şekilde web sayfasının salt metinden oluşan bir biçimi ortaya çıkarılmıştır. Bununla, içerik ile ilgili testler için yazılmış düzenli ifadelerin, HTML koduyla metinlerin iç içe kullanılabilmelerinden dolayı gerekli eşleşmelerin sağlanamamasının önüne geçilmesi amaçlanmıştır. İçerik ile ilgili değerlendirmelerde bu değişken kullanılmıştır.

- “*get_css_content*” fonksiyonu (EK B.1)

Çoklu stil sayfası tanımlamalarına göre, aynı selector'e (seçiciye) ait property (özellik) nerede son olarak tanımlanmışsa, ilgili selector'e ait property değeri bu olmaktadır [57]. Örneğin bir web sayfasının harici bir CSS dosyasında `h3 {color:yellow}` tanımlaması ile dahili CSS kodunda `h3 {color:blue}` tanımlaması yapıldığında, dahili CSS kodu, HTML kodunda sonradan yer alıyorsa, `h3` selector'üne ait renk değeri “blue” olmaktadır. Öte yandan CSSTidy'nin, anlatılan bu durumu optimize işlemlerinde dikkate aldığı görülmektedir [54]. Bu bağlamda değerlendirilen web sayfasında kullanılan harici ve dahili CSS kodlarının tanımlanma sırasına göre birleştirilmesi ve ardından CSSTidy ile optimize edilmesi sonucu, o web sayfasına ait tek bir stil sayfasının ortaya çıkarılabileceği düşünülmüştür.

Bu amaçla, bu fonksiyonda öncelikle web sayfasının harici ve dahili stil sayfaları elde edilmekte, ardından çoklu stil sayfaları durumu dikkate alınarak bu stil sayfaları birleştirilmektedir. Ancak bu stil sayfalarında import edilen başka stil sayfaları da olabileceğinden, web sayfasında yer alan bu stil sayfalarından her biri,

“check_import” fonksiyonu tarafından kontrol edilip, dönen yeni stil sayfası doğrultusunda birleştirilmiştir. Öte yandan harici ve dahili stil sayfalarının medya tiplerinin *screen* veya *all* değerine sahip olmaları dikkate alınmıştır.

Birleştirilme işlemlerinin tamamlanmasıyla, değerlendirilecek web sayfasının stil sayfası ortaya çıkarılmış ve $\$this->extintcsscont$ değişkenine aktarılmıştır.

Bununla birlikte, satırlarda (inline) tanımlanan stillerin analizi için *style* nitelik değerleri “parse_html ” fonksiyonu ile elde edilerek, bu değerlerden hafızada bir stil sayfası oluşturulmuştur ($\$this->inlinecsscont$).

HTML kodlarında tanımlanmış renkleri belirlemek için de benzer yönteme başvurulmuştur. “parse_html ” fonksiyonu ile etiketlerin *alink*, *vlink*, *link*, *text*, *color* ve *bgcolor* niteliklerine atanan renk değerleri elde edilerek, bunların analizi için hafızada başka bir stil sayfası oluşturulmuştur ($\$this->htmltmpcsscont$).

- “get_images ” fonksiyonu (EK B.1)

Bu fonksiyon web sayfasında kullanılan resimlerin her birini “create_folders” fonksiyonu ile oluşturulmuş ilgili klasöre kaydetmektedir.

- “get_w3c_html_content” fonksiyonu (EK B.1)

Bu fonksiyon web sayfasının W3C HTML geçerlik testini W3C’nin sitesinde gerçekleştirmekte, testin sonuç sayfasını ise “create_folders” fonksiyonu ile oluşturulmuş ilgili klasöre kaydetmektedir.

- “get_w3c_css_content” fonksiyonu (EK B.1)

Bu fonksiyon web sayfasının W3C CSS geçerlik testini W3C’nin sitesinde gerçekleştirmekte, testin sonuç sayfasını ise “create_folders” fonksiyonu ile oluşturulmuş ilgili klasöre kaydetmektedir.

- “*get_label_for*” fonksiyonu (EK B.1)

Bu fonksiyon web sayfasında kullanılan form elemanlarıyla ilişkilendirilmiş herhangi bir *label* etiketi olup olmadığını kontrol etmektedir.

- “*get_label_accesskey*” fonksiyonu (EK B.1)

Bu fonksiyon web sayfasında kullanılan form elemanlarıyla ilişkilendirilmiş *label* etiketlerinde *accesskey* niteliğinin kullanılıp kullanılmadığını kontrol etmektedir.

- “*get_att*” fonksiyonu (EK B.1)

Beş parametresi olan bu fonksiyon ile bir etikete ait niteliğin kullanılıp kullanılmadığı belirlenmektedir (örneğin *table* etiketinde *bgcolor* niteliğinin kullanılıp kullanılmaması gibi). Öte yandan etikete ait bir nitelik/özellik HTML kodunda veya stil sayfasında tanımlanabildiği için, bu fonksiyonda parse edilmiş hem CSS, hem de HTML kodu taranmaktadır. Parametreler sırasıyla şu şekildedir:

- Parse edilmiş CSS ($\$this->extintcsscont->css[41]$)
- Niteliğin bulunduğu etiket
- Etiket niteliklerinin bulunduğu dizi
- Niteliğin CSS yazımı (property)
- Niteliğin HTML yazımı (attribute)

Fonksiyon çalıştırıldığında öncelikle niteliğin, HTML kodunda veya satırda (inline) yapılmış bir stil tanımı varsa burada kullanılıp kullanılmadığına bakılmaktadır. Eğer kullanılmamışsa o zaman web sayfasının stil sayfası taranmaktadır ($\$this->extintcsscont->css[41]$). Stil sayfası taramalarında elementlerin özgüllük değerleri (specificity) dikkate alınamamıştır. *, tag, tag#id, .class, tag.class şeklinde tanımlanmış selector’ler kontrol edilmiştir.

- “*css_font_size*” fonksiyonu (EK B.1)

Üç parametre ile çalışan bu fonksiyon, ilk parametre ile kendisine verilen parse edilmiş CSS kodunda tanımlanan yazı tipi boyutu değerlerinin, mutlak (absolute) bir yazı tipi boyutu değeri olup olmadığını kontrol etmektedir. İkinci ve üçüncü parametreler çıkış parametresidir. İkinci parametrede kullanılan yazı tipi boyutlarının sayısı, üçüncü parametre ile bunlardan mutlak büyüklüğe sahip olanların sayısı döndürülmektedir. Döndürülen değerlerde selector sayıları baz alınmıştır.

- “*css_color*” fonksiyonu (EK B.1)

Bu fonksiyonun çalışması “*css_font_size*” fonksiyonuna benzemektedir. Fonksiyon, ilk parametre ile kendisine verilen parse edilmiş CSS kodunda tanımlanan renklerin, 216 renk paletinde olup olmadığı kontrol etmektedir. İkinci parametre ile kullanılan renklerin sayısı, son parametre ile bu renklerden 216 renk paletinde olanların sayısı döndürülmektedir.

Kodlama yapılırken geçerli ve geçersiz renkler, 216 renk paleti ile CSSTidy class’ı incelenerek belirlenmiştir. 216 renk paletinde yer alan renkler 0, 3, 6, 9, c ve f değerlerinin ikili ikili yan yana gelmesiyle oluşmaktadır [58]. Öte yandan CSSTidy, CSS kodlarını optimize ederek parse ettiği için dolayı, parse edilmiş CSS’de bazı renklerin adları, bazı renklerin ise onaltılık sayı sistemi kodları yazmaktadır. CSSTidy class’ı incelenerek optimize işleminde; renklerden hangilerinin adları, hangilerinin kodlarının yazıldığı belirlenmiş, böylece geçerli ve geçersiz renklerin tanımlamaları fonksiyonda yapılmıştır.

- “*round2*” fonksiyonu (EK B.1)

Private fonksiyonların sonuncusu olan bu fonksiyon, verilen değeri virgülden sonra 2 basamağa yuvarlamaktadır.

3.2.2.3 Public (Genel) Fonksiyonlar

Bu fonksiyonlar, web sayfasının değerlendirme işlemlerini, private fonksiyonların da yardımıyla gerçekleştirerek kullanıcıya birer dizi (array) döndürmektedir. Her dizide web sayfasının ilgili değerlendirme kriterlerinden aldığı sonuçlar (değerler, yüzdeler, başarılı, başarısız vb durumu) yer almaktadır.

Class'ta toplam olarak 22 tane public fonksiyon bulunmaktadır. Aşağıda bu fonksiyonların görevleri ve döndürdükleri dizide yer alan bilgiler açıklanmıştır. Dönen değerlere ilişkin açıklamalar "Dönen Değerler ve Sonuç Fonksiyonları" kısmında yer almaktadır.

- *"arr_tidy_gen" fonksiyonu (EK B.1)*

Bu fonksiyon Tidy ile parse edilmiş web sayfasında tespit edilen Tidy hata ve uyarı sayısı ile bu hata ve uyarıların neler olduğunu döndürmektedir.

- *"arr_tidy_acc" fonksiyonu (EK B.1)*

Web sayfasının Tidy erişilebilirlik testini gerçekleştiren bu fonksiyona parametre olarak erişilebilirlik testinin hangi seviyeye göre yapılacağı belirtilmektedir. İlgili erişilebilirlik testi uyarı sayısını ve bu uyarıların neler olduğunu geri döndürmektedir.

- *"arr_cookie " fonksiyonu (EK B.1)*

Bu fonksiyon web sayfasındaki header (başlık) bilgilerinde çerez (cookie) olup olmadığı bilgisini geri döndürmektedir.

- “*arr_w3*” fonksiyonu (EK B.1)

Bu fonksiyon, web sayfasının W3C HTML ve W3C CSS geçerlilik testi hata sayısı ile sayfada doküman tip bildirimini yapıp yapılmadığını geri döndürmektedir.

- “*arr_comment*” fonksiyonu (EK B.1)

Web sayfasında yer alan yorumları ve bu yorumların sayısını döndürmektedir.

- “*arr_title*” fonksiyonu (EK B.1)

Web sayfasında kullanılan sayfa başlığını ve bu sayfa başlığının uzunluk değerini geri döndürmektedir.

- “*arr_script*” fonksiyonu (EK B.1)

Web sayfasında kullanılan *script* ve *noscript* sayılarını döndürmektedir.

- “*arr_applet*” fonksiyonu (EK B.1)

Web sayfasında kullanılan *applet* sayısını geri döndürmektedir.

- “*arr_metatag*” fonksiyonu (EK B.1)

Web sayfasında tanımlanmış *meta* etiketlerini ve bunların sayısını geri döndürmektedir.

- “*arr_anchor*” fonksiyonu (EK B.1)

Bu fonksiyon, web sayfasında kullanılan linklerle ilgili bilgileri geri döndürmektedir. Dönen değerler şu şekildedir:

- Link sayısı
- Resim içermeyen linklerin yüzdesi
- Resim içeren linklerde, resimlerde kullanılan *alt* niteliklerinin yüzdesi
- *tabindex* niteliği kullanılan linklerin yüzdesi
- *accesskey* niteliği kullanılan linklerin yüzdesi
- Ana menü kullanımı
- HTML menü kullanımı
- Linklerin listesi

Ana menü kullanımında, link sayısı, *object* sayısı, *embed* sayısı veya *script* sayısının 0'dan büyük olması durumu, HTML menü kullanımında ise link sayısının 0'dan büyük olması ile birlikte, *object*, *embed* ve *script* sayılarının 0 olması durumu esas alınmıştır.

- “*arr_body*” fonksiyonu (EK B.1)

Bu fonksiyon, *body* etiketi ve bu etiketin *link*, *alink*, *vlink* ve *text* niteliklerinin kullanılıp kullanılmadığı bilgisini geri döndürmektedir.

- “*arr_form*” fonksiyonu (EK B.1)

Bu fonksiyon web sayfasında kullanılan formlar ve form elemanları ile ilgili çeşitli bilgiler vermektedir. Dönen dizide şunlar yer almaktadır:

- HTML form sayısı
- *object* ve *embed* sayısı
- HTML form kullanım yüzdesi
- Kullanılan *label* etiketlerinin yüzdesi
- Kullanılan *tabindex* niteliklerinin yüzdesi
- *accesskey* niteliği kullanılan *input*, *select*, *textarea* ve *button* nesnelerinin yüzdeleri

HTML form kullanım yüzdesi hesaplanırken, *object/embed* sayısı göz önünde bulundurulmuştur. Bu nedenle *object/embed* web sayfasında yer alıyorsa dönen değer NULL olmaktadır.

label etiketlerinin yüzdesi hesaplanırken *input*, *select* ve *textarea* nesnelere göre değerlendirme yapılmıştır. *input* nesnelinde *type* nitelik değerinin; *text*, *password*, *checkbox*, *radio* veya *file* olması dikkate alınmış, değeri *hidden* olan *input* nesneleri ise değerlendirme dışı bırakılmıştır.

tabindex niteliklerinin yüzdesi hesaplanırken *input*, *select*, *textarea* ve *button* nesneleri değerlendirilmeye alınmış, *input* nesnelinde *type* nitelik değeri *hidden* olanlar benzer şekilde değerlendirme dışı bırakılmıştır.

Öte yandan *accesskey* niteliklerinin kullanım yüzdeleri *input*, *select*, *textarea* ve *button* nesneleri için ayrı ayrı hesaplanmıştır. Yine *input* nesnelinde *type* nitelik değeri *hidden* olanlar değerlendirme işleminde dikkate alınmamıştır.

- “*arr_font*” fonksiyonu (EK B.1)

Bu fonksiyon ile web sayfasında *font* etiketlerinin kullanılıp kullanılmadığı bilgisi geri döndürülmektedir.

- “*arr_marquee*” fonksiyonu (EK B.1)

“*arr_font*” fonksiyonuna benzer olan bu fonksiyon ile web sayfasında kayan yazıların kullanılıp kullanılmadığı bilgisi geri döndürülmektedir.

- “*arr_map*” fonksiyonu (EK B.1)

Bu fonksiyon, web sayfasında kullanılan resim haritaları ile ilgili çeşitli bilgileri geri döndürmektedir. *area* yöntemiyle oluşturulan resim haritalarına göre kodlama yapılan bu fonksiyonda dönen değerler arasında, resim haritası sayısı, bu

haritalardaki alan sayısı, alanlarda *alt*, *tabindex* ve *accesskey* niteliklerinin kullanım yüzdeleri yer almaktadır.

- “*arr_img*” fonksiyonu (EK B.1)

Bu fonksiyon web sayfasındaki resimler ile ilgili bilgiler vermektedir. Dönen değerler şunlardır:

- Toplam resim sayısı
- İndirilen resim sayısı
- Uzantısı JPEG/GIF olan resimlerin yüzdesi
- Boyutu en fazla 30 KB olan resimlerin yüzdesi
- *width* ve *height* nitelikleri kullanılmış resimlerin yüzdesi
- *alt* niteliklerine sahip resimlerin yüzdesi
- *alt* nitelik değeri en fazla 100 karakter olan *alt* nitelikleri yüzdesi
- *alt* nitelik değeri son karakteri nokta veya boşluk olan *alt* nitelikleri yüzdesi
- *alt* nitelik değeri son karakteri nokta ve boşluk olan *alt* nitelikleri yüzdesi
- Sayfada logunun olup olmadığı
- Logo var ise logunun GIF formatında olup olmadığı
- Logo var ise logunun üzerindeki metinlerin rahatça okunup okunamadığı (NULL değeri döndürmektedir)
- Logo var ise logunun statik olup olmadığı

Web sayfasında bir logunun bulunup bulunmadığı, indirilen resimlerin taranmasıyla belirlenmiştir. Bunun için ilgili dil dosyasında (EK B.3) yer alan logo ile ilgili düzenli ifadenin ($\$locale['100']$), indirilen her resim dosyası adıyla eşleşip eşleşmediği kontrol edilmiştir. Eşleşme gerçekleşmişse, bu resim dosyalarının türünün GIF formatında olup olmadığına bakılmıştır. GIF formatında olan resim dosyası içinse logunun statik olmayabileceği belirtilmiştir.

- “*arr_table*” fonksiyonu (EK B.1)

Web sayfasında yer alan tablolarla ilgili bilgi veren bu fonksiyon; tablo sayısı, tablo sonu etiketlerinin kullanım yüzdesi ile tablo arka plan rengi ve resmi kullanılmayan tabloların yüzdelik değerlerini geri döndürmektedir.

- “*arr_color*” fonksiyonu (EK B.1)

Bu fonksiyon, web sayfasında kullanılan renklerden; 216 renk paleti içinde olanların yüzdelerini geri döndürmektedir. Fonksiyon tek parametreyle çalışmaktadır. “1” değişkeni gönderildiğinde harici ve dahili stil sayfalarından oluşturulan stil sayfasında ($\$this->extintesscont$) kullanılan renklerin yüzdesini, “2” değişkeni gönderildiğinde satırdaki (inline) stillerden oluşturulmuş stil sayfasındaki ($\$this->inlinecsscont$) renklerin yüzdesini, “3” değişkeni gönderildiğinde ise HTML niteliklerinde tanımlanmış renklerden oluşturulmuş stil sayfasındaki ($\$this->htmltmpcsscont$) renklerin yüzdesini döndürmektedir.

- “*arr_font_size*” fonksiyonu (EK B.1)

“*arr_color*” fonksiyonu ile aynı yapıda olan bu fonksiyon web sayfasında kullanılan yazı tipi büyüklüklerinden mutlak değere sahip olanların yüzdesini vermektedir. “1” değişkeni gönderildiğinde harici ve dahili stil sayfalarından oluşturulan stil sayfasında ($\$this->extintesscont$) kullanılan, “2” değişkeni gönderildiğinde ise satırdaki (inline) stillerden oluşturulmuş stil sayfasındaki ($\$this->inlinecsscont$) yazı tipi büyüklüklerinin geçerlilik yüzdelerini vermektedir.

- “*arr_page_size*” fonksiyonu (EK B.1)

Bu fonksiyon, web sayfasının resimli ve resimsiz boyutunu döndürmektedir.

- “arr_content” fonksiyonu (EK B.1)

Bu fonksiyon içerik bölümünde belirlenen kriterler ile ilgili değerlendirmeleri gerçekleştirmektedir. Daha önceden de belirtildiği gibi içerik ile ilgili kriterlerde Türkçe veya İngilizce olarak değerlendirme yapılabilmesi amaçlanmıştır. Bu nedenle içinde içerik testleri için yazılmış düzenli ifade desenlerinin bulunduğu iki adet dil dosyası oluşturulmuştur (EK B.3). Bu fonksiyon, değerlendirme diline göre bu dil dosyalarında tanımlanmış desenlerle, web sayfası içeriğinin eşleşip eşleşmemesi durumunu dikkate alarak içerik kriterleri ile ilgili değerlendirme sonuçlarını geri döndürülmektedir. Dönen dizide şunlar yer almaktadır:

- Logonun bulunup bulunmadığı ve ilgili desen (\$locale['100']) ile eşleşen resim dosyaları
- Sayfa başlığının varlığına göre kurum adının bulunup bulunmadığı ve sayfa başlığı
- Sitenin amacının sayfada yer alıp yer almadığı ve ilgili desen (\$locale['104']) ile eşleşen sözcükler
- Site içi bağlantıların var olup olmadığı ve sayfada bulunan linkler
- Duyuruların/haberlerin sayfada yer alıp almadığı ve ilgili desen (\$locale['105']) ile eşleşen sözcükler
- Arama mekanizmasının olup olmadığı ve ilgili desen (\$locale['106']) ile eşleşen sözcükler
- Yardım sayfalarına bağlantıların olup olmadığı ve ilgili desen (\$locale['101']) ile eşleşen sözcükler
- Şikayet/geri bildirim sayfalarına bağlantıların olup olmadığı ve ilgili desen (\$locale['103']) ile eşleşen sözcükler
- Adres bilgilerinin sayfada yer alıp yer almadığı ve ilgili desen (\$locale['107']) ile eşleşen sözcükler
- Telefon bilgisinin sayfada yer alıp yer almadığı ve bulunan telefonlara ait liste
- E-posta bilgisinin bulunup bulunmadığı ve bulunan e-posta adresleri
- Var olan e-postaların uygun olup olmadığı ve ilgili desen (\$locale['109']) ile eşleşen e-posta adreslerinin listesi

- “*arr_accessibility*” fonksiyonu (EK B.1)

Bu fonksiyon erişilebilirlikle ilgili web sayfasının çeşitli tarayıcılarda ve değişik çözünürlüklerde görüntülenip görüntülenmediği ile ilgili bilgi vermek için yazılmıştır. Sadece Lynx tarayıcısı için; sayfada *object*, *embed* ve resim haritası bulunup bulunmaması ile resimlerde *alt* niteliklerinin kullanılıp kullanılmaması durumuna göre bir öneri getirilmiştir. Bunun haricinde, bu fonksiyondan dönen değerlerin hepsi NULL değeri içermektedir.

3.2.2.4 Dönen Değerler ve Sonuç Fonksiyonları

Public fonksiyonlar tarafından döndürülen değerler; sayısal bir değer olabileceği gibi, bazen de NULL olabilmektedir. NULL değerler, genel olarak şu 2 durumda döndürülmektedir:

- Bir üst kritere sahip olan kriterlerde, üst kritere ilişkin bir değer olmaması durumunda (örneğin *img* etiketlerinde *alt* niteliklerinin kullanımının hesaplanmasında sayfada hiç resim olmaması)
- El yordamı kullanılarak değerlendirilmesi gereken kriterlerde (örneğin çeşitli tarayıcılarda web sayfasının aynı içeriği verebilecek şekilde tasarlanması gibi)

Sayısal değerlerden birçoğu ise yüzdesel bir oranı ifade etmektedir. Örneğin *img* etiketlerinde *alt* niteliklerinin kullanımı gibi bir bütünün parçası olan kriterlerin uygunluk düzeylerinin belirlenmesinde, yüzdesel değerler döndürülmektedir. Yüzdesel değer döndüren bu şekildeki kriterler için, dizilerin anahtarları (key) adlandırılırken, anahtar adının sonuna “per” ifadesi getirilmiştir.

Bununla birlikte bazı sayısal değerler, ilgili kriterin değerlendirilen web sayfasında sağlanıp sağlanmadığını ifade etmektedir. Bunun için her olumlu durumda “1”, her olumsuz durumda ise “0” değeri döndürülmektedir. Örneğin bir web sayfasında doküman tip bildirimini yapılması durumunda “1”, aksi takdirde

“0” deęeri döndürülecektir. El yordamı gerektiren kriterlerde de aynı durum geçerlidir. Bu tür sayısal deęer döndüren dizi anahtarları içinse, genellikle “value” ifadesi anahtar isimlerinde geçmektedir.

Öte yandan web sayfasının ilgili kriterden almış olduęu deęere göre deęerlendirme sonucunu belirlemek için, daha önceden de belirtildięi gibi test_functions.php adlı bir PHP betięi yazılmıştır (EK B.2). Public fonksiyonlar tarafından sıklıkla kullanılan bu betikteki fonksiyonlar, gönderilen deęer(ler)e ve kriterin özellięine göre ilgili dil dosyasındaki (EK B.3) sabitlerde tanımlı 6 çeşit sonuç döndürebilmektedir:

- Başarılı
- Başarısız
- Olabilir
- Olmayabilir
- Belirlenemedi
- Yok/deęerlendirme dıőı

“Başarılı” ve “Başarısız” sonuçları genel olarak el yordamı gerektirmeden deęerlendirme yapılabilen; W3C HTML geçerlilik testi hata sayısı, *img* etiketlerinde *alt* niteliklerinin kullanımı gibi kriterlerin sonucunu bildirmekte iken, “Olabilir” ve “Olmayabilir” sonuçları özellikle içerik ile ilgili el yordamı gerektiren kriterlerin durumu hakkında bilgi vermektedir. “Belirlenemedi” ve “Yok/deęerlendirme dıőı” sonuçları ise genellikle kriterin özellięine göre NULL deęerlerinde döndürülmektedir. Örneęin *img* etiketleri *alt* niteliklerinin kullanımında, sayfada resim bulunmaması halinde bu kritere iliőkin “Yok/deęerlendirme dıőı” sonucu döndürülecekken, HTML form kullanımında; sayfada *object/embed* bulunması durumunda bu kritere iliőkin “Belirlenemedi” sonucu döndürülecektir. Öte yandan public fonksiyonlarda, deęerlendirme sonuçlarını döndüren dizi anahtarları için, anahtar adlarında “test” ifadesi kullanılmıştır.

test_functions.php betiğinde, public fonksiyonlardan gönderilen değer, kriterin özelliği ile döndürülecek sonuçlar dikkate alınarak yazılmış 8 adet fonksiyon bulunmaktadır. Bu fonksiyonlar şu şekildedir:

- *“test_eq” fonksiyonu (EK B.2)*

Class içinde kriterden alınan değer ile kriterden beklenen değerın eşitliğini karşılaştırmak için kullanılan bu fonksiyon iki parametre ile çalışmaktadır. İlk parametre ile kriterin değeri (\$val1), ikinci parametre ile karşılaştırılacak değer (\$val2) gönderilmektedir. Fonksiyon, kritere ait değer NULL ise “Yok/Değerlendirme Dışı”, iki değer birbirine eşit ise “Başarılı”, değerler birbirine eşit değil ise “Başarısız” sonucunu döndürmektedir. Örneğin, web sayfasının W3C HTML geçerlilik testinden alınan hata sayısına göre başarılı, başarısız vb durumunu belirlemek için class içinde kullanılan kod: test_eq(\$htmlerrors, 0) şeklindedir.

- *“test_lt” fonksiyonu (EK B.2)*

İki parametreyle çalışan fonksiyon, ilk parametre ile kendisine gönderilen kritere ait değerın, ikinci parametre ile gönderilen diğer bir değerden küçük olup olmadığını karşılaştırmaktadır. Fonksiyon, kritere ait değer NULL ise “Yok/Değerlendirme Dışı”, kriterden alınan değer, diğer değerden küçükse “Başarılı”, aksi takdirde “Başarısız” sonucunu döndürmektedir.

- *“test_gt” fonksiyonu (EK B.2)*

“test_lt” fonksiyonuna benzer olan bu fonksiyon kriter değerının, verilen diğer bir değerden büyük olup olmadığını karşılaştırmaktadır. Fonksiyon, kritere ait değer NULL ise “Yok/Değerlendirme Dışı”, kriterden alınan değer, diğer değerden büyükse “Başarılı”, aksi takdirde “Başarısız” sonucunu döndürmektedir

- *“test_poss” fonksiyonu (EK B.2)*

Tek parametre ile çalışan bu fonksiyon özellikle içerik ile ilgili kriterler gibi el yordamı gerektiren değerlendirme sonuçlarını döndürmektedir. Fonksiyon, kritere ait değer NULL ise “Belirlenemedi”, kriterden alınan değer “1” ise “Olabilir”, aksi takdirde “Olmayabilir” sonucunu döndürmektedir.

- *“test_poss_rel” fonksiyonu (EK B.2)*

İki parametre ile çalışan bu fonksiyon, el yordamı gerektiren ve bir üst kritere sahip olan kriterlerin değerlendirme sonuçlarını döndürmektedir. İlk parametre ile kriterin değeri, ikinci parametre ile üst kriterin değeri fonksiyona gönderilmektedir. Üst kriter bulunmuyorsa “Yok/Değerlendirme Dışı” sonucunu döndüren fonksiyon, üst kriterin sağlanması durumunda “test_poss” fonksiyonunu çalıştırmaktadır.

- *“test_approp” fonksiyonu (EK B.2)*

İki parametreyle çalışan fonksiyon, sayfa başlığının uygunluğu, resim haritalarının kullanım uygunluğu gibi sayfada yer aldıktan sonra el yordamı gerektiren kriterlerin değerlendirme sonuçlarını döndürmek için yazılmıştır. İlk parametre ile öncelikle kriterin sayfada var olup olmadığına ilişkin bir değer (sayfa başlığının karakter sayısı, resim haritaları sayısı gibi) gönderilmekte, ikinci parametre ile ise kullanım uygunluğunun durumu gönderilmektedir. Ancak kullanım uygunluğu için el yordamı gerektiğinden ikinci parametre kriterin bu yöntemle değerlendirmesi yapıldıktan sonra gönderilebilmektedir. Yöntem kısmında da daha önceden belirtildiği üzere bu gibi kriterlere ait değerler el yordamı değerlendirmesinden sonra geliştirilen araçla veritabanına kaydedilebilmektedir. Geliştiren araç da, ileride değinileceği gibi bu betikte yer alan fonksiyonları kullanmakta, dolayısıyla veritabanındaki değerlere ilişkin kriterlerin değerlendirme sonuçları görüntülenebilmektedir. Bu bağlamda söz konusu bu fonksiyon class içinden ilk parametre ile, araçta ise iki parametre ile birden çalıştırılmaktadır.

Fonksiyon öncelikle uygunluk değerinin olup olmadığına (\$val2) bakmaktadır. Değer “1” ise “Başarılı”, “0” ise “Başarısız” sonucunu döndürmektedir. Eğer uygunluk değeri yok ise bu kez ilk parametre (\$val1) ile gönderilen kritere ilişkin bir verinin olup olmadığına bakılmaktadır. Bu durumda \$val1 sıfırdan büyük ise “Belirlenemedi”, aksi takdirde “Yok/Değerlendirme Dışı” sonucu döndürülmektedir.

- “*test_other*” fonksiyonu (EK B.2)

Bu fonksiyon el yordamı gerektiren, ancak değerlendirme sonucuna ilişkin; içerik ile ilgili kriterlerde olduğu gibi “Olabilir” veya “Olmayabilir” gibi bir öneri getirilemeyen kriterler için “arr_accessibility” fonksiyonu içerisinden kullanılmıştır. Tek parametre ile çalışan fonksiyon, NULL değerlerinde “Belirlenemedi”, “1” değerinde “Başarılı”, aksi takdirde “Başarısız” değerini döndürmektedir.

- “*test_form*” fonksiyonu (EK B.2)

HTML form kullanımı değerlendirme sonucunu belirlemek için yazılmış bu fonksiyonda dört parametre bulunmaktadır. İlk parametre form sayısı, ikinci ve üçüncü parametreler sırasıyla *object* ve *embed* sayısı, son parametre ise HTML form kullanımı yüzdelik oranıdır. Fonksiyon “test_approp” fonksiyonuna benzer olarak çalışmakta, sayfada *object/embed* bulunması halinde el yordamı değerlendirmesi gerektiğinden son parametreye class içinden NULL değeri gönderilmektedir.

Fonksiyon, var ise öncelikle sonra parametreye bakmakta ve oran “100” olduğunda “Başarılı”, aksi takdirde “Başarısız” sonucunu döndürmektedir. Son parametre NULL değer içerdiğinde ise fonksiyon; sayfada *object* ve *embed* nesnelere sahip olmadıkları durumlarda, HTML form sayısı 0’den büyükse “Başarılı”, değilse “Yok/Değerlendirme Dışı” sonucunu, sayfada *object* ve *embed* nesnelere sahip olduğu her durumda el yordamı gerektiği için “Belirlenemedi” sonucunu döndürmektedir.

3.2.3 Class'ın Kullanımı

Class'ın kullanımı için öncelikle class, PHP betiğine dahil edilmeli ve ardından “webpage” nesnesi oluşturularak çalıştırılmalıdır. Bu işlem constructor fonksiyonda da açıklandığı gibi; ilk parametrede değerlendirilecek web sayfasının adresi, ikinci parametre ile değerlendirme dili, üçüncü ve dördüncü parametrelerde ise sırasıyla W3C HTML ve W3C CSS geçerlilik testlerinin gerçekleştirilip gerçekleştirilmemesi belirtilerek yapılmaktadır. Ardından istenilen bir public fonksiyon kullanılarak, web sayfasının bu public fonksiyon tarafından gerçekleştirilmiş ilgili değerlendirme sonuçları elde edilebilir. Örnek bir kullanım şu şekildedir:

```
<?php
    require_once 'class.webpage.php';
    $wpage = new webpage ('http://www.balikesir.edu.tr', 'tr', true, true);
    $w3 = $wpage->arr_w3();
    print_r($w3);
?>
```

Bu örnek kodda öncelikle `require_once 'class.webpage.php';` satırında web sayfası değerlendirme class'ı PHP betiğine yüklenmekte, `$wpage = new webpage ('http://www.balikesir.edu.tr', 'tr', true, true);` satırında sırasıyla değerlendirilecek web sayfası, değerlendirme dili, W3C HTML ve W3C CSS geçerlilik testlerinin gerçekleştirilmesi belirtilerek “webpage” nesnesi oluşturulmaktadır. Ardından “arr_w3” fonksiyonu kullanılarak, değerlendirilen web sayfasının W3C testleri ile ilgili sonuçları elde edilmekte, public fonksiyonlar birer dizi döndürdüğü için print_r fonksiyonu ile de bu sonuçlar ekrana bastırılmaktadır. Çıktı şu şekildedir:

```
Array
(
    [htmlerrors] => 103
    [csserrors] => 6
    [doctypevalue] => 0
    [htmltest] => <img src='images/failed.gif' alt='Başarısız ' width='16' height='16'>
```

```
[csstest] => <img src='images/failed.gif' alt='Başarısız ' width='16' height='16'>
[doctypetest] => <img src='images/failed.gif' alt='Başarısız ' width='16' height='16'>
)
```

Çıktı incelendiğinde, değerlendirilen web sayfasının W3C HTML geçerlilik testinde 103 hatasının, W3C CSS geçerlilik testinde 6 hatasının olduğu ve doctypevalue değerinden ise sayfada doküman tip bildiriminin yapılmadığı görülmektedir. Bu değerler, önceden de belirtildiği üzere; “arr_w3” fonksiyonundan test_functions.php betiğindeki ilgili sonuç fonksiyonuna gönderilmiş ve sayfanın bu üç kriterden başarısız olduğu sonucu döndürülmüştür.

3.3 Web Sayfası Değerlendirme Aracı

Çekirdeğini web sayfası değerlendirme class'ının oluşturduğu, bu çalışmada sözü edilen araç, yöntem kısmında belirtilen noktalar doğrultusunda geliştirilmiş ve aracın çalışan bir kopyası “<http://webtest.balikesir.edu.tr>” adresine konularak EK D’de dizin ve dosya yapısı çıkarılmıştır.

Araçta, değerlendirilen web sayfasının değerlendirme sonuçlarının (kriterlerden alınan değerlerin) kaydedilmesi için veritabanı olarak MySQL, veritabanı işlemleri için de literatür taramalarında karşılan bir veritabanı class'ı [59] kullanılmıştır. Öte yandan kriterlere ilişkin veritabanında tutulan değerlerin; “başarılı”, “başarısız” vb sonuçlarının gösterilmesinde önceden de belirtildiği gibi test_functions.php betiği araç tarafından da kullanılmıştır.

Web sayfası değerlendirme class'ının içerik ile ilgili kriterlerde İngilizce değerlendirme yapabilecek şekilde kodlanması araca da yansıtılmış; hem ara yüz dili hem de değerlendirme kriterlerinin kısa açıklamaları için Türkçe ve İngilizce dil dosyaları oluşturularak araçta kullanılmıştır.

Bununla birlikte yönetici modunda, el yordamı yöntemi değerlendirme sonuçlarının veritabanına girilebilmesinin yanı sıra değerlendirilen web sayfalarının

listesi görülebilmekte, değerlendirilmiş web sayfaları ve bu sayfaların verileri silinebilmekte, daha ayrıntılı filtreleme vb işlemler yapılabilmektedir.

FreeBSD 6.0 üstünde MySQL 5.0.51a ve PHP 5.2.5 yüklü bilgisayarda çalışan bu aracın; kaynak kodu, yukarıda değinilen dosyalar, veritabanı tabloları SQL kodları ile aracın ekran görüntüleri ve özellikleri EK C'de, akış diyagramı EK E'de verilmiştir.

Geliştirilen söz konusu bu araç ile 35 kamu kurum ve kuruluşu internet sitesinin ana sayfası değerlendirilerek sonuçlar bulgular ve yorumlar kısmında sunulmuştur.

4. BULGULAR VE YORUMLAR

EK F’de listesi yer alan 35 kamu kurum ve kuruluşu internet sitesinin ana sayfası geliştirilen araç ile değerlendirilmiş, el yordamı gerektiren değerlendirmeler yapılarak veriler araca kaydedilmiş ve sonuçlar; “Değerlendirme İçin Belirlenen Kriterler” bölümünde de belirtildiği gibi “İçerik (Asgari)”, “Tasarım (Erişilebilirlik)”, “Tasarım (Teknik Özellikler)” ve “Genel Görünüm” olmak üzere dört ana başlıkta toplanmıştır.

4.1 İçerik (Asgari)

Bu bölümdeki değerlendirmeler “Genel” ve “Kuruma Erişim Bilgileri” olmak üzere iki kısımda ele alınmıştır. Tablo 4.1 ve Tablo 4.2 incelendiğinde, değerlendirilen web sayfalarının içerik ile ilgili olarak; sitenin amacı, yardım sayfalarına bağlantı ve sıkça sorulan sorular sayfalarına bağlantı kriterlerine sahip olma oranlarının oldukça düşük olduğu göze çarpmaktadır. Bunlarında dışında değerlendirilen web sayfalarının büyük bölümünün asgari içerik ile ilgili kriterleri genel olarak sağladıkları görülmektedir.

Tablo 4.1 İçerik / Genel Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama (%)	(n)
Kurum Logosu*	94	35
Kurumun Açık Adı*	100	35
Sitenin Amacı*	0	35
Site içi Bağlantılar*	100	35
Duyurular/Haberler*	100	35
Arama Mekanizması*	66	35

Tablo 4.1'in devamı

Yardım Sayfalarına Bağlantı*	3	35
S.S.S. Sayfalarına Bağlantı*	20	35
Şikâyet/Geri Bildirim Sayfalarına Bağlantı*	86	35

* El yordamı ile değerlendirilenler

Tablo 4.2 Kuruma Erişim Bilgileri Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama (%)	(n)
Adres*	83	35
Telefon*	97	35
E-posta*	83	35
E-posta Uygunluğu*	90	29

* El yordamı ile değerlendirilenler

4.2 Tasarım (Erişilebilirlik)

Bu bölümde web sayfaları; Internet Explorer, Mozilla Firefox, Konqueror ve Lynx olmak üzere dört farklı tarayıcı ile üç değişik çözünürlükte görüntülenerek değerlendirilmiştir. Tarayıcılarla ilgili değerlendirmelerde; tarayıcılar ve bunların kullanıldığı bilgisayarlardaki işletim sistemleri;

- Internet Explorer 6.0 – Windows XP Prof. 2002 SP2
- Mozilla Firefox 2.0 – Pardus 2007.3
- Konqueror 3.5 – Pardus 2007.3
- Lynx 2.8 – FreeBSD 6.0

şeklinde olup değerlendirme sonuçları Tablo 4.3'dedir.

Tablo 4.3 Tarayıcı Tipi Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama (%)	(n)
Internet Explorer*	100	35
Mozilla Firefox*	83	35
Konqueror*	77	35
Lynx*	6	35

* El yordamı ile değerlendirilenler

Tablo 4.3'den de görüldüğü gibi değerlendirilen sayfaların tamamı Internet Explorer'da sorunsuz olarak görüntülenebilmiş ancak bu oran Mozilla Firefox ve Konqueror tarayıcılarında düşmüştür. Bu tarayıcılarda sorun çıkaran sayfalarda; kodlama hatalarından veya çeşitli eklentilerden dolayı, sayfalarda kaymalar meydana gelmiş veya bazı içerikler görüntülenememiştir. Öte yandan bu problemle en çok; web sayfalarının metin tabanlı bir tarayıcı olan Lynx'de değerlendirilmesi sırasında karşılaşmıştır. Çoğu sitede; resim *alt* niteliklerinin kullanılmamasından, *embed*, *object* ve resim haritası alternatiflerinin olmamasından ve/veya metin tabanlı bir site sürümünün kullanıcıya sunulmamış olmasından dolayı, bu sitelerde genel olarak içerik ile ilgili eksikliklerin ve/veya belirsizliklerin olduğu görülmüştür.

Çözünürlük ile ilgili değerlendirmeler ise Windows XP&Firefox yüklü bir bilgisayarda yapılmıştır. Sonuçlar şu şekildedir:

Tablo 4.4 Ekran Çözünürlüğü Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama (%)	(n)
1280x1024*	100	35
1024x768*	100	35
800x600*	43	35

* El yordamı ile değerlendirilenler

Tablo 4.4’den de görüldüğü gibi 1280x1024 ve 1024x768 çözünürlüklerinde değerlendirilen sitelerin tamamında herhangi bir problemle karşılaşılmamıştır. Ancak bu oranın, sayfaların 800x600 çözünürlüğünde değerlendirilmesi sonucunda düştüğü görülmektedir. Bu çözünürlükte problem çıkaran web sayfalarında, yatay kaydırma çubukları ortaya çıkmış ve bunun sonucunda sayfada dolaşma güç hale gelmiştir.

4.3 Tasarım (Teknik Özellikler)

Bu bölümdeki değerlendirmeler, “Genel”, “Sayfa Boyutu” ve “Kodlama” olmak üzere üç alt başlıkta toplanmıştır. Tablo 4.5’de genel kısmı değerlendirme sonuçları yer almaktadır.

Tablo 4.5 Tasarım/Genel Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama	(n)
W3C HTML Geçerlilik Testi (Hata Sayısı)	146.09	35
W3C CSS Geçerlilik Testi (Hata Sayısı)	11.20	35
<i>TIDY Erişilebilirlik Testi (Uyarı Sayısı)</i>		
Seviye 1	313.03	35
Seviye 2	476	35
Seviye 3	518.34	35
<i>Javascript’ler</i>		
Javascript/Noscript	7.43/0.34	35
Script Kullanımı*	25 (%)	32
Javaapplet’ler	0	35
Meta Etiketleri	4.89	35
Çerezlerin Kullanılmaması	37 (%)	35

* El yordamı ile değerlendirilenler

Değerlendirmeler sırasında, değerlendirilen hemen her sayfanın W3C HTML geçerlilik testinde yüksek sayıda hatasının olduğu görülmüş, bu durum Tablo 4.5'den de görüldüğü gibi bu teste ait ortalama hata sayısına da yansımış ve bu değer yüksek çıkmasına neden olmuştur. Öte yandan W3C CSS geçerlilik testinde web sayfalarının, ortalama hata sayısı göz önüne alındığında, W3C HTML testine oranla stil sayfaları yönünden daha başarılı olduğu söylenebilir. Bununla birlikte değerlendirilen web sayfalarının TIDY erişilebilirlik testi ortalama uyarı sayıları, üç farklı seviyede de görüldüğü gibi oldukça yüksektir. W3C HTML geçerlilik testi ve TIDY erişilebilirlik testi bulguları, değerlendirilen web sayfalarının hem standartlar yönünden hem de erişilebilirlik yönünden eksiklerinin olduğunu ortaya koymaktadır.

“Kamu Kurumları İnternet Sitesi Kılavuzu”nda, *script* kullanımının çok gerekli durumlarla sınırlandırılması gerektiği ve *script* kullanıldığında ise kullanıcıların *script*'leri görememelerinden dolayı içerik ile ilgili sayfada herhangi bir eksikliğin olmaması gerektiği belirtilmektedir [10]. Yukarıda da görüldüğü gibi değerlendirilen web sayfalarında kullanılan ortalama *script* sayısı 7.43 iken, *script* kullanılmış sayfa sayısı 32'dir. *Script*'lerin görülememesinden dolayı *script* kullanılan bu web sayfalarında içerik ile ilgili herhangi bir problemin olup olmadığını tespit etmek için bu aşamada değerlendirilen web sayfaları görüntülenirken tarayıcının *script*'leri çalıştırma özelliği devre dışı bırakılmıştır. Yukarıdan da görüldüğü gibi *script* kullanılmış web sayfalarının sadece % 25'inde herhangi bir probleme rastlanmamıştır. Bu konuda sorun çıkaran web sayfalarında ise özellikle javascript ile hazırlanmış menülerin çalışmaması, bundan dolayı bazı menü seçeneklerinin görüntülenememesi gibi problemlerle sıklıkla karşılaşmıştır.

Tablo 4.5'den de görüldüğü gibi değerlendirilen web sayfalarının hiçbirinde javaapplet kullanılmamış ve her sayfada ortalama 4.89 adet *meta* etiketi tanımlanmıştır. Değerlendirilen web sayfalarının javaapplet'ler ve *meta* etiketler konusunda bu değerlere göre başarılı olduğu söylenebilir.

Öte yandan “Kamu Kurumları İnternet Sitesi Kılavuzu”nda, site işlevselliğinin hiçbir şekilde çerez kullanımına dayandırılmaması ve oturum kontrollerinde; sunucu taraflı oturum kontrollerinin tercih edilmesi gerektiği

belirtilmiştir [10]. Bu amaçla değerlendirilen web sayfalarının header bilgilerine bakılarak çerezlerle ilgili herhangi bir bilginin olup olmadığı kontrol edilmiştir. Yukarıdan da görüldüğü gibi bu konuda % 37'lik düşük bir başarı oranı elde edilmiştir.

Tablo 4.6 Sayfa Boyutu Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama (KB)	(n)
Resimsiz	47.42	35
Resimlerle Birlikte	373.05	35

“Kamu Kurumları İnternet Sitesi Kılavuzu”nda, kamu internet siteleri için önerilen ana sayfa dosya boyutu, tüm resimler ve *script'ler* dahil olmak üzere 50 KB'dir [10]. Bu çalışmada geliştirilen araçta, değerlendirilen web sayfalarının dosya boyutları, resimsiz ve resimlerle birlikte hesaplanmış ve elde edilen sonuçlar Tablo 4.6'da verilmiştir. Sonuçlar incelendiğinde, değerlendirilen web sayfalarının ortalama dosya boyutunun; neredeyse resimsiz olarak önerilen 50 KB sınırına yaklaştığı, resimlerle birlikte ise bu sınırın yedi katının aşılmış olduğu görülmektedir.

Web sayfalarının kodlama ile ilgili değerlendirme sonuçları ise şu şekildedir:

Tablo 4.7 Kodlama Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama	(n)
W3C Doküman Tip Bildirimi	60 (%)	35
<i>Yorumlar</i>		
Yorumların Sayısı	10.09	35
Yorumların Uygunluğu*	100 (%)	34
<i>Sayfa Başlığı</i>		
Sayfa Başlığı	100 (%)	35
Sayfa Başlığı Uzunluğu (karakter)	35.66	35

Tablo 4.7'nin devamı

Sayfa Başlığı Uygunluğu*	100 (%)	35
<i>Doğru Etiketleme</i>		
TIDY Hata Sayısı	0.71	35
TIDY Uyarı Sayısı	239.49	35
<i>Body Etiketi</i>		
Body Etiketi	100 (%)	35
Text Niteliği	11 (%)	35
Link Niteliği	9 (%)	35
Alink Niteliği	9 (%)	35
Vlink Niteliği	9 (%)	35
<i>Formlar</i>		
HTML Form Sayısı	1.23	35
Object/Embed Sayısı	1.43/1.63	35/35
HTML Form Kullanımı*	94.64 (%)	28
Label Etiketlerinin Kullanımı	16.08 (%)	25
Tabindex Niteliklerinin Kullanımı	0 (%)	26

* El yordamı ile değerlendirilenler

Doküman tip bildirim, web tarayıcılarına sayfanın nasıl yorumlanacağını ve nasıl analiz edileceğini bildirmekte ve bu tanımlamaya göre web tarayıcıları kullanıcıya bir görünüm sunmaktadır [60]. Konunun önemi bakımından “Kamu Kurumları İnternet Sitesi Kılavuzu”nda da, kamu sitelerindeki tüm dokümanlarda; tip bildirimının mutlaka yapılması gerektiği belirtilmiştir [10]. Bu amaçla W3C HTML geçerlilik testi sonuç dosyasından, değerlendirilen web sayfasında doküman tip bildiriminin yapıp yapılmadığı kontrol edilmiştir. Tablo 4.7’den de görüldüğü gibi bu önemli tanımlama, değerlendirilen web sayfalarının sadece % 60’ında yapılmıştır.

Öte yandan değerlendirilen web sayfalarının, kodlama ile ilgili “Yorumlar” ve “Sayfa Başlığı” altında yer alan değerlendirme kriterlerinde oldukça başarılı olduğu görülmektedir. Kılavuzda da belirtildiği gibi; değerlendirilen her web

sayfasındaki yorumlar teşekkür etme vb amaçla yazılmamış, her web sayfasında sayfa başlığı, sayfa içeriğini açıklayacak şekilde kullanılmış ve ortalama sayfa başlığı uzunluğu da 60 karakteri geçmemiştir [10].

Kodlama kısmı doğru etiketleme başlığı kriterlerinde ise değerlendirilen web sayfalarının genel olarak başarısız olduğu görülmektedir. “Kamu Kurumları İnternet Sitesi Kılavuzu”nda doğru etiketleme ile ilgili olarak, kodun HTML standartlarına uygun olarak yazılması, etiketlerin amaçlarına uygun olarak kullanılıp açılan etiketlerin mutlaka kapatılması, *body* etiketi ve bu etiketin *text*, *link*, *alink* ve *vlink* niteliklerinin kullanılması gerektiği belirtilmiştir [10]. Tablo 4.7’den de görüldüğü gibi değerlendirilen bütün web sayfalarında *body* etiketi kullanılmasına rağmen bu etiketin belirtilen niteliklerinin kullanım oranı oldukça düşüktür. Doğru etiketleme ile ilgili olarak ise web sayfalarının HTML kodu, TIDY ile parse edilmiş ve ortalama TIDY uyarı sayısı; yukarıdan da görüldüğü gibi oldukça yüksek çıkmıştır. Bu bulgu da değerlendirilen web sayfası kodlarında doğru etiketleme ile ilgili çeşitli problemlerin olduğunu ortaya koymaktadır.

“Kamu Kurumları İnternet Sitesi Kılavuzu”nda web sayfalarında kullanılan formların HTML olarak hazırlanması gerektiği, açıklayıcı bütün etiketlerin *label* etiketi içinde verilerek bu etiketlerin ilgili form elemanı ile ilişkilendirilmesi ve form elemanları sekme sırasının doğru dizilimde olması gerektiği belirtilmiştir [10]. Bu amaçla değerlendirilen web sayfalarındaki *object* ve *embed* sayıları da dikkate alınarak web sayfalarında kullanılan formların HTML olarak hazırlanıp hazırlanmadığı incelenmiştir. Tablo 4.7’den de görüldüğü gibi 28 web sayfasında form kullanılmış ve kullanılan bu formların büyük çoğunluğu (% 94.64’ü) HTML olarak hazırlanmıştır. Bununla birlikte form elemanları ile ilişkilendirilmiş *label* etiketlerinin kullanım yüzdesinin ise oldukça düşük olduğu (% 16.08) görülmektedir. Öte yandan form elemanları sekme sırasının doğru dizilimde olup olmadığı bu çalışmada geliştirilen araç ile belirlenemeyeceği için; form elemanlarına sekme sırası atamasında kullanılan *tabindex* niteliklerinin kullanım oranına bakılmıştır. Tablo 4.7’den de görüldüğü gibi form elemanlarının hiç birinde *tabindex* nitelikleri kullanılmamıştır.

4.4 Genel Görünüm

Bu bölümdeki bulgular “Genel”, “Logolar”, “Sayfada Gezinme”, “Metinler”, “Resimler”, “216 Renk Paletinin Kullanımı” ve “Tablolar” olmak üzere yedi alt başlıkta toplanmıştır. Tablo 4.8’de genel, Tablo 4.9’da logolar ile ilgili değerlendirme sonuçları yer almaktadır.

Tablo 4.8 Genel Görünüm/Genel Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama	(n)
Ana Menü*	100 (%)	35
Bağlantı Sayısı	76.77	35

* El yordamı ile değerlendirilenler

Tablo 4.9 Logolar Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama (%)	(n)
Logo*	94	35
Metinlerin Okunabilirliği*	82	33
GIF Formatı*	21	33
Statik Logolar*	88	33

* El yordamı ile değerlendirilenler

Tablo 4.8 ve Tablo 4.9 incelendiğinde bu başlıklar altında yer alan kriterlere genel olarak web sayfalarında uyulduğu görülmektedir. Değerlendirilen her web sayfasında bir ana menü hazırlanmış, kurum logoları değerlendirilen web sayfalarının büyük çoğunluğunda yer almış, logolar genel olarak statik ve üstündeki metinler kolayca okunabilecek şekilde tasarlanmıştır. Logoların GIF formatında hazırlanma oranının düşük olması dışında genel olarak web sayfalarının bu kriterlerden başarılı olduğu görülmektedir.

“Sayfada Gezinme” başlığı altında yer alan kriterler ve bu kriterlerden elde edilen sonuçlar ise şu şekildedir:

Tablo 4.10 Sayfada Gezinme Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama	(n)
<i>Bağlantılar</i>		
Resim İçermeyen	74.78 (%)	35
Resimlerdeki Alt Nitelikleri	39.95 (%)	34
<i>Menüler</i>		
HTML Menü*	57 (%)	35
<i>Tabindex Niteliklerinin Kullanımı</i>		
Bağlantılarda	0 (%)	35
Resim Haritası Alanlarında	0 (%)	6
<i>Accesskey Niteliklerinin Kullanımı</i>		
Bağlantılarda	0 (%)	35
Resim Haritası Alanlarında	0 (%)	6
Düğmelerde	0 (%)	0
Input Nesnelerinde	3.7 (%)	27
Textarea Nesnelerinde	0 (%)	0
Select Nesnelerinde	0 (%)	10
<i>Resim Haritaları</i>		
Harita Sayısı	0.37	35
Kullanım Uygunluğu*	0 (%)	6
Alan Sayısı	0.8	35
Alanlardaki Alt Niteliğinin Kullanımı	50 (%)	6

* El yordamı ile değerlendirilenler

“Kamu Kurumları İnternet Sitesi Kılavuzu”nda, erişilebilirliği daha kolay sağlamak amacıyla bağlantıların mümkün olduğunca metinler üzerinden verilmesi ve resim ile bağlantı verilmesi durumunda ise ilave metin bağlantısı olsa dahi

resimlerdeki *alt* niteliklerinin mutlaka kullanılması gerektiği belirtilmiştir [10]. Tablo 4.10'dan görüldüğü gibi değerlendirilen web sayfalarında resim içermeyen bağlantılar, tüm bağlantıların % 74.78'idir. Buna karşılık resim içeren bağlantılarda, resimlerdeki *alt* niteliklerinin kullanımı ise % 39.95 ile düşük bir oranda kalmıştır. Daha önceden de belirtildiği gibi, Lynx vb. metin tarayıcılarında web sayfası görüntülenmek istenildiğinde veya resimler kapatılarak bir web sayfasında dolaşmak istenildiğinde; *alt* niteliği içermeyen resimler sayfada içerik olarak belirsizliklere neden olmaktadır. Bu bağlamda elde edilen bu % 39.95'lik oran, değerlendirilen web sayfalarının birçoğunda erişilebilirlik yönünden bir takım problemlerin olabileceğini göstermektedir.

Ayrıca söz konusu kılavuz, web sayfalarında yer alan menülerin HTML olarak tasarlanmış metin menüler olmasını, sayfada HTML dışı teknikler kullanılarak hazırlanmış menülerin olması durumunda ise mutlaka alternatif bir metin menünün sayfada yer alması gerektiğini belirtmektedir [10]. Bu aşamada değerlendirilen web sayfaları, tarayıcının; java, javascript ve diğer tüm eklentileri çalıştırma özelliği devre dışı bırakılarak görüntülenmiştir. Sayfada, sayfanın alternatif bir metin versiyonunun da yer alması göz önünde bulundurularak yapılan değerlendirmeler sonucunda bu kriterle ilgili % 57'lik yüksek olmayan bir başarı oranı elde edilmiştir. Öte yandan erişilebilirlik ile ilgili önerilen *tabindex* ve *accesskey* nitelikleri [23], Tablo 4.10'dan görüldüğü gibi değerlendirilen web sayfalarında neredeyse hiç kullanılmamıştır. Bu bulgular da değerlendirilen web sayfalarında erişilebilirlik ile ilgili çeşitli problemlerin olabileceği göstermektedir.

“Kamu Kurumları İnternet Sitesi Kılavuzu”nda, resim haritaları üzerinden bağlantı verme yöntemi; il haritası üzerinden ilçelere erişim gibi resim üzerinde ayrılan bölümlerin verilen bağlantıları anlamlı kılması durumunda önerilmekte ve harita alanları için de *alt* niteliklerinin kullanılması gerektiği ifade edilmektedir [10]. Tablo 4.10'dan da görüldüğü gibi resim haritası bulunan 6 web sayfasının hiçbirinde, resim haritaları amacına uygun olarak kullanılmamış ve harita alanlarının % 50'sinde de *alt* nitelikleri tanımlanmamıştır.

Tablo 4.11 Metinler Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama (%)	(n)
Font Etiketlerini Kullanmama	37	35
Statik Yazılar	71	35
<i>Yazı Tipi Büyüklükleri</i>		
Harici-Dahili CSS	2.23	34
Satırda CSS	10.29	20

“Kamu Kurumları İnternet Sitesi Kılavuzu”nda yazı tiplerinin *font* etiketleri yerine stil sayfaları içerisinde tanımlanması, sayfada kullanılan metinlerin statik olması, yazı tipi büyüklüklerinde ise tüm tarayıcılarda tutarlı sonuçlar veren değerlerin (xx-small, x-small, small, medium, large, x-large, xx-large) kullanılması gerektiği belirtilmiştir [10]. Bu çalışmada geliştirilen araçta *font* etiketleri ile ilgili kriter için web sayfalarında *font* etiketlerinin kullanılıp kullanılmadığına, statik yazılar ile ilgili olarak da sayfada kayan yazıların kullanılıp kullanılmadığına bakılmıştır. Tablo 4.11’den de görüldüğü gibi değerlendirilen web sayfalarında *font* etiketlerinin kullanılmaması ile ilgili % 37’lik düşük bir başarı oranı elde edilirken, statik yazılar ile ilgili olarak % 71’lik bir başarı oranı elde edilmiştir.

Yöntem kısmında da belirtildiği gibi harici ve dahili stil sayfalarının birleştirilmesiyle web sayfasından tek bir stil sayfası, satırda tanımlanan stillerden de başka bir stil sayfası ortaya çıkarılmıştır. Her web sayfası için geçerli yazı tipi büyüklük değerleri yüzdesi hesaplanırken ortaya çıkarılan bu stil sayfaları dikkate alınmış ve daha önce de belirtildiği gibi bu stil sayfalarındaki selector sayılarına göre işlem yapılmıştır. Tablo 4.11’den değerlendirilen web sayfalarında tanımlanmış geçerli yazı tipi büyüklükleri ortalama yüzdelik değerleri incelendiğinde, ilgili CSS dosyalarında, belirtilen yazı tipi büyüklük değerlerinin kullanım oranının oldukça düşük olduğu görülmektedir.

Resimler ile ilgili değerlendirme sonuçları ise şu şekildedir:

Tablo 4.12 Resimler Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama	(n)
Resim Sayısı (İndirilen/Toplam)	58.69/59.09	35/35
GIF&JPEG Formatı	96.97 (%)	35
Resim Boyutu-mak. 30K	92.10 (%)	35
Width Niteliklerinin Kullanımı	70.82 (%)	35
Height Niteliklerinin Kullanımı	69.09 (%)	35
<i>Alt Nitelikleri</i>		
Kullanım	25.24 (%)	35
Uzunluk Uygunluğu-mak. 100 kar.	99.46 (%)	23
Son Karakterin Uygun. (Nokta/Boşluk)	8.61 (%)	23
Son Karakterlerin Uy. (Nokta+Boşluk)	0 (%)	23

Tablo 4.12'ten görüldüğü gibi değerlendirilen web sayfalarında kullanılan resimlerin büyük çoğunluğu “Kamu Kurumları İnternet Sitesi Kılavuzu”nda da belirtildiği gibi GIF&JPEG formatında olup resim boyutları 30 KB üst sınırındadır [10]. Ancak bu oran, *width* ve *height* niteliklerinin kullanımında düşmüş, *alt* niteliklerinin kullanımında ise % 25.24'e gerilemiştir. Öte yandan kullanılan *alt* nitelik değerlerinin uzunlukları büyük çoğunlukla 100 karakteri geçmemesine rağmen, bu niteliklerin son karakter(ler) uygunluk oranlarının oldukça düşük olduğu görülmektedir. Görme engelli kullanıcılar ya da resim yükleyemeyen/yüklemek istemeyen kullanıcılar için *alt* nitelikleri bir alternatif oluşturmakta, ayrıca *alt* niteliklerinin son karakter(ler)inin nokta ve/veya boşluk ile bitmesi sesli tarayıcılarda *alt* niteliklerinin sonraki metne karışmamasını sağlamaktadır [10]. Bu bağlamda *alt* niteliklerinden elde edilen bu bulgular, değerlendirilen web sayfalarının büyük çoğunluğunun bu kriterlerden başarısız olduğunu göstermektedir.

“Kamu Kurumları İnternet Sitesi Kılavuzu”nda bağlantılar, metinler, resimler ve arka plan renkleri için 216 renk paletinin kullanılması gerektiği belirtilmektedir [10]. Bu çalışmada geliştirilen araç ile daha öncede belirtildiği gibi; harici-dahili stil sayfalarından, satırda tanımlanmış stillerden ve bazı HTML renklerinden

oluşturulmuş stil sayfalarından yararlanılarak, bu stil sayfalarında tanımlanmış renklerin bu kritere uygunluğu belirlenmeye çalışılmıştır. Sonuçlar şu şekildedir:

Tablo 4.13 216 Renk Paletinin Kullanımı Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama (%)	(n)
Harici-Dahili CSS	53.92	34
Satırda CSS	52.08	25
HTML Niteliklerinde	47.37	32

Tablo 4.13'ten de görüldüğü gibi elden edilen stil sayfalarında kullanılan renklerin 216 renk paletinde olma yüzdeleri, tanımlanan renklere oranla yaklaşık yarı yarıyadır. Yüksek olmayan bu oranlara göre, değerlendirilen web sayfalarının 216 renk paleti kullanımında genel olarak başarılı olmadığı söylenebilir.

Değerlendirilen web sayfalarının tablolar ile ilgili değerlendirme sonuçları ise şu şekildedir:

Tablo 4.14 Tablolar Değerlendirme Sonuçları (N = 35)

Kriter	Ortalama	(n)
Tablo Sayısı	40.6	35
Tablo Sonu Etiketleri	98.67 (%)	35
Arka Plan Rengi Kullanılmaması	92.67 (%)	35
Arka Plan Resmi Kullanılmaması	97.23 (%)	35

“Kamu Kurumları İnternet Sitesi Kılavuzu”nda tabloların erişilebilirlik ile ilgili bazı güçlükler neden olduğu belirtilmekte, bunun yanında faydaları dikkate alındığında tablo kullanmamanın anlamlı olmayacağı ifade edilmekte, bu bağlamda tasarımcıların tabloları azami dikkat ile kullanmaları ve iç içe tablolardan mümkün

olduđunca sakınmaları vurgulanmaktadır [10]. Deđerlendirilen web sayfaları, Tablo 4.14’de görüldüğü gibi tablolar ile ilgili kriterlerde oldukça başarılıdır. Öte yandan ortalama tablo sayısı yukarıda da görüldüğü gibi yüksek bir değere sahiptir. Bu bulgu, tasarımcıların deđerlendirilen web sayfalarında çok sayıda iç içe tablo kullandıkları şeklinde yorumlanabilir.

5. SONUÇ VE ÖNERİLER

Bu çalışmada TÜBİTAK-MAM tarafından hazırlanan ve 2007/4 sayılı genelge ile yayımlanan “Kamu Kurumları İnternet Sitesi Kılavuzu” kapsamında kamu kurumları internet sitelerini değerlendirmeye yönelik bir araç geliştirilmeye çalışılmıştır. Ardından geliştirilen bu araç ile 35 kamu kurum ve kuruluşu internet sitesi ana sayfası değerlendirilmiştir.

5.1 Class ve Araç ile İlgili Sonuç ve Öneriler

Araç geliştirilirken direk olarak aracın kodlanmasına geçilmemiş, öncelikle belirlenen kriterler çerçevesinde değerlendirme işlemlerini yapabilen bir class yazılarak hem bu aracın çekirdeği oluşturulmuş hem de sonradan bu aracı geliştirmek isteyen geliştiricilere kolaylık sağlanmaya çalışılmıştır. Aracın kodlanmasında da aynı fonksiyonellik gösterilmeye çalışılmış, gerek ara yüz için gerekse de kriter açıklamaları için İngilizce ve Türkçe dil dosyaları oluşturularak araçta kullanılmıştır.

Bununla birlikte değerlendirme işlemlerinde kullanılmak üzere belirlenen kriterlerden bazıları el yordamı değerlendirmesi gerektirmektedir. Class, bu kriterlerde başarılı, başarısız vb kesin bir sonuç döndüremeyeceğinden, bu noktada kullanıcıya mümkün olduğunca kriterle ilgili; olabilir, olmayabilir gibi sonuçlar döndürülerek kullanıcının o kriterle ilgili bir fikre sahip olması düşünülmüştür. Öte yandan araçta da bu durum göz önünde bulundurulmuş ve el yordamı değerlendirilmesinden sonra belirlenen sonucun aracın veritabanına girilebilmesine olanak sağlanmıştır (yönetici modu).

Ayrıca geliştirilen araç ile bir web sayfası değerlendirildikten sonra o web sayfasının değerlendirme sonuçları, değerlendirilmiş bütün web sayfalarından elde edilen ortalama sonuç değerleriyle karşılaştırılabilmekte, ortalama sonuç değerleri

de alan adı ve/veya ülke koduna göre filtre edilebilmektedir. Bununla kullanıcıların, kendi alan adları grubu içinde durum değerlendirmeleri yapabilmesi amaçlanmıştır. Ortalama sonuç değerlerinin ise güncel olabilmesi açısından, web sayfalarının en son değerlendirildiği tarihteki değerlendirme sonuçları ortalamaya dahil edilmiştir.

Class ve araç geliştirilirken her aşamada sağlıklı sonuçların elde edilmesi için yüzlerce internet sitesi değerlendirilmiş, sonuçlar bu web sayfalarındaki HTML ve CSS verileriyle karşılaştırılmış ve mümkün olduğunca stabil (kararlı) bir class ve araç geliştirilmeye çalışılmıştır. Ancak gerek web sayfalarında bir işlemin birden çok teknolojiyle veya birden fazla yöntem ile yapılabilmesi, gerekse konunun ve kodlamanın da zorluğu dikkate alındığında geliştirilen class'ın ve aracın eksiklerinin olması kaçınılmaz olmaktadır. Bilinen eksikler, getirilen yorumlar ve öneriler şu şekildedir:

- Web sayfası değerlendirmelerinde salt HTML ve CSS kodlarına göre işlem yapılmıştır. Bu nedenle web sayfalarında javascript vb ile yapılanlar doğal olarak değerlendirme dışı kalmıştır. Bu durum class'ın ve/veya aracın bir eksiği olarak düşünülebilir. Ancak javascript kodları arasından veri çekebilme, bu dilin özelliğinden dolayı çok da olanaklı görünmemektedir. Kaldı ki “Kamu Kurumları İnternet Sitesi Kılavuzu”nda da *script* kullanımının çok gerekli durumlarla sınırlandırılması ve kullanıcıların *script*'leri görememesinden dolayı herhangi bir içerik eksikliğinin yaşanmaması gerektiği belirtilmektedir [10].
- Class'ın yazılmasında en çok CSS verisi gerektiren işlemlerde zorlanılmıştır. Tablolarda arka plan resminin kullanılmaması, resimlerde *width* niteliklerinin kullanılması gibi bazı kriterlerde hem HTML hem de CSS kodunun taranması gerekmektedir. Class'ta bu işlem “get_att” fonksiyonu ile (EK B.1) yapılmasına rağmen CSS tarafında; yaygın ancak sınırlı sayıda selector tanımlamalarına göre tarama yapılmıştır. Bu noktada CSS selector'leri özgüllük değerlerinin (specificity) hesaplanarak işlem yapılmasının daha kesin ve daha sağlıklı sonuçlar vereceği düşünülmektedir. Ancak buna karşılık da ilgili elementin parent, sibling vb elementlerinin HTML kodunda

analizi gerekmektedir. Class'taki "get_nodes2" fonksiyonunda (EK B.1), bununla ilgili bazı adımlar atılmış ancak devamı getirilememiştir.

- 216 renk paletinde yer alan renklerin onaltılık sayı sistemi kodları bilinmesine rağmen "css_color" fonksiyonunda (EK B.1), renkler tanımlanırken; CSSTidy'nin parse ve optimize işlemleri sonucu bu class'a göre tanımlama yapmak durumunda kalınmıştır. Bu nedenle CSSTidy class'ı güncelleştirildiğinde bu fonksiyondaki renk tanımlamalarının yeniden yapılması gerekebilir.
- "check_import" fonksiyonunda (EK B.1), import işlemlerinin CSS kodlarının başında yapıldığı kabul edilerek sadece "@import ..." biçimindeki kodlamalar için düzenli ifade yazılmıştır. Bu fonksiyon ile import işlemlerinin CSS kodunun başında yapılıp yapılmadığının kontrolü ve "@media ..." şeklinde import edilen CSS kodlarının da dikkate alınmasıyla daha verimli sonuçlar alınabilir.
- "arr_map" fonksiyonunda (EK B.1), istemci taraflı resim haritaları için *area* yöntemiyle oluşturulan resim haritalarına göre kodlama yapılmıştır. Dolayısıyla diğer yöntemlerle oluşturulan resim haritaları bu fonksiyon tarafından görülemeyecektir. Fonksiyon bu yönde geliştirilerek diğer yöntemlerle oluşturulan resim haritaları ile ilgili bilgilerin de elde edilmesi sağlanabilir.
- *frame* ve *iframe* içeren web sayfalarında, *frame* ve *iframe*'ler için ayrıca analiz gerektiğinden class'ta bunlar göz ardı edilmiştir. Doğal olarak bunları içeren web sayfaları değerlendirildiğinde sağlıklı sonuçlar alınamayacaktır. Bu durum bir eksiklik olarak görülse de "Kamu Kurumları İnternet Sitesi Kılavuzu"nda sitelerde zorunlu olmadıkça çerçevelerin kullanılmaması, kullanılması durumunda ise sitenin çerçeve içermeyen bir sürümünün hazırlanması gerektiği belirtilmektedir [10].
- Yönlendirmelerin olduğu bazı web sayfalarında, yönlendirilen sayfa yerine yönlendirme kodunun olduğu sayfa elde edildiğinden doğal olarak değerlendirilmek istenilen gerçek web sayfası değerlendirilememektedir. Bu duruma yönlendirilen sayfa adresinin değerlendirilmesi geçici bir çözüm olsa da, class'ın bu yönde geliştirilmesinin araca esneklik sağlayacağı düşünülmektedir.

- HTML Tidy'nin özelliğinden dolayı niteliklere atanan boş değerlerde nitelikler tanımlanmamış olarak kabul edilmektedir. Bu durum özellikle *alt* niteliklerine atanan boş değerlerde bir eksiklik olarak düşünülebilir. Ancak web sayfalarında resimler kapatılarak dolaşımak istenildiğinde veya Lynx vb. metin tarayıcılarında web sayfaları görüntülendiğinde bu durumun içerik olarak eksikliklere neden olabileceği unutulmamalıdır.
- Bazı web sayfaları değerlendirilirken nadiren de olsa “allowed memory size of ...” hatasıyla karşılaşmıştır. Bu, geliştirilen aracın 256 MB belleğe sahip donanımsal olarak zayıf bir sunucuda çalışmasından kaynaklanmıştır.
- Class'ın bel kemiği, HTML ve CSS kodlarının en iyi şekilde parse edilmesi olduğu için class'ta yeni bir parser yazmak yerine bu işlemleri gerçekleştirebilen araçlar (HTML Tidy ve CSS Tidy) tercih edilmiştir.
- Bir web sayfası değerlendirilmek istenildiğinde bu web sayfasına ait verilerin kaydedilmesi için class tarafından otomatik olarak “www” klasörü ve bu klasörün içinde çeşitli klasörler oluşturulmaktadır. Bu nedenle “www” klasörüne yazma yetkisi verilmesi ve sahibinin “www” kullanıcısı olması gerekmektedir.
- İçerik ile ilgili kriterler için düzenli ifade desenleri yazılmıştır. Bu desenlerden bazıları, kriterle ilgili anahtar kelimeler eşleştiğinde, eşleşen kelimeleri geri döndürmektedir. Bu desenler geliştirilip değerlendirilen web sayfasında anahtar kelimelerin geçtiği cümleler geri döndürülebilirse kullanıcıya daha zengin bir içerik sunulabilir.
- Class'ta, değerlendirilen web sayfasının resimsiz ve resimlerle birlikte boyutu hesaplanmıştır. Web sayfalarında kullanılan diğer eklentiler göz önünde bulundurulmamıştır.
- “arr_accessibility” fonksiyonunda (EK B.1), web sayfasının sadece Lynx tarayıcısında görüntülenebilmesi ile ilgili öneri getirilmiştir. Bu fonksiyon geliştirilerek hem diğer tarayıcılar için hem de değişik çözünürlükler için öneri getirilmeye çalışılabilir.
- Çeşitli dillerde dil dosyaları hazırlanarak hem class'ın hem aracın bu yönde geliştirilmesi desteklenebilir.

5.2 Değerlendirilen Web Sayfaları ile İlgili Sonuç ve Öneriler

Class'ın ve aracın geliştirilmesinin ardından 35 kamu kurum ve kuruluşu internet sitesi ana sayfası “İçerik (Asgari)”, “Tasarım (Erişilebilirlik)”, “Tasarım (Teknik Özellikler)” ve “Genel Görünüm” olmak üzere dört ana başlıkta toplanan kriterlere göre değerlendirilmiştir. Değerlendirmeler sonucunda, bu web sayfalarının asgari içerik ile ilgili kriterleri büyük ölçüde sağladıkları ancak diğer üç ana başlıkta yer alan kriterlerde genel olarak başarısız oldukları görülmüştür. Özellikle W3C HTML geçerlilik testi ve TIDY erişilebilirlik testi bulguları, değerlendirilen web sayfalarının hem standartlar yönünden hem de erişilebilirlik yönünden eksiklerinin olduğunu ortaya koymuştur. Bu bulguları da sayfa boyutlarından, javascript ve tablo sayılarından, *label* etiketlerinden, *alt*, *accesskey* ve *tabindex* niteliklerinden ve yazı tipi boyutlarından elde edilen bulgular başta olmak üzere diğer bulgular da desteklemiştir.

Değerlendirmeler sırasında özellikle gov.tr uzantılı web sayfalarında, hem içeriksel yoğunluktan hem de kullanılan javascript, flash vb eklentilerin çokluğundan dolayı ortalamaların üzerinde bulgulara rastlanmıştır. Bu konuda k12.tr uzantılı web sayfalarının ise -bu sitelerin bazılarında kullanılan içerik yönetim sistemlerinden dolayı- daha başarılı olduğu görülmüştür.

Bununla birlikte değerlendirmeler sırasında bazı web sayfalarında; “bu site en iyi ... tarayıcısında görüntülenir” veya “bu site en iyi ... çözünürlüğünde görüntülenir” vb ifadelerle karşılaşmıştır. Web standartları ve erişilebilirlik gibi çok önemli konular göz önüne alındığında ve “Kamu Kurumları İnternet Sitesi Kılavuzu”nun da yayımlanmış olmasına rağmen, bu ve benzeri ifadelerin web sayfalarında halâ yer alıyor olması son derece üzücüdür.

Bu çalışmada, değerlendirilen web sayfaları yukarıda da belirtildiği gibi geliştirilen araç ile değerlendirilmiş, gereken kriterlerde ise el yordamı değerlendirilmesi yapılmıştır. Öte yandan otomatik araçların; erişilebilirlik ile ilgili genel bir fikir verdiği, daha çok sorunlu olan bölümleri gösterdiği, bazen bu araçların bulamayacağı problemlerin olabildiği veya sorun olmayacak bir şeyi sorunlu olarak

gösterebildiđi belirtilmektedir. Ayrıca bir sitenin tüm erişilebilirlik araçlarında başarılı olmasının sitenin tamamen erişilebilir olduğunu göstermediđi ve el yordamı ile müdahalenin gerektiđi vurgulanmaktadır [49]. Bu bağlamda ve web sayfalarının sık sık güncellenmesi durumu da göz önünde bulundurulduğunda, tasarımcılara standartların ve erişilebilirliđin sağlanması yönünde temel olarak řu önerilerde bulunmaktadır [10, 28]:

- Web sayfalarında HTML metin menüleri kullanılmalıdır.
- *label* etiketleri ile *alt*, *accesskey* ve *tabindex* nitelikleri ilgili etiketlerde kullanılmalıdır.
- *alt* nitelikleri mümkün olduğunca anlamlı yazılmalıdır. Örneđin “resim 3” yerine “Nokia N70 Cep Telefonu Resmi .” gibi.
- *alt* nitelikleri, nokta ve boşluk karakterleri ile sona ermelidir.
- Tablo sayısı en az sayıda tutulmalıdır.
- Özellikle Lynx vb. metin tarayıcılarında web sayfaları test edilmelidir.
- Görsel tarayıcılarda eklentiler kapatılarak web sayfaları test edilmelidir.

Yaklaşık bir sene gibi uzunca bir kodlama sürecinin ardından bu çalışmada geliştirilen class'ın ve aracın eksikleri olmasına rağmen, web sayfaları kılavuz kapsamındaki birçok kriter bakımından geliştirilen bu araç ile değerlendirilebilmektedir. Yukarıda otomatik araçlar ile ilgili belirtilenleri ve el yordamı değerlendirmelerinin de önemini dikkate almak kaydıyla, tasarımcılar web sayfalarını geliştirilen bu araç ile zaman zaman değerlendirerek kılavuz kapsamındaki birçok noktada fikir sahibi olabilirler. Öte yandan söz konusu kılavuzun erişilebilirlik ve web standartları gibi çok önemli konuları içermesinden dolayı, bu çalışmada geliştirilen araçtan; kamu kurum ve kuruluşlarının yanı sıra diđer siteler de faydalanabilirler.

EK A KAMU KURUMLARI İNTERNET SİTESİ KILAVUZU

1. GİRİŞ

e-Dönüşüm Türkiye Projesi 2005 Eylem Planı'nda 38 nolu "Kamu Kurumları İnternet Sitesi Kılavuzunun Hazırlanması" eylemine yer verilmiştir. TÜBİTAK Marmara Araştırma Merkezi sorumluluğunda Ağustos 2005'te başlayan çalışmalar tamamlanarak 2006 yılının ilk yarısında bu kılavuz hazırlanmıştır.

Kılavuzun temel amacı, kamu kurumları internet sitelerinde sunulması gereken asgari içeriğin ve sunum şekillerinin belirlenmesidir. Ortaya çıkan kılavuz kamu kurumları internet sitelerinin sağlaması gereken asgari içeriği tanımlamakta, bunun yanında bu sitelerin en geniş kitlelerce erişilebilir olması amacıyla site tasarımına dönük öneriler içermektedir.

Herhangi bir kamu kurumu, özel sektörden farklı olarak, genelde ilgili olduğu alanda hizmetin alınabileceği yegane kurumdur. Bu nedenle o alanda hizmet almak isteyen tüm bireylerin başvurabileceği tek mercidir. Bu durum kamu kurumlarına kendilerinden hizmet almak isteyen ya da hizmet alması gerekli olan herkese hizmet verebilecek alt yapıları sunma zorunluluğunu getirir. Kamu kurumları sunacakları hizmet için hedef kitlelerini sınırlandırma seçeneğine sahip olmadığı gibi toplumun tüm kesimlerine dil, din, ırk, cinsiyet, eğitim durumu, engellilik durumu ayrımı yapmadan hizmet sunmak durumundadır.

Kamu kurumlarının internet siteleri için de durum farklı değildir. Kurumlar internet siteleri üzerinden sundukları hizmeti de toplumun her kesimine ulaştırabilmelidir. Kamu internet siteleri kullandığı tarayıcı tipinden ve işletim sisteminden bağımsız olarak tüm kullanıcılara aynı işlevselliği sağlamak zorundadır. Diğer taraftan, interneti bazı erişim kolaylaştırıcı uygulamalar yardımıyla kullanabilen engelli kullanıcılar için tüm kamu internet sitelerine ulaşabilmek ve içeriğini eksiksiz okuyabilmek temel bir haktır. Kamu kurumları internet siteleri, engelli kullanıcıların siteye erişimini sağlayabilecek her tür önlemi de almak zorundadır.

Geniş erişilebilirlik düzeyini sağlamak bazı noktalardan ödün vermeyi gerektirebilecektir. Ödün verilmesi gerekebilecek en temel alan ise sitenin görsel tasarımı olacaktır. Kamu kurumları internet siteleri, yaygın erişilebilirliği sağlayabilmek amacıyla site görsel tasarımından bazı feragatlerde bulunabileceklerdir.

1.1 Kılavuzun Hazırlanış Esasları

"Kamu Kurumları İnternet Siteleri Kılavuzu", tüm kamu kurumu internet sitelerinin sağlaması gereken asgari içerik ve sunum şekillerini belirleme amacıyla, uluslararası standartlar ve diğer ülkelerde yapılan benzer çalışmaların çıktıları temel alınarak hazırlanmıştır.

Kılavuz sekiz ana başlık altında toplanmıştır.

1. *Giriş*: Çalışmanın dayanak ve amaçları
2. *İçerik*: Kamu kurumu internet sitelerinin sunmaları gereken içeriğe dönük öneriler
3. *Tasarım*: İçeriğin sunum şekline dönük öneriler
4. *Yayımlama*: Sitenin internet ortamında sunumuna yönelik öneriler
5. *Erişilebilirlik Kontrolü*: Sitenin erişilebilirliğinin artmasını sağlamak üzere yayımlanmış kontrol listeleri
6. *Kısaltmalar*: Kılavuzda kullanılan Türkçe ve İngilizce kısaltmalar

Kılavuzda teknik terimler mümkün olduğunca Türkçeleştirilerek kullanılmakla birlikte, bazı Türkçe terimlerin kullanıldığı ilk noktada, anlam kargaşasına yol açmamak için İngilizce terim

de parantez içinde verilmiştir. Bazı terimlerin henüz uygun Türkçe karşılıkları oluşmadığından, doğrudan tırnak içinde İngilizceleri kullanılmıştır. Bu kılavuz internet teknolojilerine ait bir referans doküman değildir. Site tasarımcıları, tasarım için gereken her tür teknik alt yapıya sahip olmalı, karşılaştıkları sorunlarda kullandıkları teknolojilere ait referanslara başvurmalıdır. Kılavuz kapsamlı teknik ayrıntılar içermemekte, bunun yerine her konu için erişilebilirliği artırıcı tavsiyelerde bulunmaktadır. Kılavuz, tek başına tüm teknik yönleriyle internet sitesi hazırlama kılavuzu olarak değerlendirilmemelidir.

1.2. Uygulama Esasları

Kamu kurumlarınca, internet sitelerinin bu kılavuzda yer alan tavsiyelere göre yeniden gözden geçirilmesi, gerek kamu kurumları arası içerik uyumunun sağlanması, gerekse site erişilebilirliğinin artırılabilmesi açısından önem taşımaktadır.

Bu kılavuz kamu kurumları internet siteleri için en temelde basitlik ve sadeliği önermektedir. Bu yüzden kılavuzda yer alan hususları sağlayabilmek için site tasarımcılarının yeni teknik bilgiler öğrenmeleri gerekmekte, tam aksine zaten sahip oldukları bilgi birikimlerini kullanarak, sitelerinin basit ve sade hale getirilebilmesi, dolayısıyla kolay erişilebilir olmasını sağlamaları gerekmektedir.

Kılavuzda önerilen tasarım değişikliklerinin tüm kamu kurumları için aynı anda yapılması olası gözükmemektedir. Ancak, kamu kurumları internet sitesi tasarımcılarının, sitelerinin bu Kılavuzun önerileriyle uyuşmayan yanlarını belirlemeleri ve bu uyuşmazlıkları bir plan çerçevesinde gidermeleri beklenmektedir.

1.3. Kaynaklar

Kılavuzun hazırlanmasında, bu konuda daha önce çalışmalar yapmış olan ülke örneklerinden, konuya ilişkin uluslararası standartlardan ve ilgili mevzuattan faydalanılmıştır. Kılavuzun yazılması sırasında faydalanılan dokümanların ve çevrimiçi kaynakların listesi aşağıda verilmektedir.

Standartlar	
Web Accessibility Initiative (WAI)	http://www.w3.org/WAI/
Web Content Accessibility Guidelines (WCAG) Overview	http://www.w3.org/WAI/intro/wcag.php
Uluslararası Benzer Çalışmalar	
New Zeland Government Web Guidelines Version 2.1	http://www.e.govt.nz/standards/web-guidelines/web-guidelines-v-2-1/web-guidelines.pdf
Guidelines for UK Government websites v 1.0	http://www.cabinetoffice.gov.uk/e-government/docs/resources/web_guideline_handbook/pdf/wgl_azindex.pdf
Kanunlar	
4982 sayılı Bilgi Edinme Hakkı Kanunu	
5432 sayılı Bilgi Edinme Hakkı Kanununda Değişiklik Yapılmasına Dair Kanun	
4688 Sayılı Kamu Görevlileri Sendikaları Kanunu	
Diğer Kaynaklar	
Bilişim Sözlüğü 2005, Prof. Dr. Bülent Sankur, Pusula Yayıncılık	http://www.bilisimsozlugu.com
“.tr” Alan Adları : Politikalar, Kurallar, İşleyiş	http://www.nic.tr
2005/20 sayılı Başbakanlık Genelgesi ek'i e-Dönüşüm Türkiye Projesi Birlikte Çalışabilirlik Esasları Rehberi, DPT Müsteşarlığı, 2005	http://www.bilgitoplumu.gov.tr/yayin/

2. İÇERİK

Bir internet sitesinin en temel öğeleri sunduğu bilgi ve hizmetlerdir. Site, sunduklarının ilgi çekiciliği, işe yararlığı, doğruluğu, güncelliği ölçüsünde kullanıcıların ilgisini çekebilecektir. Doyurucu içeriğe sahip olmayan bir internet sitesinin, sadece etkileyici tasarımıyla fayda sağlaması ve hatta bilinirlik kazanması olası değildir.

Bu bölümde kamu internet sitelerinin içeriğinin oluşturulmasına yönelik tavsiyeler yer almaktadır. Site içeriğinin nasıl yönetileceği ile değişik organizasyonlar için gerekli olan en az içerik yine bu bölümde tanımlanmaktadır.

2.1 Temel Beklentiler

Bir kamu internet sitesinin içeriğinden beklenen temel özellikler şöyle sıralanabilir:

- İçeriğin doğruluğu
- İçeriğin güncelliği
- İçerik doyuruculuğu
- Açıklık, kolay anlaşılabilirlik
- İmla hataları ve yazım yanlışlarından arındırılmış olması
- Kolay ve yaygın erişilebilirlik

2.2 Kullanıcılar

Kamu kurumu internet sitesi içeriği belirlenirken göz önünde tutulması gereken ilk nokta sitenin hedef kullanıcı profilidir. Hedef kullanıcı profili ve bu kullanıcıların ihtiyaçlarının belirlenmesi, sitenin bu ihtiyaçları karşılayabilecek içeriği sunmasının ilk adımıdır.

Kullanıcılar ve kullanıcı profillerinin belirlenmesi, sitenin neleri içermesi gerektiğini ve kullanıcıların siteden beklentilerini de ortaya çıkaracaktır. Kullanıcı profilinin belirlenmesi sırasında göz önüne alınması gereken örnek sorular aşağıda verilmiştir:

- Kullanıcıların siteyi ziyaret etme sıklığı ne olacaktır? Eğer kullanıcıların siteyi çok sık ziyaret etmeleri beklenmiyorsa, akılda kalıcılık açısından sitenin görsel tasarımı önemli olacaktır. Kullanıcıların siteye sürekli erişimi bekleniyorsa (örneğin; etkileşimli hizmetler sunan kurumların siteleri), bu tür kullanıcılar için işlevsellik görselliğin önüne geçecektir.
- Kullanıcılar sitedeki bilgiyi nasıl ve ne amaçla kullanacaklardır? Sadece bilgilendirme amaçlı mı, yoksa etkileşimli hizmetler de sunulabilecek midir?
- Kullanıcıların eğitim düzeyi nedir?
- Siteden, diğer kamu kurumlarının siteleri veya özel sitelere bağlantı verilmesi gerekli midir?
- Site içi arama işlevi olmalı mıdır?
- Kullanıcılardan geri besleme alınmalı mıdır?
- Sitenin güncellenme sıklığı ne olmalıdır?

2.3 Bilginin Elektronik Yayınlanması

Elektronik yayımlama, bilgilerin elektronik ortamda üretilmesi ve dağıtılması anlamını taşır. Günümüzde bilgiler çoğunlukla elektronik ortamlarda yazılmakta ve saklanmaktadır. Elektronik ortamda üretilen belgelerin internet ortamında paylaşımı da yapıları gereği basılı belgelere göre çok daha kolay olmaktadır.

Dokümanların elektronik olarak yayınlanması beraberinde birçok avantajlar getirmektedir. Bu avantajlar şu başlıklar ile özetlenebilir:

- Düşük Maliyet
- Erişim Kolaylığı
- Kullanım Kolaylığı
- Güncellenme Kolaylığı
- Dağıtım Esnekliği

Elektronik yayımlama, bilgiye istenildiği anda, istenildiği yerden erişimi mümkün kılmakta, erişimi kolaylaştırmakta ve ucuzlatmaktadır. Basılı dokümanlara göre, bilgilerin güncellenmesi çok daha kolay ve daha az maliyetle yapılabilmektedir.

Kamu kurumlarının interneti etkin olarak kullanmaları gerek kurumlar, gerekse kullanıcılar açısından elektronik yayımlamanın avantajlarından en üst düzeyde faydalanılması sonucunu doğuracaktır.

2.4 Kamu İnternet Sitelerinde Yer Alması Beklenen Asgari İçerik

Kamu kurumları görevleri, sorumlulukları, büyüklükleri açısından farklılık gösterdiklerinden, internet sitelerinin içermeleri gereken bilgi düzeyi de farklı olmaktadır. Bu bölümde bir kamu sitesinde yer alması gerekli en az bilgi düzeyi tanıtılacaktır. Kurumlar, kendi ihtiyaçlarına göre bu içeriği artırabilecektir.

Beklenen asgari içerik, asgari site içeriği ve asgari ana sayfa içeriği olarak iki bölüm halinde değerlendirilecektir.

27/04/2004 tarihli ve 25445 Sayılı Resmi Gazete'de yayımlanarak yürürlüğe giren Bilgi Edinme Hakkı Kanununun Uygulanmasına İlişkin Esas ve Usuller Hakkında Yönetmeliğin 6'ncı maddesinde "...Kurum ve kuruluşlar, Kanun ve bu Yönetmelik hükümleri çerçevesinde, bilgi edinme hakkının etkin olarak kullanılabilmesi ve bilgi edinme başvurularından kaynaklanan iş yükünün en aza indirilebilmesi amacıyla kurumsal internet sayfalarını bu madde hükümlerine göre yeniden şekillendirir. Bu Yönetmelik kapsamındaki kurum ve kuruluşlar, bu bilgileri tek tek birimler bazında, aynı kurumsal internet sayfası üzerinden; zorunlu hallerde kurumsal internet sayfasından link verilmek suretiyle birime ait internet sayfası üzerinden kamuoyunun bilgisine sunarlar" hükmü yer almaktadır. Kamu kurum ve kuruluşlarının internet sitelerinin bu hükme uygun hazırlanması zorunludur.

2.4.1 Asgari Site İçeriği

Kamu kurumları internet siteleri; ait olduğu kuruma ilişkin olan, kamuya açıklanmasında sakınca görülmeyen ve ilgili yasalarla açıklanması zorunlu kılınan her türlü bilgiyi kolay erişilebilir, anlaşılabilir, doğru, güncel, kolay okunabilir bir biçimde sunmak zorundadır. Sitelerin içeriği kurumların özelliklerine göre büyük farklılıklar gösterebilecektir. Kimi site, birkaç bilgi sayfasından oluşurken, kimisi çok sayıda etkileşimli hizmet sunan büyük bir portal yapısında olabilecektir. Bu bölümde her türlü kamu kurumuna ait sitenin sunması gereken minimum içeriğin çerçevesi çizilecektir. Kurumlar kendi ihtiyaçlarına göre, bu içeriği artırabilecektir.

Bir kamu internet sitesi en az aşağıdaki öğeleri içermelidir:

- Kurum adı, logosu
- Sitenin amacı
- Kurum yapılanması
- Kurumun sorumlulukları ve amaçları
- Kurum yöneticileri
- Kurumun alt birimleri ve bu birimlerin görevleri
- Kurum ile alakalı mevzuat
- Kurumun sunduğu hizmetleri açıklayıcı sayfalar
- Eğer varsa sunulan etkileşimli hizmetler
- Kurumun üretmekle yükümlü olduğu her tür rapor
- İstatistiksel bilgiler
- Kuruma ait duyurular, haberler
- Kuruma ait basın açıklamaları
- Basında kurum ile ilgili çıkan haberler
- İnternet üzerinden sunulabilecek her tür form
- Kuruma ait erişim bilgileri (telefon, adres, e-posta)
- Şikayet ve geri bildirim mekanizması
- Site ile ilgili yardım sayfaları
- Sıkça Sorulan Sorular sayfaları

- 4688 Sayılı Kamu Görevlileri Sendikaları Kanunu'nun 16'ncı maddesi gereğince kurumun sendikali personeline ait veriler
- İhtiyaca göre yabancı dilde kurumun tanıtımı, teşkilat yapısı, görev ve sorumlulukları vb.

2.4.2 Asgari Ana Sayfa İçeriği

Ana sayfa, bir internet sitesinin en temel elemanıdır. Sitenin internet adresi ile ("www.kamukurumu.gov.tr") ulaşılan ilk sayfadır. Sitenin içeriğinden habersiz olan ve sadece alan adını bilen tüm kullanıcıların ulaşacakları ilk yer sitenin ana sayfasıdır. Kullanıcıların site hakkındaki ilk izlenimleri ana sayfayı görmeleri ile oluşacaktır.

Genel anlamda sitenin en fazla bağlantı alan sayfası da ana sayfadır. Kullanıcılar çoğunlukla alt sayfaların adreslerini yazmak yerine ana sayfa bağlantılarını kullanmayı tercih ettiklerinden siteye bağlanan her kullanıcı ana sayfayı en az bir kez ziyaret etmiş olacaktır.

Ana sayfa sitenin tüm temel alanlarını, bu alanlara nasıl ulaşılabileceğini ve sitenin temel var olma amacını içermelidir. Ana sayfa merkezi konumu nedeniyle kimi zaman bazı özel amaçlı bilgileri vermede kullanılsa da (kurumdan haberler gibi) genellikle içeriği sadedir.

Ana sayfa, site içi dolaşımı kolaylaştırabilecek bir bağlantı ve menü yapısına sahip olmalıdır. Site ana sayfası, kullanıcıların site içinde aradıklarını bulmalarını sağlamaya dönük bir geçiş sayfası olarak düşünülmeli, mümkün olduğunca basit, küçük boyutlu tutulmalıdır.

Kamu internet siteleri ana sayfaları aşağıdaki özellikleri içermelidir:

- Kurum logosu ve açık adı (eğer bağlı olunan başka bir kurum varsa o kuruma ait bilgiler ile birlikte)
- Sitenin amacı
- Site içi bağlantılar
- Duyurular, yenilikler
- Erişim bilgileri (e-posta, telefon, adres)
- Arama mekanizması
- Yardım sayfalarına bağlantı
- Sıkça Sorulan Sorular sayfalarına bağlantı
- Şikayet ve geri bildirim sayfalarına bağlantı
- Önemli görülen diğer sayfalara doğrudan bağlantı

2.4.3 Bilgi Edinme Hakkı Kanunu

4982 sayılı Bilgi Edinme Hakkı Kanunu; kamu kurum ve kuruluşlarını, kanunda yer alan istisnalar dışındaki her türlü bilgi veya belgeyi başvuranların yararlanmasına sunmak ve bilgi edinme başvurularını etkin, süratli ve doğru sonuçlandırmak üzere, gerekli idarî ve teknik tedbirleri almakla yükümlü kılmaktadır.

Kanun, bilgi edinme başvurularının, başvuru sahibinin kimliğinin tespitine yarayacak bilgilerin saklanması koşuluyla elektronik ortamda yapılabilmesine olanak tanımaktadır. Bu nedenle, kamu internet sitelerinin, bilgi edinme hakkını kullanmak isteyen kullanıcılar için dinamik bir başvuru formu hazırlamaları yerinde olacaktır. Form işlenirken, başvuru sahibinin kimlik ve erişim bilgilerinin yanı sıra, başvurunun yapıldığı tarih, saat ve IP adresi de saklanmalıdır.

2.4.4 E-Posta Adresleri

Gerek site ile ilgili erişim adresi olarak, gerekse site içinde gerekebilecek diğer noktalarda verilecek erişim adreslerinde özel e-posta adresleri kullanılmamalıdır. Site içinde hiçbir noktada kurum çalışanlarına ait olan, ancak özel e-posta hizmet sağlayıcılarından alınmış e-posta adresleri kullanılmamalıdır.

Site içinde verilecek e-posta adresleri, mümkün olduğu sürece kurum alan adını içermelidir. E-posta adresleri kurum alan adı içermelerinin yanı sıra, göreve dayalı olarak belirlenmiş olmalıdır.

bilgi@kamukurumu.gov.tr, sikayet@kamukurumu.gov.tr gibi genel adresler, ali.cicek@kamukurumu.gov.tr gibi kişisel adreslere tercih edilmelidir.

Bu sayede görevli kişinin değişmesi durumunda iletişim kopukluğu yaşanması önlenmiş olur ve uyum sorunu yaşanmaz. İlgili birimde yeni görevlendirilen çalışan aynı adresi kullanmaya devam edebilir. Tüm sitenin taranarak eski çalışanın adreslerinin güncellenmesi gerekmez ve kişiye bağımlılık engellenmiş olur.

2.4.5 Etkileşimli Hizmetler

Yukarıda da değinildiği gibi, kamu internet sitelerinin içerikleri, kurumsal gerekliliklere göre çeşitlenmektedir. Bölüm 2.4.1’de sitelerden beklenen minimum içerik tanıtılmıştır. Minimum içerik doğal olarak sitelerin sunması gereken en az bilgi düzeyini tanımlamakta, ancak kurumun sunması gerekli olan etkileşimli hizmetleri sınırlamamaktadır.

Kamu kurumları günümüzde kimi hizmetlerini etkileşimli olarak çevrimiçi sunmaktadır. Bu hizmetler vatandaşlık numarasının öğrenilmesinden pasaport başvurusuna, bildirge doldurulmasından sınav sonuçlarının duyurulmasına kadar çeşitlenmektedir.

Etkileşimli hizmetler sunan kurumlar, sitenin çok geniş bir kullanıcı grubu tarafından kullanılabilir olmasına önem vermelidir. Site hizmetlerinden faydalanmak tek tür tarayıcı ya da işletim sistemi kullanımını zorunlu kılmamalı, engelli kullanıcılar için uygun erişim yöntemleri de dikkate alınmalıdır.

Etkileşimli hizmetler sunan kurumlar bu belgenin 3 no’lu tasarım bölümünde yer alan önerileri dikkate almalıdır. Siteye yaygın erişimin sağlanabilmesi, kurumların en önemli önceliklerinden biri olmalıdır.

Etkileşimli hizmetlerin sunumu için kullanılacak veri tabanı ya da programlama dillerinin değerlendirilmesi bu dokümanın kapsamı dışındadır. Bu amaçla, yaygın erişim imkanı sağlayan her tür teknoloji den faydalanılabilir.

2.4.6 Etkileşimli Ortamlar

Kamu internet sitelerinde etkileşimli tartışma, sohbet ve danışma ortamları kullanılmak isteniyor ise, kullanıcıların göndereceği mesajların içeriğinin herhangi bir suç içermemesi, kurum amaçları ile ters düşmemesi önemlidir. İçerik kontrolünün nasıl yapılacağı belirlenmesi, bu tür ortamlar yaratılmadan önce kararlaştırılması gereken en önemli konudur. Aşağıda iki temel etkileşimli tartışma ortamına dönük öneriler özetlenmiştir.

2.4.6.1 Tartışma Forumları

Kamu internet sitelerinde tartışma forumları kurulması düşünülüyorsa, bu forumlar mutlaka kontrollü (moderated) olmalıdır. Forum kullanıcıları kayıtlı kullanıcılar olmalı, kayıt işlemi sırasında verilen bilgilerin mantıklı olup olmadığı kontrol edilmelidir. Organizasyonun kararına bağlı olarak kayıtlı kullanıcıların foruma mesaj yollamalarına izin verilebilir ve forum yöneticisi belli aralıklar ile mesajları tarayarak içerik olarak uygunsuz olanları silebilir, sorumluların üye kaydını iptal edebilir. Daha yüksek dereceli kontrol uygulanmak isteniyor ise, foruma yollanan her mesaj ancak forum yöneticisi tarafından denetlendikten sonra yayımlanabilir. Bu durumda forumun işlerliği kısıtlanacak, ancak sorunlu mesajların yayınlanma ihtimali en aza indirilmiş olacaktır.

2.4.6.2 Sohbet Odaları

Kamu internet sitelerinde sohbet odalarına yer verilmemelidir.

2.5 İçerik Yönetimi

Bu bölümde internet ortamında sunulacak dokümanların taşınması gereken temel özelliklere değinilmektedir. “Doküman” terimi hem doğrudan HTML sayfalarda sunulan metinleri, hem de çeşitli biçimlerde hazırlanmış dosya şeklinde sunulan bilgiyi (pdf, doc vb.) kapsamaktadır. Yani bir anlamda doküman terimi ile sitenin tüm içeriğine değinilmektedir.

2.5.1 Yayınlanacak Dokümanlar

Kamu kurumlarının internet üzerinden erişime sunacağı dokümanlar, kurumların hizmet alanlarına göre çeşitlilik gösterecektir. Her kurum, ilgili yasa ve yönetmeliklerinin gerektirdiği tüm dokümanları internet ortamında erişime açmalıdır. Kurum yetkilileri, site içeriğini bu çerçevede belirlemelidir.

2.5.2 Dokümanların Kalitesi

İnternet ortamında yayınlanacak tüm dokümanlar, kurumun diğer basılı dokümanları ile aynı titizlikte hazırlanmalıdır. Dokümanlar gerek içerik, gerekse biçimsel anlamda hiçbir hata içermemelidir. Eksik, yanlış, eski bilgi ve dokümanlara kamu internet sitelerinde kesinlikle yer verilmemelidir.

İnternet üzerinde yayınlanacak dokümanlar da, aynen basılı dokümanlarda olduğu gibi kurumun kalite yöntemleri ve onay süreçlerine göre denetlenmelidir.

2.5.3 Dokümanların Güncellenmesi

Kamu internet sitelerinde yayımlanan her doküman mutlaka en güncel sürüm olarak yayınlanmalıdır. İlgili dokümanın güncellenmesi durumunda, güncel sürüm vakit geçirmeden internet sitesine de eklenmelidir.

Eğer bazı dokümanların eski sürümlerinin güncel sürüm ile birlikte tutulması isteniyorsa, dosyalar bu durumu açıklayacak şekilde adlandırılmalıdır. Kullanıcı özel bir çaba harcamadan güncel sürüme ulaşabilmeli, dilerse eski sürümlere de erişebilmelidir.

2.5.4 Dokümanların Güvenliği

Eğer kurum bazı dokümanları herkesin değil, sadece belli kişi veya kurumların erişimi için internet ortamına sunmak istiyorsa, bu kısıtlı erişimi sağlayacak mekanizma titizlikle kurulmalıdır.

Bu tür bir erişim denetim yöntemi kullanılıyor ise, erişim yetkisi olmayan kullanıcılara bu durum detaylı mesajlar ile açıklanmalıdır. "Yetkiniz yok" gibi kısa ifadeler yerine erişimin neden kısıtlandığı açıklanmalıdır.

2.5.5 Dokümanların Yapılandırılması

Dokümanların hazırlanması sırasında biçimler (style) ve şablonlardan faydalanılmalıdır. Kelimelerin doğrudan görünüşünü değiştirmek yerine ilgili biçimi seçme, gerekiyorsa biçim üzerinde değişiklik yapma yolu seçilmelidir. Kullanılan kelime işlemci desteklediği sürece, biçim kullanımından vazgeçilmemelidir.

Doküman yazılmaya başlanmadan önce doküman yapısı kurulmalıdır. Kullanılacak biçimler ve şablonlar belirlenmeli, ardından içeriğin yazılmasına geçilmelidir.

Dokümanın her parçası kendi içinde bir bütün oluşturmalıdır. Kullanıcıların aradıkları bilgiye ulaşana kadar tüm dokümanı okumaları beklenemeyeceği gibi, arama motorları da günümüzde doküman içeriği üzerinde arama yapabilmekte, doğrudan ilgili bölüme bağlantı verebilmektedir. Bölümlerin kendi içinde bütünlük taşıması bu noktada kullanıcının tatmin edilebilmesi açısından önemlidir.

2.6 Bilgi Organizasyonu

İnternet siteleri eğer özellikle geniş içeriğe sahiplerse, sunulan bilgilere kolay erişimi sağlamanın ilk kuralı bilgilerin doğru organize edilmesidir. Bilgilerin doğru organize edilmesi, doğru dolaşım sisteminin kurulmasını ve dolayısı ile de kullanıcıların bilgiye kolay erişimini sağlayacaktır. Bilgi organizasyonunu üç ana başlıkta toplamak mümkündür:

- Bilginin mantıksal olarak bölünmesi
- Bilginin önem sırasına göre dizimi
- Bilgi blokları arasında hiyerarşik yapının kurulması

Sitede sunulacak bilgiler tek bir bütün halinde bulunmamalı, küçük boyutlu parçalara bölünmelidir. Bu bölme işlemi mantıksal olarak yapılmalıdır.

Site içeriğini oluşturan bilgi blokları arasındaki mantıksal ilişkiler incelenerek hiyerarşik bir yapı kurulmalıdır. Bu yapı kurulurken bilgilerin önem sırası da göz ardı edilmemelidir. Kullanıcılar için büyük önem arz eden bilgiler kolay erişilebilir noktalarda tutulmalıdır.

Sonuçta, bir internet sitesini oluşturan bilgilere ait organizasyonel yapı elde edilecektir.



Şekil 1 Bilgi Yapılandırması

Site içinde sunulan bilgiler mantıksal anlamda tutarlı olacak şekilde organize edilmeli ve önem düzeyi de dikkate alınarak yapılandırılmalıdır. Şekil 1 'de örnek bir bilgi yapılandırılması verilmiştir. Bu yapı düzenli olup, her dokümanın kolay bulunabilmesini ve kolay geri dönüş imkanı sağlamaktadır.

Şekil 2 'de ise asla kurulmaması gereken bir bilgi organizasyonu gösterilmektedir. Bu örnekte, mantıksal bölümlenme yapılmamış, bağlantılar gelişigüzel verilmiştir. Sağlıklı dolaşım imkanı yoktur.



Şekil 2 Karmaşık yapı

2.6.1 Bilgi Organizasyonu Türleri

Bilginin en basit olarak yapılandırılması lineer bir dizi şeklindedir. Bilgilerin mantıklı bir sıra halinde olması lineer bir dizin için ideal durumdur. Kronolojik ya da alfabetik olarak tutulabilen bilgiler dizi yapısında tutulmaya uygundur.

Bu yapıda her sayfa bir önceki ve sonraki sayfaya bağlanmakta, tüm bilgiler bir dizi şeklinde tutulmaktadır. Bilgi boyutu büyüdükçe, dizi yapısının kullanımı anlamsızlaşacaktır. Yine de her sayfadan ana sayfaya bağlantı kurulması koşulu ile kullanılabilir.



Şekil 3 Bilgi Dizisi

Kullanılan bilgi organizasyonu türlerinden biri de Izgara Yapısı'dır. Bu yapıda dokümanlar arası bağlantılar bir ızgara yapısı şeklinde kurulur. Bağlantılar Bilgi Dizisi'nde olduğu gibi tek boyutlu olmayıp, her doküman tüm komşularına bağlantı içerir. Bu tür bir yapının kurulması, bazı kullanıcılar için anlama ve kullanma zorluğu yaratabilir.



Şekil 4 Izgara Yapısı

Ağ yapısı türünde bilgi organizasyonu, tüm öğelerin birbirlerine bağlı olduğu bir ağ yapısıdır. Burada amaç öğeler arası dolaşımı hızlandırmak ve kolaylaştırmaktır. Her dokümandan bir diğerine tek bir bağlantı izlenerek erişilebilir.

Bu yapı getirdiği erişim kolaylığının yanında, karmaşıklığı artırmakta ve bakımı zorlaştırmaktadır.



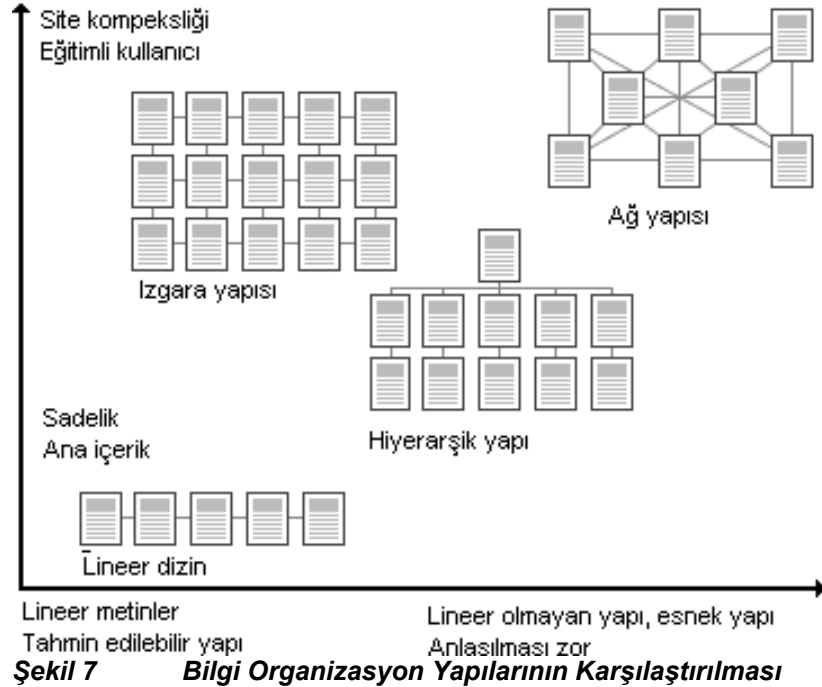
Şekil 5 Ağ Yapısı

Hiyerarşik bilgi organizasyonu yapısı, öğelerin bir hiyerarşik düzen içinde tutulmasına dayanır. Kurulan hiyerarşi mantıksal bütünlüğe ve önem sırasına göredir. Kullanıcı ilgili dokümanlara daha kolay erişebilir. Bağlantılar sadece aynı hiyerarşik grup arasında olduğundan karmaşıklık da çok fazla değildir.



Şekil 6 Hiyerarşik Yapı

Aşağıdaki grafik bu dört tip yapılandırmanın çeşitli kriterler açısından karşılaştırmasını yapmaktadır. Grafikte yatay eksenle ilerledikçe anlaşılma zorluğu artmakta buna karşın esneklik kazanılmaktadır. Dikey eksenle ilerlemek ise karmaşıklığı artırmakta, kullanıcı bilgi düzeyinin artmasını gerektirmektedir.



Şekil 7 Bilgi Organizasyon Yapılarının Karşılaştırılması

2.6.2 Site İçi Dolaşım

İnternet siteleri kullanıcılarına kapsamlı bilgi sunmanın yanında bu bilgilere kolay erişim yollarını da sunmak zorundadır. Sitenin sunduğu bilgiler ancak kolay erişilebilir olmaları durumunda anlamlı olacaktır.

Site içi erişim için kullanıcıya uygun bir menü sistemi sunulmalıdır. 0 no'lu başlıkta menüler ile ilgili tasarım önerileri verilmektedir. Tasarımdan bağımsız olarak menü yapıları incelenecek olursa, sitelerde kurulu menüler üç ana türde gruplanabilmektedir:

Sığ Menüler:

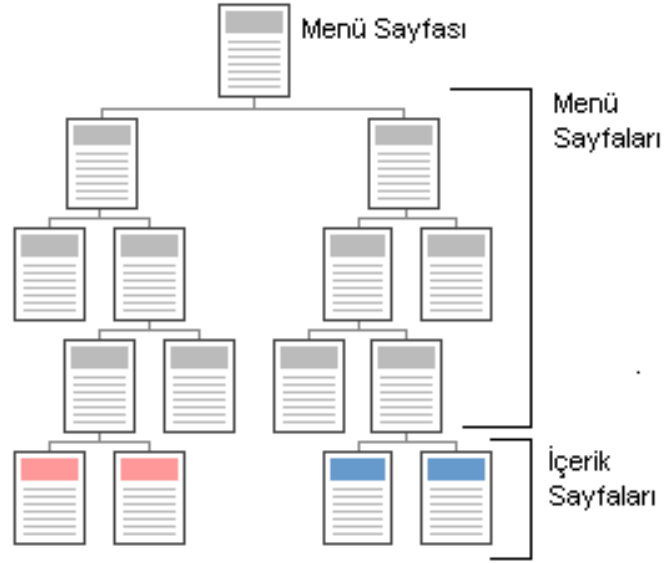
Hiyerarşik kademelerin iyi tanımlanmadığı, tüm site içeriğine doğrudan ana sayfadan bağlantı verilen yapılardır.



Şekil 8 Sığ Menü Yapısı

Derin Menüler:

Esas içeriğe ulaşım için çok sayıda ara sayfadan geçilmesi gereken, gereğinden fazla kademelere ayrılmış yapılardır. Ara sayfalar bağlantı vermek dışında bir işleve sahip değildir.



Şekil 9 Derin Menü Yapısı

Dengeli Menüler:

Dengeli bir ağaç yapısında tasarlanmış, çok derin olmayan ve her seviyede yeterli düzeyde bilgiye erişilebilen menülerdir. İnternet sitelerinde kullanılan dolaşım yapısının dengeli menü benzeri olması kullanıcılar açısından olumlu olacaktır.



Şekil 10 Dengeli Menü Yapısı

Kamu kurumları internet sitelerinde site içi dolaşım sistemlerinin dengeli yapıda olmasına gayret gösterilmelidir. Kamu internet sitelerinde yer alan tüm sayfalar en az bir başka sayfaya bağlantı içermelidir. Her sayfa hiyerarşik yapıda bir üstünde olan sayfaya ya da ana sayfaya bağlanabilmelidir. Başka hiçbir sayfaya bağlantı içermeyen sayfalar “Çıkmaz Sokak” (Dead-End) olarak adlandırılır ve kullanıcı bu sayfalardan ancak tarayıcının sağladığı geri dönüş imkanıyla ayrılabilir. Kamu internet siteleri Çıkmaz Sokak içermemelidir.

3. TASARIM

Bu bölümde kamu sektörü internet sitelerinin hazırlanmasında uyulması önerilen tasarım kriterleri yer almaktadır.

3.1 Erişilebilirlik

Herhangi bir kamu kurumu, özel sektörden farklı olarak, genelde ilgili olduğu alanda hizmetin alınabileceği yegane kurumdur. Bu nedenle o alanda hizmet almak isteyen tüm bireylerin başvurabileceği tek mercidir. Sunulan hizmetlerin toplumun tüm kesimlerinin erişimine açık olması vazgeçilmez bir ilkedir. İnternet siteleri aracılığı ile sunulan hizmetlerde de, herkesin erişiminin sağlanması en önemli beklentidir.

Engelli kullanıcılar, bazı özel teknolojiler sayesinde siteye bağlanıyor olabilir. Görme engelli kullanıcılar sesli tarayıcılar sayesinde siteye erişirken, eski bilgisayar sistemlerine sahip kullanıcılar eski tarayıcılar kullanıyor olabilir. Bazı kullanıcılar sadece metin gösteren tarayıcılar üzerinden erişirken kimi kullanıcılar en güncel teknolojiden faydalanan olabilir.

Kullanıcıların siteye erişimde farklı düzeylerde teknolojiler kullanabilmelerinin yanı sıra, kullanıcıların teknik bilgi düzeyleri arasında büyük oranda farklılıklar da olabilir.

Kamu internet siteleri tasarlanırken tüm bu teknoloji ya da bilgi düzeyi farklılıklarının kullanıcıların alacağı hizmette değişiklik yaratmaması, herkese aynı hizmetin sunulabilmesi amaçlanmalıdır. Site tasarımında esneklik ve uyulanabilirlik özellikleri dikkate alınmalıdır.

Geniş kullanıcı kesiminin siteye erişebilir duruma getirilmesi, site tasarımının görsel özelliklerinden ödün vermeyi gerektirse dahi, yaygın erişilebilirliğin sağlanması kamu internet sitelerinin temel varolma nedenidir. Özellikle engelli kullanıcıların siteye erişiminin sağlanması, site tasarımına başlarken dikkate alınması gereken bir husustur. Tasarımı bitmiş bir sitenin, sesli tarayıcılar gibi engelli destek araçlarına uygun hale getirilmesine çalışmak yerine, site bu gereklilik göz önüne alınarak tasarlanmalıdır.

3.1.1 Tarayıcılar

Kamu internet siteleri hiçbir zaman tek bir tarayıcı tipi için tasarlanmamalıdır. Bir tarayıcının kullanım yaygınlığının yüksek oranlara ulaşması, o tarayıcıyı kamu internet sayfaları için hedef konuma getirmez. Kamu internet sitesi tasarlayıcıları kullanım yaygınlığına bakmadan mümkün olduğunca çok sayıda tarayıcıda aynı hizmeti, aynı içerik ve görünümde sunabilecek siteler hazırlamakla yükümlüdür.

Tarayıcı uyumluluğunun sağlanmasının en temel yolu, gerek tasarım sırasında, gerekse tasarım sonrasında, sitenin değişik tarayıcılar ile uyumunun test edilmesi ve görülen sorunların giderilmesidir.

Kamu internet sayfalarında “Bu sayfalar en iyi xxx tarayıcısında görüntülenir” gibi belli bir ürünü adresleyen ibareler yer almamalıdır.

3.1.2 Ekran Çözünürlüğü

Kamu internet siteleri, farklı ekran çözünürlüklerinde sorunsuz okunabilecek şekilde tasarlanmalıdır. Tek bir çözünürlük seçilerek, o çözünürlükte mükemmel sonuç almak, buna karşılık diğer çözünürlüklerde kaymalar ve problemlerle karşılaşma durumu yaşanmamalıdır.

Farklı ekran çözünürlüklerinde sorunsuz görüntülenmeyi elde edebilmek için “Katlı Sayfa Yönergeleri” (Cascading Style Sheets-CSS) gibi sayfa yönergesi tekniklerinden faydalanılmalı, ayrıca çok iyi test edilmelidir.

Sayfanın altına “Bu sayfalar en iyi xxx ekran çözünürlüğünde görüntülenir” gibi ibareler koymaktan da ayrıca sakınılmalıdır.

3.2 Teknik Özellikler

3.2.1 Bağlantılı Metin Dili

Kamu internet siteleri, açık internet standartları kullanılarak tasarlanmalıdır (örneğin; The World Wide Web Consortium -HTML).

Aksini gerektiren özel bir durum olmadıkça, bağlantılı metin dili olarak HTML kullanılmalıdır. Çeşitli nedenlerle başka bir teknolojinin kullanılması gerekliliği durumunda dahi, sitenin alternatif bir HTML sürümü mutlaka hazırlanmalıdır. HTML'in geçerli en yeni sürümü olan HTML 4.01 kullanılmalıdır.¹

Hazırlanan sitede HTML'in doğru kullanılması erişilebilirliği artırmada önemli etkenlerden biri olacaktır. Hazırlanan sitenin HTML kullanımını açısından kontrolü için W3C'nin çevrimiçi geçerlilik kontrol sayfası önerilmektedir:

W3C HTML Geçerlilik Kontrol Sayfası : <http://validator.w3.org>

3.2.2 Sayfa Yönergeleri (Style Sheets)

Sayfa yönergeleri, tarayıcıların internet sayfalarının görünümünü değiştirmede kullandığı metin dosyalarıdır. Sayfa yönergeleri, aynı içeriğin farklı amaçlar için farklı şekillerde görüntülenmesini sağlar ve bunu yaparken de içeriğin yeniden oluşturulmasını gerektirmez.

W3C'nin önerisi olan Katlı Sayfa Yönergelerinin (Cascading Style Sheets) kullanılması, siteyi daha uyarlanabilir dolayısıyla da daha kolay erişilebilir kılacaktır. Bunun yanında sayfa yönergelerinin kullanılması site bakımını ve güncellenmesini de kolaylaştıracaktır. Sayfa yönergisinde yapılacak bir değişiklik, sitedeki tüm sayfaları birden etkileyecektir.

Sayfa yönergeleri genellikle daha yeni olan tarayıcılar tarafından desteklenmektedir. Görsel olmayan, metin tabanlı ya da sesli tarayıcılar üzerinde etkileri olmamaktadır.

Sayfa yönergeleri, HTML ile birlikte kullanılmalıdır. Görünüm doğrudan yönergelerin kontrolüne bırakılmamalı, metine dönük HTML kontrolü yapılmalıdır (Başlık ve vurgu etiketlerinin kullanımı gibi : <h1>, <h2>, ,). Bu sayede görsel olmayan ya da sayfa yönergelerini desteklemeyen tarayıcıları kullananlar için bir sorun oluşmayacaktır.

Yaygın kullanımı bulunmayan sayfa yönergesi tekniklerinden sakınılmalı, hazırlanan sayfalar çeşitli tarayıcılarda test edilmelidir.

Hazırlanan sitenin sayfa yönergeleri kullanımı açısından kontrolü için W3C'nin CSS geçerlilik kontrol sayfası önerilmektedir:

*W3C CSS Geçerlilik Kontrol Sayfası :
<http://jigsaw.w3.org/css-validator/validator-uri.html>*

3.2.3 Betikleme (Scripting)

Betikleme dillerinin kullanımı, internet sitesinin kullanıcılarına daha gelişmiş sunum ve dolaşım özellikleri ile birlikte etkileşimli içerik de sunabilmesini sağlamaktadır. Bu getirilerin yanında betikleme kullanımının site erişilebilirliğini azaltma riski de bulunmaktadır.

Bazı kurumlar ve kullanıcılar tarayıcılarında betiklerin işlemesine izin vermemekte, bazı tarayıcılar bazı tür betikleri desteklememektedir. Betikleme ile yaratılan dinamik aşağı açılan menüler (drop-down menu) hareket ve görme kısıtlı (örneğin; ekran büyüteçleri kullanılması durumu) kullanıcılarda önemli erişim sorunları yaratmaktadır.

¹ e-Dönüşüm Türkiye Projesi Birlikte Çalışabilirlik Esasları Rehberi , T.C. Başbakanlık DPT Müsteşarlığı, 2005

Betikleme dilleri hakkında temel bir endişe de bir tür güvenlik açığına neden olma olasılıklarıdır.

Bu nedenlerden dolayı, kamu sitelerinde betikleme kullanımı çok gerekli olan durumlarla sınırlandırılmalıdır. Betikleme kullanılması durumunda, mutlaka sayfanın metin tabanlı alternatif sürümü de var olmalıdır. Kullanıcının betikleri görememesi, herhangi bir içerik eksikliğine neden olmamalıdır.

3.2.4 Yardımcı Veri (Metadata)

Sayfaların arama motorlarına kaydını kolaylaştırabilmek için, site ana sayfasında, ana bölüm sayfalarında ve arama motorlarına kaydolması gerektiği düşünülen diğer tüm sayfalarda yardımcı veri kullanılması gereklidir.

Yardımcı veri, HTML biçimindeki her belgenin başlık bölümüne eklenebilir. Özellikle tablo içeren HTML sayfalarında yardımcı veri kullanımı gereklidir.

Sitelerde yardımcı veri kullanımında, e-Devlet Metaveri Standardı çalışmasına² uyum gözetilmesi gerekmektedir.

3.2.5 Dosyalar

3.2.5.1 Dosya İsimlendirme

Sitenin içerdiği dosyaların isimlendirilmesinde aşağıdaki kurallara uyulması önemlidir:

- Dosya isimleri küçük harflerden oluşmalıdır.
- Dosya ismi boşluk karakteri içermemelidir.
- “_” karakteri dışında herhangi bir noktalama işareti dosya isminde yer almamalıdır.
- Dosya isimleri 50 karakterden uzun olmamalıdır.
- Dosya isimleri dosyayı tanımlayıcı olmalıdır.
- HTML dosya uzantıları (htm, html, shtml) mutlaka kullanılmalıdır. Tüm site kapsamında aynı tür dosyalar için aynı dosya uzantısı kullanılmalıdır (örneğin; tüm HTML dosyalar için “.html” uzantısı kullanılması).

3.2.5.2 Dosya Boyutları

Düşük bağlantı hızlı kullanıcılar için, dosya boyutlarının büyüklüğü, uzun indirme süresi sorunu doğurabilir. Bunun önüne geçmenin yolu, internet sitesini oluşturan dosyaların boyutlarını mümkün olduğunca küçük tutmaktır.

Bir internet dosyasının boyutu, ilgili HTML dosyasının, içerdiği resimlerin ve betiklerin büyüklüklerinin toplamından oluşmaktadır.

Ana sayfa, kullanıcının site içinde gitmek istediği yönü belirlemesini sağladığından ana sayfanın boyutu mümkün olduğunca küçük olmalıdır. Bu sayede kendi erişim hızı ne kadar yavaş olursa olsun, tüm kullanıcılar ana sayfaya kolaylıkla erişebileceklerdir.

Siteyi oluşturan sayfalarda, bilgilerin boyutu büyük sayfalarda toplanmasından ziyade, kolay erişilecek şekilde küçük boyutlu sayfalara bölünmesi ve bölünen bu sayfalar arasındaki bağlantıların da çok iyi kurulması gerekmektedir.

İdeal koşullarda, 50 Kb'lik bir internet sayfasının yüklenmesi, 14.4 kbps'lik bağlantıda 27 sn, 56 kbps'lik bir bağlantıda ise 7 sn sürmektedir.

Bu değerlendirmeler ışığında, kamu internet sitelerinde yer alan dosyalar için şu önerilerde bulunulabilir:

- Ana sayfa için önerilen dosya boyutu tüm resimler, betikler dahil olmak üzere en fazla 50 Kb olmalıdır.
- Diğer bilgi sayfalarının dosya boyutu en fazla 120 Kb olmalıdır.

² Bu amaçla e-Dönüşüm Türkiye 2005 Eylem Planı'nda 35 No'lu "Birlikte Çalışabilirlik İçin Veri Paylaşımı" eylemine yer verilmiş ve çalışmalar sürdürülmektedir.

- Özel durumlarda (raporlar, istatistiksel veriler gibi kullanıcının bir seferde basmak isteyeceği bilgiler) toplam dosya boyutu 300 Kb'yi geçmemelidir. Bu durumlarda, kullanıcı dosya boyutu için uyarılmalıdır.

3.2.6 Kodlama

İnternet sayfaları ister görsel tasarım araçları sayesinde hazırlansın, isterse doğrudan bir metin editöründe HTML olarak yazılsın bir HTML kodundan oluşmaktadır. Bu kodun W3C önerilerine uygun olarak HTML 4.01 standardında yazılmış olması gerekmektedir.

Çoğu tarayıcı HTML spesifikasyonlarına tam uymayan sayfaları da gösterebilmektedir. Ancak, kamu kurumları açısından, bir sayfanın tarayıcılar tarafından görüntülenebiliyor olması yeterli değildir. Gerekli olan sayfaların HTML kurallarına, W3C önerilerine tam uygun olarak hazırlanması, sitenin bütünlüğünü, güvenilirliğini ve tanınırlığını zora sokmamasıdır.

Tasarımcılar, kodlamada belli bir yazım düzeni içinde hareket etmeli, kısa vadeli çözümlere kaçmamalıdır. Kurallara, standartlara uygun ve belli bir düzen içinde hazırlanmış olan sayfaların bakımı ve güncellenmesi de şüphesiz çok daha kolay olacaktır.

3.2.6.1 Doküman Tipi Bildirimi

Doküman tipi bildirim, dokümanın içerdiği etiketlerin nasıl yorumlanacağını belirtmektedir. Tip bildirim bir doküman tip tanımını ve bu tanımın nerede bulunduğunu göstermektedir. Kamu kurumları internet sitelerinin içerdiği tüm dokümanlar tip bildirimini mutlaka barındırmalıdır. Örnek tip bildirim aşağıda yer almaktadır:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

3.2.6.2 Sayfa Başlığı

Sayfa başlığı etiketlerinin kullanılması, hem kullanıcının site içi dolaşımını, hem arama motorlarına kaydı, hem de site yönetimini kolaylaştırmaktadır. Bu yüzden, sayfanın başlık alanında (<HEAD>) doküman başlığı her zaman yer almalıdır (<TITLE>). Kullanılan başlık, dokümanın amacını açıklamaya yeterli olmalı, dokümanla uyumsuz olmamalıdır. Sitenin tüm sayfalarında aynı başlık kullanılmamalı, sayfaya özel, açıklayıcı bir başlık seçilmelidir.

Başlık bilgisi site içi sayfalar arasında tutarlı ve bir kurala bağlı olmalıdır. Başlık, sayfa içeriğini açıklayabilmelidir. Örneğin; "Kurum-Bölüm-Sayfa Adı" gibi, siteye ait tüm sayfalarda kullanılacak bir söz dizimi belirlenmeli ve istisnasız tüm sayfalarda kullanılmalıdır.

Başlıklar mümkün olan durumlarda 30 karakterden, en kötü durumda 60 karakterden uzun olmamalıdır. Uzun başlıkların tarayıcılar tarafından kesilebileceği ve tam açıklayıcı olmayan "Sık Kullanılanlar" kısa yollarına neden olabileceği unutulmamalıdır.

3.2.6.3 Doğru Etiketleme

Kodun HTML standardına uygun yazılması doğru etiketleme yapılmasını da gerektirmektedir. Tüm etiketler amaçlarına uygun olarak kullanılmalıdır. Açılan tüm etiketler mutlaka kapanmalıdır. Etiket öznitelikleri uygun sırada yazılmalı ve kullanılan bu sıra, site içindeki tüm dokümanlar arasında tutarlı olmalıdır.

BODY etiketleri sayfada yer almalıdır. BODY etiketinde text, link, vlink, alink öznitelikleri de kullanılmalıdır.

Doğru etiketlemenin uzun vadede zaman ve emek tasarrufu sağlayacağı unutulmamalıdır.

3.2.6.4 Yorumlar (Comments)

Yorum kodları sitenin bakımı ve güncellenmesi sırasında bakım ekibine kolaylık sağlamaktadır. Yorum kodları gerekli görülen, açıklama ihtiyacı duyulan yerlerde kullanılmalıdır. Doküman boyutunu büyütmemek amacıyla gerekli görülmeyen yerlerde yorum kodu kullanılmamalıdır. Açıklama ihtiyacı ve doküman boyutunun büyümesi göz önünde tutularak yorum kodlarının gereğinden fazla olmaması sağlanmalıdır. Yorum

kodlarında birey ya da şirket isimleri yer almamalı, teşekkür etme gibi amaçlarla yazılmamalıdır.

3.2.6.5 Şablonlar ve Kod Dizimi (Templates)

Hazırlanan HTML kodları, yeniden kullanılabilirliği sağlayabilmek amacıyla şablon yapısında tasarlanmalıdır. Hazırlanan sayfalar içinde yorum kodları kullanılarak içerik, işlev ve sürüm bilgileri verilmelidir.

Şablon yapısında hazırlanan sayfalardan yeni sayfalar türetilmesi çok daha hızlı ve kolay olacaktır.

Kod içinde etiketler tüm sayfalarda tutarlı bir sırada dizilmelidir. Gerek etiketlerin kullanımı, gerekse etiket özniteliklerinin kullanımında tutarlı bir sıranın izlenmesi sayfaların güncellenmesini ve bakımını kolaylaştıracaktır. Kod kalitesini kontrol etmek amacıyla açık kaynak kodlu HTML Tidy editörü kullanılabilir.

HTML Tidy: <http://tidy.sourceforge.net/>

3.2.7 Formlar

Ekranda doldurulması ve çevrimiçi olarak gönderilmesi amacıyla tasarlanan formlar mutlaka HTML olarak hazırlanmalıdır. Düzgün olarak tasarlanmış HTML formların görme engelli kişilerce de doldurulabileceği, ama bu kişilerin basılı formları kullanmadığı göz önünde tutularak formun erişilebilirliğine azami dikkat gösterilmelidir.

Form elemanları basit bir şekilde yerleştirilmelidir. İki sütundan oluşan ve etiketlerin solda, doldurulacak alanların sağda olduğu bir yerleşim hem görme engelli hem de diğer kullanıcılar için yeterli olacaktır.

<fieldset> etiketi ilgili form elemanlarını gruplamak amacıyla kullanılmalıdır. Bu etiketin, "legend" özniteliği gruba başlık atamak için kullanılabilir.

Tüm açıklayıcı etiketler <label> etiketi içinde verilmeli ve alan adı ile ilişkilendirilmelidir.

Alan içindeki sekme sırası kontrol edilmeli ve doğru dizilimde olmalıdır. Formda doldurulması mecburi olan alanların ya da önemli bilgilerin gösteriminde sadece farklı renk kullanımına güvenilmemelidir. Örneğin; kırmızı metinler renk körü kullanıcılar için anlamlı olmayacaktır.

Form ile ilgili tüm yardım bilgileri de çevrimiçi olarak verilmelidir.

3.2.7.1 Formların İşlenmesi

Kullanıcı doldurduğu formun işleme durumu ile ilgili bilgilendirilmelidir. Form içeriği alındığında kullanıcıya geri bildirimde bulunulmalı, örneğin bu durumu bildiren bir sayfa gösterilmelidir.

3.2.8 Çerezler (Cookies)

Kamu internet siteleri mecbur kalmadıkça çerez kullanmamalıdır. Site işlevselliği hiçbir şekilde çerez kullanımına dayanmamalıdır. Site kullanımının izlenmesi amacıyla da çerezlerden faydalanılmamalıdır. Oturum kontrolü yapılması gerektiğinde çerez kullanımı yerine, olabildiğince sunucu tarafında oturum kontrolü tercih edilmelidir.

3.3 Genel Görünüm

3.3.1 Ana Sayfa

Bir internet sitesi, bir alan adı ile ilişkilendirilmiş ve ana sayfadan başlayarak bir dizi bağlantı sonucunda tümüne ulaşılabilen bir internet sayfaları koleksiyonu olarak değerlendirilebilir.

Ana sayfa, bir internet sitesinin en temel elemanı ve sitenin internet adresi ile ("www.kamukurumu.gov.tr") ulaşılan ilk sayfasıdır. Sitenin içeriğinden habersiz olup sadece alan adını bilen tüm kullanıcıların ulaşacakları ilk yer sitenin ana sayfasıdır. Kullanıcıların site

hakkındaki ilk izlenimleri ana sayfayı görmeleri ile oluşacaktır. Genel anlamda sitenin en fazla bağlantı alan sayfası da ana sayfadır. Kullanıcılar çoğunlukla alt sayfaların adreslerini yazmak yerine ana sayfa bağlantılarını kullanmayı tercih ettiklerinden siteye bağlanan her kullanıcı ana sayfayı en az bir kez ziyaret etmiş olacaktır.

Ana sayfa sitenin tüm temel alanlarını, bu alanlara nasıl ulaşılabileceğini ve sitenin temel var olma amacını içermelidir. Ana sayfa merkezi konumu nedeni ile kimi zaman kurumdan haberler gibi bazı özel amaçlı bilgileri vermede kullanılsa da genellikle içeriği sade olmalıdır.

Genelde ana sayfadan bağlantı verilen sayfalar daha geniş içeriğe ve daha az bağlantıya sahip olmaktadır. Ana sayfadan bağlanılan sayfalar çoğunlukla her sayfada tekrarlanan bir dizi dolaşım ögesi içermektedir.

Ana sayfaya ulaşmak için doğrudan site alan adının yazılması yeterli olmalıdır (örneğin; www.kamukurumu.gov.tr). Dosya adının adrese eklenmesi gerekmemelidir (örneğin; www.kamukurumu.gov.tr/index.html).

Ana sayfa hariç sitenin içerdiği tüm sayfalar, ana sayfaya bağlantı içermelidir. Bu bağlantının kurum logosu üzerinden verilmesi genel yoldur.

3.3.2 Logolar

Kurumun logosunun en yeni sürümü sitenin ana sayfasında yer almalıdır. Site bütünlüğünü sağlayabilmek amacıyla siteye ait tüm sayfalarda da aynı logo ya da küçültülmüş hali veya kurum ismi bulunmalıdır.

Kullanılan logo üzerinde yer alan tüm metinler okunabilir olmalıdır. Logo çok renkli olmadığı sürece GIF resim formatında olmalıdır. Logonun saydam (transparent) olması durumunda, sayfanın arka plan rengi kullanılarak logonun örtüşmesi önlenmelidir (anti-alias).

3.3.3 Çerçeveseler (Frames)

Kamu internet siteleri tasarımında çok zorunlu olmadıkça çerçeveler kullanılmamalıdır.

Site tasarımında çerçevelerin kullanılması erişilebilirlik sorunlarına yol açmaktadır. Çerçeveler bazı eski tarayıcılarda desteklenmemekte, sesli ya da metin tabanlı tarayıcılarda ise sorunlar yaratmaktadır. Ayrıca site için çerçeveler kullanarak bir dolaşım sistemi kurulması, site boyutunu artırıcı bir etkidir.

Çerçeveler bir doküman olmayıp, bir barındırıcı görevi görmektedir. Bir internet adresinin doğrudan bir dokümanı işaretlemesi yerine içi boş olabilecek bir taşıyıcıyı adreslemesi istenen bir durum değildir.

Çerçeve kullanılması durumunda oluşan içeriğin ve dolaşım öğelerinin ayrı dokümanlarda tutulması, internet dokümanlarının beklenen yapısına ters düşmektedir.

Yol açtıkları bütün bu olumsuz durumlar nedeniyle kamu internet sitelerinde çerçeveler kullanılmamalıdır.

Yine de çeşitli nedenlerle site tasarımında çerçevelerin kullanılması gerekiyorsa, bu durumda da sitenin içerik olarak aynı özelliklere sahip çerçeve içermeyen bir –No Frames- sürümü de hazırlanmalıdır. Çerçeve içermeyen sürüme, ana sayfada sol üst köşeden bağlantı verilmelidir.

3.3.4 Site İçi Dolaşım

Kamu internet sitelerinde, site içinde dolaşım kolay olmalıdır. Sitenin genel görünümü yanında, site içi dolaşım da göz ardı edilmemelidir. Kolay bir dolaşım sisteminin varlığı sitenin işlevselliğini artırıcı temel öğelerden biridir. Unutulmamalıdır ki, sunulan bilgi ve servisler ancak ulaşılabilir olmaları durumunda faydalıdır.

Kullanıcılar site içi dolaşımı sağlayabilmek için herhangi bir ilave yazılım bileşeni yüklemek zorunda kalmamalıdır.

Kamu internet sitelerinin şu temel dolaşım kriterlerini sağlaması gerekmektedir:

- Her sayfadan ana sayfaya kolay görünür bir bağlantı olmalı
- Ana sayfa site içeriğine ilişkin ana menüyü içermeli
- Ana sayfadan sitenin tüm temel bölümlerine bir tıklama ile erişim mümkün olmalı
- Kurumun logosu veya adı sitede bulunan her sayfada yer almalı
- Sayfalarda yer alan bağlantılar, bağlantının ait olduğu yere ilişkin kısa açıklamalar içermeli, sadece “tıkla” gibi ibareler olmamalı
- Bağlantılar mümkün olduğunca metin üzerinden verilmeli

3.3.4.1 Bağlantılar (Links)

Site içi bağlantılar mümkün olduğunca metinler üzerinden verilmelidir. 8 karakterli bir kelimenin resim olması durumunda boyutu en az 50 kat fazla olacaktır. Ayrıca engelli kullanıcılar metni kendi ihtiyaçlarına göre büyütebilir, renklendirebilir, okutabilir. Resimler üzerinde bu tür işlemler mümkün olmayacaktır. Görsel gerekler nedeniyle bağlantı vermede resim kullanılması gerekmekte ise, aynı bağlantının ayrıca bir metin üzerinden de verilmesi resim görüntüleyemeyen kullanıcıların bağlantıya erişimini mümkün kılacaktır. Tüm kullanıcıların internet sitelerini resimler ile birlikte görüntülemedikleri veya görüntüleyemedikleri unutulmamalı, resim üzerinden bağlantı vermekten mümkün oldukça kaçınılmalıdır.

Resim ile bağlantı verilmesi durumunda “Alt” etiketleri mutlaka kullanılmalıdır. İlave metin bağlantısı verilmesi durumunda dahi bu uygulamadan vazgeçilmemelidir.

3.3.4.2 Menüler

Kamu sitelerinde kullanılan menülerin HTML olarak tasarlanmış metin menüler olması önerilmektedir. Çeşitli nedenlerle, özellikle estetik çekinceler ve HTML dışı teknikler ile oluşturulan menüler kullanılması durumunda (örneğin; Javascript, Shockwave) standart HTML kullanılarak alternatif bir menü de hazırlanmalıdır.

Metinden oluşan menüler, genelde bir grup bağlantı içermektedir. Bu bağlantılar arasında doğru noktalama işaretleri kullanılması erişilebilirlik teknolojilerini kullanan okuyucular için çok önemlidir. WAI Erişilebilirlik Rehberi, menü öğelerinin “|” karakteri ile ayrılmasını önermektedir. Eğer bu karakterin kullanımının menünün genel görünümünü bozduğu düşünülüyorsa, bu karakter, arka plan rengi ile aynı renkte yapılabilir. Bu sayede, çoğu kullanıcı tarafından görülmeyecek, ancak örneğin hareket engelli kullanıcıların işleri de kolaylaşacaktır.

Metin menüsünü oluşturan öğelere sekme numarası değerleri atanması, fare kullanamayan kullanıcıların menüyü klavye yardımı ile kullanabilmelerini sağlayacaktır.

Javascript menüler, bazı kullanıcılar tarafından görüntülenmek istenmeyebilir ya da görüntülenemeyebilir, “ateş duvarları”(fire-wall) tarafından bloklanabilir, fare kullanmakta zorlanan kullanıcılar için kullanım zorluğu yaratabilir, bazı eski tarayıcılarda hatalara yol açabilir. Bu tür bir menü kullanılmasının eksileri ile getirileri karşılaştırılmalı ve site için uygun çözüm bulunmalıdır.

3.3.4.3 Resim Haritası (Imagemap)

Resim haritası üzerinden bağlantı verme yöntemi, ancak resim üzerinde ayrılan bölümlerin verilen bağlantıları anlamlı kılması durumunda (örneğin; ülke haritası üzerinden bölgelere erişim) kullanılmalıdır. Resim haritası kullanılması durumunda sayfada bir metin alternatifi de yer almalıdır.

Resim haritası üzerindeki tüm bağlantı seçenekleri için “Alt” etiketleri mutlaka kullanılmalıdır.

3.3.5 Metin

HTML özünde zengin metin işleme özelliklerine sahip olan bir metin gösterim standardıdır. Metin işleme etiketlerinin kurallara uygun olarak kullanılması, HTML sayfaların erişilebilirliğini arttıran özelliklerden biridir.

Bir internet sayfasının yazımının iyi olması, zayıf görme yeteneği olan kullanıcılarca okunabilirliği arttırabilmek adına yazı büyüklüklerinin ve renklerinin değiştirilebilmesi ama yine de elemanları arasında görsel uyumun korunabilmesini gerektirir.

Kamu internet sayfalarında yer alması istenen tüm kritik bilgiler metin/HTML formatında verilmelidir. Bilgiler resim içinde yer almamalıdır. Çeşitli nedenlerle resim içinde verilmesi gereken bilgilerin metin alternatifi de sayfada sunulmalıdır.

Metnin renk özelliği kullanılarak vurgular yapılmamalıdır. Renklerin tüm kullanıcılarca görüntülenememe riski olduğu unutulmamalıdır.

3.3.5.1 Yazı Tipi (Font)

Kamu internet sayfalarında yer alacak yazı tipleri, yaygın kullanım alanı olan yazı tipleri arasından seçilmelidir. Kullanıcıların sayfalardaki metinleri görebilmek için yeni yazı tipleri ya da yazılımlar yüklemeleri gerekmemelidir.

Yazı tiplerinin etiketleri kullanmak yerine sayfa yönergeleri içinde verilmeleri tercih edilmelidir. Yazı tipleri tercih sırasına göre alternatifli olarak belirlenmelidir (örneğin; Arial, Helvetica, Sans-Serif) .

Bir metnin yazımında özel bir yazı tipi kullanımı gerekiyorsa, bu durumda metin bir resim olarak hazırlanmalı ve sayfaya metin alternatifi ile birlikte bu resim yerleştirilmelidir. Bu türden resim kullanımı asgari düzeyde tutulmalıdır.

Yazı Tipi Büyüklüğü:

Yazı tipi büyüklüğü, kullanıcıların kendi ihtiyaçlarına göre değiştirebilmesine imkan verecek şekilde her zaman belirtilmelidir. Tarayıcılar arasında farklı sonuçlara yol açabilecek yüzde (%) gibi bağıl büyüklükler yerine tüm tarayıcılarda tutarlı sonuçlar veren boyut isimleri (xx-small, x-small, small, medium, large, x-large, xx-large) kullanılmalıdır.

Yazı Tipi Rengi:

Yazı tipi rengi, sayfa arka plan rengine zıt bir renkte seçilmelidir. Okuma imkanını sınırlandırdığından, desenli arka plan kullanımından sakınılmalıdır. Özellikle sınırlı görme yeteneği olan kullanıcılar için desenli arka plan kullanılmaması önemlidir. Bunun yanında arka planda metinlerin okunmasını engelleyebilecek arka plan resimleri kullanılmasından da kaçınılmalıdır.

Sayfanın içerdiği metnin anlamının güçlendirilmesinde renk kullanımı tek yöntem olarak düşünülmemelidir. Örneğin; önemli bir bilginin kırmızı renk ile yazılması renk körü bir kullanıcı ya da siyah beyaz monitör kullanan bir kullanıcı için hiçbir anlam ifade etmeyecektir. Bu gibi durumlarda metnin vurgusunun etiketi ile güçlendirilmesi daha anlamlı olacaktır.

Beyaz renkli metin kullanımından çok gerekli olmadıkça kaçınılmalıdır. Eğer arka plan renkleri kullanılmadan yazıcı çıktısı alınmak istenirse, beyaz renkli metin basılamayacaktır.

Diğer Yazı Tipi Özellikleri:

Kamu internet sayfalarında yer alan metinler statik olmalıdır. Kayan ya da parlayan yazı etiketleri tarayıcı bağımlı olduklarından kullanılmamalıdır.

Bağlantılar ile karışmaması için altı çizili metin kullanımından sakınılmalıdır.

Vurgu yaratmak amacıyla harfler arasında boşluk bırakılmamalıdır.

Tümüyle büyük harf kullanımı ve italik yazı kullanımı çok yoğun olmamalıdır. İki tekniğin fazla kullanılması da sayfanın okunabilirliğini zorlaştıracaktır.

3.3.6 Resimler

Sayfalarda kullanılacak resimlerin GIF veya JPG (JPEG) resim formatlarında olması tavsiye edilmektedir. PNG ve SVG resim formatlarının kullanımları sınırlı olduğundan bu aşamada önerilmemektedir.

Sınırlı sayıda renk içeren resimler ve tek renkli geniş alanlar içeren resimler için GIF formatı kullanılmalıdır.

Hareketli GIF kullanılması önerilmemektedir, ancak kullanılması durumunda toplam boyut 30K'yı geçmemelidir. Hareketli GIF'ler durmadan önce en fazla dört kez dönmelidir. Hareketli GIF'ler logolarda asla kullanılmamalıdır.

Boyut:

Resim boyutunun 10K'nın altında kalmasına çalışılmalı, en kötü durumda 30K üst limit olarak kabul edilmelidir. Büyük boyutlu resimler ana sayfada asla kullanılmamalıdır. Daha büyük boyutlu resimlerin kullanılması gerektiğinde tercihen pul imge (thumbnail) yöntemi kullanılmalıdır. Resmin boyutu yazılmalı, uyarı verilmelidir.

Hem GIF hem de JPEG resimler, boyutlarının düşürülmesi için kaliteyi bozmayacak şekilde sıkıştırılmalıdır.

“Alt” Etiketlerinin Kullanımı:

Tüm resimler için “Alt Text” formunda metin karşılıkları mutlaka kullanılmalıdır. Görme engelli kullanıcılar ya da resim yükleyemeyen/yüklemek istemeyen kullanıcılar için alt etiketleri resme bir alternatif oluşturacaktır.

Resimler için kullanılacak alt etiketleri o resmin, o sayfaya niçin konulduğunu açıklamalı ve 100 karakteri geçmemelidir. Alt etiketleri sesli tarayıcılarda sonraki metne karışmaması için nokta ve boşluk karakteri ile bitmelidir.

Tümüyle dekoratif amaçlar için kullanılan ve sayfa içeriği ile doğrudan ilgili olmayan resimlerin alt etiketleri “ ” (aç kapa çift tırnak) karakterlerinden oluşmalıdır.

Sayfa içi dolaşım için resim kullanılmaması önerilmesine rağmen, yine de kullanılması durumunda bu resimlere ait alt etiketleri resmin içeriği ile aynı olmalıdır.

Madde imli listelerde madde imi olarak resimler kullanılması durumunda, bu resimlerin alt etiketleri “ ” (aç kapa çift tırnak) veya “madde imi” olmalıdır.

Sayfada alt etiketi konmamış hiçbir resim olmamalıdır.

Resim İçinde Metin Kullanımı:

Alternatif bir HTML metin versiyonu olmadığı sürece, resimler metin içermemelidir. Bu duruma tek istisna, Latin alfabesi dışındaki karakterlerin yabancı dilde bilgi verme amacıyla kullanılmasının gerekmesidir.

Diğer Özellikler:

 etiketi içinde yükseklik (height) ve genişlik (width) öznitelikleri kullanılmalıdır. Bu sayede tarayıcıların sayfanın metin kısmını işleme süreleri kısılacak ve düşük bağlantı hızlı kullanıcılar resimler yüklenmeden metinleri okumaya başlayabileceklerdir.

3.3.7 Renkler

Bağlantılar ve metinler için 216 renkten oluşan Renk Paleti kullanılmalıdır. Mümkün olduğu durumlarda, hazırlanan resimler de Renk Paleti kullanılarak üretilmelidir. Arka plan renkleri yine Renk Paletinden seçilmelidir.

Arka plan renkleri ile metin renkleri arasında uyum olmalıdır. Renk körü kullanıcıların karşılaşılabileceği problemleri önlemek amacıyla kırmızı ve yeşil renklerin bir arada kullanılmasından sakınılmalıdır.

Sayfa tasarımında kullanılacak renkler; arka plan-metin-grafik renkleri; organizasyonun kurumsal kimliğine ters düşmemelidir. Kullanılan renkler, sayfa içeriğinin okunmasına engel olmamalı ve gözleri yormamalıdır.

Sayfa içindeki bölümlerden renkler kullanılarak bahsedilmemelidir. Örneğin, “kırmızı tabloda görüldüğü şekilde...” gibi ifadelerden sakınılmalıdır.

3.3.8 Yerleşim

İnternet sayfası esnek bir ortamdır. Sabit olmayan daha esnek tasarımların kullanılması sayfaları daha verimli yapacaktır. Esnek tasarımlı sayfalar, kullanıcı ihtiyaçlarına cevap verebilen, tarayıcı penceresi boyutu değiştirildiğinde sütun genişliklerini yeniden ayarlayan, metinlerin yerleşimini tekrar düzenleyen sayfalardır. Metin boyutu büyütüldüğünde ya da küçültüldüğünde görsel olarak sorunsuz çalışmaya devam ederler.

Esnek tasarımlı sayfalarda eleman öznitelikleri (boyut, konum, genişlik) mutlak değil, bağlı olarak tanımlanır.

3.3.9 Tablolar

Tabloların kullanımı bazı tarayıcılarda ve özellikle Erişilebilirlik Teknolojilerinden faydalanan kullanıcılarda bazı güçlükler yaratabilmektedir. Bunun yanında sağladığı faydalar dikkate alındığında, tablo kullanılmamasının beklenmesi anlamlı olmayacaktır.

Tasarımcılardan beklenen tablolarını azami dikkat ile kullanmaları ve mümkün olduğunca geniş bir tarayıcı ve erişim yöntemi için testlerini yapmalarıdır. Tablolar ile ilgili uyulması önerilen diğer yöntemler şunlardır:

- Sütun sayısı minimumda tutulmalıdır.
- İç içe tablo kullanımından mümkün mertebe sakınılmalıdır. Ancak alternatif yöntem olmaması durumunda kullanılmalıdır.
- Değişik tarayıcılar tabloları değişik şekillerde işleyebildiğinden, uyumluluk mutlaka sınanmalıdır.
- Table etiketinde *Col/s* özneliği mutlaka kullanılmalıdır.
- Eğer sabit genişlikli bir tablo kullanılıyorsa, genişlik hiçbir zaman 535 pixel'i geçmemelidir. Bu sayede tablonun standart kağıda çıktısı alınabilir.
- Tablo sonu etiketleri her zaman kullanılmalıdır.
- Bazı eski tarayıcılar tablo arka plan resimleri kullanımını desteklememektedir. Mümkün olduğunca bu yöntemden kaçınılmalıdır.
- Bazı eski tarayıcılar tablo içinde arka plan rengi kullanımını desteklememektedir.

3.3.10 Kurumsal Kimlik

Bir kamu kurumunun internet sitesi, o kurumun kimliğini yansıtmalıdır. Kullanıcı internet sitesinde, kurumun binalarında, dokümanlarında gördüğü renkleri, amblemleri, logoları arayacaktır. Bu tür aşinalık oluşturan öğelerin sitede yer alması, sitenin kurumu tanımlayan renkler kullanılarak hazırlanması sitenin kuruma aidiyetini güçlendirecektir.

Site içeriğinin kalitesine çok özen gösterilmelidir. Site içeriği imla hataları, yazım yanlışları, bozuk anlatımlar açısından çok sıkı denetlenmelidir. İçerik her zaman güncel tutulmalıdır.

Sitenin çevrimiçi dünyada kurumun başlıca temsilcisi olduğu hiçbir zaman unutulmamalıdır.

3.4 Test Yöntemleri

Kamu internet sitelerinde doğru ve doyurucu içerik sunmanın yanında gelen en temel beklenti sitenin mümkün olabilecek en geniş kullanıcı kitlesi tarafından erişilebilmesidir. Kullanıcıların sahip oldukları araçlar, bilgi düzeyleri, fiziksel engelleri siteye erişmelerini kısıtlamamalı, kamu internet siteleri mümkün olduğunca geniş bir kitle tarafından aynı içerikle görüntülenebilmelidir.

Bu temel hedefe ulaşmanın en önemli adımı, site tasarımının gerçekleştirilmesi sırasında, sitenin gerçek dünyada nasıl görüntüleneceğinin test edilmesidir. Bu testlerde öncelik, estetik

kaygılardan ziyade içeriğin değişik tarayıcılarda ve değişik teknolojilerde kayıpsız görüntülenebilmesine verilmelidir.

Özellikle, engelli kullanıcıların siteye erişimde sorun yaşamamaları temel amaç olmalıdır. Site, ekran okuyucular gibi erişim destekleyici teknolojilerde sorunsuz çalışabilmelidir.

Kamu internet siteleri üzerinde uygulanabilecek testler üç ana aşamada toplanabilir:

Teknik Testler:

Site kodunun kullanılan teknolojilerin gereklerine uygun olarak yazıldığını sınamak için yapılacak olan kontrollerdir. Yazılan kod; HTML, CSS kullanıldı ise Javascript ve diğer betikleme (script) dilleri gereklerini tam olarak sağlamalıdır.

Teknik testler, site yayına geçmeden önce yapılmalı, sorunlu görülen hiçbir sayfa sorunlar giderilmeden yayına çıkarılmamalıdır. HTML ve CSS kullanım kontrolü için 3.4.1 Araçlar bölümünde belirtilen internet sayfaları kullanılabilir.

Erişilebilirlik Testleri:

Sitenin gerek değişik tarayıcı tiplerinde, değişik işletim sistemlerinde ve ekran çözünürlüklerinde sorunsuzluğunu denetlemek için; gerekse destek teknolojileri kullanan kullanıcıların siteye sorunsuz erişip erişmediğini denetlemek için yapılması gerekli olan testlerdir.

Bu testler, çeşitli tarayıcılar ve destek teknolojileri ile sayfaların görüntülenmesi sınanarak yapılabileceği gibi, çevrimiçi bazı test sayfalarından da destek alınabilir. 3.4.1 Araçlar bölümünde örnekleri yer alan test sayfaları siteyi W3C Erişilebilirlik Kılavuzu'na göre sınayabilmektedir.

Kullanışlılık Testleri:

Sitenin kullanıcıların amaçlarına uygunluğunu ve kullanılabilirliğini sınamak amacıyla yapılacak olan testlerdir. Kullanışlılık testleri geliştirme sürecinin erken evrelerinde yapılmalı, bu sayede sorunlar olması durumunda tasarım fazla maliyet kaybı olmadan güncellenebilmelidir.

Testlerde mümkün olduğunca çeşitli tarayıcı tipleri, işletim sistemleri, bağlantı hızları ve bilgisayarlar kullanılarak gerçek dünyaya yaklaşılmaya çalışılmalıdır.

3.4.1 Araçlar

Hazırlanan sitenin HTML kullanımını açısından kontrolü için W3C'nin çevrimiçi geçerlilik kontrol sayfası önerilebilir.

W3C HTML Geçerlilik Kontrol Sayfası : <http://validator.w3.org>

Hazırlanan sitenin sayfa yönergeleri kullanımını açısından kontrolü için W3C'nin CSS geçerlilik kontrol sayfası önerilebilir.

W3C CSS Geçerlilik Kontrol Sayfası : <http://jigsaw.w3.org/css-validator/validator-uri.html>

Kod kalitesini kontrol etmek amacıyla açık kaynak kodlu HTML Tidy editörü kullanılabilir.

HTML Tidy: <http://tidy.sourceforge.net/>

Sitenin erişilebilirlik gereklerini sağlayıp sağlamadığını sınamak için WEBXACT sayfası kullanılabilir. Sayfa erişilebilirliğin yanında kalite ve güvenlik hususlarını da sınayabilmektedir.

WEBXACT : <http://webxact.watchfire.com/>

W3C'nin sayfalarında erişilebilirlik kontrolü için kullanılacak çok sayıda araç tanıtılmaktadır.

<http://www.w3.org/WAI/ER/existingtools.html>

4. YAYIMLAMA

Bu bölümde, kamu kurumları internet sitelerinin yayımlanmasına ilişkin öneriler yer almaktadır.

4.1 Alan Adı Belirlenmesi

Kamu kurumları internet sitelerinin kullanılacağı adresler mutlaka “.tr” uzantısı ile bitmelidir. Kamu kurumları, sınıflarına göre aşağıda verilen alan adı uzantılarını kullanabilir:

“.gov.tr”	T.C. Devlet kurum ve kuruluşları için alt alan adı
“.bel.tr”	T.C. İl, ilçe ve belde belediye teşkilatları için alt alan adı
“.pol.tr”	T.C. Emniyet teşkilatı için alt alan adı
“.mil.tr”	T.C. Askeri kurum ve kuruluşlar için alt alan adı
“.k12.tr”	T.C. MEB’ce onaylanmış ilköğretim ve liseler için alt alan adı
“.edu.tr”	T.C. YÖK’çe onaylanmış yüksek öğretim kurumları için alt alan adı
“.org.tr”	Vakıf, dernek, sivil toplum örgütü vb. kuruluşlar için alt alan adı

Türkiye’de alan adlarının dağıtım işlemleri Ortadoğu Teknik Üniversitesi tarafından yürütülmektedir (www.nic.tr). Kamu kurumları kendilerine uygun olan alan adlarını ilgili kurumun belirlediği kurallar çerçevesinde temin etmelidir.

4.2 İç Kaynak / Dış Kaynak Kullanımı

Kamu kurumu internet sayfalarına ilişkin içeriğin hazırlanması, sayfaların tasarlanması, programlanması, bir sunucuya yerleştirilerek internete açılması işlemlerinin gerçekleştirilmesinde kurumlar kendi iç kaynaklarından (personel, sunucu v.d.) yararlanabileceği gibi kurum kaynaklarının (özellikle insan kaynağı) yetersiz kalması durumunda, site tasarımı ve yayımı işlemleri için hizmet alımı yoluyla da gerçekleştirilebilir. Bu hizmetlerin alımı, kurumun tabi olduğu ihale hükümleri çerçevesinde yürütülecektir.

4.2.1 Site Tasarımı İçin Hizmet Alımı

Kamu kurumları, kendi internet sitesinin tasarımı ve yönetimini yapabilecek birimi ve yeterli insan gücü bulunmuyor ise bu hizmeti bir özel sektörden veya bir başka kamu kuruluşundan (üniversite, enstitü v.d.) alabilir. Hizmet alımlarında aşağıdaki hususlar dikkate alınmalıdır.

Hizmet alan kamu kurumu ile hizmet sağlayıcı arasında yapılan sözleşmede şu noktadaki sorumluluklar ayrıntılı şekilde belirtilmelidir:

- Tasarım
- İçerik belirleme
- Site bakımı ve güncellenmesi
- Site tanıtımı

Kamu kurumu, içeriğin ne şekilde sağlanacağı, güncellemelerin nasıl iletileceği gibi alanlarda kendine düşen sorumluluğu tanımlamalı, hizmet sağlayıcıdan beklentilerini açık olarak belirlemelidir.

Bu tür bir hizmet alımında, bu Kılavuzun gerekliliklerinin sağlanması ön koşullardan biri olmalıdır.

Hizmet sağlayıcı ile yapılacak olan bakım sözleşmesi aşağıdaki alanları mutlaka kapsamalıdır:

- Sözleşme süresi
- Güncelleme sıklığı, sayısı, aciliyeti
- Sözleşme süresince tasarımın kaç kez yenilenebileceği
- Hazırlanacak tasarım örneklerinin sayısı
- Sözleşme kapsamı dışında kalan, ancak ihtiyaç duyulabilecek ilave tasarımların maliyeti

Site tasarımının özel bir kuruluşa yaptırılması durumunda, sayfalara tasarımcı kuruluşa ait reklam içeriği bulunmamalıdır. Sayfalar tasarımcı kuruluşun logosunu ve/veya adını içermemelidir. Sadece site ile ilgili sorunlarda erişim adresi olarak tasarımcı kuruluşa ait bir e-posta adresi verilebilir.

Bu tür dış hizmet alımları özellikle etkileşimli hizmetler sunan kurumlar için uygulanabilir gözükmemektedir. Ancak, statik bilgi sunan küçük boyutlu kamu kurumları (örneğin; küçük belediyeler gibi) tarafından kullanılabilir.

4.2.2 Site Yayımlanması İçin Hizmet Alımı

Kamu kurumları, kendi geliştirdikleri siteleri, kendi sunucularında yayımlayabileceği gibi site barındırma hizmeti veren kamu veya özel kuruluşlardan da hizmet alabilecektir.

Site yayımlanması için alınacak hizmetin sağlaması gereken kriterler, sağlanamaması durumunda uygulanacak cezai yaptırımlar sözleşmede net şekilde belirlenmelidir.

Bir site yayımlama sözleşmesinde kapsanması gereken asgari hususlar aşağıda verilmektedir.

- **Bağlantı sürekliliği garantisi:** Verilecek bağlantı sürekliliği garantisi %98'den az olmamalıdır. Bağlantı sürekliliğinin nasıl ölçüleceği, ceza durumları ayrıca belirtilmelidir.
- **Güvenlik:** Sunucu güvenliğinin, kurum için gerekli seviyede olduğu teyit edilmelidir.
- **Yedekleme:** Hizmet sağlayıcı, alıcının belirleyeceği sıklıkta yedekleme yapacağını ve hata durumunda belirlenen sürede yedekten geri dönebileceğinin garantisini vermelidir.
- **Veritabanı entegrasyonu:** Veritabanı kullanımı gerektiren etkileşimli hizmet sunuluyor ya da yakın gelecekte sunulması öngörülüyor ise sunucu gerekli desteği vermelidir.
- **Betikleme:** Sunucu site için gerekli betikleme desteğini sağlayabilmelidir.
- **Sunucu İstatistikleri:** Sunucuya ait istatistikler belli periyotlarda alınabilmelidir.
- **Bant genişliği:** Site için gerekli bant genişliği belirlenmeli, ilave bant genişliğinin maliyeti kontrata dahil edilmelidir. Kurum ihtiyaçları ve hizmet sağlayıcının sunduğu kapasiteler kıyaslanmalı, en optimum bant genişliği çözümüne karar verilmelidir (Sınırsız ya da sabit miktar + sabit ücretli ilaveler).
- **Teknik Destek:** Hizmet sağlayıcının sunduğu teknik destek hizmeti değerlendirilmelidir.
- **Fiyatlandırma:** Hizmet sağlayıcının sunduğu fiyatlandırma politikası incelenmeli, gizli maliyetler içermediğinden emin olunmalıdır. Gizli maliyet doğurabilecek alanlar şunlardır:
 - İlave bant genişliği gereksinimi
 - İlave disk alanı gereksinimi
 - İlave yazılım gereksinimi

5. ERİŞİLEBİLİRLİK KONTROLÜ

WAI (WEB Accessibility Initiative), uluslararası geniş katılım ile kurulmuş olan ve engelli kullanıcıların internet erişilebilirliğinin sağlanmasına dönük çalışmalar yapan; kılavuzlar, destek materyalleri hazırlayan bir organizasyondur. (<http://www.w3.org/WAI/>)

WAI'nin hazırladığı rehberlerden biri İnternet Sayfaları Erişilebilirlik Rehberi'dir (Web Content Accessibility Guidelines - WCAG). Bu rehberin amacı engelli kullanıcılar tarafından erişilebilir internet sayfaları hazırlanmasına yol göstermektir. Rehber, internet sayfaları ya da

uygulamalarında yer alan tüm içerik için, yani metinler, resimler, formlar, sesler ve diğer öğeler için erişilebilirliği artırıcı önerilerde bulunmaktadır.

WCAG'ın şu an güncel olan 1.0 sürümü 14 adet temel öneri alanı içermekte ve her alanda da bir ya da daha fazla denetim noktası tanımlamaktadır. Denetim noktaları üç derece üzerinden önceliklendirilmiştir.

1.Öncelik

Birinci öncelik mutlaka sağlanmalıdır. Sağlanmaması durumunda, bir ya da daha fazla kullanıcı grubu dokümandaki bilgiye erişemeyecektir. Denetim noktasının gerektirdiklerinin sağlanması, bazı grupların sayfaları kullanabilmelerinin temel koşuludur.

2.Öncelik

İkinci öncelik sağlanmalıdır. Sağlanmaması durumunda, bir ya da daha fazla kullanıcı grubu dokümandaki bilgiye zorlukla erişebileceklerdir. Denetim noktasının gerektirdiklerinin sağlanması, bazı grupların sayfalara erişiminin önündeki büyük engelleri kaldıracaktır.

3.Öncelik

Üçüncü öncelik sağlanabilir. Sağlanmaması durumunda, bir ya da daha fazla grup, dokümandaki bilgiye erişimde bazı sorunlar yaşayabilecektir. Denetim noktasının gerektirdiklerinin sağlanması, sayfalara erişimi kolaylaştıracaktır.

5.1 WCAG'ın Temel Önerileri

WAI Internet Sayfaları Erişilebilirlik Rehberi (WCAG) 1.0 sürümü 14 adet temel öneri alanı tanımlamaktadır. Bu öneriler aşağıda özetlenmektedir:

Sesli ve Görsel İçerik:

Sitede sunulan sesli veya görsel içeriğin aynı işlevsellikte metin eşdeğeri de sağlanmalıdır.

Renkler:

Sadece renklere dayanarak anlatımda bulunulmamalı, metin ve resimlerin renksiz görünümde de anlaşılabilir olması sağlanmalıdır.

Bağlantılı Metin Dili ve Sayfa Yönergeleri:

Bağlantılı metin dili (HTML) kurallarına uygun olarak kullanılmalıdır. Sayfa görünümü için görünüm belirleyen etiketler ve öznitelikleri yerine sayfa yönergelerinden faydalanılmalıdır.

Doğal Dil Kullanımı:

Doküman içinde farklı dilde alanlar mevcut ise, bu durum bağlantılı metin dili etiketleri ile belirlenmelidir. Bu sayede ses sentezleyiciler veya ekran okuyucular dil değişimine adapte olabilirler.

Tablolar:

Tablolar erişilebilirlik teknolojilerinin işleyebilmesi için kurallara uygun olarak gerekli tüm etiketler kullanılarak yaratılmalıdır.

Yeni Teknolojiler:

Sayfalarda kullanılan yeni teknolojiler, bu teknolojileri desteklemeyen araçlarda sorun yaratmamalıdır.

Kullanıcı Kontrollü İçerik:

Kullanıcılar kayan, parlayan yazıları açıp kapatabilmeli, akan metinlerin akış hızını değiştirebilmelidir.

Gömülü Nesnelere:

Bir gömülü nesne kendi arayüzüne sahipse, bu arayüz erişilebilir olmalıdır. Eğer gömülü nesnenin arayüzü erişilebilir yapılamıyorsa, alternatif bir erişim çözümü sağlanmalıdır. Kullanıcı arayüzü erişilebilir tasarım kurallarını sağlamalıdır. Çeşitli araçlar kullanılarak içeriğe ulaşılabilirlik mümkün olmalıdır.

Farklı Giriş Araçları:

Sayfaların kullanımı farklı giriş cihazları üzerinden yapılabilmelidir. Kullanıcı, fare, klavye, ses, baş hareketi gibi çeşitli giriş cihazlarını kullanabilmelidir.

Geçici Çözümler:

Eski tarayıcıların ve erişim destek teknolojilerinin sorunsuz çalışabilmesi için geçici çözümler kullanılmalıdır. Geçici çözümler, gelecekte kullanılması gerekmeyecek olan, ancak şu anda erişim destek teknolojilerinin ihtiyaç duyduğu çözümleri tanımlamaktadır.

W3C Teknolojileri ve Rehberleri:

W3C'nin önerdiği teknolojiler kullanılmalı ve erişilebilirlik rehberlerine uyulmalıdır. W3C teknolojilerin kullanımının uygun olmadığı durumlarda, bir alternatif versiyon hazırlanmalıdır.

Bağlam ve Yönelim Bilgisi:

Karmaşık sayfa ve elemanların anlaşılabilmesi için bağlam ve yönelim bilgileri sağlanmalıdır.

Basit Dolaşım Mekanizması:

Sitede basit ve tutarlı bir dolaşım mekanizması olmalıdır, bu sayede kullanıcılar aradıklarını daha kolay bulabileceklerdir.

Temiz ve Basit Dokümanlar:

Üretilen sayfalar temiz ve basit olmalı, bu sayede kolaylıkla anlaşılabilenlerdir.

5.2 Kontrol Listesi

Aşağıda WCAG rehberine göre hazırlanmış, öncelik derecelerine göre düzenlenmiş kontrol listeleri yer almaktadır.

Bu Kılavuzun önceki bölümlerinde önerilen tavsiyeler, aşağıdaki kontrol listelerinde yer alan önerileri de kapsamaktadır. Yine de WCAG önerilerinin toplu olarak verilmesinin faydalı olacağı düşüncesiyle bu listeler Kılavuza eklenmiştir.

Öncelik 1 tablosunda yer alan önerilerin mutlaka sağlanması gerektiği,

Öncelik 2 tablosunda yer alan önerilerin sağlanması gerektiği,

Öncelik 3 tablosunda verilen önerilerin sağlanmasının iyi olacağı

unutulmamalıdır.

Tabloların orijinallerine <http://www.w3.org/TR/WCAG10/full-checklist.html> adresinden erişilebilir. Aynı sayfadaki bağlantılar kullanılarak bu önerilerin nasıl hayata geçirileceğine dair yöntem tavsiyelerine de ulaşılabilir.

Tablo 1 - Öncelik 1 Kontrol Listesi

Öncelik 1	
Genel Özellikler	
1.1	Metin olmayan her ögenin bir metin eşleniği sunulmalıdır (Örneğin, alt, longdesc etiketleri kullanılarak). Bu gereklilik resimleri, metnin simgesel gösterimlerini (semboller dahil), resim haritası bölgelerini, animasyonları, hareketli GIF'leri, uygulama parçacıklarını (applet), programlanabilir nesnelere, ses dosyalarını, görüntü dosyalarını, görüntü dosyalarının ses kısmını, madde imi olarak kullanılan resimleri, grafik düğmeleri, çerçeveleri, betikleri kapsar.
2.1	Renkli sunulan tüm bilgiler, renksiz okunduğunda da erişilebilir olmalıdır.
4.1	Doküman içinde dil değişimleri dikkatle belirtilmelidir.
6.1	Dokümanlar, sayfa yönergeleri kullanılmadan da okunabilir olmalıdır. Örneğin, bir HTML sayfası ilgili sayfa yönergesi yoksa da okunabilmelidir.
6.2	Dinamik içerik değiştiğinde, dinamik içeriğin eşdeğer sunumu da güncelleniyor olmalıdır.
7.1	Ekranın kırışmasından (flickering) kaçınılmalıdır.
14.1	Site içeriğine uygun olan en açık ve basit dil kullanılmalıdır.
Resim ve Resim Haritaları	
1.2	Sunucu tarafında yer alan resim haritalarının her aktif bölgesi için metinsel bağlantı verilmelidir.
9.1	Bölgeler geometrik şekiller ile tanımlanabildiği sürece sunucu tarafı yerine istemci tarafında çalışan resim haritaları kullanılmalıdır.
Tablolar	
5.1	Veri tabloları için, satır ve sütun başlıkları tanımlanmalıdır.
5.2	İki veya daha fazla mantıksal seviye içeren satır veya sütun başlıkları olan veri tablolarında, veri hücreleri ve başlık hücreleri ilişkilendirilmelidir.
Çerçeveler	
12.1	Çerçeve tanımlanmasını ve dolaşımı kolaylaştırabilmek için her çerçevenin başlığı yazılmalıdır.
Betikler ve Uygulama Parçacıkları	
6.3	Betikler, uygulama parçacıkları ya da diğer programlanabilir nesnelere desteklenmediğinde ya da kapatıldığında, sayfalar yine de kullanılabilir olmalıdır. Eğer bu durum mümkün değil ise, eşdeğer içeriğe sahip erişilebilir bir sayfa hazırlanmalıdır.
Çoklu ortam	
1.3	Bir görüntü dosyasının görsel kısmının içerdiği önemli bilgilerin, işitsel eşdeğeri de kullanıma sunulmalıdır.
1.4	Zamana dayalı tüm çoklu ortam sunumlarının (ör. film ya da animasyon) eşdeğer alternatifleri de (başlıklar veya görsel kısma dair işitsel açıklamalar) senkronize olarak sunulmalıdır.
Erişilebilirlik Sağlanamıyor ise	
11.4	Eğer tüm çabalara rağmen sayfaya erişilebilirlik sağlanamıyorsa, sayfanın eş içeriğe (veya işlevselliğe) sahip, aynı sıklıkta güncellenen W3C teknolojilerini kullanan erişilebilir bir eşdeğeri hazırlanmalıdır.

Tablo 2 - Öncelik 2 Kontrol Listesi

Öncelik 2	
Genel Özellikler	
2.2	Arka plan rengi ve yazı rengi ikilileri, görme sorunu yaşayan kullanıcılarca görüntülendiğinde ya da siyah beyaz ekranda görüntülendiğinde yeterli kontrastı içerecek şekilde seçilmelidir.
3.1	Bilgi sunumunda resim yerine metin kullanımı tercih edilmelidir.
3.2	Dilbilgisi kurallarına uygun dokümanlar hazırlanmalıdır.
3.3	Görünüm ve sunumu denetlemek için sayfa yönergeleri kullanılmalıdır.
3.4	Etiket öznitelik değerlerinde ve sayfa yönergesi özellik değerlerinde mutlak yerine bağıl ölçüler kullanılmalıdır.
3.5	Doküman yapısını oluştururken başlık elemanları, kurallarına uygun olarak kullanılmalıdır
3.6	Listeler ve liste elemanları düzenli olarak etiketlenmelidir.
3.7	Alıntılar etiketlenmelidir. Alıntı etiketleri biçimlendirme amaçlı olarak kullanılmamalıdır.
6.5	Dinamik içerik erişilebilir olmalı ya da alternatif sayfalar hazırlanmalıdır.
7.2	Yanıp sönen (blink) öğeler kullanılmamalıdır.
7.4	Periyodik olarak kendini yenileyen (refresh) sayfalar yapılmamalıdır.
7.5	Kod içinde otomatik sayfa yönlendirmesi yapılmamalıdır. Bunun yerine yönlendirme ihtiyaçları sunucu konfigürasyonu üzerinden karşılanmalıdır.
10.1	Kullanıcının istemi dışında "Pop-up" pencereler ya da yeni pencereler açılmamalı ve kullanıcının bilgisi dışında aktif pencere değiştirilmemelidir.
11.1	Mümkün olduğu ve hedeflenen işe uygun olduğu sürece W3C teknolojileri en güncel sürümleri ile kullanılmalıdır.
11.2	W3C teknolojilerinin kullanım dışı kalmış özelliklerinden sakınılmalıdır.
12.3	Uygun olduğu sürece geniş bilgi öbekleri daha küçük, kontrol edilebilir gruplara bölünmelidir.
13.1	Verilen her bağlantı noktasının hedefi açıkça tanımlanmalıdır.
13.2	Sayfalara ve siteye anlamsal bilgi eklemek için yardımcı veriler (metadata) kullanılmalıdır.
13.3	Sitenin genel içeriği ile ilgili bilgi sağlanmalıdır. (Örneğin, site haritası ya da içerik tablosu)
13.4	Kullanılan dolaşım mekanizması tutarlı olmalıdır.
Tablolar	
5.3	Doğrusallaştırıldığında anlam kaybına yol açtığı sürece tablolar yerleşim amaçlı olarak kullanılmamalıdır. Aksi durumda eşdeğer bir alternatifi de hazırlanmalıdır. (doğrusallaştırılmış versiyon)
5.4	Bir tablonun yerleşim amaçlı kullanılması durumunda, görsel biçimlendirme amacıyla yapısal etiketler kullanılmamalıdır.
Çerçeveler	
12.2	Çerçeve başlıkları açıkça belli etmiyorsa, çerçevelerin amacı ve birbirleri ile ilişkileri tanımlanmalıdır.
Formlar	
10.2	Form elemanlarına ait etiketler doğru yerleştirilmelidir.
12.4	Etiketler, ilgili form elemanları ile ilişkilendirilmelidir.
Betikler ve Uygulamacıklar	
6.4	Betik ve uygulamacıklar içinde kullanılan olay işleyiciler (event handler) giriş aracından bağımsız olmalıdır.
7.3	Sayfalarda hareketten kaçınılmalıdır.
8.1	Betikler ve uygulama parçacıkların erişim destek teknolojileri ile erişilebilir ve bu teknolojilere uyumlu olması sağlanmalıdır.
9.2	Kendi arayüzü olan her elemanın cihaz-bağımsız tarzda çalıştığından emin olunmalıdır.

9.3	Betikler için cihaza bağımlı olay işleyiciler yerine mantıksal olay işleyiciler kullanılmalıdır.
-----	--

Tablo 3 - Öncelik 3 Kontrol Listesi

Öncelik 3	
Genel Özellikler	
4.2	Doküman içinde her kısaltmanın geçtiği ilk noktada, o kısaltmanın uzun hali de verilmelidir.
4.3	Dokümanın doğal dili belirtilmelidir.
9.4	Sayfa içinde linkler, form kontrolleri ve nesnelere arasında mantıksal bir sekme sırası kurulmalıdır.
9.5	Önemli bağlantılar, form kontrolleri ve form kontrol grupları için klavye kısa yol tuşları tanımlanmalıdır.
10.5	Bitişik bağlantılar arasında boşluk karakterleri ile çevrelenmiş basılabilir karakterler kullanılmalıdır. Doğrudan bağlantılar bitleştirilmemelidir.
11.3	Kullanıcıların sadece kendi tercihlerine uygun (ör. dil, içerik tipi vb.) dokümanları almalarını sağlamak için dokümanlar ile ilgili bilgiler verilmelidir.
13.5	Dolaşım mekanizmasının kullanımı için dolaşım çubukları, menüler hazırlanmalıdır.
13.6	İlgili bağlantılar gruplanmalı, gruplar tanımlanmalı ve tüm grubu atlayabilmek için bir yol tanımlanmalıdır.
13.7	Eğer arama mekanizması kurulmuş ise, değişik tercih ve yetenek düzeyleri için farklı arama türleri sağlanmalıdır.
13.8	Ayır edici bilgiler sayfaların, paragrafların başlangıcına yerleştirilmelidir.
13.9	Doküman koleksiyonları ile ilgili bilgi verilmelidir.
13.10	Çok satırlı ASCII grafik kullanılması durumunda, bunları atlayabilmek için bir yol tanımlanmalıdır.
14.2	Sayfanın içeriğini zenginleştireceği durumlarda, metinlere resim veya sesli sunumlar eklenebilir.
14.3	Sayfalar arasında tutarlılığı olan bir sunum stili yaratılmalıdır.
Resimler ve Resim Haritaları	
1.5	İstemci tarafında yer alan resim haritalarında her aktif bölge için alternatif bir metin bağlantısı da sağlanmalıdır.
Tablolar	
5.5	Tablolar için özet bilgi verilmelidir.
5.6	Başlık etiketlerinde kısaltmalar kullanılmalıdır.
10.3	Paralel ya da sonraki satıra kaydırmalı kolonlarda metin içeren tabloların, lineer metin halinde alternatifi sağlanmalıdır.
Formlar	
10.4	Metin kutuları ve alanlarına varsayılan değerler atanmalıdır.

5.3 Site Erişilebilirlik Kontrolü

Hazırlanan internet sitesinin erişilebilirlik yeteneğini sınamak için kullanılacak bazı internet kaynakları mevcuttur. Bu sayfalar kullanılarak, sitenin WAI denetim noktalarını sağlayıp sağlamadığı öğrenilebilir.

Bu amaçla kullanılacak sayfalardan biri WEBXACT'dir. Bu sayfada sitenin hangi öncelikte hangi kontrolleri sağlamadığı çevrimiçi olarak görülebilir. Sayfa erişilebilirliğin yanında kalite ve güvenlik hususlarını da sınavabilmektedir.

WEBXACT : <http://webxact.watchfire.com/>

Ayrıca, W3C'nin sayfalarında erişilebilirlik kontrolü için kullanılacak çok sayıda araç tanıtılmaktadır. Bu araçlar içinden amaca daha uygun başka bir araç seçmek de mümkündür.

<http://www.w3.org/WAI/ER/existingtools.html>

6. KISALTMALAR

TÜRKÇE

BTE : Bilişim Teknolojileri Enstitüsü
MAM : Marmara Araştırma Merkezi
TÜBİTAK : Türkiye Bilimsel ve Teknolojik Araştırma Kurumu

İNGİLİZCE

CSS : Cascading Style Sheets
DOC : Document
GIF : Graphics Interface Format
HTML : HyperText Markup Language
IP : Internet Protocol
JPEG : Joint Pictures Experts Group
Kb : Kilobyte
Kbps : Kilobit per second
PDF : Portable Document Format
PNG : Portable Network Graphics
SVG : Scalable Vector Graphics
W3C : World Wide Web Consortium
WAI : WEB Accessibility Initiative
WCAG : Web Content Accessibility Guidelines

EK B WEB SAYFASI DEĞERLENDİRME CLASS'I

EK B.1 Kaynak Kodu (class.webpage.php)

```

<?php
/*-----+
| WEB PAGE EVALUATION CLASS
+-----+
| This class has been written for m.sc. thesis.
+-----+
| Thesis Subject: The Development of a Tool Aimed at The Evaluation of The Public Institutions
Websites Under The Circular 2007/4
+-----+
| Erkan DURAN
| Balıkesir University, Institute of Science
| Department of Computer Education and Instructional Technology
+-----+
| M. Sc. Thesis Supervisor: Assist. Prof. Dr. Taner TANRISEVER
| taner@balikesir.edu.tr
+-----+
| Balıkesir-Turkey, 2008
+-----+
| Released under the terms & conditions of v2 of the
| GNU General Public License. For details refer to
| the included gpl.txt file or visit http://gnu.org
+-----*/

//In this class, for parsing HTML pages, HTML TIDY has been used.
//TIDY is a binding for the Tidy HTML clean and repair utility which allows you to not only clean
and otherwise manipulate HTML documents, but also traverse the document tree.
//In this class, the purpose of the TIDY is parsing HTML codes.
//Before you run this class, install HTML TIDY package/extension.
//For details: php.net/tidy or tidy.sourceforge.net
//Also in this class, CSSTidy class has been used for parsing CSS files.
//CSSTidy class is an opensource CSS parser and optimiser.
//For details : csstidy.sourceforge.net

require_once ('csstidy/class.csstidy.php');
class webpage
{
//Settings
private $settings = array
(
//Folders. These are created automatically.
'fo_www' => 'www', // It is main folder that contains all of the evaluated pages. Path:
/www
'fo_url' => "", // It is URL folder that contains evaluated page. Path: /www/url
'fo_css' => 'css', //It is CSS folder that contains evaluated page's external&imported CSS
files .Path: /www/url/css
'fo_images' => 'images', //It is image folder that contains evaluated page's images. Path:
/www/url/images

```

```
'fo_w3c_html' => 'w3c_html', //It is W3C HTML folder that contains W3C HTML validation
test result of web page. Path: /www/url/w3c_html
'fo_w3c_css' => 'w3c_css', //It is W3C CSS folder that contains W3C CSS validation test
result of web page. Path: /www/url/w3c_css
```

//preg_match patterns for content tests (Web page has a logo/e-mail ? etc) These values are in language files in 'locale_set' folder.

```
'p_logo' => "", //For logo
'p_help' => "", //For link to help pages
'p_faq' => "", //For link to faq pages
'p_feed' => "", //For link to feedback pages
'p_aim' => "", //For aims
'p_anno' => "", //For announcements
'p_search' => "", //For search
'p_address' => "", //For addresses
'p_phone' => "", //For phone
'p_appropmail' => "", //For the appropriate of mails .
```

```
//Perform W3C test?
'test_w3c_html'=> false,
'test_w3c_css' => false
```

```
);
```

```
var $url;
var $domain; // Domain of url
var $path; //Path of url
private $fcont; // Web page content ( Code)
private $fcont2; //Web page content without comments, scripts and styles (Code)
private $fcont3; // Web page texts without HTML codes.
private $w3chtmlfcont; //File content of W3C HTML validation test result
private $w3ccssfcont; //File content of W3C CSS validation test result
private $i = 0; //External and imported CSS file count
private $extintesscont; //External and internal CSS Content
private $inlinecsscont; //Inline CSS content
private $htmltmpesscont; //CSS content for HTML color values
private $tidy; // TIDY object
private $tidyhtml; //Tidy nodes
private $tidyerrors; //Tidy error count
private $tidywars; //Tidy warning count
```

```
/*-----*/
/*-----*/
/*-----> Constructor Function <----- */
/*-----*/
/*-----*/
```

```
function webpage ($url, $lan = false, $test_w3c_html = false, $test_w3c_css = false)
{
```

```
    //Settings
    if ($lan) {$lan = $lan;} else $lan = 'en';
    require "locale_set/" . $lan . '.php';
    require_once "test_functions.php";
```

```
    $this->settings['p_logo'] = $locale['100'];
    $this->settings['p_help'] = $locale['101'];
    $this->settings['p_faq'] = $locale['102'];
    $this->settings['p_feed'] = $locale['103'];
    $this->settings['p_aim'] = $locale['104'];
    $this->settings['p_anno'] = $locale['105'];
```

```

$this->settings['p_search'] = $locale['106'];
$this->settings['p_address'] = $locale['107'];
$this->settings['p_phone'] = $locale['108'];
$this->settings['p_appropmail'] = $locale['109'];

$this->settings['test_w3c_html'] = $test_w3c_html;
$this->settings['test_w3c_css'] = $test_w3c_css;

//Check URL
$url = preg_match('/^http:\V(.*)/iX', $url, $url) ? $url[1] : "";
if ($url == "")
{
    echo "Invalid URL";
    exit;
}
$this->url = 'http://'.rtrim($url, '/');

// Domain and Path
$spurl = parse_url($this->url); //Host, path etc.
$this->domain = $spurl['scheme'].'://'.$spurl['host'];
if (array_key_exists('path', $spurl))
{
    $pinf = pathinfo($spurl['path']);
    if ($pinf['dirname'] == '')
    {
        $this->path = array_key_exists('extension', $pinf) ? '' : $spurl['path'];
    } else
    {
        $this->path = array_key_exists('extension', $pinf) ? $pinf['dirname'] :
$pinf['dirname'].'/'.$pinf['basename'];
    }
} else $this->path = null;

//Create Folders
$this->create_folders($url);

//Get URL Content
$this->get_url_content($this->url);

//TIDY Object
$config = array('output-encoding' => 'utf-8');
$this->tidy = new tidy();
$this->tidy = tidy_parse_string($this->fcont, $config);
$this->tidyerrors = tidy_error_count($this->tidy);
$this->tidywars = tidy_warning_count($this->tidy);
$this->tidyhtml = $this->tidy->html();

//Get CSS Content
$this->i = 0;
$this->get_css_content();

//CSSTidy Objects
$this->extintcsscont = $this->parse_css($this->extintcsscont);
$this->inlinecsscont = $this->parse_css($this->inlinecsscont);
$this->htmltmpcsscont = $this->parse_css($this->htmltmpcsscont);

//Get Images
$this->get_images();

```

```

//Get W3C HTML validation test result page
if ($this->settings['test_w3c_html']) $this->get_w3c_html_content($this->url);
//Get W3C CSS validation test result page
if ($this->settings['test_w3c_css']) $this->get_w3c_css_content($this->url);
}

/*-----*/
/*-----*/
/*--> PRIVATE FUNCTIONS<-- */
/*-----*/
/*-----*/

//This function returns an array of nodes under TIDY nodes ($this->tidyhtml) with the specified
id/type (based by brutuscat, php.net)
private function get_nodes2($node, $type, $parent, $sibling, &$res)
{
    if($node != null)
    {
        if(@$node->id == $type)
        {
            $res[] = array
            (
                'value'    => $node->value, //The value of the node
                'name'     => $node->name, //The name of the tag (e.g. html, a, etc..)
                'attribute' => $node->attribute, //An array with the attributes of the current node,
or NULL if there aren't any
                'line'     => $node->line, //The line where the node starts
                'column'   => $node->column, //The column where the node starts
                'parent'   => $parent, // Not used
                'sibling'  => $sibling //Not used
            );
        }

        if($node->child != null)
        {
            foreach ($node->child as $child)
            {
                $sibl .= $child->name ? $child->name."|": "";
                $this->get_nodes2($child, $type, $parent.$node->name."|", $sibl, &$res);
            }
        }
    }
}

//Other HTML parsing function (based by Lyubinskiy, I.S.)
private function parse_html ($html, $tag, $type)
{
    $pstring1 = "[^]*";
    $pstring2 = "[^"]*";
    $pnstring = "[^\\>]";
    $pintag = "(?:$pstring1|$pstring2|$pnstring)*";
    $pattrs = "(?:\\s$pintag){0,1}";

    if ($type == "b")
    {
        $ptag = $this->enclose("<$tag$pattrs>", "<", "\\V$tag>");
        $string = preg_match("/$ptag/iXs", $html, $string) ? $string[1] : "";
        return $string; //First element content (between start tag and end tag) with specified tag
    } else if ($type == "ba")

```

```

{
    $ptag = $this->enclose("<$tag$pattrs>", "<", "\\$tag>");
    preg_match_all("/$ptag/iXs", $html, $array);
    $array = $array[1];
    return $array; //Array of all elements' contents with specified tag. Not used.
} else if ($type == "a")
{
    $ptag = "<$tag(?:\s(?:.*?)){0,1}>";
    preg_match_all("/$ptag/iXs", $html, $array);
    $array = $array[0];
    return $array; // Array of all tags with specified tag
} else
{
    $ptag = "<$tag(?:\s(?:.*?)){0,1}>";
    $pattr1 = "(?:[^\s]*)$type\s*=\s*["'"]{1}([^\s">]*)";
    $pattr2 = "(?:[^\s]*)$type\s*=\s*([^\s'>]+)";
    preg_match_all("/$ptag/iXs", $html, $array);
    $array = $array[0];
    $newarray = array ();
    foreach ($array as $arr) {
        if (preg_match("/$pattr1/iXs", $arr, $x)) { array_push($newarray, $x[1]); }
        elseif (preg_match("/$pattr2/iXs", $arr, $y)) { array_push($newarray, $y[1]); }
    }
    return $newarray; //Array of attribute values of all elements with specified tag and specified
type (attribute)
}
}

//This function returns preg_match pattern which finds value between start tag and end tag (based by
Lyubinskiy, I.S.)
private function enclose($start, $end1, $end2)
{
    return "$start(?:.*?)$end1$end2";
    //return "$start(?:[$end1]|$end1(?:!$end2))*$end1$end2";
}

//CSS parsing function
private function parse_css ($content)
{
    $css = new csstidy();
    $css->set_cfg('merge_selectors', TRUE);
    $css->set_cfg('lowercase_s', TRUE);
    $css->parse($content);
    return $css;
}

//This function prevents unwanted chars for the URL folder name and image file names and returns to
new name.
private function set_name($name)
{
    $name = preg_replace('/(\/|\\|\\:|\\*|\\?|\\|<|>|\\|~)/', '_', $name);
    return $name;
}

//This function returns real/full path of external CSS files, imported CSS files and image files.
private function check_url($url)
{
    $purl = parse_url($url);

```

```

if (array_key_exists('host', $purl))
{
    $tmp = $url;
} else
{
    if (substr($url, 0, 1) == '/') //Root
    {
        $tmp = $this->domain.$url;
    }
    elseif (substr($url, 0, 2) == './') //Working directory
    {
        $tmp = $this->domain.$this->path.'/'.substr($url, 2, strlen($url)-2);
    } else
    {
        $tmp = $this->domain.$this->path.'/'.$url;
    }
}

$arr = explode('/', $tmp);
foreach ($arr as $k => $v)
{
    if ($v == '..') //Up
    {
        unset($arr[$k]);
        prev($arr);
        unset($arr[key($arr)]);
    }
}
$tmp = implode('/', $arr);
return $tmp;
}

//This function first checks recursively imported CSS files for screen/all/not exist media type in
external or in internal or in imported CSS files, after that merges checked.
private function check_import($content, &$newcontent)
{
    $this->css_write($content);
    preg_match_all("/@import\s+(?:url\()?(?:[\"'\"])(.*)?(?:[\"'\"])?(?:\))?(.*);/iX", $content, $imports);
    $content = preg_replace("/@import(?:[\"'\"]*)/i", "", $content);

    if (count($imports[1]) > 0)
    {
        foreach ($imports[1] as $impkey => $impval)
        {
            if (preg_match("/(screen|all)/iX", $imports[2][$impkey]) ||
trim($imports[2][$impkey]) == "")
            {
                if ($file = file_get_contents($this->check_url($impval)))
                {
                    $info = /* IMPORT: '$impval.'.$imports[2][$impkey].' */;
                    $this->check_import($info.$file, $newcontent);
                }
            }
        }
    }
    $newcontent .= $content;
}
}

```



```

//This function saves CSS files/codes at web page. Path: /www/url/css
private function css_write($content)
{
    $this->i++;
    chdir($this->settings['fo_css']);
    touch($this->i.'.css') or die ("Empty CSS file can not be created.");
    $file = fopen($this->i.'.css', 'w') or die ("Empty CSS file can not be opened in writing mode.");
    fwrite($file, $content) or die ("CSS Content can not be written to empty CSS file.");
    fclose($file);
}

//This function scans files where a directory and its subdirectories recursively. Returns array (Files'
//path, name, extension, size). (based by lixlpixel.org)
private function scan_directory_recursively($directory, $filter=FALSE, &$d_tree)
{
    if(substr($directory,-1) == '/')
    {
        $directory = substr($directory,0,-1);
    }
    if(!file_exists($directory) || !is_dir($directory))
    {
        return FALSE;
    } elseif(is_readable($directory))
    {
        $directory_list = opendir($directory);
        while (FALSE !== ($file = readdir($directory_list)))
        {
            if($file != '.' && $file != '..')
            {
                $path = $directory.'/'.$file;
                if(is_readable($path))
                {
                    $subdirectories = explode('/', $path);
                    if(is_dir($path))
                    {
                        $this->scan_directory_recursively($path, $filter, $d_tree);
                    } elseif(is_file($path))
                    {
                        $extension = end(explode('.', end($subdirectories)));
                        if($filter === FALSE || $filter == $extension)
                        {
                            $d_tree[] = array(
                                'path'    => $path,
                                'name'    => end($subdirectories),
                                'extension'=> $extension,
                                'size'    => filesize($path));
                        }
                    }
                }
            }
        }
        closedir($directory_list);
        return $directory_tree;
    } else
    {
        return FALSE;
    }
}

```

```

//This function creates the folders
private function create_folders($url)
{
    $foldername = $this->set_name(rtrim($url, '/'));
    $wd = dirname(__FILE__);

    //Main folder
    if (!file_exists($wd.'/'.$this->settings['fo_www'])) mkdir($wd.'/'.$this->settings['fo_www']) or
die ($wd.'/'.$this->settings['fo_www']. " folder can not be created.");

    //URL folder. Path: www/url
    if (!file_exists($wd.'/'.$this->settings['fo_www'].'/'.$foldername))
    {
        mkdir($wd.'/'.$this->settings['fo_www'].'/'.$foldername) or die ($this->wd.'/'.$this-
>settings['fo_www'].'/'.$foldername. " folder can not be created.");
    } else
    {
        $this->scan_directory_recursively($wd.'/'.$this->settings['fo_www'].'/'.$foldername, false,
$files);
        foreach ($files as $file) unlink($file['path']);
    }
    $this->settings['fo_url'] = $wd.'/'.$this->settings['fo_www'].'/'.$foldername;

    //CSS folder. Path: www/url/css
    if (!file_exists($this->settings['fo_url'].'/'.$this->settings['fo_css'])) mkdir($this-
>settings['fo_url'].'/'.$this->settings['fo_css']) or die ($this->settings['fo_url'].'/'.$this-
>settings['fo_css']. " folder can not be created.");
    $this->settings['fo_css'] = $this->settings['fo_url'].'/'.$this->settings['fo_css'];

    //Image folder. Path: www/url/images
    if (!file_exists($this->settings['fo_url'].'/'.$this->settings['fo_images'])) mkdir($this-
>settings['fo_url'].'/'.$this->settings['fo_images']) or die ($this->settings['fo_url'].'/'.$this-
>settings['fo_images']. " folder can not be created.");
    $this->settings['fo_images'] = $this->settings['fo_url'].'/'.$this->settings['fo_images'];

    //W3C HTML folder. Path: www/url/w3c_html
    if (!file_exists($this->settings['fo_url'].'/'.$this->settings['fo_w3c_html'])) mkdir($this-
>settings['fo_url'].'/'.$this->settings['fo_w3c_html']) or die ($this->settings['fo_url'].'/'.$this-
>settings['fo_w3c_html']. " folder can not be created.");
    $this->settings['fo_w3c_html'] = $this->settings['fo_url'].'/'.$this->settings['fo_w3c_html'];

    //W3C CSS folder. Path: www/url/w3c_css
    if (!file_exists($this->settings['fo_url'].'/'.$this->settings['fo_w3c_css'])) mkdir($this-
>settings['fo_url'].'/'.$this->settings['fo_w3c_css']) or die ($this->settings['fo_url'].'/'.$this-
>settings['fo_w3c_css']. " folder can not be created.");
    $this->settings['fo_w3c_css'] = $this->settings['fo_url'].'/'.$this->settings['fo_w3c_css'];
}

//This function gets web page content
private function get_url_content($url)
{
    // Web page content
    $this->fcont = file_get_contents($url) or die ("URL content can not be taken.");
    chdir($this->settings['fo_url']);
    touch('index.htm') or die ("index.htm file can not be created for URL content.");
    $file = fopen('index.htm', 'w') or die ("index.htm file can not be opened in writing mode for URL
content.");
    fwrite($file, $this->fcont) or die ("URL content can not be written to 'index.htm'.");
    fclose($file);
}

```

```

// Web page content without comments, scripts and styles
$psstring1 = "[^]*";
$psstring2 = "[^]*";
$pnstring = "[^>]";
$psintag = "(?:$psstring1|$psstring2|$pnstring)*";
$psattrs = "(?:\\s$psintag){0,1}";
$pscomtag = $this->enclose('<!--', '-', '->');
$pscrtag = $this->enclose("<script$psattrs>", "<", "\\script>");
$psstytag = $this->enclose("<style$psattrs>", "<", "\\style>");

$this->fcont2 = preg_replace("/(?:$pscomtag|$pscrtag|$psstytag)/iXs", "", $this->fcont);

//Texts without HTML codes
$this->fcont3 = preg_replace("/(?:is)<.*?>/iX", "", $this->fcont2);
}

//This function first gets external and internal cascading style sheets which have screen/all media type
after merges them for parsing.
//Also this function;
//Gets and merges inline CSS codes for parsing.
//Gets and merges color values in 'alink', 'bgcolor', 'color', 'link', 'text', 'vlink' attributes for parsing.
private function get_css_content()
{
    //Gets external CSS files.
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_LINK, "", "", $links);
    if (count($links) > 0)
    {
        foreach ($links as $link)
        {
            if (is_array($link['attribute']))
            {
                if (array_key_exists('rel', $link['attribute']) && array_key_exists('href',
$link['attribute']))
                {
                    if ((strtolower($link['attribute']['rel']) == 'stylesheet') &&
(!array_key_exists('media', $link['attribute']) || (array_key_exists('media',
$link['attribute']) && preg_match("/(screen|all)/iX", $link['attribute']['media']))) // Is media type
screen, all or not exist?
                    {
                        if ($tempextcss = file_get_contents($this-
>check_url($link['attribute']['href'])))
                        {
                            $info = /* EXTERNAL: '$link['attribute']['href'].' */;
                            $this->check_import($info.$tempextcss, $extcss); // This file
contains imported CSS files ?
                            $sscont[] = array
                                (
                                    'line' => $link['line'],
                                    'column' => $link['column'],
                                    'content' => $extcss
                                );
                        }
                    }
                }
            }
        }
    }
}

```

```

//Get internal CSS codes
$this->get_nodes2($this->tidyhtml, TIDY_TAG_STYLE, "", $styles);
if (count($styles) > 0)
{
    foreach ($styles as $style)
    {
        if (is_array($style['attribute']))
        {
            if (!array_key_exists('media', $style['attribute']) || (array_key_exists('media',
                $style['attribute']) && preg_match("/(screen|all)/iX", $style['attribute']['media']))) // Is media type
                screen, all or not exist?
            {
                $tempintcss = $this->parse_html($style['value'], 'style', 'b');
                $tempintcss = preg_replace("/(<!-->)/iX", "", $tempintcss); //For old
                browsers
                $info = '/* INTERNAL */';
                $this->check_import($info.$tempintcss, $intcss); //This code contains
                imported CSS files ?
                $csscont[] = array
                    (
                        'line'    => $style['line'],
                        'column' => $style['column'],
                        'content' => $intcss);
                    }
            }
        }
    }
}

//Merge external and internal CSS codes
if (@count($csscont) > 0)
{
    foreach ($csscont as $key => $row)
    {
        $line[$key] = $row['line'];
        $column[$key] = $row['column'];
        $content[$key] = $row['content'];
    }
    array_multisort($line, SORT_ASC, $column, SORT_ASC, $csscont);

    $this->extintcsscont = "";
    for ($i = 0; $i < count($csscont); $i++)
    {
        $this->extintcsscont .= $csscont[$i]['content'];
    }
} else $this->extintcsscont = "";

// Get inline CSS codes
$inliness = $this->parse_html($this->fcont2, '[a-z]+', 'style');
if (count($inliness) > 0)
{
    $this->inlinesscont = "";
    for ($i = 0; $i < count($inliness); $i++)
    {
        $this->inlinesscont .= "x$i {".$inliness[$i]."}"; //Merge them
    }
} else $this->inlinesscont = "";

//Get HTML colors
$colatts = array('alink', 'bgcolor', 'color', 'link', 'text', 'vlink');

```

```

$this->htmltmpcsscont = "";
$j = -1;

foreach ($colatts as $colatt)
{
    $j++;
    $htmltmpcss = $this->parse_html($this->fcont2, '[a-z]+', "$colatt");
    if (count($htmltmpcss) > 0)
    {
        for ($i = 0; $i < count($htmltmpcss); $i++)
        {
            $this->htmltmpcsscont .= "x$j$i {color: ".$htmltmpcss[$i].'}'; //Merges them
        }
    }
}

//This function gets images from the web page and saves them into 'images' folder. Path:
//www/url/images.
private function get_images ()
{
    chdir($this->settings['fo_images']);

    $this->get_nodes2($this->tidyhtml, TIDY_TAG_IMG, "", "", $imags);

    $i = 0;
    if (count($imags) > 0)
    {
        foreach ($imags as $img)
        {
            $i++;
            if (is_array($img['attribute']))
            {
                if (array_key_exists('src', $img['attribute']))
                {
                    if ($img_ = file_get_contents($this->check_url($img['attribute']['src'])))
                    {
                        $filename = $this->set_name($i.'.'.basename($this->check_url($img['attribute']['src'])));

                        touch($filename) or die ("Empty image file can not be created.");
                        $file = fopen($filename, 'w') or die ("Empty image file can not be
opened in writing mode for image content.");
                        fwrite($file, $img_) or die ("Image content can not be written to
empty image file.");
                        fclose($file);
                    }
                }
            }
        }
    }
}

//This function performs W3C HTML validation test and saves the test's result page into 'w3c_html'
folder. Path: /www/url/w3c_html
private function get_w3c_html_content($url)
{
    $this->w3chtmlfcont = file_get_contents('http://validator.w3.org/check?uri='.$url) or die ("URL
content can not be taken at W3C HTML test.");
}

```

```

chdir($this->settings['fo_w3c_html']);
touch('index.htm') or die ("index.htm' file can not be created for W3C HTML test.");
$file = fopen('index.htm', 'w') or die ("index.htm' file can not be opened in writing mode for
W3C HTML test.");
fwrite($file, $this->w3chtmlfcont) or die ("W3C HTML test content can not be written to
index.htm.");
fclose($file);
}

//This function performs W3C CSS validation test and saves the test's result page into 'w3c_css'
folder. Path: /www/url/w3c_css
private function get_w3c_css_content($url)
{
    $this->w3ccssfcont = file_get_contents('http://jigsaw.w3.org/css-validator/validator?uri='.$url)
or die ("URL content can not be taken at W3C CSS test.");
    chdir($this->settings['fo_w3c_css']);
    touch('index.htm') or die ("index.htm' file can not be created for W3C CSS test.");
    $file = fopen('index.htm', 'w') or die ("index.htm' file can not be opened in writing mode for
W3C CSS test.");
    fwrite($file, $this->w3ccssfcont) or die ("W3C CSS test content can not be written to index.htm
file.");
    fclose($file);
}

//This function checks associated label element with specified form element id. Return true or false.
private function get_label_for ($labels, $id)
{
    $found = 0;
    foreach ($labels as $lab)
    {
        if ($lab['attribute']['for'] == $id)
        {
            $found = 1;
            break;
        } else $found = 0;
    }
    return $found == 1 ? true : false;
}

//This function checks acceskey attributes in associated label element with specified form element id.
Return true or false.
private function get_label_accesskey ($labels, $id)
{
    $found = 0;
    foreach ($labels as $lab)
    {
        if (array_key_exists('accesskey', $lab['attribute']) && $lab['attribute']['for'] == $id)
        {
            $found = 1;
            break;
        } else $found = 0;
    }
    return $found == 1 ? true : false;
}

```

//This function checks tag's attribute/property in CSS and HTML code.
 /*, tag, #id, tag#id, .class and tag.class selectors are checked only in CSS code. Returns true or false.
 private function get_att (\$content, \$tag, \$attlist, \$cssprop, \$htmlprop)

```
{
    $found = false;
    $arr = array('*', $tag); // * and $tag
    if (is_array($attlist))
    {
        if (array_key_exists('id', $attlist))
        {
            array_push($arr, '#' . strtolower($attlist['id'])); // #id
            array_push($arr, $tag . '#' . strtolower($attlist['id'])); // element#id
        }
        if (array_key_exists('class', $attlist))
        {
            array_push($arr, '.' . strtolower($attlist['class'])); // .class
            array_push($arr, $tag . '.' . strtolower($attlist['class'])); // element.class
        }

        //In HTML code or inline css
        if (array_key_exists($htmlprop, $attlist) ||
            (array_key_exists('style', $attlist) && preg_match("/$cssprop(?:\\s*)([0-9A-Z_]+(?:\\s*)/iX", $attlist['style'])))
            $found = true;
    }

    if ($found == false)
    {
        if (is_array($content))
        {
            foreach ($content as $selectors => $selarr)
            {
                if ($selarr[$cssprop])
                {
                    $sels = explode(',', $selectors);
                    foreach ($sels as $sel)
                    {
                        if (in_array(trim($sel), $arr))
                        {
                            $found = true;
                            break;
                        }
                    }
                }
                if ($found) break;
            }
        }
    }
    return $found;
}
```

//This function returns total font-size property count and total valid font-size property count in CSS code.

```
private function css_font_size ($content, &$count, &$valid)
{
    $count = 0;
    $valid = 0;

    $abs_size = 'xx-small|x-small|small|medium|large|x-large|xx-large';
```

```

$rel_size = 'larger|smaller';
$sabs_unit = 'in|cm|mm|pt|pc';
$rel_unit = 'em|ex|px';
$per_unit = '%';
$number = '(?:\.)?\d+(?:\.)?(?:\d+)?';
$length = $number."(?:$rel_unit|$sabs_unit|$per_unit)";
$style = '(normal|italic|oblique)?(?:\s+)?';
$variant = '(normal|small-caps)?(?:\s+)?';
$weight = '(normal|bold|bolder|lighter|100|200|300|400|500|600|700|800|900)?(?:\s+)?';
$sysfont = 'caption|icon|menu|message-box|small-caption|status-bar';
$svw = "(?:$style$variant$weight)?"
$fsz = "($sabs_size|$rel_size|$length)(?:\s+)?";
$lhg = "(\/(?:normal(?:$number(?:$rel_unit|$sabs_unit|$per_unit)?)))(?:\s+)?";
$fml = '((?:.*)*.*)';
$sec1 = "(?:$svw$fsz$lhg$fml)";
$sec2 = "(?:$sysfont)";
$font = "($sec2|$sec1)";

foreach ($content as $selectors => $selarr)
{
    if ($selarr['font'] || $selarr['font-size'])
    {
        $selcount = count(explode(',', $selectors));

        foreach ($selarr as $prop => $val)
        {
            $val = strtolower($val);
            if ($prop == 'font')
            {
                preg_match("/$font/iX", $val, $font);
                if (count($font) == 8) //If font-size isn't system font.
                {
                    $count += $selcount;
                    if (in_array($font[5], explode('|', $sabs_size))) $valid += $selcount;
                }
            } elseif ($prop == 'font-size')
            {
                $count += $selcount;
                if (preg_match("/($sabs_size)/iX", $val) && !in_array($val, explode('|',
$rel_size)) && !in_array($val, explode('|', $sysfont))) $valid += $selcount;
            }
        }
    }
}
}

```

//This function returns total color property count and total valid color property count in CSS code.

```

private function css_color ($content, &$count, &$valid)
{
    $sarr_valid1 = array ('0','3','6','9','c','f');
    $sarr_valid2 = array ('red', 'blue', 'aqua', 'lime');
    $sarr_invalid = array ('maroon', 'orange', 'olive', 'purple', 'green', 'navy', 'teal', 'silver', 'gray', 'grey',
'darkgrey', 'darkslategrey', 'lightgray', 'lightslategrey', 'slategrey', 'dimgrey');
    $count = 0;
    $valid = 0;

    foreach ($content as $selectors => $selarr)
    {
        $selcount = count(explode(',', $selectors));
    }
}

```



```

$tidy = tidy_parse_string($this->fcont, $config);
//$tidy->CleanRepair();
$tidy->diagnose();
$tidyaccwars = tidy_access_count($tidy);
preg_match_all("(/(?..*)access(?:.*)/iX", $tidy->errorBuffer, $bufferaccs);
foreach ($bufferaccs[0] as $buf) $buffer .= $buf."\\n";

return array
(
    'warnings' => $tidyaccwars, //Warning count
    'test'     => test_eq($tidyaccwars, 0),
    'buffer'   => $tidyaccwars > 0 ? htmlspecialchars($buffer) : NULL //Warnings
);
}

//Cookies
function arr_cookie ()
{
    $headers = get_headers($this->url, 1) or die ("Headers can not be taken for cookies..");
    $sval = array_key_exists('Set-Cookie', $headers) ? 0 : 1;

    return array
    (
        'value' => $sval, //1: Succesfull, 0: Failed
        'test'  => test_eq($sval, 1)
    );
}

//W3C tests
function arr_w3 ()
{
    if ($this->settings['test_w3c_html'])
    {
        $htmlerrors = preg_match('/(\\d+) Errors/', $this->w3htmlfcont, $htmlerrors) ?
        $htmlerrors[1] : "";
        $htmlerrors = $htmlerrors <> " ? $htmlerrors : 0;

        $doctypeval = preg_match('/^(no Doctype found)!/', $this->w3htmlfcont) ? 0 : 1;
    }

    if ($this->settings['test_w3c_css'])
    {
        $csserrors = preg_match('/Errors \\((\\d+)\\)/', $this->w3ccssfcont, $csserrors) ? $csserrors[1]
: ";
        $csserrors = $csserrors <> " ? $csserrors : 0;
    }

    return array
    (
        'htmlerrors' => isset($htmlerrors) ? $htmlerrors : NULL, //HTML error count
        'csserrors'  => isset($csserrors) ? $csserrors : NULL, //CSS error count
        'doctypevalue' => isset($doctypeval) ? $doctypeval : NULL, //Doctype. //1:
Succesfull, 0: Failed

        'htmltest'   => test_eq($htmlerrors, 0),
        'csstest'    => test_eq($csserrors, 0),
        'doctypestest' => test_eq($doctypeval, 1)
    );
}

```

```

//Comments
function arr_comment ()
{
    $pcomtag = $this->enclose('<!--', '!', '->');
    preg_match_all("/$pcomtag/iXs", $this->fcont, $comments_);
    $comcount = count($comments_[1]);
    foreach ($comments_[1] as $comment) $comments .= $comment."\\n";
    return array
    (
        'count'          => $comcount, //Comment count
        'appropstest'    => test_approp($comcount), //Appropriate of usage test
        'commentstest'   => $comcount > 0 ? htmlspecialchars($comments) : NULL //Comments
    );
}

//Page Title
function arr_title ()
{
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_TITLE, " ", $titles);
    $title = $this->parse_html($titles[0]['value'], "title", "b");
    $length = strlen($title);

    return array
    (
        'length'         => $length, //Title length
        'test'           => test_gt($length, 0),
        'lengthtest'     => test_lt($length, 61), //Length test
        'appropstest'    => test_approp($length), //Appropriate test
        'title'          => $length > 0 ? $title : NULL //Title
    );
}

//Scripts
function arr_script ()
{
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_SCRIPT, " ", $scripts);
    $scount = count($scripts);

    $this->get_nodes2($this->tidyhtml, TIDY_TAG_NOSCRIPT, " ", $noscripts);
    $noscount = count($noscripts);

    return array
    (
        'scriptcount'    => $scount, //Script count
        'noscriptcount' => $noscount, //Noscript count
        'appropstest'    => test_approp($scount)
    );
}

//Applets
function arr_applet ()
{
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_APPLET, " ", $applets);
    $aplcount = count($applets);

    return array
    (
        'count'          => $aplcount, //Applet count

```

```

        'test'      => test_eq($aplcount, 0)
    );
}

//Metatags
function arr_metatag ()
{
    $this->get_nodes2($this->tidyhtml,TIDY_TAG_META, " ", $metatags_);
    $mtgcount = count($metatags_);

    foreach ($metatags_ as $metatag) $metatags .= $metatag['value']."\n";

    return array
    (
        'count'      => $mtgcount, //Metatag count
        'test'       => test_gt($mtgcount, 0),
        'metatags'   => $mtgcount > 0 ? htmlspecialchars($metatags) : NULL, //Meta tags
    );
}

//Links
function arr_anchor ()
{
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_A, " ", $anchors);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_OBJECT, " ", $objects);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_EMBED, " ", $embeds);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_SCRIPT, " ", $scripts);

    $saccount = count($anchors);
    $shrefs = "";
    $stabidx = 0;
    $sacckey = 0;

    $swithimg = 0;
    $simgcount = 0;
    $simgalt = 0;

    if ($saccount > 0)
    {
        foreach ($anchors as $sanc)
        {
            if (is_array($sanc['attribute']))
            {
                //Hrefs
                if (array_key_exists('href', $sanc['attribute'])) $shrefs .=
                $sanc['attribute']['href']."\n";
                //Tabindex
                if (array_key_exists('tabindex', $sanc['attribute'])) $stabidx ++;
                //Accesskey
                if (array_key_exists('accesskey', $sanc['attribute'])) $sacckey ++;
            }
        }

        //Images
        $tidyanc = tidy_parse_string($sanc['value']);
        $this->get_nodes2($tidyanc->html(), TIDY_TAG_IMG, " ", $sancimgs);
        if (is_array($sancimgs) && count($sancimgs) > 0)
        {
            $swithimg ++;
            $simgcount += count($sancimgs);
        }
    }
}

```

```

        foreach ($ancings as $ancing)
        {
            if (is_array($ancing['attribute']) && array_key_exists('alt',
$ancing['attribute'])) $imgalt ++;
        }
    }
    $ancings = "";
}
}

$withoutimgper = $anccount > 0 ? $this->round2((( $anccount - $withimg) / $anccount) * 100) :
NULL;
$imgaltper = $imgcount > 0 ? $this->round2(($imgalt / $imgcount) * 100) : NULL;
$tabidxper = $anccount > 0 ? $this->round2(($tabidx / $anccount) * 100) : NULL;
$acckeyper = $anccount > 0 ? $this->round2(($acckey / $anccount) * 100) : NULL;
$mainmenuval = ($anccount > 0 || count($objects) > 0 || count($embeds) > 0 || count($scripts) >
0) ? 1 : 0;
$htmlmenuval = ($anccount > 0 && count($objects) == 0 && count($embeds) == 0 &&
count($scripts) == 0) ? 1 : 0;

return array
(
    'count'           => $anccount, //Link count
    'withoutimgper'  => $withoutimgper, //Percentage of links without image
    'imgaltper'      => $imgaltper, //Percentage of alt attributes in links with images
    'tabidxper'      => $tabidxper, //Percentage of links with tabindex attributes
    'acckeyper'      => $acckeyper, //Percentage of links with accesskey attributes
    'mainmenuvalue'  => $mainmenuval, // 1: Possible, 0: Not possible
    'htmlmenuvalue' => $htmlmenuval, // 1: Possible, 0: Not possible

    'imgaltest'     => test_eq($imgaltper, 100), // Test of alt attributes in links with images
    'tabidxtest'    => test_eq($tabidxper, 100), //Test of links with tabindex attributes
    'acckeytest'    => test_eq($acckeyper, 100), //Test of links with accesskey attributes
    'mainmenutest'  => test_poss($mainmenuval), //Main menu test
    'htmlmenutest'  => test_poss($htmlmenuval), //HTML menu test

    'hrefs'         => $anccount > 0 ? $hrefs : NULL //Hrefs
);
}

//Body tag and its attributes
function arr_body ()
{
    $body = $this->parse_html($this->fcont2, 'body', 'a');
    $val = count($body) > 0 ? 1 : 0;

    $body = "";
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_BODY, "", $body);
    if (is_array($body) && is_array($body[0]['attribute']))
    {
        if (array_key_exists('text', $body[0]['attribute'])) {$textval = 1;} else {$textval = 0;}
        if (array_key_exists('link', $body[0]['attribute'])) {$linkval = 1;} else {$linkval = 0;}
        if (array_key_exists('alink', $body[0]['attribute'])) {$alinkval = 1;} else {$alinkval = 0;}
        if (array_key_exists('vlink', $body[0]['attribute'])) {$vlinkval = 1;} else {$vlinkval = 0;}
    } else
    {
        $textval = 0;
        $linkval = 0;
        $alinkval = 0;
    }
}

```

```

    $vlinkval = 0;
}
return array
(
    'value'           => $val, //Body value. 1: Successfull , 0: Failed
    'textvalue'      => $textval, //1: Successfull , 0: Failed
    'linkvalue'      => $linkval, //1: Successfull , 0: Failed
    'alinkvalue'     => $alinkval, //1: Successfull , 0: Failed
    'vlinkvalue'     => $vlinkval, //1: Successfull , 0: Failed

    'test'           => test_eq($val, 1),
    'texttest'       => test_eq($textval, 1),
    'linktest'       => test_eq($linkval, 1),
    'alinktest'      => test_eq($alinkval, 1),
    'vlinktest'      => test_eq($vlinkval, 1)
);
}

//Forms and form elements
function arr_form ()
{
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_OBJECT, " ", $objects); $objcount =
count($objects);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_EMBED, " ", $embeds); $sembcount =
count($embeds);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_FORM, " ", $forms); $frmcount =
count($forms);

    if ($objcount + $sembcount == 0)
    {
        $val = $frmcount > 0 ? 100 : NULL;
    } else if ($objcount + $sembcount > 0)
    {
        $val = $frmcount > 0 ? NULL : NULL;
    }

    $this->get_nodes2($this->tidyhtml, TIDY_TAG_INPUT, " ", $inputs); $inpcount =
count($inputs);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_SELECT, " ", $selects); $selcount =
count($selects);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_TEXTAREA, " ", $textareas); $stareacount =
count($textareas);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_BUTTON, " ", $buttons); $butcount =
count($buttons);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_LABEL, " ", $labels); $lblcount =
count($labels);

    $sinptypeforlbl = array('text', 'password', 'checkbox', 'radio', 'file'); //Input type for labels
    $inpforlbl = 0;

    $lbl = 0;
    $stabdx = 0;
    $inpacckey = 0;
    $butacckey = 0;
    $stareaacckey = 0;
    $selacckey = 0;

    if ($inpcount > 0)
    {

```

```

foreach ($inputs as $inp)
{
    //Label
    if ((is_array($inp['attribute']) && array_key_exists('type', $inp['attribute']) &&
        $inp['attribute']['type'] <> 'hidden' && in_array($inp['attribute']['type'],
$sinptypeforlbl)) || (is_array($inp['attribute']) && !array_key_exists('type', $inp['attribute']) ||
!is_array($inp['attribute'])))
    {
        $inpforlbl ++;
        if (is_array($inp['attribute']) && array_key_exists('id', $inp['attribute']) &&
$lblcount > 0 && $this->get_label_for($labels, $inp['attribute']['id'])) $lbl ++;
    }

    if (is_array($inp['attribute']))
    {
        //Tabindex
        if (((array_key_exists('type', $inp['attribute']) && $inp['attribute']['type'] <>
'hidden') || !array_key_exists('type', $inp['attribute'])) && array_key_exists('tabindex',
$inp['attribute'])) $tabidx ++;

        //Accesskey
        if ((array_key_exists('accesskey', $inp['attribute']) && ((array_key_exists('type',
$inp['attribute']) && $inp['attribute']['type'] <> 'hidden') || (!array_key_exists('type',
$inp['attribute']))) || (array_key_exists('id', $inp['attribute']) && $lblcount > 0 && $this-
>get_label_accesskey($labels, $inp['attribute']['id']))) $inpacckey ++;

        if (array_key_exists('type', $inp['attribute']) && $inp['attribute']['type'] ==
'hidden') $inpcount --;
    }
}

if ($selcount > 0)
{
    foreach ($selects as $sel)
    {
        if (is_array($sel['attribute']))
        {
            //Label
            if ($lblcount > 0 && array_key_exists('id', $sel['attribute']) && $this-
>get_label_for($labels, $sel['attribute']['id'])) $lbl ++;

            //Tabindex
            if (array_key_exists('tabindex', $sel['attribute'])) $tabidx ++;

            //Accesskey
            if (array_key_exists('accesskey', $sel['attribute']) || (array_key_exists('id',
$sel['attribute']) && $lblcount > 0 && $this->get_label_accesskey($labels, $sel['attribute']['id'])))
$selacckey ++;
        }
    }
}

if ($stareacount > 0)
{
    foreach ($textareas as $starea)
    {
        if (is_array($starea['attribute']))
        {

```

```

        //Label
        if ($lblcount > 0 && array_key_exists('id', $starea['attribute']) && $this->get_label_for($labels, $starea['attribute']['id'])) $lbl ++;

        //Tabindex
        if (array_key_exists('tabindex', $starea['attribute'])) $tabidx ++;

        //Accesskey
        if (array_key_exists('accesskey', $starea['attribute']) || (array_key_exists('id', $starea['attribute']) && $lblcount > 0 && $this->get_label_accesskey($labels, $starea['attribute']['id'])))
            $stareaacckey ++;
    }
}

if ($butcount > 0)
{
    foreach ($buttons as $but)
    {
        if (is_array($but['attribute']))
        {
            //Accesskey
            if (array_key_exists('accesskey', $but['attribute'])) $butacckey ++;
            //Tabindex
            if (array_key_exists('tabindex', $but['attribute'])) $tabidx ++;
        }
    }
}

$lblcount = $inpforlbl + $selcount + $stareacount;
$labelper = $frmcount > 0 && $lblcount > 0 ? $this->round2(($lbl / $lblcount) * 100) :
NULL;

$tabidxcount = $inpcount + $selcount + $stareacount + $butcount;
$tabidxper = $frmcount > 0 && $tabidxcount > 0 ? $this->round2(($tabidx /
$tabidxcount) * 100) : NULL;

$inpacckeyper = $inpcount > 0 ? $this->round2(($inpacckey / $inpcount) * 100) : NULL;
$selacckeyper = $selcount > 0 ? $this->round2(($selacckey / $selcount) * 100) : NULL;
$stareaacckeyper = $stareacount > 0 ? $this->round2(($stareaacckey / $stareacount) * 100) : NULL;
$butacckeyper = $butcount > 0 ? $this->round2(($butacckey / $butcount) * 100) : NULL;

return array
(
    'count'           => $frmcount, //HTML form count
    'objectcount'    => $objcount,
    'embedcount'     => $embcount,
    'value'          => $val,
    'labelper'       => $labelper, //Percentage of label tags
    'tabidxper'      => $tabidxper, //Percentage of tabindex attributes
    'inpacckeyper'   => $inpacckeyper, //Percentage of inputs with accesskey attribute
    'selacckeyper'   => $selacckeyper, //Percentage of selects with accesskey attribute
    'tareaacckeyper' => $stareaacckeyper, //Percentage of textareas with accesskey attribute
    'butacckeyper'   => $butacckeyper, //Percentage of buttons with accesskey attribute

    'test'           => test_form($frmcount, $objcount, $embcount, $val), //Form test
    'labeltest'      => test_eq($labelper, 100),
    'tabidxtest'     => test_eq($tabidxper, 100),
    'inpacckeytest' => test_eq($inpacckeyper, 100),

```



```

        'selacckeytest'    => test_eq($selacckeyper, 100),
        'tareaacckeytest' => test_eq($tareaacckeyper, 100),
        'butacckeytest'  => test_eq($butacckeyper, 100)
    );
}

//Font tags
function arr_font ()
{
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_FONT, "", $fonts);
    $sval = count($fonts) == 0 ? 1 : 0;

    return array
    (
        'value'    => $sval, // 1: Successfull, 0: Failed
        'test'     => test_eq($sval, 1)
    );
}

//Marquee tags
function arr_marquee ()
{
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_MARQUEE, "", $marquees);
    $sval = count($marquees) == 0 ? 1 : 0;

    return array
    (
        'value'    => $sval, // 1: Successfull, 0: Failed
        'test'     => test_eq($sval, 1)
    );
}

//Image maps
function arr_map ()
{
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_MAP, "", $maps);
    $mapcount = count($maps);

    $sareaalt = 0;
    $sareatabidx = 0;
    $sareaacckey = 0;

    if ($mapcount > 0)
    {
        $this->get_nodes2($this->tidyhtml, TIDY_TAG_AREA, "", $areas);
        $sareacount = count($areas);

        if ($sareacount > 0)
        {
            foreach ($areas as $sarea)
            {
                if (is_array($sarea['attribute']))
                {
                    //Tabindex
                    if (array_key_exists('tabindex', $sarea['attribute'])) $sareatabidx ++;
                    //Accesskey
                    if (array_key_exists('accesskey', $sarea['attribute'])) $sareaacckey ++;
                    //Alt
                    if (array_key_exists('alt', $sarea['attribute'])) $sareaalt ++;
                }
            }
        }
    }
}

```

```

    }
  }
} else $areacount = 0;

$areaaltper = ($mapcount > 0 && $areacount > 0) ? $this->round2(($areaalt / $areacount) * 100)
: NULL;
$areatabidxper = ($mapcount > 0 && $areacount > 0) ? $this->round2(($areatabidx /
$areacount) * 100) : NULL;
$areaacckeyper = ($mapcount > 0 && $areacount > 0) ? $this->round2(($areaacckey /
$areacount) * 100) : NULL;

return array
(
  'mapcount'    => $mapcount, //Image map count
  'areacount'   => $areacount, //Area count
  'areaaltper'  => $areaaltper, //Percentage of areas with alt attribute
  'areatabidxper' => $areatabidxper, //Percentage of areas with tabindex attribute
  'areaacckeyper' => $areaacckeyper, //Percentage of areas with accesskey attribute

  'areaalttest'  => test_eq($areaaltper, 100),
  'areatabidxtest' => test_eq($areatabidxper, 100),
  'areaacckeytest' => test_eq($areaacckeyper, 100),
  'apptest'     => test_approp($mapcount) //Appropriate test
);
}

//Images
function arr_img ()
{
  $this->get_nodes2($this->tidyhtml, TIDY_TAG_IMG, " ", $imgs);
  $imgcount = count($imgs);

  $imgexttype = 0;
  $imgfilesize = 0;
  $imgwidth = 0;
  $imgheight = 0;
  $imgalt = 0;
  $imgaltlength = 0;
  $imgaltendchar = 0;
  $imgaltendchars = 0;
  $logos = false;
  $giflogo = false;
  $statlogo = true;

  if ($imgcount > 0)
  {
    foreach ($imgs as $img)
    {
      if (is_array($img['attribute']))
      {
        if (array_key_exists('alt', $img['attribute']))
        {
          $imgalt++;
          $endchar = substr($img['attribute']['alt'], -1);
          if (strlen($img['attribute']['alt']) <= 100) $imgaltlength ++;
          if ($endchar == '!' || $endchar == ')') $imgaltendchar ++;
          if (preg_match("/\s+$/iX", $img['attribute']['alt'])) $imgaltendchars ++;
        }
      }
    }
  }
}

```

```

        if ($this->get_att($this->extintesscont->css[41], 'img', $img['attribute'], 'width',
'width')) $imgwidth ++;
        if ($this->get_att($this->extintesscont->css[41], 'img', $img['attribute'], 'height',
'height')) $imgheight ++;
    }
}

$this->scan_directory_recursively($this->settings['fo_images'], false, $imgfiles);
$downloadedimg = count($imgfiles);

if ($downloadedimg > 0)
{
    foreach ($imgfiles as $imgfile)
    {
        $pfile = $this->settings['fo_images'].'.$imgfile['name'];

        if ((exif_imagetype($pfile) == IMAGETYPE_GIF || (exif_imagetype($pfile) ==
IMAGETYPE_JPEG)) $imgexttype ++;
        if ($imgfile[size] <= 30720) $imgfilesize ++;

        if (preg_match("/(?:".$this->settings['p_logo'].")/iX", $imgfile['name']))
        {
            $logos = true;
            if ((exif_imagetype($pfile) == IMAGETYPE_GIF))
            {
                $giflogo = true;
                $statlogo = false;
            }
        }
    }
}

$imgexttypeper = $downloadedimg > 0 ? $this->round2(($imgexttype / $downloadedimg) *
100) : NULL;
$imgfilesizeper = $downloadedimg > 0 ? $this->round2(($imgfilesize / $downloadedimg) * 100)
: NULL;
$imgwidthper = $imgcount > 0 ? $this->round2(($imgwidth / $imgcount) * 100) : NULL;
$imgheightper = $imgcount > 0 ? $this->round2(($imgheight / $imgcount) * 100) : NULL;
$imgaltper = $imgcount > 0 ? $this->round2(($imgalt / $imgcount) * 100) : NULL;
$imgaltlengthper = $imgalt > 0 ? $this->round2(($imgaltlength / $imgalt) * 100) : NULL;
$imgaltendcharper = $imgalt > 0 ? $this->round2(($imgaltendchar / $imgalt) * 100) : NULL;
$imgaltendcharsper = $imgalt > 0 ? $this->round2(($imgaltendchars / $imgalt) * 100) : NULL;
$logoval = $logos ? 1 : 0;
$giflogoval = $logos == true ? ($giflogo == true ? 1 : 0) : NULL;
$textreadlogoval = $logos == true ? NULL : NULL;
$statlogoval = $logos == true ? ($statlogo == true ? 1 : 0) : NULL;

return array
(
    'count'           => $imgcount, // Image count
    'downloaded'     => $downloadedimg, //Downloaded image count
    'exttypeper'     => $imgexttypeper, //Percentage of JPEG&GIF
    'filesizeper'    => $imgfilesizeper, //Percentage of images with 30K max size
    'widthper'       => $imgwidthper, //Percentage of images with 'with' attribute
    'heightper'      => $imgheightper, //Percentage of images with height attribute
    'altper'         => $imgaltper, //Percentage of images with alt attribute
    'altlengthper'   => $imgaltlengthper, //Percentage of alt attributes with 100 max chars

```

```

    'altendcharper'    => $imgaltendcharper, //Percentage of alt attributes with last char is dot or
space
    'altendcharsper'  => $imgaltendcharsper, //Percentage of alt attributes with last chars are dot
and space
    'logoval'         => $logoval, //1: Possible 0: Not possible
    'giflogoval'      => $giflogoval, //1: Possible 0: Not possible
    'textreadlogoval' => $textreadlogoval, //1: Possible 0: Not possible
    'statlogoval'     => $statlogoval, //1: Possible 0: Not possible

    'exttypetest'     => test_eq($imgexttypeper, 100),
    'filesizetest'    => test_eq($imgfilesizeper, 100),
    'widthtest'       => test_eq($imgwidthper, 100),
    'heighttest'      => test_eq($imgheightper, 100),
    'alttest'         => test_eq($imgaltper, 100),
    'allengthtest'    => test_eq($imgallengthper, 100),
    'altendchartest'  => test_eq($imgaltendcharper, 100),
    'altendchartest'  => test_eq($imgaltendcharsper, 100),
    'logotest'        => test_poss($logoval),
    'giflogotest'     => test_poss_rel($giflogoval, $logoval),
    'textreadlogotest' => test_poss_rel($textreadlogoval, $logoval),
    'statlogotest'    => test_poss_rel($statlogoval, $logoval)
);
}

//Tables
function arr_table ()
{
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_TABLE, "", $Tables);
    $tblcount = count($Tables);
    $tblendtags = $this->parse_html($this->fcont2, 'table', 'a');
    $tblendtagcount = count($tblendtags);
    $tblendtagcount = $tblendtagcount > $tblcount ? $tblcount : $tblendtagcount;

    $tblbgcolor = 0;
    $tblbground = 0;

    if ($tblcount > 0)
    {
        foreach ($Tables as $table)
        {
            if (!$this->get_att($this->extintcsscont->css[41], 'table', $table['attribute'], 'background-
color', 'bgcolor')) $tblbgcolor ++;
            if (!$this->get_att($this->extintcsscont->css[41], 'table', $table['attribute'], 'background-
image', 'background')) $tblbground ++;
        }
    }

    $tblendtagper = $tblcount > 0 ? $this->round2(($tblendtagcount / $tblcount) * 100) : NULL;
    $tblbgcolorper = $tblcount > 0 ? $this->round2(($tblbgcolor / $tblcount) * 100) : NULL;
    $tblbgroundper = $tblcount > 0 ? $this->round2(($tblbground / $tblcount) * 100) : NULL;

    return array
    (
        'count'          => $tblcount, //Table count
        'endtagper'      => $tblendtagper, //Percentage of table with end tag
        'bgcolorper'     => $tblbgcolorper, //Percentage of table without background color
        'bgroundper'     => $tblbgroundper, //Percentage of table without background image
    )
}

```

```

        'endtagtest' => test_eq($tblendtagper, 100),
        'bgcolortest' => test_eq($tblbgcolorper, 100),
        'bgroundtest' => test_eq($tblbgroundper, 100)
    );
}

//Colors
function arr_color ($val)
{
    // 1: External and Internal, 2: Inline, 3: HTML Color
    $val = $val == 1 ? $this->extintsscont : ($val == 2 ? $this->inlinecsscont : $this->htmltmpesscont);

    if (count($val->css) > 0)
    {
        $this->css_color($val->css[41], $count, $valid);
    } else
    {
        $count = 0;
        $valid = 0;
    }

    $per = $count > 0 ? $this->round2(($valid / $count) * 100) : NULL;

    return array
    (
        'per' => $per,
        'test' => test_eq($per, 100)
    );
}

//Font Sizes
function arr_font_size($val)
{
    // 1: External and Internal, 2: Inline,
    $val = $val == 1 ? $this->extintsscont : $this->inlinecsscont;

    if (count($val->css) > 0)
    {
        $this->css_font_size($val->css[41], $count, $valid);
    } else
    {
        $count = 0;
        $valid = 0;
    }

    $per = $count > 0 ? $this->round2(($valid / $count) * 100) : NULL;

    return array
    (
        'per' => $per,
        'test' => test_eq($per, 100)
    );
}

//Page Size
function arr_page_size ()
{
    chdir($this->settings['fo_url']);
}

```

```

$psize = filesize('index.htm');

$imgtotsize = 0;
$this->scan_directory_recursively($this->settings['fo_images'], false, $imgfiles);

if (count($imgfiles) > 0) foreach ($imgfiles as $imgfile) $imgtotsize += $imgfile['size'];
$psizewithimgs = $this->round2(($psize + $imgtotsize) / 1024);

return array
(
    'size'           => $this->round2($psize / 1024), //Page size
    'sizewithimgs'  => $psizewithimgs, //Page size with images
    'sizewithimgstest' => test_lt($psizewithimgs, 50.01)
);
}

//Content tests
function arr_content ()
{
    //Logos
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_IMG, "", $imgs);
    $imgcount = count($imgs);

    if ($imgcount > 0)
    {
        foreach ($imgs as $img)
        {
            if (is_array($img['attribute']) && array_key_exists('src', $img['attribute']))
            {
                if (preg_match("/(?:\".$this->settings['p_logo'].)/iX", $img['attribute']['src']))
                {
                    $logos .= $img['attribute']['src']."\n";
                }
            }
        }
    }
}

$logoval = isset($logos) ? 1 : 0;

//Institution Name
$this->get_nodes2($this->tidyhtml, TIDY_TAG_TITLE, "", $titles);
$instname = $this->parse_html($titles[0]['value'], 'title', 'b');

$instname = strlen($instname) > 0 ? $instname : NULL;
$instnameval = isset($instname) ? 1 : 0;
//Anchors, Helps, FAQs, Feeds...
$this->get_nodes2($this->tidyhtml, TIDY_TAG_A, "", $anchors);
$anccount = count($anchors);

if ($anccount > 0)
{
    foreach ($anchors as $anc)
    {
        if (is_array($anc['attribute']) && array_key_exists('href', $anc['attribute']))
        {
            $hrefs .= $anc['attribute']['href']."\n";

            $btwanc = $this->parse_html($anc['value'], 'a', 'b');

```

```

//Helps
preg_match_all("/(?:". $this->settings['p_help']. ")/iX", $btwanc, $helps_);
if (count($helps_[0]) > 0) foreach ($helps_[0] as $help) $helps .= $help. "\n";

//FAQs
preg_match_all("/(?:". $this->settings['p_faq']. ")/iX", $btwanc, $faqs_);
if (count($faqs_[0]) > 0) foreach ($faqs_[0] as $faq) $faqs .= $faq. "\n";

//Feedback
preg_match_all("/(?:". $this->settings['p_feed']. ")/iX", $btwanc, $feeds_);
if (count($feeds_[0]) > 0) foreach ($feeds_[0] as $feed) $feeds .= $feed. "\n";
    }
}
}

$ancval = isset($hrefs) ? 1 : 0;
$helpval = isset($helps) ? 1 : 0;
$faqval = isset($faqs) ? 1 : 0;
$feedval = isset($feeds) ? 1 : 0;

//Aims
preg_match_all("/(?:". $this->settings['p_aim']. ")/iX", $this->fcont3, $aims_);
if (count($aims_[0]) > 0) foreach ($aims_[0] as $aim) $aims .= $aim. "\n";

$aimval = isset($aims) ? 1 : 0;

//Announcements
preg_match_all("/(?:". $this->settings['p_anno']. ")/iX", $this->fcont3, $annos_);
if (count($annos_[0]) > 0) foreach ($annos_[0] as $anno) $annos .= $anno. "\n";

$annoval = isset($annos) ? 1 : 0;

//Search Mec.
$this->get_nodes2($this->tidyhtml, TIDY_TAG_FORM, " ", $forms);

if (count($forms) > 0)
{
    foreach ($forms as $form)
    {
        $val = $this->parse_html($form['value'], 'form', 'b');
        preg_match_all("/(?:". $this->settings['p_search']. ")/iX", $val, $searchs_);
        if (count($searchs_[0]) > 0) foreach ($searchs_[0] as $search) $searchs .=
$search. "\n";
    }
}
$searchval = isset($searchs) ? 1 : 0;

//Address
preg_match_all("/(?:". $this->settings['p_address']. ")/iX", $this->fcont3, $addresses_);
if (count($addresses_[0]) > 0) foreach ($addresses_[0] as $address) $addresses .= $address. "\n";

$addressval = isset($addresses) ? 1 : 0;

//Telephone
preg_match_all("/(?:?:?". $this->settings['p_phone']. ")\s*:\s*\+\s*[0-9]\s*[0-9]\s*(?:\s*[0-9]{3,4}\s*)?\s*[0-9]{3}\s*[0-9]{2}\s*[0-9]{2}/iX", $this->fcont3, $phones_);
if (count($phones_[0]) > 0) foreach ($phones_[0] as $phone) $phones .= $phone. "\n";

```

```

$phoneval = isset($phones) ? 1 : 0;

//e-Mails
preg_match_all('/(?:([a-z0-9_-\.\+])@(?:[-0-9a-z_-\.\+]{1,50}[a-z0-9]{2,4})/iX', $this-
>fcont2.$this->fcont3, $mails_);
if (count($mails_[0]) > 0)
{
    foreach ($mails_[0] as $k => $v)
    {
        $mails .= $v."\\n";
        if (preg_match("/(?:\".$this->settings['p_appropmail'].\")/", $mails_[1][$k]))
        $appropmails .= $v."\\n";
    }
}

$mailval = isset($mails) ? 1 : 0;
$appropmailval = $mailval == 1 ? (isset($appropmails) ? 1 : 0) : NULL;

return array
(
    'logovalue'           => $logoval, //1: Possible, 0: Not Possible
    'instnamevalue'     => $instnameval, //1: Possible, 0: Not Possible
    'aimvalue'          => $aimval, //1: Possible, 0: Not Possible
    'ancevalue'         => $anceval, //1: Possible, 0: Not Possible
    'annovalue'         => $annoval, //1: Possible, 0: Not Possible
    'searchvalue'       => $searchval, //1: Possible, 0: Not Possible
    'helpvalue'         => $helpval, //1: Possible, 0: Not Possible
    'faqvalue'          => $faqval, //1: Possible, 0: Not Possible
    'feedvalue'         => $feedval, //1: Possible, 0: Not Possible
    'addressvalue'      => $addressval, //1: Possible, 0: Not Possible
    'phonevalue'        => $phoneval, //1: Possible, 0: Not Possible
    'mailvalue'         => $mailval, //1: Possible, 0: Not Possible
    'appropmailvalue'   => $appropmailval, //1: Possible, 0: Not Possible
    'logos'             => $logos,
    'instname'          => $instname,
    'aims'              => $aims,
    'hrefs'             => $hrefs,
    'annos'             => $annos,
    'searchs'           => $searchs,
    'helps'             => $helps,
    'faqs'              => $faqs,
    'feeds'             => $feeds,
    'addresses'         => $addresses,
    'phones'            => $phones,
    'mails'             => $mails,
    'appropmails'       => $appropmails,
    'logotest'          => test_poss($logoval),
    'instnametest'      => test_poss($instnameval),
    'aimtest'           => test_poss($aimval),
    'anctest'           => test_poss($anceval),
    'annotest'          => test_poss($annoval),
    'searchtest'        => test_poss($searchval),
    'helptest'          => test_poss($helpval),
    'faqtest'           => test_poss($faqval),
    'feedtest'          => test_poss($feedval),
    'adresstest'        => test_poss($addressval),
    'phonetest'         => test_poss($phoneval),
    'mailtest'          => test_poss($mailval),
    'appropmailtest'    => test_poss_rel($appropmailval, $mailval)
)

```



```

    );
}

//Accessibility test
function arr_accessibility ()
{
    //Browsers
    $ieval = NULL; //Internet Explorer value
    $ffval = NULL; //Firefox value
    $konval = NULL; //Konqueror value

    //For Lynx
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_OBJECT, " ", $objects); $objcount =
count($objects);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_EMBED, " ", $embeds); $sembcount =
count($embeds);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_IMG, " ", $imgs); $imgcount = count($imgs);
    $this->get_nodes2($this->tidyhtml, TIDY_TAG_MAP, " ", $maps); $mapcount =
count($maps);

    $imgalt = 0;
    if ($imgcount > 0) foreach ($imgs as $img) if (is_array($img['attribute']) &&
array_key_exists('alt', $img['attribute'])) $imgalt++;
    $imgaltper = $imgcount > 0 ? $this->round2(($imgalt / $imgcount) * 100) : NULL;

    $lyval = (($imgaltper == 100 || $imgaltper == NULL) && $objcount == 0 && $sembcount == 0
&& $mapcount == 0) ? 1 : 0; //Lynx value

    //Resolution 1: 1280x1024, 2: 1024x768, 3: 800x600
    $res1val = NULL;
    $res2val = NULL;
    $res3val = NULL;

    return array
    (
        'ievalue'      => $ieval,
        'ffvalue'     => $ffval,
        'konvalue'    => $konval,
        'lyvalue'     => $lyval,
        'res1value'   => $res1val,
        'res2value'   => $res2val,
        'res3value'   => $res3val,
        'ietest'      => test_other($ieval),
        'fftest'      => test_other($ffval),
        'kontest'     => test_other($konval),
        'lytest'      => test_poss($lyval),
        'res1test'    => test_other($res1val),
        'res2test'    => test_other($res2val),
        'res3test'    => test_other($res3val)
    );
}
} //Class
?>

```

EK B.2 Sonuç Fonksiyonları (test_functions.php)

```

<?php

//Test result functios...

//Two values of the equivalent
function test_eq ($val1, $val2)
{
    return isset($val1) ? ($val1 == $val2 ? SUCCESSFUL : FAILED) : NULL_X;
}

//First value is less than second value
function test_lt ($val1, $val2)
{
    return isset($val1) ? ($val1 < $val2 ? SUCCESSFUL : FAILED) : NULL_X;
}

//First value is greater than second value
function test_gt ($val1, $val2)
{
    return isset($val1) ? ($val1 > $val2 ? SUCCESSFUL : FAILED) : NULL_X;
}

//Possible or not possible
function test_poss ($val)
{
    return isset($val) ? ($val == 1 ? POSSIBLE : NOT_POSSIBLE) : N_A;
}

function test_poss_rel ($val1, $val2)
{
    return $val2 == 1 ? test_poss($val1) : NULL_X;
}

//Appropriate test
function test_approp ($val1, $val2 = false)
{
    return $val2 == '1' ? SUCCESSFUL : ($val2 == '0' ? FAILED : ($val1 > 0 ? N_A : NULL_X));
}

function test_other($val)
{
    return isset($val) ? ($val == 1 ? SUCCESSFUL : FAILED) : N_A;
}

//Form test
function test_form($f, $o, $e, $fv = false)
{
    if (isset($fv))
    {
        return $fv == 100 ? SUCCESSFUL : FAILED;
    } else
    {
        if ($o + $e == 0)
        {
            return $f > 0 ? SUCCESSFUL : NULL_X;
        } else if ($o + $e > 0)
    }
}

```

```

    {
        return $f > 0 ? N_A : N_A;
    }
}
?>

```

EK B.3 İçerik Testleri Dil Dosyaları

EK B.3.1 Türkçe Dil Dosyası (/locale_set/tr.php)

```

<?php

define("SUCCESSFUL", "<img src='images/successful.gif' alt='Başarılı. ' width='16' height='16'>");
define("FAILED", "<img src='images/failed.gif' alt='Başarısız. ' width='16' height='16'>");
define("N_A", "<img src='images/n_a.gif' alt='Belirlenemedi. ' width='17' height='17'>");
define("NULL_X", "<img src='images/null.gif' alt='Yok. ' width='15' height='12'>");
define("POSSIBLE", "<img src='images/possible.gif' alt='Olabilir. ' width='16' height='16'>");
define("NOT_POSSIBLE", "<img src='images/notpossible.gif' alt='Olmayabilir. ' width='16' height='16'>");

$locale['100'] = "lg|logo";
$locale['101'] = "yardım";
$locale['102'] = "sss\s\s\s|sıkça\s+sorulan\s+SORU";
$locale['103'] = "bilgi\s+edinme|bize\s+ulaşın";
$locale['104'] = "amaç|amac";
$locale['105'] = "duyuru|haber|yenilik";
$locale['106'] = "ara";
$locale['107'] = "adres|cadde|cad\|cd\|mahalle|mah\|sem|sokak|sok\|sk\.";
$locale['108'] = "tel|telefon";
$locale['109'] = "bilgi|iletisim|şikayet|yardım|[0-9]+";

?>

```

EK B.3.2 İngilizce Dil Dosyası (/locale_set/en.php)

```

<?php

define("SUCCESSFUL", "<img src='images/successful.gif' alt='Successful. ' width='16' height='16'>");
define("FAILED", "<img src='images/failed.gif' alt='Failed. ' width='16' height='16'>");
define("N_A", "<img src='images/n_a.gif' alt='N/A. ' width='17' height='17'>");
define("NULL_X", "<img src='images/null.gif' alt='NULL. ' width='15' height='12'>");
define("POSSIBLE", "<img src='images/possible.gif' alt='Possible. ' width='16' height='16'>");
define("NOT_POSSIBLE", "<img src='images/notpossible.gif' alt='Not Possible. ' width='16' height='16'>");

$locale['100'] = "lg|logo";

```

```
$locale['101'] = "help";  
$locale['102'] = "faq|f\.a\.q|frequently\s+asked\s+question";  
$locale['103'] = "feedback";  
$locale['104'] = "aim|goal|intention|purpose|objective";  
$locale['105'] = "annos|announcement";  
$locale['106'] = "search";  
$locale['107'] = "address|avenue|district|street|st\.";  
$locale['108'] = "phone";  
$locale['109'] = "support|info|complaint|help|[0-9]+";
```

```
?>
```

EK C WEB SAYFASI DEĞERLENDİRME ARACI

EK C.1 Kaynak Kodu (check.php)

```

<?php
ob_start();
session_start();

//Temp directories removing function (based by lixlpixel.org)
function rm_dir_recursively($directory)
{
    if(substr($directory,-1) == '/')
    {
        $directory = substr($directory,0,-1);
    }
    if(!file_exists($directory) || !is_dir($directory))
    {
        return FALSE;
    } elseif(is_readable($directory))
    {
        $directory_list = opendir($directory);
        while (FALSE !== ($file = readdir($directory_list)))
        {
            if($file != '.' && $file != '..')
            {
                $path = $directory.'/'.$file;
                if(is_readable($path))
                {
                    if(is_dir($path))
                    {
                        if(!rmdir($path))
                        {
                            rm_dir_recursively($path);
                            rmdir($path);
                        }
                    } elseif(is_file($path))
                    {
                        unlink($path);
                    }
                }
            }
        }
        closedir($directory_list);
    } else return FALSE;
}

//Trim function (by Nick Jones)
function trimlink($text, $length)
{
    $dec = array("&", "\"", "'", "\\", "\'", "\", "<", ">");
    $enc = array("&amp;", "&quot;", "&#39;", "&#92;", "&quot;", "&#39;", "&lt;", "&gt;");
}

```

```

$text = str_replace($enc, $dec, $text);
if (strlen($text) > $length) $text = substr($text, 0, ($length-3))."...";
$text = str_replace($dec, $enc, $text);
return $text;
}

// Universal page pagination function (based by CrappoMan)
function makepagenav($start,$count,$total,$range=0,$link="")
{
    $res="";
    $pg_cnt=ceil($total / $count);
    if ($pg_cnt > 1) {
        $idx_back = $start - $count;
        $idx_next = $start + $count;
        $cur_page=ceil(($start + 1) / $count);
        $res.="$cur_page/$pg_cnt \n";
        if ($idx_back >= 0) {
            if ($cur_page > ($range + 1)) $res. "<a href='$link'.\"rowstart=0\">&lt;&lt;</a>\n";
            $res. "<a href='$link'.\"rowstart=$idx_back\">&lt;</a>\n";
        }
        $idx_fst=max($cur_page - $range, 1);
        $idx_lst=min($cur_page + $range, $pg_cnt);
        if ($range==0) {
            $idx_fst = 1;
            $idx_lst=$pg_cnt;
        }
        for($i=$idx_fst;$i<=$idx_lst;$i++) {
            $offset_page=( $i - 1 ) * $count;
            if ($i==$cur_page) {
                $res. "<b>$i</b>\n";
            } else {
                $res. "<a href='$link'.\"rowstart=$offset_page\">$i</a>\n";
            }
        }
        if ($idx_next < $total) {
            $res. "<a href='$link'.\"rowstart=$idx_next\">&gt;</a>\n";
            if ($cur_page < ($pg_cnt - $range)) $res. "<a href='$link'.\"rowstart=".($pg_cnt-1)*$count.\">&gt;&gt;</a>\n";
        }
    }
    return $res;
}

function edit_frm_input($field, $crit1 = false)
{
    global $data;
    if (isset($_SESSION['admin']))
    {
        $ret = "<input type='text' size='5' class='textbox' name='". $field. "\"
value='". $data[$field]. ">";
        return !$crit1 ? $ret : ($data[$crit1] > 0 ? $ret : "");
    } else return "";
}

function round_cvt_per($val)
{
    return (round($val, 2)*100).'%';
}

```

```

function round_per($val)
{
    return round($val, 2).' (%)';
}

function round_number($val)
{
    return round($val, 2);
}

//Begin
$thisfile = basename(__file__);

if(!isset($_SESSION['language']))
{
    $sarr_lan = array('tr', 'en');
    $lan = @$_GET['lan'];
    if (!isset($lan) || !in_array($lan, $sarr_lan)) $lan = 'tr';
} else $lan = $_SESSION['language'];

require_once 'locale_set/'.$lan.'.php';
require_once 'locale_res/'.$lan.'.php';
require_once 'locale_cri/'.$lan.'.php';
require_once 'class.webpage.php';
require_once 'test_functions.php';
require_once 'class.database.php';

echo "
<!DOCTYPE HTML PUBLIC '-//W3C//DTD HTML 4.01 Transitional//EN'>
<html>
<head>
    <title>".$locale['e100']. "</title>
    <meta http-equiv='Content-Type' content='text/html; charset=iso-8859-9'>
    <meta name='description' content='".$locale['e100']."'>
    <meta name='keywords' content='".$locale['e101']."'>
    <link rel='stylesheet' href='style.css' type='text/css'>
    <script type='text/javascript'>
        function popUp(URL, W, H)
        {
            eval("window.open(URL, ", 'toolbar = 1, scrollbars = 1, location = 0, statusbar = 0,
menubar = 0, resizable = 0, width = \'' + W + '\', height = \'' + H + '\', left = 128, top = 128);");
        }
    </script>
</head>
<body text='black' link='black' alink='black' vlink='black'>\n";

$db = new database("mysql", "localhost", "WEBTEST", "webtest", "erkan");
$db->debug_mode(false);
$db->query(file_get_contents('sql_info.txt'));
$db->query(file_get_contents('sql_webpage.txt'));

$db->query("SELECT * FROM info");
if ($db->count_rows() == 0)
{
    $db->query("INSERT INTO info(total, visited, last_ip) VALUES(0, 0, NULL)");
    $db->query("SELECT * FROM info");
}
$info_row = $db->get_row();

```

```

if (isset($_SESSION['admin']))
{
    $site_id = @$_GET['site_id'];
    if (isset($site_id)) $_SESSION['web_site_id'] = $site_id;
}

if(!isset($_SESSION['web_site_id']))
{
    if (!isset($_POST['ok']))
    {
        echo "<p align = 'right'>
            <a href='\".$thisfile.\"?lan=tr' accesskey='t' tabindex='1'>Türkçe</a> |
            <a href='\".$thisfile.\"?lan=en' accesskey='e' tabindex='2'>English</a>
        </p>";
        echo "<center>".$_locale['e102']."$_locale['e103']."$_locale['e104']."$_locale['e105'];
        $_SESSION['security_code'] = substr(md5(rand(0,999)), 10, 5);

        echo " <form method='post' action='\".$thisfile.\"?lan=\".$lan.\">
            <div class='fraent'>
                <div class='ent1'> </div>
                <div class='ent2'> \"$_SESSION['security_code'].\" </div>

                <div class='ent1'><label for='sec_code'>".$_locale['e110'].\"</label></div>
                <div class='ent2'> <input type='password' id='sec_code' class='textbox'
name='sec_code' accesskey='s' tabindex='3' value='\"></div>

                <div class='ent1'><label for='url'>".$_locale['e111'].\"</label></div>
                <div class='ent2'><input type='text' id='url' class='textbox' name='url'
tabindex='4' accesskey='u' value='http://\"> </div>

                <div class='ent1'><label for='w3c_h'>".$_locale['e112'].\"</label></div>
                <div class='ent2'><input type='checkbox' id='w3c_h' class='textbox'
name='w3c_h' accesskey='h' tabindex='5' checked> </div>

                <div class='ent1'><label for='w3c_c'>".$_locale['e113'].\"</label></div>
                <div class='ent2'><input type='checkbox' id='w3c_c' class='textbox'
name='w3c_c' accesskey='c' tabindex='6' checked> </div>

                <div class='ent1'> </div>
                <div class='ent2'><input type='submit' class='textbox' name='ok' accesskey='o'
tabindex='7' value='\".$_locale['e114'].\"></div>
                \".$_locale['e115'].\"
            </div>
        </form></center>";

        if ($info_row['last_ip'] <> $_SERVER["REMOTE_ADDR"]) $db->query("UPDATE info
SET last_ip = \"$_SERVER['REMOTE_ADDR'].\", visited = \".( $info_row['visited'] + 1).\"");
        echo "<hr><p align='center'>".$_locale['e116'].\"
        \".$info_row['visited'].\"</p></body></html>";
    }
    elseif (isset($_POST['ok']))
    {
        $sec_code = isset($_POST['sec_code']) ? $_POST['sec_code'] : "";
        $who = $sec_code == "f".$_SESSION['security_code']."l" ? 'admin' : 'user';

        if ($who == 'user')
        {
            if ($sec_code <> $_SESSION['security_code'])
            {

```



```

        echo $locale['e117'];
        exit;
    }
}

$url = @$_POST['url'];
$w3c_h = @$_POST['w3c_h'] ? true : false;
$w3c_c = @$_POST['w3c_c'] ? true : false;

$site = new webpage ($url, $lan, $w3c_h, $w3c_c);
$site = array
(
    'url'      => $site->url,
    'dom'     => $site->domain,
    'path'    => $site->path,
    'cont'    => $site->arr_content(),
    'acce'    => $site->arr_accessibility(),
    'w3c'     => $site->arr_w3(),
    'acc1'    => $site->arr_tidy_acc(1),
    'acc2'    => $site->arr_tidy_acc(2),
    'acc3'    => $site->arr_tidy_acc(3),
    'scr'     => $site->arr_script(),
    'apl'     => $site->arr_applet(),
    'mtg'     => $site->arr_metatag(),
    'coo'     => $site->arr_cookie(),
    'com'     => $site->arr_comment(),
    'tit'     => $site->arr_title(),
    'tid'     => $site->arr_tidy_gen(),
    'bod'     => $site->arr_body(),
    'for'     => $site->arr_form(),
    'anc'     => $site->arr_anchor(),
    'map'     => $site->arr_map(),
    'fon'     => $site->arr_font(),
    'mar'     => $site->arr_marquee(),
    'img'     => $site->arr_img(),
    'tab'     => $site->arr_table(),
    'col1'    => $site->arr_color(1), // External and Internal
    'col2'    => $site->arr_color(2), // Inline
    'col3'    => $site->arr_color(3), // HTML Colors
    'fs1'     => $site->arr_font_size(1), // External and Internal
    'fs2'     => $site->arr_font_size(2), // Inline
    'size'    => $site->arr_page_size()
);

//Write Database
$db->query("SELECT MAX(id) as last_id FROM webpage");
$my_row = $db->get_row();
$id = $my_row['last_id'] + 1;

preg_match("/^http:\/\/(?:www|[a-z0-9_-\.\.])\.(?:[a-z0-9_-\.\.])\.(com|gen|org|biz|info|av|dr|pol|bel|mil|bbs|k12|edu|name|net|gov|web|tel|tv|aero|coop|int|jobs|muse
um|pro|travel)(?:\.)?([a-z]{2})?/iX", $site['dom'], $domp);

$db->query("INSERT INTO webpage(id, site, sub1, sub2, type, country, path, time, who)
VALUES('".$id."', '".$site['url']."', '".$domp[1]."', '".$domp[2]."', '".$domp[3]."', '".$domp[4]."',
'".$site['path']."', '".$time()."', '".$who."'");

$db->query("UPDATE webpage SET
tid_warnings = '".$site['tid']['warnings']."'");

```

```

tid_errors = ".$site[tid]['errors'].",
tid_errorbuffer = ".(isset($site[tid]['errorbuffer']) ?
'''.mysql_real_escape_string($site[tid]['errorbuffer']).'''' : 'NULL').",
tid_warningbuffer = ".(isset($site[tid]['warningbuffer']) ?
'''.mysql_real_escape_string($site[tid]['warningbuffer']).'''' : 'NULL').",

acc1_warnings = ".$site[acc1]['warnings'].",
acc2_warnings = ".$site[acc2]['warnings'].",
acc3_warnings = ".$site[acc3]['warnings'].",
acc1_buffer = ".(isset($site[acc1]['buffer']) ? '''.mysql_real_escape_string($site[acc1]['buffer']).'''' :
'NULL').",

acc2_buffer = ".(isset($site[acc2]['buffer']) ? '''.mysql_real_escape_string($site[acc2]['buffer']).'''' :
'NULL').",
acc3_buffer = ".(isset($site[acc3]['buffer']) ? '''.mysql_real_escape_string($site[acc3]['buffer']).'''' :
'NULL').",

coo_value = ".$site[coo]['value'].",

w3c_htmlerrors = ".(isset($site[w3c]['htmlerrors']) ? $site[w3c]['htmlerrors'] : 'NULL').",
w3c_csserrors = ".(isset($site[w3c]['csserrors']) ? $site[w3c]['csserrors'] : 'NULL').",
w3c_doctypevalue = ".(isset($site[w3c]['doctypevalue']) ? $site[w3c]['doctypevalue'] : 'NULL').",

com_count = ".$site[com]['count'].",
com_comments = ".(isset($site[com]['comments']) ?
'''.mysql_real_escape_string($site[com]['comments']).'''' : 'NULL').",

tit_length = ".$site[tit]['length'].",
tit_value = ".($site[tit]['length'] > 0 ? 1 : 0).",
tit_title = ".(isset($site[tit]['title']) ? '''.mysql_real_escape_string($site[tit]['title']).'''' : 'NULL').",

scr_scriptcount = ".$site[scr]['scriptcount'].",
scr_noscriptcount = ".$site[scr]['noscriptcount'].",

apl_count = ".$site[apl]['count'].",

mtg_count = ".$site[mtg]['count'].",
mtg_metatags = ".(isset($site[mtg]['metatags']) ?
'''.mysql_real_escape_string($site[mtg]['metatags']).'''' : 'NULL').",

anc_count = ".$site[anc]['count'].",
anc_withoutimgper = ".(isset($site[anc]['withoutimgper']) ? $site[anc]['withoutimgper'] :
'NULL').",
anc_imgaltper = ".(isset($site[anc]['imgaltper']) ? $site[anc]['imgaltper'] : 'NULL').",
anc_tabidxper = ".(isset($site[anc]['tabidxper']) ? $site[anc]['tabidxper'] : 'NULL').",
anc_acckeyper = ".(isset($site[anc]['acckeyper']) ? $site[anc]['acckeyper'] : 'NULL').",
anc_mainmenuvalue = ".$site[anc]['mainmenuvalue'].",
anc_htmlmenuvalue = ".$site[anc]['htmlmenuvalue'].",

bod_value = ".$site[bod]['value'].",
bod_textvalue = ".$site[bod]['textvalue'].",
bod_linkvalue = ".$site[bod]['linkvalue'].",
bod_alinkvalue = ".$site[bod]['alinkvalue'].",
bod_vlinkvalue = ".$site[bod]['vlinkvalue'].",

for_count = ".$site[for]['count'].",
for_objectcount = ".$site[for]['objectcount'].",
for_embedcount = ".$site[for]['embedcount'].",
for_value = ".(isset($site[for]['value']) ? $site[for]['value'] : 'NULL').",

```

```

for_labelper = ".(isset($site['for']['labelper']) ? $site['for']['labelper'] : 'NULL').",
for_tabidxper = ".(isset($site['for']['tabidxper']) ? $site['for']['tabidxper'] : 'NULL').",
for_inpacckeyper = ".(isset($site['for']['inpacckeyper']) ? $site['for']['inpacckeyper'] : 'NULL').",
for_selacckeyper = ".(isset($site['for']['selacckeyper']) ? $site['for']['selacckeyper'] : 'NULL').",
for_tareaacckeyper = ".(isset($site['for']['tareaacckeyper']) ? $site['for']['tareaacckeyper'] : 'NULL').",
for_butacckeyper = ".(isset($site['for']['butacckeyper']) ? $site['for']['butacckeyper'] : 'NULL').",

fon_value = ".$site['fon']['value'].",

mar_value = ".$site['mar']['value'].",

map_mapcount = ".$site['map']['mapcount'].",
map_areacount = ".$site['map']['areacount'].",
map_areaaltper = ".(isset($site['map']['areaaltper']) ? $site['map']['areaaltper'] : 'NULL').",
map_areatabidxper = ".(isset($site['map']['areatabidxper']) ? $site['map']['areatabidxper'] : 'NULL').",
map_areaacckeyper = ".(isset($site['map']['areaacckeyper']) ? $site['map']['areaacckeyper'] : 'NULL').",

img_count = ".$site['img']['count'].",
img_downloaded = ".$site['img']['downloaded'].",
img_exttypeper = ".(isset($site['img']['exttypeper']) ? $site['img']['exttypeper'] : 'NULL').",
img_filesizeper = ".(isset($site['img']['filesizeper']) ? $site['img']['filesizeper'] : 'NULL').",
img_widthper = ".(isset($site['img']['widthper']) ? $site['img']['widthper'] : 'NULL').",
img_heightper = ".(isset($site['img']['heightper']) ? $site['img']['heightper'] : 'NULL').",
img_altper = ".(isset($site['img']['altper']) ? $site['img']['altper'] : 'NULL').",
img_altlengthper = ".(isset($site['img']['altlengthper']) ? $site['img']['altlengthper'] : 'NULL').",
img_altendcharper = ".(isset($site['img']['altendcharper']) ? $site['img']['altendcharper'] : 'NULL').",
img_altendcharsper = ".(isset($site['img']['altendcharsper']) ? $site['img']['altendcharsper'] : 'NULL').",
img_logovalue = ".$site['img']['logovalue'].",
img_giflogovalue = ".(isset($site['img']['giflogovalue']) ? $site['img']['giflogovalue'] : 'NULL').",
img_textreadlogovalue = ".(isset($site['img']['textreadlogovalue']) ? $site['img']['textreadlogovalue'] : 'NULL').",
img_statlogovalue = ".(isset($site['img']['statlogovalue']) ? $site['img']['statlogovalue'] : 'NULL').",

tab_count = ".$site['tab']['count'].",
tab_endtagper = ".(isset($site['tab']['endtagper']) ? $site['tab']['endtagper'] : 'NULL').",
tab_bgcolorper = ".(isset($site['tab']['bgcolorper']) ? $site['tab']['bgcolorper'] : 'NULL').",
tab_bgroundper = ".(isset($site['tab']['bgroundper']) ? $site['tab']['bgroundper'] : 'NULL').",

col1_per = ".(isset($site['col1']['per']) ? $site['col1']['per'] : 'NULL').",
col2_per = ".(isset($site['col2']['per']) ? $site['col2']['per'] : 'NULL').",
col3_per = ".(isset($site['col3']['per']) ? $site['col3']['per'] : 'NULL').",

fs1_per = ".(isset($site['fs1']['per']) ? $site['fs1']['per'] : 'NULL').",
fs2_per = ".(isset($site['fs2']['per']) ? $site['fs2']['per'] : 'NULL').",

size_size = ".$site['size']['size'].",
size_sizewithimgs = ".$site['size']['sizewithimgs'].",

cont_logovalue = ".$site['cont']['logovalue'].",
cont_instnamevalue = ".$site['cont']['instnamevalue'].",
cont_aimvalue = ".$site['cont']['aimvalue'].",
cont_ancvalue = ".$site['cont']['ancvalue'].",
cont_annovalue = ".$site['cont']['annovalue'].",
cont_searchvalue = ".$site['cont']['searchvalue'].",
cont_helpvalue = ".$site['cont']['helpvalue'].",
cont_faqlvalue = ".$site['cont']['faqlvalue'].",
cont_feedvalue = ".$site['cont']['feedvalue'].",
cont_addressvalue = ".$site['cont']['addressvalue'].",

```

```

cont_phonevalue = ".$site['cont']['phonevalue'].",
cont_mailvalue = ".$site['cont']['mailvalue'].",
cont_appropmailvalue = ".(isset($site['cont']['appropmailvalue']) ? $site['cont']['appropmailvalue'] :
'NULL').",

cont_logos = ".(isset($site['cont']['logos']) ? ""mysql_real_escape_string($site['cont']['logos'])."" :
'NULL').",
cont_instname = ".(isset($site['cont']['instname']) ?
""mysql_real_escape_string($site['cont']['instname'])."" : 'NULL').",
cont_aims = ".(isset($site['cont']['aims']) ? ""mysql_real_escape_string($site['cont']['aims'])."" :
'NULL').",
cont_hrefs = ".(isset($site['cont']['hrefs']) ? ""mysql_real_escape_string($site['cont']['hrefs'])."" :
'NULL').",
cont_annos = ".(isset($site['cont']['annos']) ? ""mysql_real_escape_string($site['cont']['annos'])."" :
'NULL').",
cont_searchs = ".(isset($site['cont']['searchs']) ?
""mysql_real_escape_string($site['cont']['searchs'])."" : 'NULL').",
cont_helps = ".(isset($site['cont']['helps']) ? ""mysql_real_escape_string($site['cont']['helps'])."" :
'NULL').",
cont_faqs = ".(isset($site['cont']['faqs']) ? ""mysql_real_escape_string($site['cont']['faqs'])."" :
'NULL').",
cont_feeds = ".(isset($site['cont']['feeds']) ? ""mysql_real_escape_string($site['cont']['feeds'])."" :
'NULL').",
cont_addresses = ".(isset($site['cont']['addresses']) ?
""mysql_real_escape_string($site['cont']['addresses'])."" : 'NULL').",
cont_phones = ".(isset($site['cont']['phones']) ? ""mysql_real_escape_string($site['cont']['phones']).""
: 'NULL').",
cont_mails = ".(isset($site['cont']['mails']) ? ""mysql_real_escape_string($site['cont']['mails'])."" :
'NULL').",
cont_appropmails = ".(isset($site['cont']['appropmails']) ?
""mysql_real_escape_string($site['cont']['appropmails'])."" : 'NULL').",
acce_lyvalue = ".$site['acce']['lyvalue'].
        WHERE id = ".$Sid.""");

    $_SESSION['web_site_id'] = $id;
    $_SESSION['language'] = $lan;
    if ($who == 'admin') $_SESSION['admin'] = true;

    $db->query("UPDATE info SET total = ".$info_row['total'] + 1."");

    header("location: ".$thisfile."");
}
} else
{
    $site_id = $_SESSION['web_site_id'];
    $sev = @$_GET['ev'];
    $arr_ev = array ('lo', 'co', 'ac', 'te', 'ge', 'data');
    if (!isset($sev) || !in_array($sev, $arr_ev)) $sev = 'co';

    if ($sev == 'lo')
    {
        unset($_SESSION['admin']);
        unset($_SESSION['web_site_id']);
        unset($_SESSION['language']);
        header("location: ".$thisfile."?lan=$lan");
    }

    /**/
    $filt_type = isset($_POST['filt_type']) ? $_POST['filt_type'] : @$_GET['filt_type'];

```

```

$filt_country = isset($_POST['filt_country']) ? $_POST['filt_country'] : @$_GET['filt_country'];

$filt_type = $filt_type <> 'all' ? $filt_type : NULL;
$filt_country = $filt_country <> 'all' ? $filt_country : NULL;

$filt_link = isset($filt_type) ? "&filt_type=$filt_type" : "";
$filt_link = isset($filt_country) ? $filt_link."&filt_country=$filt_country" : $filt_link;

$filt_sql = isset($filt_type) && $filt_type <> 'all' ? " AND type = '". $filt_type. "' : ";
$filt_sql = isset($filt_country) && $filt_country <> 'all' ? $filt_sql." AND country =
'". $filt_country. "' : $filt_sql;

/**/

if (isset($_SESSION['admin']))
{
    $action = @$_GET['action'];
    if (!isset($action)) $action = "";

    if ($action == 'delete')
    {
        $val = @$_GET['val'];
        if ($val == 'this')
        {
            $db->query("DELETE FROM webpage WHERE id = $site_id");
        } elseif ($val == 'old')
        {
            $db->query("DELETE FROM webpage WHERE id NOT IN (SELECT id
FROM (SELECT max(id) as id FROM webpage GROUP BY site) AS wpage)");
        } elseif ($val == 'temp')
        {
            rm_dir_recursively("www");
        }
        header("location: ".$thisfile."?ev=$ev");
    } elseif ($action == 'edit')
    {
        switch ($ev)
        {
            case 'co':

                $edit_sql = "
cont_logovalue = " . ($_POST['cont_logovalue'] <> " ? $_POST['cont_logovalue'] : 'NULL').",
cont_instnamevalue = " . ($_POST['cont_instnamevalue'] <> " ? $_POST['cont_instnamevalue'] :
'NULL').",
cont_aimvalue = " . ($_POST['cont_aimvalue'] <> " ? $_POST['cont_aimvalue'] : 'NULL').",
cont_ancvalue = " . ($_POST['cont_ancvalue'] <> " ? $_POST['cont_ancvalue'] : 'NULL').",
cont_annovalue = " . ($_POST['cont_annovalue'] <> " ? $_POST['cont_annovalue'] : 'NULL').",
cont_searchvalue = " . ($_POST['cont_searchvalue'] <> " ? $_POST['cont_searchvalue'] : 'NULL').",
cont_helpvalue = " . ($_POST['cont_helpvalue'] <> " ? $_POST['cont_helpvalue'] : 'NULL').",
cont_faqlvalue = " . ($_POST['cont_faqlvalue'] <> " ? $_POST['cont_faqlvalue'] : 'NULL').",
cont_feedvalue = " . ($_POST['cont_feedvalue'] <> " ? $_POST['cont_feedvalue'] : 'NULL').",
cont_addressvalue = " . ($_POST['cont_addressvalue'] <> " ? $_POST['cont_addressvalue'] :
'NULL').",
cont_phonevalue = " . ($_POST['cont_phonevalue'] <> " ? $_POST['cont_phonevalue'] : 'NULL').",
cont_mailvalue = " . ($_POST['cont_mailvalue'] <> " ? $_POST['cont_mailvalue'] : 'NULL').",
cont_appropmailvalue = " . ($_POST['cont_mailvalue'] == 1 && $_POST['cont_appropmailvalue'] <>
" ? $_POST['cont_appropmailvalue'] : 'NULL')."";

                break;

```

```

        case 'ac':

            $edit_sql = "
acce_ievalue = " . ($_POST['acce_ievalue'] <> " ? $_POST['acce_ievalue'] : 'NULL').",
acce_ffvalue = " . ($_POST['acce_ffvalue'] <> " ? $_POST['acce_ffvalue'] : 'NULL').",
acce_konvalue = " . ($_POST['acce_konvalue'] <> " ? $_POST['acce_konvalue'] : 'NULL').",
acce_lyvalue = " . ($_POST['acce_lyvalue'] <> " ? $_POST['acce_lyvalue'] : 'NULL').",
acce_res1value = " . ($_POST['acce_res1value'] <> " ? $_POST['acce_res1value'] : 'NULL').",
acce_res2value = " . ($_POST['acce_res2value'] <> " ? $_POST['acce_res2value'] : 'NULL').",
acce_res3value = " . ($_POST['acce_res3value'] <> " ? $_POST['acce_res3value'] : 'NULL')."";
            break;

        case 'te':

            $edit_sql = "
scr_appropvalue = " . ($_POST['scr_appropvalue'] <> " ? $_POST['scr_appropvalue'] : 'NULL').",
com_appropvalue = " . ($_POST['com_appropvalue'] <> " ? $_POST['com_appropvalue'] : 'NULL').",
tit_appropvalue = " . ($_POST['tit_appropvalue'] <> " ? $_POST['tit_appropvalue'] : 'NULL').",
for_value = " . ($_POST['for_value'] <> " ? $_POST['for_value'] : 'NULL')."";
            break;

        case 'ge':

            $edit_sql = "
img_logovalue = " . ($_POST['img_logovalue'] <> " ? $_POST['img_logovalue'] : 'NULL').",
img_textreadlogovalue = " . ($_POST['img_logovalue'] == 1 && $_POST['img_textreadlogovalue'] <>
" ? $_POST['img_textreadlogovalue'] : 'NULL').",
img_giflogovalue = " . ($_POST['img_logovalue'] == 1 && $_POST['img_giflogovalue'] <> " ?
$_POST['img_giflogovalue'] : 'NULL').",
img_statlogovalue = " . ($_POST['img_logovalue'] == 1 && $_POST['img_statlogovalue'] <> " ?
$_POST['img_statlogovalue'] : 'NULL').",
anc_htmlmenuvalue = " . ($_POST['anc_htmlmenuvalue'] <> " ? $_POST['anc_htmlmenuvalue'] :
'NULL').",
anc_mainmenuvalue = " . ($_POST['anc_mainmenuvalue'] <> " ? $_POST['anc_mainmenuvalue'] :
'NULL').",
map_appropvalue = " . ($_POST['map_appropvalue'] <> " ? $_POST['map_appropvalue'] : 'NULL')."";
            break;
    }
    $db->query("UPDATE webpage SET ".$edit_sql.", who = 'admin' WHERE id =
$site_id");
    header("location: ".$thisfile."?ev=$ev");
} else
{
    $filt_sub1 = isset($_POST['filt_sub1']) ? $_POST['filt_sub1'] : @$_GET['filt_sub1'];
    $filt_sub2 = isset($_POST['filt_sub2']) ? $_POST['filt_sub2'] : @$_GET['filt_sub2'];
    $filt_path = isset($_POST['filt_path']) ? $_POST['filt_path'] : @$_GET['filt_path'];
    $filt_who = isset($_POST['filt_who']) ? $_POST['filt_who'] : @$_GET['filt_who'];

    $filt_sub1 = $filt_sub1 <> " ? $filt_sub1 : NULL;
    $filt_sub2 = $filt_sub2 <> " ? $filt_sub2 : NULL;
    $filt_path = $filt_path <> " ? $filt_path : NULL;
    $filt_who = $filt_who <> 'all' ? $filt_who : NULL;

    $filt_link = isset($filt_sub1) ? $filt_link."&filt_sub1=$filt_sub1" : $filt_link;
    $filt_link = isset($filt_sub2) ? $filt_link."&filt_sub2=$filt_sub2" : $filt_link;
    $filt_link = isset($filt_path) ? $filt_link."&filt_path=$filt_path" : $filt_link;
    $filt_link = isset($filt_who) ? $filt_link."&filt_who=$filt_who" : $filt_link;

    $filt_sql = isset($filt_sub1) ? $filt_sql." AND sub1 = '$filt_sub1.'" : $filt_sql;

```

```

    $filt_sql = isset($filt_sub2) ? $filt_sql." AND sub2 = '". $filt_sub2.'"' : $filt_sql;
    $filt_sql = isset($filt_path) ? $filt_sql." AND path = '". $filt_path.'"' : $filt_sql;
    $filt_sql = isset($filt_who) && $filt_who <> 'all' ? $filt_sql." AND who =
    '". $filt_who.'"' : $filt_sql;

    $frm_adm_sub1 = "<input type='text' name='filt_sub1' value='". $filt_sub1.'"
class='textbox' size='10'>\n";
    $frm_adm_sub2 = "<input type='text' name='filt_sub2' value='". $filt_sub2.'"
class='textbox' size='10'>\n";
    $frm_adm_path = "<input type='text' name='filt_path' value='". $filt_path.'"
class='textbox' size='10'>\n";

    $frm_adm_who = "<select name='filt_who' class='textbox'>";
    $frm_adm_who .= "<option value='all'>all</option>\n";
    $frm_adm_who .= "<option value='admin'".(isset($filt_who) && $filt_who ==
'admin' ? " selected" : "").">admin</option>\n";
    $frm_adm_who .= "<option value='user'".(isset($filt_who) && $filt_who == 'user' ? "
selected" : "").">user</option>\n";
    $frm_adm_who .= "</select>\n";

    $frm_adm_clean = "<a href='". $thisfile.'"?ev=$ev&action=delete&val=this'
onClick='return DeleteMessage();>'. $locale['adm100']. "</a> ";
    $frm_adm_clean .= "<a href='". $thisfile.'"?ev=$ev&action=delete&val=old'
onClick='return DeleteMessage();>'. $locale['adm101']. "</a> ";
    $frm_adm_clean .= "<a href='". $thisfile.'"?ev=$ev&action=delete&val=temp'
onClick='return DeleteMessage();>'. $locale['adm102']. "</a> ";

    echo "
<script type='text/javascript'>
    function DeleteMessage()
    {
        return confirm(\"'". $locale['adm103']. "'\");
    }
</script>\n";

    $records = "<a ".($ev == 'data' ? "class='act'":
'').">". $thisfile.'"?ev=data$filt_link'>". $locale['adm104']. "</a> | ";

    $edit_frm_s = "<form method='post' action='". $thisfile.'"?ev=$ev&action=edit'>";
    $edit_frm_e = "<div class='post'><input type='submit' class='textbox' value='OK'
name='ok'></div></form>";
    }
} else
{
    $frm_adm_sub1 = "";
    $frm_adm_sub2 = "";
    $frm_adm_path = "";
    $frm_adm_who = "";
    $frm_adm_clean = "";

    $records = "";
    $edit_frm_s = "";
    $edit_frm_e = "";
}

$db->query("SELECT * FROM webpage where id=$site_id");
$data = $db->get_row();

```

```

$ldb->query("SELECT count(*),AVG(tid_warnings), AVG(tid_errors), AVG(acc1_warnings),
AVG(acc2_warnings), AVG(acc3_warnings), AVG(coo_value), AVG(w3c_htmlerrors),
AVG(w3c_csserrors), AVG(w3c_doctypevalue), AVG(com_count), AVG(com_appropvalue),
AVG(tit_length), AVG(tit_value), AVG(tit_appropvalue), AVG(scr_appropvalue),
AVG(scr_scriptcount), AVG(scr_noscriptcount), AVG(apl_count), AVG(mtg_count),
AVG(anc_count), AVG(anc_withoutimgper), AVG(anc_imgaltper),AVG(anc_tabidxper),
AVG(anc_acckeyper), AVG(anc_mainmenuvalue), AVG(anc_htmlmenuvalue), AVG(bod_value),
AVG(bod_textvalue), AVG(bod_linkvalue), AVG(bod_alinkvalue), AVG(bod_vlinkvalue),
AVG(for_count), AVG(for_objectcount), AVG(for_embedcount), AVG(for_value),
AVG(for_labelper), AVG(for_tabidxper), AVG(for_inpacckeyper), AVG(for_selacckeyper),
AVG(for_tareaacckeyper), AVG(for_butacckeyper), AVG(fon_value), AVG(mar_value),
AVG(map_mapcount), AVG(map_areacount), AVG(map_areaaltper), AVG(map_areatabidxper),
AVG(map_areaacckeyper), AVG(map_appropvalue), AVG(img_count), AVG(img_downloaded),
AVG(img_exttypeper), AVG(img_filesizeper), AVG(img_widthper),AVG(img_heightper),
AVG(img_altper), AVG(img_altlengthper), AVG(img_altendcharper), AVG(img_altendcharsper),
AVG(img_logovalue), AVG(img_giflogovalue), AVG(img_textreadlogovalue),
AVG(img_statlogovalue), AVG(tab_count), AVG(tab_endtagper), AVG(tab_bgcolorper),
AVG(tab_bgroundper), AVG(col1_per), AVG(col2_per), AVG(col3_per), AVG(fs1_per),
AVG(fs2_per), AVG(size_size), AVG(size_sizewithimgs), AVG(cont_logovalue),
AVG(cont_instnamevalue), AVG(cont_aimvalue), AVG(cont_ancvalue), AVG(cont_annovalue),
AVG(cont_searchvalue), AVG(cont_helpvalue), AVG(cont_faqlvalue), AVG(cont_feedvalue),
AVG(cont_addressvalue), AVG(cont_phonevalue), AVG(cont_mailvalue),
AVG(cont_appropmailvalue), AVG(acce_ievalue), AVG(acce_ffvalue), AVG(acce_konvalue),
AVG(acce_lyvalue), AVG(acce_res1value), AVG(acce_res2value), AVG(acce_res3value)
FROM webpage WHERE 1 ".$filt_sql." AND id IN (SELECT max(id) FROM
webpage GROUP BY site)");

```

```
$avg_data = $ldb->get_row();
```

```

$ldb->query("SELECT count(tid_warnings), count(tid_errors), count(acc1_warnings),
count(acc2_warnings), count(acc3_warnings), count(coo_value), count(w3c_htmlerrors),
count(w3c_csserrors), count(w3c_doctypevalue), count(com_count), count(com_appropvalue),
count(tit_length), count(tit_value), count(tit_appropvalue), count(scr_appropvalue),
count(scr_scriptcount), count(scr_noscriptcount), count(apl_count), count(mtg_count),
count(anc_count), count(anc_withoutimgper), count(anc_imgaltper),count(anc_tabidxper),
count(anc_acckeyper), count(anc_mainmenuvalue), count(anc_htmlmenuvalue), count(bod_value),
count(bod_textvalue), count(bod_linkvalue), count(bod_alinkvalue), count(bod_vlinkvalue),
count(for_count), count(for_objectcount), count(for_embedcount), count(for_value),
count(for_labelper), count(for_tabidxper), count(for_inpacckeyper), count(for_selacckeyper),
count(for_tareaacckeyper), count(for_butacckeyper), count(fon_value), count(mar_value),
count(map_mapcount), count(map_areacount), count(map_areaaltper), count(map_areatabidxper),
count(map_areaacckeyper), count(map_appropvalue), count(img_count), count(img_downloaded),
count(img_exttypeper), count(img_filesizeper), count(img_widthper),count(img_heightper),
count(img_altper), count(img_altlengthper), count(img_altendcharper), count(img_altendcharsper),
count(img_logovalue), count(img_giflogovalue), count(img_textreadlogovalue),
count(img_statlogovalue), count(tab_count), count(tab_endtagper), count(tab_bgcolorper),
count(tab_bgroundper), count(col1_per), count(col2_per), count(col3_per), count(fs1_per),
count(fs2_per), count(size_size), count(size_sizewithimgs), count(cont_logovalue),
count(cont_instnamevalue), count(cont_aimvalue), count(cont_ancvalue), count(cont_annovalue),
count(cont_searchvalue), count(cont_helpvalue), count(cont_faqlvalue), count(cont_feedvalue),
count(cont_addressvalue),count(cont_phonevalue), count(cont_mailvalue),
count(cont_appropmailvalue), count(acce_ievalue), count(acce_ffvalue), count(acce_konvalue),
count(acce_lyvalue), count(acce_res1value), count(acce_res2value), count(acce_res3value)
FROM webpage WHERE 1 ".$filt_sql." AND id IN (SELECT max(id) FROM
webpage GROUP BY site)");

```

```
$count_data = $ldb->get_row();
```

```
$co_array = array
```

```
(
```

```
0 => array
```



```

(
  //info, subj, class, test, value, detail, avg, n
  0 => array ("", 'c100', "", "", "", "", ""),
  1 => array ('ic101', 'c101', 'sub1', test_poss($data['cont_logovalue']),
edit_frm_input('cont_logovalue'), (isset($data['cont_logos']) ? 'cont_logos' : NULL),
round_cvt_per($avg_data['AVG(cont_logovalue)'], $count_data['count(cont_logovalue)']),

  2 => array ('ic102', 'c102', 'sub1', test_poss($data['cont_instnamevalue']),
edit_frm_input('cont_instnamevalue'), (isset($data['cont_instname']) ? 'cont_instname' : NULL),
round_cvt_per($avg_data['AVG(cont_instnamevalue)'], $count_data['count(cont_instnamevalue)']),

  3 => array ('ic103', 'c103', 'sub1', test_poss($data['cont_aimvalue']),
edit_frm_input('cont_aimvalue'), (isset($data['cont_aims']) ? 'cont_aims' : NULL),
round_cvt_per($avg_data['AVG(cont_aimvalue)'], $count_data['count(cont_aimvalue)']),

  4 => array ('ic104', 'c104', 'sub1', test_poss($data['cont_ancvalue']),
edit_frm_input('cont_ancvalue'), (isset($data['cont_hrefs']) ? 'cont_hrefs' : NULL),
round_cvt_per($avg_data['AVG(cont_ancvalue)'], $count_data['count(cont_ancvalue)']),

  5 => array ('ic105', 'c105', 'sub1', test_poss($data['cont_annovalue']),
edit_frm_input('cont_annovalue'), (isset($data['cont_annos']) ? 'cont_annos' : NULL),
round_cvt_per($avg_data['AVG(cont_annovalue)'], $count_data['count(cont_annovalue)']),

  6 => array ('ic106', 'c106', 'sub1', test_poss($data['cont_searchvalue']),
edit_frm_input('cont_searchvalue'), (isset($data['cont_searchs']) ? 'cont_searchs' : NULL),
round_cvt_per($avg_data['AVG(cont_searchvalue)'], $count_data['count(cont_searchvalue)']),

  7 => array ('ic107', 'c107', 'sub1', test_poss($data['cont_helpvalue']),
edit_frm_input('cont_helpvalue'), (isset($data['cont_helps']) ? 'cont_helps' : NULL),
round_cvt_per($avg_data['AVG(cont_helpvalue)'], $count_data['count(cont_helpvalue)']),

  8 => array ('ic108', 'c108', 'sub1', test_poss($data['cont_faqvalue']),
edit_frm_input('cont_faqvalue'), (isset($data['cont_faqs']) ? 'cont_faqs' : NULL),
round_cvt_per($avg_data['AVG(cont_faqvalue)'], $count_data['count(cont_faqvalue)']),

  9 => array ('ic109', 'c109', 'sub1', test_poss($data['cont_feedvalue']),
edit_frm_input('cont_feedvalue'), (isset($data['cont_feeds']) ? 'cont_feeds' : NULL),
round_cvt_per($avg_data['AVG(cont_feedvalue)'], $count_data['count(cont_feedvalue)'])
),
1 => array
(
  0 => array ("", 'c200', 'sub1', "", "", "", ""),
  1 => array ('ic201', 'c201', 'sub1', test_poss($data['cont_addressvalue']),
edit_frm_input('cont_addressvalue'), (isset($data['cont_addresses']) ? 'cont_addresses' : NULL),
round_cvt_per($avg_data['AVG(cont_addressvalue)'], $count_data['count(cont_addressvalue)']),

  2 => array ('ic202', 'c202', 'sub1', test_poss($data['cont_phonevalue']),
edit_frm_input('cont_phonevalue'), (isset($data['cont_phones']) ? 'cont_phones' : NULL),
round_cvt_per($avg_data['AVG(cont_phonevalue)'], $count_data['count(cont_phonevalue)']),

  3 => array ('ic203', 'c203', 'sub1', test_poss($data['cont_mailvalue']),
edit_frm_input('cont_mailvalue'), (isset($data['cont_mails']) ? 'cont_mails' : NULL),
round_cvt_per($avg_data['AVG(cont_mailvalue)'], $count_data['count(cont_mailvalue)']),

  4 => array ('ic204', 'c204', 'sub1', test_poss_rel($data['cont_appropmailvalue'],
$data['cont_mailvalue']), edit_frm_input('cont_appropmailvalue'), (isset($data['cont_appropmails']) ?
'cont_appropmails' : NULL), round_cvt_per($avg_data['AVG(cont_appropmailvalue)'],
$count_data['count(cont_appropmailvalue)'])
)
)

```

```

    )
);

$ac_array = array
(
    0 => array
    (
        0 => array ("", 'a100', "", "", "", "", ""),
        1 => array ('ia101', 'a101', 'sub1', test_other($data['acce_ievalue']),
edit_frm_input('acce_ievalue'), NULL, round_cvt_per($avg_data['AVG(acce_ievalue)']),
$count_data['count(acce_ievalue)']),

        2 => array ('ia102', 'a102', 'sub1', test_other($data['acce_ffvalue']),
edit_frm_input('acce_ffvalue'), NULL, round_cvt_per($avg_data['AVG(acce_ffvalue)']),
$count_data['count(acce_ffvalue)']),

        3 => array ('ia103', 'a103', 'sub1', test_other($data['acce_konvalue']),
edit_frm_input('acce_konvalue'), NULL, round_cvt_per($avg_data['AVG(acce_konvalue)']),
$count_data['count(acce_konvalue)']),

        4 => array ('ia104', 'a104', 'sub1', test_poss($data['acce_lyvalue']),
edit_frm_input('acce_lyvalue'), NULL, round_cvt_per($avg_data['AVG(acce_lyvalue)']),
$count_data['count(acce_lyvalue)']),
    ),

    1 => array
    (
        0 => array ("", 'a200', "", "", "", "", ""),
        1 => array ('ia201', 'a201', 'sub1', test_other($data['acce_res1value']),
edit_frm_input('acce_res1value'), "<a href='\"javascript:popUp(\".$data['site'].\"\", 1280,
1024)\">\".$locale['r130'].\"</a>\"", round_cvt_per($avg_data['AVG(acce_res1value)']),
$count_data['count(acce_res1value)']),

        2 => array ('ia202', 'a202', 'sub1', test_other($data['acce_res2value']),
edit_frm_input('acce_res2value'), "<a href='\"javascript:popUp(\".$data['site'].\"\", 1024,
768)\">\".$locale['r130'].\"</a>\"", round_cvt_per($avg_data['AVG(acce_res2value)']),
$count_data['count(acce_res2value)']),

        3 => array ('ia203', 'a203', 'sub1', test_other($data['acce_res3value']),
edit_frm_input('acce_res3value'), "<a href='\"javascript:popUp(\".$data['site'].\"\", 800,
600)\">\".$locale['r130'].\"</a>\"", round_cvt_per($avg_data['AVG(acce_res3value)']),
$count_data['count(acce_res3value)']),
    )
);

$te_array = array
(
    0 => array
    (
        0 => array ("", 't100', "", "", "", "", ""),
        1 => array ('it101', 't101', 'sub1', test_eq($data['w3c_htmlerrors'], 0),
$data['w3c_htmlerrors'], NULL, round_number($avg_data['AVG(w3c_htmlerrors)']),
$count_data['count(w3c_htmlerrors)']),

        2 => array ('it102', 't102', 'sub1', test_eq($data['w3c_csserrors'], 0),
$data['w3c_csserrors'], NULL, round_number($avg_data['AVG(w3c_csserrors)']),
$count_data['count(w3c_csserrors)']),

        3 => array ('it103', 't103', 'sub1head', "", "", NULL, "", ""),

```

```

4 => array ("", 't104', 'sub2', test_eq($data['acc1_warnings'], 0),
$data['acc1_warnings'], (isset($data['acc1_buffer']) ? 'acc1_buffer' : NULL),
round_number($avg_data['AVG(acc1_warnings)']), $count_data['count(acc1_warnings)']),

5 => array ("", 't105', 'sub2', test_eq($data['acc2_warnings'], 0),
$data['acc2_warnings'], (isset($data['acc2_buffer']) ? 'acc2_buffer' : NULL),
round_number($avg_data['AVG(acc2_warnings)']), $count_data['count(acc2_warnings)']),

6 => array ("", 't106', 'sub2', test_eq($data['acc3_warnings'], 0),
$data['acc3_warnings'], (isset($data['acc3_buffer']) ? 'acc3_buffer' : NULL),
round_number($avg_data['AVG(acc3_warnings)']), $count_data['count(acc3_warnings)']),

7 => array ('it110', 't110', 'sub1head', "", NULL, "", ""),
8 => array ("", 't111', 'sub2', NULL,
$data['scr_scriptcount'], $data['scr_noscriptcount'], NULL,
round_number($avg_data['AVG(scr_scriptcount)']), round_number($avg_data['AVG(scr_noscriptcount)']),
$count_data['count(scr_scriptcount)'], $count_data['count(scr_noscriptcount)']),

9 => array ("", 't112', 'sub2', test_approp($data['scr_scriptcount'],
$data['scr_appropvalue']), edit_frm_input('scr_appropvalue', 'scr_scriptcount'), NULL,
round_cvt_per($avg_data['AVG(scr_appropvalue)']), $count_data['count(scr_appropvalue)']),

10 => array ('it120', 't120', 'sub1', test_eq($data['apl_count'], 0),
$data['apl_count'], NULL, round_number($avg_data['AVG(apl_count)']),
$count_data['count(apl_count)']),

11 => array ('it121', 't121', 'sub1', test_gt($data['mtg_count'], 0),
$data['mtg_count'], (isset($data['mtg_metatags']) ? 'mtg_metatags' : NULL),
round_number($avg_data['AVG(mtg_count)']), $count_data['count(mtg_count)']),

12 => array ('it122', 't122', 'sub1', test_eq($data['coo_value'], 1), NULL, NULL,
round_cvt_per($avg_data['AVG(coo_value)']), $count_data['count(coo_value)']),
),

1 => array
(
0 => array ("", 't200', "", "", "", "", ""),
1 => array ("", 't201', 'sub1', "", $data['size_size'], NULL,
round_number($avg_data['AVG(size_size)']), $count_data['count(size_size)']),

2 => array ('it202', 't202', 'sub1', test_lt($data['size_sizewithings'], 50.01),
$data['size_sizewithings'], NULL, round_number($avg_data['AVG(size_sizewithings)']),
$count_data['count(size_sizewithings)']),
),

2 => array
(
0 => array ("", 't300', "", "", "", "", ""),
1 => array ('it301', 't301', 'sub1', test_eq($data['w3c_doctypevalue'], 1), "", NULL,
round_cvt_per($avg_data['AVG(w3c_doctypevalue)']), $count_data['count(w3c_doctypevalue)']),

2 => array ('it310', 't310', 'sub1head', "", NULL, "", ""),
3 => array ("", 't311', 'sub2', "", $data['com_count'], NULL,
round_number($avg_data['AVG(com_count)']), $count_data['count(com_count)']),
4 => array ("", 't312', 'sub2', test_approp($data['com_count'],
$data['com_appropvalue']), edit_frm_input('com_appropvalue', 'com_count'),
(isset($data['com_comments']) ? 'com_comments' : NULL),
round_cvt_per($avg_data['AVG(com_appropvalue)']), $count_data['count(com_appropvalue)']),

```

```

5 => array ('t320', 't320', 'sub1head', "", "", NULL, "", ""),
6 => array ("", 't321', 'sub2', test_gt($data['tit_length'], 0), "", NULL,
round_cvt_per($avg_data['AVG(tit_value)']), $count_data['count(tit_value)']),

7 => array ("", 't322', 'sub2', test_lt($data['tit_length'], 61), $data['tit_length'],
NULL, round_number($avg_data['AVG(tit_length)']), $count_data['count(tit_length)']),

8 => array ("", 't323', 'sub2', test_approp($data['tit_length'],
$data['tit_appropvalue']), edit_frm_input('tit_appropvalue', 'tit_length'), (isset($data['tit_title']) ?
'tit_title' : NULL), round_cvt_per($avg_data['AVG(tit_appropvalue)']),
$count_data['count(tit_appropvalue)']),

9 => array ('t330', 't330', 'sub1head', "", "", NULL, "", ""),
10 => array ("", 't331', 'sub2', test_eq($data['tid_errors'], 0),
$data['tid_errors'], (isset($data['tid_errorbuffer']) ? 'tid_errorbuffer' : NULL),
round_number($avg_data['AVG(tid_errors)']), $count_data['count(tid_errors)']),

11 => array ("", 't332', 'sub2', test_eq($data['tid_warnings'], 0),
$data['tid_warnings'], (isset($data['tid_warningbuffer']) ? 'tid_warningbuffer' : NULL),
round_number($avg_data['AVG(tid_warnings)']), $count_data['count(tid_warnings)']),

12 => array ('t340', 't340', 'sub2head', "", "", NULL, "", ""),
13 => array ("", 't341', 'sub3', test_eq($data['bod_value'], 1), "", NULL,
round_cvt_per($avg_data['AVG(bod_value)']), $count_data['count(bod_value)']),

14 => array ("", 't342', 'sub3', test_eq($data['bod_textvalue'], 1), "", NULL,
round_cvt_per($avg_data['AVG(bod_textvalue)']), $count_data['count(bod_textvalue)']),

15 => array ("", 't343', 'sub3', test_eq($data['bod_linkvalue'], 1), "", NULL,
round_cvt_per($avg_data['AVG(bod_linkvalue)']), $count_data['count(bod_linkvalue)']),

16 => array ("", 't344', 'sub3', test_eq($data['bod_alinkvalue'], 1), "", NULL,
round_cvt_per($avg_data['AVG(bod_alinkvalue)']), $count_data['count(bod_alinkvalue)']),

17 => array ("", 't345', 'sub3', test_eq($data['bod_vlinkvalue'], 1), "", NULL,
round_cvt_per($avg_data['AVG(bod_vlinkvalue)']), $count_data['count(bod_vlinkvalue)']),

18 => array ("", 't350', 'sub1head', "", "", NULL, "", ""),
19 => array ("", 't351', 'sub2', "", $data['for_count'], NULL,
round_number($avg_data['AVG(for_count)']), $count_data['count(for_count)']),

20 => array ("", 't352', 'sub2', "",
$data['for_objectcount'].'.$data['for_embedcount'], NULL,
round_number($avg_data['AVG(for_objectcount)']).'.'.round_number($avg_data['AVG(for_embedcount)']),
$count_data['count(for_objectcount)'].'.$count_data['count(for_embedcount)']),

21 => array ('t353', 't353', 'sub2', test_form($data['for_count'],
$data['for_objectcount'], $data['for_embedcount'], $data['for_value']), $data['for_value'].'
'.edit_frm_input('for_value'), NULL, round_per($avg_data['AVG(for_value)']),
$count_data['count(for_value)']),

22 => array ('t354', 't354', 'sub2', test_eq($data['for_labelper'], 100),
$data['for_labelper'], NULL, round_per($avg_data['AVG(for_labelper)']),
$count_data['count(for_labelper)']),
23 => array ('t355', 't355', 'sub2', test_eq($data['for_tabidxper'], 100),
$data['for_tabidxper'], NULL, round_per($avg_data['AVG(for_tabidxper)']),
$count_data['count(for_tabidxper)'])
)
);

```

```

$ge_array = array
(
    0 => array
        (
            0 => array ("', 'g100', ", ", ", ", ", ", ),
            1 => array ('ig101', 'g101', 'sub1', test_poss($data['anc_mainmenuvalue']),
edit_frm_input('anc_mainmenuvalue'), NULL,
round_cvt_per($avg_data['AVG(anc_mainmenuvalue)']), $count_data['count(anc_mainmenuvalue)']),

            2 => array ("', 'g102', 'sub1', "", $data['anc_count'], NULL,
round_number($avg_data['AVG(anc_count)']), $count_data['count(anc_count)']),
        ),

    1 => array
        (
            0 => array ("', 'g200', ", ", ", ", ", ", ),
            1 => array ("', 'g201', 'sub1', test_poss($data['img_logovalue']),
edit_frm_input('img_logovalue'), NULL, round_cvt_per($avg_data['AVG(img_logovalue)']),
$count_data['count(img_logovalue)']),

            2 => array ('ig202', 'g202', 'sub1', test_poss_rel($data['img_textreadlogovalue'],
$data['img_logovalue']), edit_frm_input('img_textreadlogovalue'), NULL,
round_cvt_per($avg_data['AVG(img_textreadlogovalue)']),
$count_data['count(img_textreadlogovalue)']),

            3 => array ('ig203', 'g203', 'sub1', test_poss_rel($data['img_giflogovalue'],
$data['img_logovalue']), edit_frm_input('img_giflogovalue'), NULL,
round_cvt_per($avg_data['AVG(img_giflogovalue)']), $count_data['count(img_giflogovalue)']),

            4 => array ('ig204', 'g204', 'sub1', test_poss_rel($data['img_statlogovalue'],
$data['img_logovalue']), edit_frm_input('img_statlogovalue'), NULL,
round_cvt_per($avg_data['AVG(img_statlogovalue)']), $count_data['count(img_statlogovalue)']),
        ),

    2 => array
        (
            0 => array ("', 'g300', ", ", ", ", ", ", ),
            1 => array ("', 'g310', 'sub1head', "", ", NULL, ", ", ),
            2 => array ('ig311', 'g311', 'sub2', "", $data['anc_withoutimgper'], NULL,
round_per($avg_data['AVG(anc_withoutimgper)']), $count_data['count(anc_withoutimgper)']),

            3 => array ('ig312', 'g312', 'sub2', test_eq($data['anc_imgaltper'], 100),
$data['anc_imgaltper'], NULL, round_per($avg_data['AVG(anc_imgaltper)']),
$count_data['count(anc_imgaltper)']),

            4 => array ("', 'g320', 'sub1head', "", ", NULL, ", ", ),
            5 => array ('ig321', 'g321', 'sub2', test_poss($data['anc_htmlmenuvalue']),
edit_frm_input('anc_htmlmenuvalue'), NULL,
round_cvt_per($avg_data['AVG(anc_htmlmenuvalue)']), $count_data['count(anc_htmlmenuvalue)']),

            6 => array ('ig330', 'g330', 'sub1head', "", ", NULL, ", ", ),
            7 => array ("', 'g331', 'sub2', test_eq($data['anc_tabidxper'], 100),
$data['anc_tabidxper'], NULL, round_per($avg_data['AVG(anc_tabidxper)']),
$count_data['count(anc_tabidxper)']),

            8 => array ("', 'g332', 'sub2', test_eq($data['map_areatabidxper'], 100),
$data['map_areatabidxper'], NULL, round_per($avg_data['AVG(map_areatabidxper)']),
$count_data['count(map_areatabidxper)']),
        ),
)

```

```

9 => array ('g340', 'g340', 'sub1head', "", "", NULL, "", ""),
10 => array ("", 'g341', 'sub2', test_eq($data['anc_acckeyper'], 100),
$data['anc_acckeyper'], NULL, round_per($avg_data['AVG(anc_acckeyper)']),
$count_data['count(anc_acckeyper)']),

11 => array ("", 'g342', 'sub2', test_eq($data['map_areaacckeyper'], 100),
$data['map_areaacckeyper'], NULL, round_per($avg_data['AVG(map_areaacckeyper)']),
$count_data['count(map_areaacckeyper)']),

12 => array ("", 'g343', 'sub2', test_eq($data['for_butacckeyper'], 100),
$data['for_butacckeyper'], NULL, round_per($avg_data['AVG(for_butacckeyper)']),
$count_data['count(for_butacckeyper)']),

13 => array ("", 'g344', 'sub2', test_eq($data['for_inpacckeyper'], 100),
$data['for_inpacckeyper'], NULL, round_per($avg_data['AVG(for_inpacckeyper)']),
$count_data['count(for_inpacckeyper)']),

14 => array ("", 'g345', 'sub2', test_eq($data['for_tareaacckeyper'], 100),
$data['for_tareaacckeyper'], NULL, round_per($avg_data['AVG(for_tareaacckeyper)']),
$count_data['count(for_tareaacckeyper)']),

15 => array ("", 'g346', 'sub2', test_eq($data['for_selacckeyper'], 100),
$data['for_selacckeyper'], NULL, round_per($avg_data['AVG(for_selacckeyper)']),
$count_data['count(for_selacckeyper)']),

16 => array ("", 'g350', 'sub1head', "", "", NULL, "", ""),
17 => array ("", 'g351', 'sub2', "", $data['map_mapcount'], NULL,
round_number($avg_data['AVG(map_mapcount)']), $count_data['count(map_mapcount)']),

18 => array ('g352', 'g352', 'sub2', test_approp($data['map_mapcount'],
$data['map_appropvalue']), edit_frm_input('map_appropvalue', 'map_mapcount'), NULL,
round_cvt_per($avg_data['AVG(map_appropvalue)']), $count_data['count(map_appropvalue)']),

19 => array ("", 'g353', 'sub2', "", $data['map_areacount'], NULL,
round_number($avg_data['AVG(map_areacount)']), $count_data['count(map_areacount)']),

20 => array ('g354', 'g354', 'sub2', test_eq($data['map_areaaltper'], 100),
$data['map_areaaltper'], NULL, round_per($avg_data['AVG(map_areaaltper)']),
$count_data['count(map_areaaltper)'])
),
3 => array
(
0 => array ("", 'g400', "", "", "", "", ""),
1 => array ('g401', 'g401', 'sub1', test_eq($data['fon_value'], 1), NULL, NULL,
round_cvt_per($avg_data['AVG(fon_value)']), $count_data['count(fon_value)']),

2 => array ('g402', 'g402', 'sub1', test_eq($data['mar_value'], 1), NULL, NULL,
round_cvt_per($avg_data['AVG(mar_value)']), $count_data['count(mar_value)']),

3 => array ('g410', 'g410', 'sub1head', "", "", NULL, "", ""),
4 => array ("", 'g411', 'sub2', test_eq($data['fs1_per'], 100), $data['fs1_per'],
NULL, round_per($avg_data['AVG(fs1_per)']), $count_data['count(fs1_per)']),

5 => array ("", 'g412', 'sub2', test_eq($data['fs2_per'], 100), $data['fs2_per'],
NULL, round_per($avg_data['AVG(fs2_per)']), $count_data['count(fs2_per)'])
),
4 => array

```

```

(
  0 => array ("", 'g500', "", "", "", "", ""),
  1 => array ("", 'g501', 'sub1', "", $data['img_downloaded'], $data['img_count'],
NULL,
round_number($avg_data['AVG(img_downloaded)'], round_number($avg_data['AVG(img_count)']
), $count_data['count(img_downloaded)'], $count_data['count(img_count)']),

  2 => array ('ig502', 'g502', 'sub1', test_eq($data['img_exttypeper'], 100),
$data['img_exttypeper'], NULL, round_per($avg_data['AVG(img_exttypeper)']),
$count_data['count(img_exttypeper)']),

  3 => array ('ig503', 'g503', 'sub1', test_eq($data['img_filesizeper'], 100),
$data['img_filesizeper'], NULL, round_per($avg_data['AVG(img_filesizeper)']),
$count_data['count(img_filesizeper)']),

  4 => array ('ig504', 'g504', 'sub1', test_eq($data['img_widthper'], 100),
$data['img_widthper'], NULL, round_per($avg_data['AVG(img_widthper)']),
$count_data['count(img_widthper)']),

  5 => array ('ig505', 'g505', 'sub1', test_eq($data['img_heightper'], 100),
$data['img_heightper'], NULL, round_per($avg_data['AVG(img_heightper)']),
$count_data['count(img_heightper)']),

  6 => array ("", 'g510', 'sub1head', "", "", NULL, "", ""),
  7 => array ('ig511', 'g511', 'sub2', test_eq($data['img_altper'], 100),
$data['img_altper'], NULL, round_per($avg_data['AVG(img_altper)']),
$count_data['count(img_altper)']),

  8 => array ('ig512', 'g512', 'sub2', test_eq($data['img_altlengthper'], 100),
$data['img_altlengthper'], NULL, round_per($avg_data['AVG(img_altlengthper)']),
$count_data['count(img_altlengthper)']),

  9 => array ('ig513', 'g513', 'sub2', test_eq($data['img_altendcharper'], 100),
$data['img_altendcharper'], NULL, round_per($avg_data['AVG(img_altendcharper)']),
$count_data['count(img_altendcharper)']),

  10 => array ('ig514', 'g514', 'sub2', test_eq($data['img_altendcharsper'], 100),
$data['img_altendcharsper'], NULL, round_per($avg_data['AVG(img_altendcharsper)']),
$count_data['count(img_altendcharsper)']),
),

5 => array
(
  0 => array ("", 'g600', "", "", "", "", ""),
  1 => array ('ig601', 'g601', 'sub1', test_eq($data['col1_per'], 100),
$data['col1_per'], NULL, round_per($avg_data['AVG(col1_per)']), $count_data['count(col1_per)']),

  2 => array ('ig602', 'g602', 'sub1', test_eq($data['col2_per'], 100),
$data['col2_per'], NULL, round_per($avg_data['AVG(col2_per)']), $count_data['count(col2_per)']),

  3 => array ('ig603', 'g603', 'sub1', test_eq($data['col3_per'], 100),
$data['col3_per'], NULL, round_per($avg_data['AVG(col3_per)']), $count_data['count(col3_per)']),
),

6 => array
(
  0 => array ("", 'g700', "", "", "", "", ""),
  1 => array ("", 'g701', 'sub1', "", $data['tab_count'], NULL,
round_number($avg_data['AVG(tab_count)']), $count_data['count(tab_count)']),

```

```

                2 => array ('g702', 'g702', 'sub1', test_eq($data['tab_endtagper'], 100),
$data['tab_endtagper'], NULL, round_per($avg_data['AVG(tab_endtagper)'],
$count_data['count(tab_endtagper)']),

                3 => array ('g703', 'g703', 'sub1', test_eq($data['tab_bgcolorper'], 100),
$data['tab_bgcolorper'], NULL, round_per($avg_data['AVG(tab_bgcolorper)'],
$count_data['count(tab_bgcolorper)']),

                4 => array ('g704', 'g704', 'sub1', test_eq($data['tab_bgroundper'], 100),
$data['tab_bgroundper'], NULL, round_per($avg_data['AVG(tab_bgroundper)'],
$count_data['count(tab_bgroundper)'])
            ),
        );

        echo "<center>
            <div class='fra'>
                <div class='suml'>
                    ".$locale['r100']." ".$info_row['total']."<br>
                    ".$locale['r101']." ".$avg_data['count(*)']."
                </div>
                <div class='sumr'>
                    <form method='post' action=".$thisfile."?ev=$ev$filt_link>
                        $frm_adm_sub1 $frm_adm_sub2
                        <select name='filt_type' class='textbox'>\n
                            <option value='all'>all</option>\n";
                        $db->query("SELECT DISTINCT type FROM
webpage");
                            while ($tydata = $db->get_row())
                            {
                                echo "<option
value=".$tydata['type']."". (isset($filt_type) && $filt_type == $tydata['type'] ? " selected" :
"". ">".$tydata['type']."</option>\n";
                            }
                        echo "</select>\n

                            <select name='filt_country' class='textbox'>\n
                            <option value='all'>all</option>\n";
                            $db->query("SELECT DISTINCT country FROM
webpage");
                            while ($codata = $db->get_row())
                            {
                                echo "<option
value=".$codata['country']."". (isset($filt_country) && $filt_country == $codata['country'] ? "
selected" : ""). ">".$codata['country']."</option>\n";
                            }
                        echo "</select>\n

                            $frm_adm_path $frm_adm_who
                            <input type='submit' class='textbox' value='OK' name='ok'>\n
                            $frm_adm_clean
                        </form>
                    </div>
                </div>

            <h2><a href=".$data['site']."" target='blank'>".$data['site']."</a> ".$locale['r110']."</h2>";
            echo "$records
                <a ".$( $ev == 'co' ? "class='act'": "" )."href=".$thisfile."?ev=co$filt_link'
accesskey='c'>".$locale['r111']."</a> |

```



```

        <a ".($sev == 'ac' ? "class='act'": "")."href=".$thisfile."?ev=ac$filt_link'
accesskey='a'>".$locale['r112'].</a> |
        <a ".($sev == 'te' ? "class='act'": "")."href=".$thisfile."?ev=te$filt_link'
accesskey='t'>".$locale['r113'].</a> |
        <a ".($sev == 'ge' ? "class='act'": "")."href=".$thisfile."?ev=ge$filt_link'
accesskey='v'>".$locale['r114'].</a> |
        <a href=".$thisfile."?ev=lo">".$locale['r115'].</a>;

if (isset($_SESSION['admin']) && $sev == 'data')
{
    echo "<p><div class='ana'>";
    echo "<div class='hd1'>".$locale['adm105'].</div>\n <div
class='hd2'>".$locale['adm106'].</div>\n <div class='hd2'>".$locale['adm107'].</div>\n";

    $rowstart = @$_GET['rowstart']; if (!isset($rowstart)) $rowstart = 0;
    $db->query("SELECT * FROM webpage WHERE 1 ".$filt_sql." AND id IN (SELECT
max(id) FROM webpage GROUP BY site) ORDER BY time DESC LIMIT $rowstart,10");
    while ($filt_data = $db->get_row())
    {
        echo "<div class='d1'><a href=".$thisfile."?site_id=".$filt_data['id'].'"
alt='".$filt_data['site']."'>".trimlink($filt_data['site'], 45).</a></div>";
        echo "<div class='d2'>.date('d/m/y h:i', $filt_data['time']).</div>";
        echo "<div class='d2'>".$filt_data['who'].</div>";
    }
    echo "</div>";
    echo "<p><div align='center' style='margin-top:5px;clear:both;'>\n".$locale['adm108'].
".makepagenav($rowstart,10,$avg_data['count(*)'],10,'".$thisfile."?ev=data$filt_link&")."\n</div>\n"
;
} else
{
    $det = @$_GET['det'];
    $info = @$_GET['info'];
    if (isset($det) || isset($info))
    {
        echo "<p><div class='fra'><div class='det'>";
        if (isset($det))
        {
            echo nl2br($data[$det]);
        } elseif (isset($info))
        {
            echo $locale[$info];
        }
        echo "</div></div>";
    }

    $view = $sev == 'co' ? $co_array : ($sev == 'ac' ? $ac_array : ($sev == 'te' ? $te_array :
$ge_array));

    for ($i = 0; $i < count($view); $i++)
    {
        echo "<p><div class='fra'>";
        if ($i == 0) echo $edit_frm_s;
        for ($j = 0; $j < count($view[$i]); $j++)
        {
            if ($j == 0)
            {
                echo "<div class='head1'>".$locale[$view[$i][$j][1]].</div>\n
                <div class='head2'>".$locale['r120'].</div>\n
                <div class='head3'>".$locale['r121'].</div>\n

```

```

        <div class='head4'>".$locale['r122']."</div>\n
        <div class='head5'>".$locale['r123']."</div>\n
        <div class='head6'>".$locale['r124']."</div>\n";
    } else
    {
        echo "<div class='t1'><span class='". $view[$i][$j][2]."'>".($view[$i][$j][0]
<> " ? "<a href='". $thisfile."?ev=$ev&info=".$view[$i][$j][0]."'
class='info'>". $locale[$view[$i][$j][1]]."</a>" : $locale[$view[$i][$j][1]]."</span> </div>\n";
        echo "<div class='t2'>". $view[$i][$j][3]."</div>\n";
        echo "<div class='t3'>". $view[$i][$j][4]."</div>\n";
        echo "<div class='t4'>";
        if (isset($view[$i][$j][5]))
        {
            if ($view[$i][$j][0] == 'ia201' || $view[$i][$j][0] == 'ia202' ||
$view[$i][$j][0] == 'ia203')
            {
                echo $view[$i][$j][5];
            } else echo "<a
href='". $thisfile."?ev=$ev&det=".$view[$i][$j][5]."'>". $locale['r130']."</a>";
            } else echo ";
        echo "</div>\n";
        echo "<div class='t5'>". $view[$i][$j][6]."</div>\n";
        echo "<div class='t6'>". $view[$i][$j][7]."</div>\n";
    }
}
if ($i == count($view) - 1) echo $edit_frm_e;
echo "</div>";
}

echo "<p><div class='fra'>
    <div class='expl'>". $locale['r133']."<br>". $locale['r134']."</div>
    <div class='expl'>". $locale['r135']."<br>". $locale['r136']."</div>
    <div class='expl'>". $locale['r131']."<br>". $locale['r132']."</div>
</div>";
}
echo" </center>
</body>
</html>";

ob_end_flush();

}

?>

```

EK C.2 Veritabanı Tablolarını Oluşturan SQL Kodları

EK C.2.1 Info Tablosu (sql_info.txt)

```

CREATE TABLE IF NOT EXISTS `info` (
    `total` int(11) default NULL,
    `visited` int(11) default NULL,
    `last_ip` varchar(15) collate latin1_general_ci default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;

```

EK C.2.2 Webpage Tablosu (sql_webpage.txt)

```

CREATE TABLE IF NOT EXISTS `webpage` (
  `id` int(11) NOT NULL auto_increment,
  `site` varchar(100) collate latin1_general_ci NOT NULL,
  `sub1` varchar(25) collate latin1_general_ci default NULL,
  `sub2` varchar(25) collate latin1_general_ci default NULL,
  `type` varchar(6) collate latin1_general_ci default NULL,
  `country` varchar(2) collate latin1_general_ci default NULL,
  `path` varchar(42) collate latin1_general_ci default NULL,
  `time` int(11) NOT NULL,
  `who` varchar(5) collate latin1_general_ci NOT NULL,

  `tid_warnings` int(11) default NULL,
  `tid_errors` int(11) default NULL,
  `tid_errorbuffer` mediumtext collate latin1_general_ci default NULL,
  `tid_warningbuffer` mediumtext collate latin1_general_ci default NULL,

  `acc1_warnings` int(11) default NULL,
  `acc2_warnings` int(11) default NULL,
  `acc3_warnings` int(11) default NULL,
  `acc1_buffer` mediumtext collate latin1_general_ci default NULL,
  `acc2_buffer` mediumtext collate latin1_general_ci default NULL,
  `acc3_buffer` mediumtext collate latin1_general_ci default NULL,

  `coo_value` char(1) collate latin1_general_ci default NULL,

  `w3c_htmlerrors` int(11) default NULL,
  `w3c_csserrors` int(11) default NULL,
  `w3c_doctypevalue` char(1) collate latin1_general_ci default NULL,

  `com_count` int(11) default NULL,
  `com_appropvalue` char(1) collate latin1_general_ci default NULL,
  `com_comments` mediumtext collate latin1_general_ci default NULL,

  `tit_length` int(11) default NULL,
  `tit_value` char(1) collate latin1_general_ci default NULL,
  `tit_appropvalue` char(1) collate latin1_general_ci default NULL,
  `tit_title` tinytext collate latin1_general_ci default NULL,

  `scr_scriptcount` int(11) default NULL,
  `scr_noscriptcount` int(11) default NULL,
  `scr_appropvalue` char(1) collate latin1_general_ci default NULL,

  `apl_count` int(11) default NULL,

  `mtg_count` int(11) default NULL,
  `mtg_metatags` mediumtext collate latin1_general_ci default NULL,

  `anc_count` int(11) default NULL,
  `anc_withoutimgper` float default NULL,
  `anc_imgaltper` float default NULL,
  `anc_tabidxper` float default NULL,
  `anc_acckeyper` float default NULL,
  `anc_mainmenuvalue` char(1) collate latin1_general_ci default NULL,
  `anc_htmlmenuvalue` char(1) collate latin1_general_ci default NULL,

  `bod_value` char(1) collate latin1_general_ci default NULL,

```

```

`bod_textvalue` char(1) collate latin1_general_ci default NULL,
`bod_linkvalue` char(1) collate latin1_general_ci default NULL,
`bod_alinkvalue` char(1) collate latin1_general_ci default NULL,
`bod_vlinkvalue` char(1) collate latin1_general_ci default NULL,

`for_count` int(11) default NULL,
`for_objectcount` int(11) default NULL,
`for_embedcount` int(11) default NULL,
`for_value` float default NULL,
`for_labelper` float default NULL,
`for_tabidxper` float default NULL,
`for_inpacckeyper` float default NULL,
`for_selacckeyper` float default NULL,
`for_tareaacckeyper` float default NULL,
`for_butacckeyper` float default NULL,

`fon_value` char(1) collate latin1_general_ci default NULL,

`mar_value` char(1) collate latin1_general_ci default NULL,

`map_mapcount` int(11) default NULL,
`map_areacount` int(11) default NULL,
`map_areaaltper` float default NULL,
`map_areatabidxper` float default NULL,
`map_areaacckeyper` float default NULL,
`map_appropvalue` char(1) collate latin1_general_ci default NULL,

`img_count` int(11) default NULL,
`img_downloaded` int(11) default NULL,
`img_exttypeper` float default NULL,
`img_filesizeper` float default NULL,
`img_widthper` float default NULL,
`img_heightper` float default NULL,
`img_altper` float default NULL,
`img_altlengthper` float default NULL,
`img_altendcharper` float default NULL,
`img_altendcharsper` float default NULL,
`img_logovalue` char(1) collate latin1_general_ci default NULL,
`img_giflogovalue` char(1) collate latin1_general_ci default NULL,
`img_textreadlogovalue` char(1) collate latin1_general_ci default NULL,
`img_statlogovalue` char(1) collate latin1_general_ci default NULL,

`tab_count` int(11) default NULL,
`tab_endtagper` float default NULL,
`tab_bgcolorper` float default NULL,
`tab_bgroundper` float default NULL,

`col1_per` float default NULL,
`col2_per` float default NULL,
`col3_per` float default NULL,

`fs1_per` float default NULL,
`fs2_per` float default NULL,

`size_size` float default NULL,
`size_sizewithimgs` float default NULL,

`cont_logovalue` char(1) collate latin1_general_ci default NULL,
`cont_instnamevalue` char(1) collate latin1_general_ci default NULL,

```

```

`cont_aimvalue` char(1) collate latin1_general_ci default NULL,
`cont_ancvalue` char(1) collate latin1_general_ci default NULL,
`cont_annovalue` char(1) collate latin1_general_ci default NULL,
`cont_searchvalue` char(1) collate latin1_general_ci default NULL,
`cont_helpvalue` char(1) collate latin1_general_ci default NULL,
`cont_faqlvalue` char(1) collate latin1_general_ci default NULL,
`cont_feedvalue` char(1) collate latin1_general_ci default NULL,
`cont_addressvalue` char(1) collate latin1_general_ci default NULL,
`cont_phonevalue` char(1) collate latin1_general_ci default NULL,
`cont_mailvalue` char(1) collate latin1_general_ci default NULL,
`cont_appropmailvalue` char(1) collate latin1_general_ci default NULL,

`cont_logos` text collate latin1_general_ci default NULL,
`cont_instname` tinytext collate latin1_general_ci default NULL,
`cont_aims` text collate latin1_general_ci default NULL,
`cont_hrefs` mediumtext collate latin1_general_ci default NULL,
`cont_annos` text collate latin1_general_ci default NULL,
`cont_searchs` text collate latin1_general_ci default NULL,
`cont_helps` text collate latin1_general_ci default NULL,
`cont_faqs` text collate latin1_general_ci default NULL,
`cont_feeds` text collate latin1_general_ci default NULL,
`cont_addresses` text collate latin1_general_ci default NULL,
`cont_phones` text collate latin1_general_ci default NULL,
`cont_mails` text collate latin1_general_ci default NULL,
`cont_appropmails` text collate latin1_general_ci default NULL,

`acce_ievalue` char(1) collate latin1_general_ci default NULL,
`acce_ffvalue` char(1) collate latin1_general_ci default NULL,
`acce_konvalue` char(1) collate latin1_general_ci default NULL,
`acce_lyvalue` char(1) collate latin1_general_ci default NULL,

`acce_res1value` char(1) collate latin1_general_ci default NULL,
`acce_res2value` char(1) collate latin1_general_ci default NULL,
`acce_res3value` char(1) collate latin1_general_ci default NULL,

PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci
AUTO_INCREMENT=1;

```

EK C.3 Kullanılan Veritabanı Class'ı (class.database.php)

```

<?php
/**
 * Database class
 *
 * @author Sven Wagener <wagener_at_indot_dot_de>
 * @copyright Sven Wagener
 * @include Funktion: _include_
 */
class database{
    var $database_types="";

    var $db_connect="";
    var $db_close="";
    var $db_select_db="";

```

```

var $db_query="";
var $db_fetch_array="";
var $db_num_rows="";

var $host;
var $database;
var $user;
var $password;
var $port;
var $database_type;
var $dsn;

var $sql;

var $con; // variable for connection id
var $con_string; // variable for connection string
var $query_id; // variable for query id

var $errors; // variable for error messages
var $error_count=0; // variable for counting errors
var $error_nr;
var $error;

var $debug=false; // debug mode off

/**
 * Constructor of class - Initializes class and connects to the database
 * @param string $database_type the name of the database
 * (ifx=Informix,msql=MiniSQL,mssql=MS SQL,mysql=MySQL,pg=Postgres SQL,sybase=Sybase)
 * @param string $host the host of the database
 * @param string $database the name of the database
 * @param string $user the name of the user for the database
 * @param string $password the password of the user for the database
 * @desc Constructor of class - Initializes class and connects to the database.
 *
 * You can use this shortcuts for the database type:
 *
 * ifx -> INFORMIX
 * msql -> MiniSQL
 * mssql -> Microsoft SQL Server
 * mysql -> MySQL
 * odbc -> ODBC
 * pg -> Postgres SQL
 * sybase -> Sybase
 */
function database($database_type,$host,$database,$user,$password,$port=false,$dsn=false){
    $database_type=strtolower($database_type);
    $this->host=$host;
    $this->database=$database;
    $this->user=$user;
    $this->password=$password;
    $this->port=$port;
    $this->dsn=$dsn;

    $this->database_types=array("ifx","msql","mssql","mysql","odbc","pg","sybase");

    // Setting database type and connect to database
    if(in_array($database_type,$this->database_types)){
        $this->database_type=$database_type;
    }
}

```

```

$this->db_connect=$this->database_type."_connect";
$this->db_close=$this->database_type."_close";
$this->db_select_db=$this->database_type."_select_db";

if($database_type=="odbc"){
    $this->db_query=$this->database_type."_exec";
    $this->db_fetch_array=$this->database_type."_fetch_row";
} else {
    $this->db_query=$this->database_type."_query";
    $this->db_fetch_array=$this->database_type."_fetch_array";
}

$this->db_num_rows=$this->database_type."_num_rows";

return $this->connect();
} else {
    $this->halt("Database type not supported");
    return false;
}
}

/**
 * This function connects the database
 * @return boolean $is_connected Returns true if connection was successful otherwise false
 * @desc This function connects to the database which is set in the constructor
 */
function connect(){
    // Selecting connection function and connecting

    if($this->con==""){
        // INFORMIX
        if($this->database_type=="ifx"){
            $this->con=call_user_func($this->db_connect,$this->database."@".$this-
>host,$this->user,$this->password);
        } else if($this->database_type=="mysql"){
            // With port
            if(!$this->port){
                $this->con=call_user_func($this->db_connect,$this->host.":".$this-
>port,$this->user,$this->password);
            }
            // Without port
            else {
                $this->con=call_user_func($this->db_connect,$this->host,$this-
>user,$this->password);
            }
            // mSQL
        } else if($this->database_type=="msql"){
            $this->con=call_user_func($this->db_connect,$this->host,$this->user,$this-
>password);
            // MS SQL Server
        } else if($this->database_type=="mssql"){
            $this->con=call_user_func($this->db_connect,$this->host,$this->user,$this-
>password);
            // ODBC
        } else if($this->database_type=="odbc"){
            $this->con=call_user_func($this->db_connect,$this->dsn,$this->user,$this-
>password);
            // Postgres SQL
        } else if($this->database_type=="pg"){

```

```

        // With port
        if(!$this->port){
            $this->con=call_user_func($this->db_connect,"host=".$this->host."
port=".$this->port." dbname=".$this->database." user=".$this->user." password=".$this->password);
        }
        // Without port
        else{
            $this->con=call_user_func($this->db_connect,"host=".$this->host."
dbname=".$this->database." user=".$this->user." password=".$this->password);
        }
        // Sybase
    }else if($this->database_type=="sybase"){
        $this->con=call_user_func($this->db_connect,$this->host,$this->user,$this-
>password);
    }

    if(!$this->con){
        $this->halt("Wrong connection data! Can't establish connection to host.");
        return false;
    }else{
        if($this->database_type!="odbc"){
            if(!call_user_func($this->db_select_db,$this->database,$this->con)){
                $this->halt("Wrong database data! Can't select database.");
                return false;
            }else{
                return true;
            }
        }
    }
}else{
    $this->halt("Already connected to database.");
    return false;
}
}

/**
 * This function disconnects from the database
 * @desc This function disconnects from the database
 */
function disconnect(){
    if(@call_user_func($this->db_close,$this->con)){
        return true;
    }else{
        $this->halt("Not connected yet");
        return false;
    }
}

/**
 * This function starts the sql query
 * @param string $sql_statement the sql statement
 * @return boolean $successfull returns false on errors otherwise true
 * @desc This function disconnects from the database
 */
function query($sql_statement){
    $this->sql=$sql_statement;
    if($this->debug){
        printf("SQL statement: %s\n",$this->sql);
    }
}

```



```

if($this->database_type=="odbc"){
    // ODBC
    if(!$this->query_id=call_user_func($this->db_query,$this->con,$this->sql)){
        $this->halt("No database connection exists or invalid query");
    }else{
        if (!$this->query_id) {
            $this->halt("Invalid SQL Query");
            return false;
        }else{
            return true;
        }
    }
}
}
}
}

/**
 * This function returns a row of the resultset
 * @return array $row the row as array or false if there is no more row
 * @desc This function returns a row of the resultset
 */
function get_row(){
    if($this->database_type=="odbc"){
        // ODBC database
        if($row=call_user_func($this->db_fetch_array,$this->query_id)){
            for ($i=1; $i<=odbc_num_fields($this->query_id); $i++) {
                $fieldname=odbc_field_name($this->query_id,$i);
                $row_array[$fieldname]=odbc_result($this->query_id,$i);
            }
            return $row_array;
        }else{
            return false;
        }
    }else{
        // All other databases
        $row=call_user_func($this->db_fetch_array,$this->query_id);
        return $row;
    }
}

/**
 * This function returns number of rows in the resultset
 * @return int $row_count the nuber of rows in the resultset
 * @desc This function returns number of rows in the resultset
 */
function count_rows(){
    $row_count=call_user_func($this->db_num_rows,$this->query_id);
}

```

```

        if($row_count>=0){
            return $row_count;
        }else{
            $this->halt("Can't count rows before query was made");
            return false;
        }
    }
}
/**
 * This function returns all tables of the database in an array
 * @return array $tables all tables of the database in an array
 * @desc This function returns all tables of the database in an array
 */
function get_tables(){
    if($this->database_type=="odbc"){
        // ODBC databases
        $tablelist=odbc_tables($this->con);

        for($i=0;odbc_fetch_row($tablelist);$i++) {
            $tables[$i]=odbc_result($tablelist,3);
        }
        return $tables;
    }else{
        // All other databases
        $tables = "";
        $sql="SHOW TABLES";
        $this->query_id($sql);
        for($i=0;$data=$this->get_row();$i++){
            $tables[$i]=$data["Tables_in_". $this->database];
        }
        return $tables;
    }
}

/**
 * Prints out a error message
 * @param string $message all occurred errors as array
 * @desc Returns all occurred errors
 */
function halt($message){
    if($this->debug){
        printf("Database error: %s\n", $message);
        if($this->error_nr!="" && $this->error!=""){
            printf("MySQL Error: %s (%s)\n",$this->error_nr,$this->error);
        }
        die ("Session halted.");
    }
}

/**
 * Switches to debug mode
 * @param boolean $switch
 * @desc Switches to debug mode
 */
function debug_mode($debug=true){
    $this->debug=$debug;
}
}
?>

```

EK C.4 Arayüz Dil Dosyaları

EK C.4.1 Türkçe Dil Dosyası (/locale_res/tr.php)

<?php

```

$locale['e100'] = "Web Sayfası Değerlendirme Aracı - Erkan DURAN";
$locale['e101'] = "Yüksek Lisans, Tez, Web Sayfası Değerlendirme, Araç, 2007/4, Genelge, W3C,
Web Standartları, Erişilebilirlik, e-Devlet, Erkan DURAN, Taner TANRISEVER, Balıkesir
Üniversitesi";
$locale['e102'] = "KAMU KURUM VE KURULUŞLARI İNTERNET SİTELERİNİN<br>2007/4
SAYILI GENELGE KAPSAMINDA DEĞERLENDİRİLMESİNE YÖNELİK<br>BİR ARAÇ
GELİŞTİRİLMESİ <p>&nbsp;</p>";
$locale['e103'] = "Erkan DURAN<br>Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü<br>Bilgisayar
ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı<p>&nbsp;</p>";
$locale['e104'] = "(Yüksek Lisans Tezi / Tez Danışmanı: Yrd. Doç. Dr. Taner
TANRISEVER)<p>&nbsp;</p>";
$locale['e105'] = "Balıkesir, 2008<hr>";

$locale['e110'] = "Güvenlik Kodu";
$locale['e111'] = "Web Sayfası";
$locale['e112'] = "W3C HTML Testi";
$locale['e113'] = "W3C CSS Testi";
$locale['e114'] = "Tamam";
$locale['e115'] = "Doğru sonuçlar için yönlendirilen sayfa adresleri yerine sayfanın tam adresi giriniz
ve çerçeve içermeyen sayfaları test ediniz.";
$locale['e116'] = "Toplam Ziyaret.";
$locale['e117'] = "Güvenlik Kodu Hatalı";

$locale['adm100'] = "[Bu siteyi sil]";
$locale['adm101'] = "[Eski siteleri sil]";
$locale['adm102'] = "[Geçici dosyaları/klasörleri sil]";
$locale['adm103'] = "Emin misiniz?";
$locale['adm104'] = "[Siteler]";
$locale['adm105'] = "Site";
$locale['adm106'] = "Tarih";
$locale['adm107'] = "Değerlendiren";
$locale['adm108'] = "Sayfa";

$locale['r100'] = "Değerlendirilen Toplam Site.";
$locale['r101'] = "Ortalamaya Dahil Edilen (N).";
$locale['r110'] = "Değerlendirme Sonuçları";
$locale['r111'] = "İçerik (Asgari)";
$locale['r112'] = "Tasarım (Erişilebilirlik)";
$locale['r113'] = "Tasarım (Teknik Özellikler)";
$locale['r114'] = "Genel Görünüm";
$locale['r115'] = "[Çıkış]";
$locale['r120'] = "Sonuç";
$locale['r121'] = "Değer";
$locale['r122'] = "Görüntüle";
$locale['r123'] = "G. Ortalama";
$locale['r124'] = "(n)";

$locale['r130'] = "<img src='images/view.gif' alt='Görüntüle.' width='16' height='16'>";
$locale['r131'] = SUCCESSFUL. " Başarılı";
$locale['r132'] = FAILED. " Başarısız";

```

```

$locale['r133'] = N_A. " Belirlenemedi";
$locale['r134'] = NULL_X. " Yok/Değ. dışı";
$locale['r135'] = POSSIBLE. " Olabilir ";
$locale['r136'] = NOT_POSSIBLE. " Olmayabilir";

$locale['c100'] = "Genel";
$locale['c101'] = "Kurum Logosu";
$locale['c102'] = "Kurumun Açık Adı";
$locale['c103'] = "Sitenin Amacı";
$locale['c104'] = "Siteiçi Bağlantılar";
$locale['c105'] = "Duyurular/Haberler";
$locale['c106'] = "Arama Mekanizması";
$locale['c107'] = "Yardım Sayfalarına Bağlantı";
$locale['c108'] = "S.S.S. Sayfalarına Bağlantı";
$locale['c109'] = "Şikayet/Geri Bildirim Say. Bağ.";

$locale['c200'] = "Kuruma Erişim Bilgileri";
$locale['c201'] = "Adres";
$locale['c202'] = "Telefon";
$locale['c203'] = "E-posta";
$locale['c204'] = "E-posta Uygunluğu";

$locale['a100'] = "Tarayıcı Tipi";
$locale['a101'] = "Internet Explorer";
$locale['a102'] = "Mozilla Firefox";
$locale['a103'] = "Konqueror";
$locale['a104'] = "Lynx";

$locale['a200'] = "Ekran Çözünürlüğü";
$locale['a201'] = "1280x1024";
$locale['a202'] = "1024x768";
$locale['a203'] = "800x600";

$locale['t100'] = "Genel";
$locale['t101'] = "W3C HTML Testi (Hata Sayısı)";
$locale['t102'] = "W3C CSS Testi (Hata Sayısı)";
$locale['t103'] = "TIDY Erişebilirlik Testi (Uyarı Sayısı)";
$locale['t104'] = "Seviye 1";
$locale['t105'] = "Seviye 2";
$locale['t106'] = "Seviye 3";

$locale['t110'] = "Javascript'ler";
$locale['t111'] = "Javascript/Noscript (Adet)";
$locale['t112'] = "Script Kullanımı";

$locale['t120'] = "Java Applet'ler (Adet)";
$locale['t121'] = "Meta Tag'lar(Adet)";
$locale['t122'] = "Çerezler";

$locale['t200'] = "Sayfa Boyutu (KB)";
$locale['t201'] = "Resimsiz";
$locale['t202'] = "Resimlerle Birlikte";

$locale['t300'] = "Kodlama";
$locale['t301'] = "W3C Doctype";

$locale['t310'] = "Yorumlar";
$locale['t311'] = "Yorumların Sayısı";
$locale['t312'] = "Yorumların Uygunluğu";

```

\$locale['t320'] = "Sayfa Başlığı";
 \$locale['t321'] = "Sayfa Başlığı";
 \$locale['t322'] = "Sayfa Başlığı Uzunluğu (Karakter)";
 \$locale['t323'] = "Sayfa Başlığının Uygunluğu";

\$locale['t330'] = "Doğru Etiketleme";
 \$locale['t331'] = "TIDY Hata Sayısı";
 \$locale['t332'] = "TIDY Uyarı Sayısı";

\$locale['t340'] = "Body Etiketi";
 \$locale['t341'] = "Body Etiketi";
 \$locale['t342'] = "Text Niteliği";
 \$locale['t343'] = "Link Niteliği";
 \$locale['t344'] = "Alink Niteliği";
 \$locale['t345'] = "Vlink Niteliği";

\$locale['t350'] = "Formlar";
 \$locale['t351'] = "HTML Form Sayısı";
 \$locale['t352'] = "Object/Embed Sayısı";
 \$locale['t353'] = "HTML Form Kullanımı(%)";
 \$locale['t354'] = "Label Etiketleri (%)";
 \$locale['t355'] = "Tabindex Nitelikleri (%)";

\$locale['g100'] = "Genel";
 \$locale['g101'] = "Ana Menü";
 \$locale['g102'] = "Bağlantı Sayısı";

\$locale['g200'] = "Logolar";
 \$locale['g201'] = "Logo";
 \$locale['g202'] = "Metinlerin Okunabilirliği";
 \$locale['g203'] = "GIF Formatı";
 \$locale['g204'] = "Statik Logolar";

\$locale['g300'] = "Sayfada Gezinme";
 \$locale['g310'] = "Bağlantılar";
 \$locale['g311'] = "Resim İçermeyen (%)";
 \$locale['g312'] = "Resimlerdeki Alt Niteliği (%)";

\$locale['g320'] = "Menüler";
 \$locale['g321'] = "HTML Menü";

\$locale['g330'] = "Tabindex Niteliklerinin Kullanımı (%)";
 \$locale['g331'] = "Bağlantılarda";
 \$locale['g332'] = "Resim Haritası Alanlarında";

\$locale['g340'] = "Accesskey Niteliklerinin Kullanımı (%)";
 \$locale['g341'] = "Bağlantılarda";
 \$locale['g342'] = "Resim Haritası Alanlarında";
 \$locale['g343'] = "Düğmelerde";
 \$locale['g344'] = "Input Nesnelerinde";
 \$locale['g345'] = "Textarea Nesnelerinde";
 \$locale['g346'] = "Select Nesnelerinde";

\$locale['g350'] = "Resim Haritaları";
 \$locale['g351'] = "Harita Sayısı";
 \$locale['g352'] = "Kullanım Uygunluğu";
 \$locale['g353'] = "Alan Sayısı";
 \$locale['g354'] = "Alanlardaki Alt Niteliğinin Kullanımı (%)";

```

$locale['g400'] = "Metinler";
$locale['g401'] = "Font Etiketleri";
$locale['g402'] = "Statik Yazılar";

$locale['g410'] = "Yazı Tipi Büyüklükleri (%)";
$locale['g411'] = "Harici-Dahili CSS";
$locale['g412'] = "Satırda CSS";

$locale['g500'] = "Resimler";
$locale['g501'] = "Resim Sayısı (İndirilen/Toplam)";
$locale['g502'] = "GIF/JPEG Formatı (%)";
$locale['g503'] = "Resim Boyutu - mak. 30K (%)";
$locale['g504'] = "Width Niteliklerinin Kullanımı (%)";
$locale['g505'] = "Height Niteliklerinin Kullanımı (%)";

$locale['g510'] = "Alt Nitelikleri (%)";
$locale['g511'] = "Kullanım";
$locale['g512'] = "Uzunluk Uygunluğu";
$locale['g513'] = "Son Karakterin Uygunluğu";
$locale['g514'] = "Son Karakterlerin Uygunluğu";

$locale['g600'] = "216 Renk Paletinin Kullanımı (%)";
$locale['g601'] = "Harici-Dahili CSS";
$locale['g602'] = "Satırda CSS";
$locale['g603'] = "HTML Niteliklerinde";

$locale['g700'] = "Tablolar";
$locale['g701'] = "Tablo Sayısı";
$locale['g702'] = "Tablo Sonu Etiketleri (%)";
$locale['g703'] = "Arka Plan Rengi Kullanılmaması (%)";
$locale['g704'] = "Arka Plan Resmi Kullanılmaması (%)";

?>

```

EK C.4.2 İngilizce Dil Dosyası (/locale_res/en.php)

```
<?php
```

```

$locale['e100'] = "Web Page Evaluation Tool - Erkan DURAN";
$locale['e101'] = "M. Sc., Thesis, Web Page Evaluation, Tool, 2007/4, Circular, W3C, Web Standards, Accessibility, e-Government, Erkan DURAN, Taner TANRISEVER, Balıkesir University";
$locale['e102'] = "THE DEVELOPMENT OF A TOOL AIMED AT <br>THE EVALUATION OF THE PUBLIC INSTITUTIONS WEBSITES <br>UNDER THE CIRCULAR 2007/4 <p>&nbsp;</p>";
$locale['e103'] = "Erkan DURAN<br>Balıkesir University, Institute of Science<br>Department of Computer Education and Instructional Technology<p>&nbsp;</p>";
$locale['e104'] = "(M.Sc. Thesis / Supervisor: Assist. Prof. Dr. Taner TANRISEVER)<p>&nbsp;</p>";
$locale['e105'] = "Balıkesir-Turkey, 2008<hr>";

$locale['e110'] = "Security Code";
$locale['e111'] = "Web Page";
$locale['e112'] = "W3C HTML Test";
$locale['e113'] = "W3C CSS Test";

```

```

$locale['e114'] = "OK";
$locale['e115'] = "For correct results, you should write web page's full address instead of the
redirected address and test the web pages without frame.";
$locale['e116'] = "Total Visit.";
$locale['e117'] = "Invalid Security Code";

$locale['adm100'] = "[Delete this site]";
$locale['adm101'] = "[Delete old sites]";
$locale['adm102'] = "[Delete temp files/folders]";
$locale['adm103'] = "Are you sure?";
$locale['adm104'] = "[Sites]";
$locale['adm105'] = "Site";
$locale['adm106'] = "Time";
$locale['adm107'] = "Evaluated by";
$locale['adm108'] = "Page";

$locale['r100'] = "Evaluated Total Web Site.";
$locale['r101'] = "Included in Average (N).";

$locale['r110'] = "Evaluation Results";
$locale['r111'] = "Content (Minumum)";
$locale['r112'] = "Design (Accessilibity)";
$locale['r113'] = "Design (Technical Properties)";
$locale['r114'] = "General View";
$locale['r115'] = "[Exit]";

$locale['r120'] = "Result";
$locale['r121'] = "Value";
$locale['r122'] = "Show";
$locale['r123'] = "G. Average";
$locale['r124'] = "(n)";

$locale['r130'] = "<img src='images/view.gif alt='View. ' width='16' height='16'>";
$locale['r131'] = SUCCESSFUL. " Successful";
$locale['r132'] = FAILED. " Failed";
$locale['r133'] = N_A. " N/A ";
$locale['r134'] = NULL_X. " Null";
$locale['r135'] = POSSIBLE. " Possible";
$locale['r136'] = NOT_POSSIBLE. " Not Possible";

$locale['c100'] = "General";
$locale['c101'] = "Institution Logo";
$locale['c102'] = "Institution Name";
$locale['c103'] = "Purpose of the site";
$locale['c104'] = "Links in site";
$locale['c105'] = "Announcements/News";
$locale['c106'] = "Search Mechanism";
$locale['c107'] = "Link to Help Pages";
$locale['c108'] = "Link to FAQ Pages";
$locale['c109'] = "Link to Complaint/Feedback Pages";

$locale['c200'] = "Contact Information";
$locale['c201'] = "Address";
$locale['c202'] = "Phone";
$locale['c203'] = "E-mail";
$locale['c204'] = "Appropriate of the e-mail";

$locale['a100'] = "Browser Type";
$locale['a101'] = "Internet Explorer";

```

```

$locale['a102'] = "Mozilla Firefox";
$locale['a103'] = "Konqueror";
$locale['a104'] = "Lynx";

$locale['a200'] = "Display Resolution";
$locale['a201'] = "1280x1024";
$locale['a202'] = "1024x768";
$locale['a203'] = "800x600";

$locale['t100'] = "General";
$locale['t101'] = "W3C HTML Test (Error Count)";
$locale['t102'] = "W3C CSS Test (Error Count)";
$locale['t103'] = "TIDY Accessibility Test (Warning Count)";
$locale['t104'] = "Level 1";
$locale['t105'] = "Level 2";
$locale['t106'] = "Level 3";

$locale['t110'] = "Javascrpts";
$locale['t111'] = "Javascript/Noscript (Count)";
$locale['t112'] = "Usage of Javascrpts";

$locale['t120'] = "Java Applets (Count)";
$locale['t121'] = "Meta Tags (Count)";
$locale['t122'] = "Cookies";

$locale['t200'] = "Page Size (KB)";
$locale['t201'] = "Without Image";
$locale['t202'] = "With Images";

$locale['t300'] = "Coding";
$locale['t301'] = "W3C Doctype";

$locale['t310'] = "Comments";
$locale['t311'] = "Comment Count";
$locale['t312'] = "Appropriate of The Comments";

$locale['t320'] = "Page Title";
$locale['t321'] = "Page Title";
$locale['t322'] = "Length of The Page Title (Character)";
$locale['t323'] = "Appropriate of The Page Title";

$locale['t330'] = "Correct Tagging";
$locale['t331'] = "TIDY Error Count";
$locale['t332'] = "TIDY Warning Count";

$locale['t340'] = "Body Tag";
$locale['t341'] = "Body Tag";
$locale['t342'] = "Text Attribute";
$locale['t343'] = "Link Attribute";
$locale['t344'] = "Alink Attribute";
$locale['t345'] = "Vlink Attribute";

$locale['t350'] = "Forms";
$locale['t351'] = "HTML Form Count";
$locale['t352'] = "Object/Embed Count";
$locale['t353'] = "HTML Form Usage (%)";
$locale['t354'] = "Label Tags (%)";
$locale['t355'] = "Tabindex Attributes (%)";
$locale['g100'] = "General";

```



```

$locale['g101'] = "Main Menu";
$locale['g102'] = "Link Count";

$locale['g200'] = "Logos";
$locale['g201'] = "Logo";
$locale['g202'] = "Readability of Texts";
$locale['g203'] = "GIF Format";
$locale['g204'] = "Static Logos";

$locale['g300'] = "Moving on Page";

$locale['g310'] = "Links";
$locale['g311'] = "Without Image (%)";
$locale['g312'] = "Alt Attribute of Images (%)";

$locale['g320'] = "Menus";
$locale['g321'] = "HTML Menu";

$locale['g330'] = "Tabindex Attributes Usage (%)";
$locale['g331'] = "Links";
$locale['g332'] = "Areas of Image Map";

$locale['g340'] = "Accesskey Attributes Usage(%)";
$locale['g341'] = "Links";
$locale['g342'] = "Areas of Image Map";
$locale['g343'] = "Buttons";
$locale['g344'] = "Inputs";
$locale['g345'] = "Textareas";
$locale['g346'] = "Selects";

$locale['g350'] = "Image Maps";
$locale['g351'] = "Map Count";
$locale['g352'] = "Appropriate of Usage";
$locale['g353'] = "Area Count";
$locale['g354'] = "Usage of Alt Attribute of Areas (%)";

$locale['g400'] = "Texts";
$locale['g401'] = "Font Tags";
$locale['g402'] = "Static Texts";

$locale['g410'] = "Font Size (%)";
$locale['g411'] = "External&Internal CSS";
$locale['g412'] = "Inline CSS";

$locale['g500'] = "Images";
$locale['g501'] = "Image Count (Downloaded/Total)";
$locale['g502'] = "GIF or JPEG Format (%)";
$locale['g503'] = "Size - max 30K (%)";
$locale['g504'] = "Width Attributes Usage (%)";
$locale['g505'] = "Height Attributes Usage (%)";

$locale['g510'] = "Alt Attributes (%)";
$locale['g511'] = "Usage";
$locale['g512'] = "Appropriate of Length";
$locale['g513'] = "Appropriate of Last Character";
$locale['g514'] = "Appropriate of Last Characters";

$locale['g600'] = "216 Colors Palette Usage (%)";
$locale['g601'] = "External&Internal CSS";

```

```

$locale['g602'] = "Inline CSS";
$locale['g603'] = "HTML Attributes";

$locale['g700'] = "Tables";
$locale['g701'] = "Table Count";
$locale['g702'] = "End Tags (%)";
$locale['g703'] = "No Usage of Bgcolor Attribute (%)";
$locale['g704'] = "No Usage of Background Attribute (%)";

?>

```

EK C.5 Kriterlerle İlgili Dil Dosyaları

EK C.5.1 Türkçe Dil Dosyası (/locale_cri/tr.php)

```

<?php

$locale['ic101'] = "Ana sayfada kurum logosu bulunmalıdır.";
$locale['ic102'] = "Ana sayfada kurumun açık adı belirtilmelidir.";
$locale['ic103'] = "Sitenin amacı ana sayfada yer almalıdır.";
$locale['ic104'] = "Ana sayfada sitenin tüm temel bölümlerine bağlantılar bulunmalıdır.";
$locale['ic105'] = "Kuruma ait duyurular/haberler ana sayfada yer almalıdır.";
$locale['ic106'] = "Ana sayfada arama mekanizması bulunmalıdır.";
$locale['ic107'] = "Ana sayfada yardım sayfalarına bağlantı bulunmalıdır.";
$locale['ic108'] = "Ana sayfada sıkça sorulan sorular sayfasına bağlantı bulunmalıdır.";
$locale['ic109'] = "Ana sayfada şikayet/geri bildirim sayfalarına bağlantılar bulunmalıdır.";

$locale['ic201'] = "Kuruma ait erişim bilgileri; telefon, adres, e-posta ana sayfada yer almalıdır.";
$locale['ic202'] = "Kuruma ait erişim bilgileri; telefon, adres, e-posta ana sayfada yer almalıdır.";
$locale['ic203'] = "Kuruma ait erişim bilgileri; telefon, adres, e-posta ana sayfada yer almalıdır.";
$locale['ic204'] = "E-posta adreslerinde kişisel adresler yerine göreve dayalı adresler tercih edilmelidir.<br>Örnek: bilgi@kamukurumu.gov.tr";

$locale['ia101'] = "Web sayfaları çeşitli tarayıcılarda rahatça görüntülenebilecek şekilde tasarlanmalıdır. Web sayfaları tek bir tarayıcı tipi için tasarlanmamalıdır.";
$locale['ia102'] = "Web sayfaları çeşitli tarayıcılarda rahatça görüntülenebilecek şekilde tasarlanmalıdır. Web sayfaları tek bir tarayıcı tipi için tasarlanmamalıdır.";
$locale['ia103'] = "Web sayfaları çeşitli tarayıcılarda rahatça görüntülenebilecek şekilde tasarlanmalıdır. Web sayfaları tek bir tarayıcı tipi için tasarlanmamalıdır.";
$locale['ia104'] = "Web sayfaları çeşitli tarayıcılarda rahatça görüntülenebilecek şekilde tasarlanmalıdır. Web sayfaları tek bir tarayıcı tipi için tasarlanmamalıdır.";

$locale['ia201'] = "Web sayfaları farklı ekran çözünürlüklerinde sorunsuzca görünülenebilecek şekilde tasarlanmalıdır.";
$locale['ia202'] = "Web sayfaları farklı ekran çözünürlüklerinde sorunsuzca görünülenebilecek şekilde tasarlanmalıdır.";
$locale['ia203'] = "Web sayfaları farklı ekran çözünürlüklerinde sorunsuzca görünülenebilecek şekilde tasarlanmalıdır.";

$locale['it101'] = "Web sayfaları, HTML kullanımı açısından W3C'nin HTML geçerlilik testinden geçebilmelidir.";
$locale['it102'] = "Web sayfaları, CSS kullanımı açısından W3C'nin CSS geçerlilik testinden geçebilmelidir.";

```

\$locale['it103'] = "TIDY ile 3 farklı seviye için yapılan erişebilirlik testi sonuçları...";

\$locale['it110'] = "Web sayfalarında script kullanımı çok gerekli durumlarda sınırlandırılmalıdır. Script kullanılması durumunda sayfanın metin tabanlı alternatif sürümü var olmalıdır.";

\$locale['it120'] = "Web sayfalarında applet'ler kullanılmamalıdır.";

\$locale['it121'] = "Web sayfalarının arama motorlarına kaydını kolaylaştırmak için 'meta tag'lar web sayfalarında kullanılmalıdır.";

\$locale['it122'] = "Web sayfalarında mecbur kalmadıkça çerez kullanılmamalıdır. Oturum kontrolü yapılması gerektiğinde çerez kullanımı yerine, olabildiğince sonucu tarafında oturum kontrolü tercih edilmelidir.";

\$locale['it202'] = "Ana sayfa dosya boyutu tüm resimler, scriptler dahil olmak üzere en fazla 50 KB olmalıdır. Diğer bilgi sayfalarının dosya boyutu en fazla 120 KB olmalıdır.";

\$locale['it301'] = "Web sayfaları, doküman tip bildirimini mutlaka barındırmalıdır.";

\$locale['it310'] = "Web sayfalarında gereğinden fazla yorum kodları kullanılmamalıdır. Yorumlarda birey ya da şirket isimleri yer almamalı ve yorumlar teşekkür etme gibi amaçlarla yazılmamalıdır.";

\$locale['it320'] = "Web sayfalarında sayfa başlığı 'title' kullanılmalıdır. Sayfa başlığı 60 karakterden uzun olmamalı ve başlık sayfanın içeriğine uygun olmalıdır.";

\$locale['it330'] = "Kodlama HTML standartlarına uygun olarak yapılmalıdır.";

\$locale['it340'] = "Web sayfalarında 'body' etiketi ve bu etiketin 'text', 'link', 'alink' ve 'vlink' nitelikleri kullanılmalıdır.";

\$locale['it353'] = "Web sayfalarındaki formlar HTML formatında hazırlanmalıdır.";

\$locale['it354'] = "Form elamanlarına ait tüm açıklama etiketleri 'label' etiketi içinde verilmeli ve alan adı ile ilişkilendirilmelidir.";

\$locale['it355'] = "Form elamanlarına ait sekme sırası kontrol edilmeli ve doğru dizilimde olmalıdır.";

\$locale['ig101'] = "Ana sayfada site içeriğine ilişkin ana menü bulunmalıdır.";

\$locale['ig202'] = "Logo üzerindeki tüm metinler kolayca okunabilmelidir.";

\$locale['ig203'] = "Logolar GIF formatında olmalıdır.";

\$locale['ig204'] = "Hareketli GIF'ler logolarda kullanılmamalıdır.";

\$locale['ig311'] = "Bağlantılar mümkün olduğunca metin tabanlı olmalıdır.";

\$locale['ig312'] = "Resim ile bağlantı verilmesi durumunda 'alt' nitelikleri mutlaka kullanılmalıdır.";

\$locale['ig321'] = "Kullanılan menüler HTML olarak tasarlanmış metin menüler olmalıdır.";

\$locale['ig330'] = "'tabindex' niteliklerinin kullanılması erişilebilirliği arttıracaktır.";

\$locale['ig340'] = "'accesskey' nitelikleri web sayfasını daha fazla erişebilir kılmaktadır.";

\$locale['ig352'] = "Resim haritası üzerinden bağlantı verme yöntemi, ancak resim üzerinde ayrılan bölümlerin verilen bağlantıları anlamlı kılması durumunda kullanılmalıdır. Örnek: Ülke haritası üzerinden bölgelere erişim gibi.";

\$locale['ig354'] = "Resim haritası üzerindeki tüm alanlar için 'alt' nitelikleri kullanılmalıdır.";

\$locale['ig401'] = "Yazı tipleri 'font' etiketleri yerine CSS içinde kullanılmalıdır.";

\$locale['ig402'] = "Web sayfalarında kayan yazılar kullanılmamalıdır.";

\$locale['ig410'] = "Yazı tipi büyüklüklerinde xx-small, x-small, small, medium, large, x-large, xx-large gibi değerler kullanılmalıdır.";

\$locale['ig502'] = "Web sayfasındaki resimler GIF/JPEG formatında olmalıdır.";

\$locale['ig503'] = "Resim boyutu 30 KB'yi geçmemelidir.";

\$locale['ig504'] = "Resimlerde 'width' nitelikleri kullanılmalıdır.";

```

$locale['ig505'] = "Resimlerde 'height' nitelikleri kullanılmalıdır.";

$locale['ig511'] = "Resimlerde 'alt' nitelikleri kullanılmalıdır.";
$locale['ig512'] = "Resimlerde 'alt' nitelik değeri en fazla 100 karakter olmalıdır.";
$locale['ig513'] = "Resimlerde 'alt' nitelik değeri son karakteri nokta veya boşluk olmalıdır.";
$locale['ig514'] = "Resimlerde 'alt' nitelik değeri son karakterleri nokta ve boşluk olmalıdır.";

$locale['ig601'] = "Bağlantılar ve metinler için 216 renkten oluşan renk paleti kullanılmalıdır.";
$locale['ig602'] = "Bağlantılar ve metinler için 216 renkten oluşan renk paleti kullanılmalıdır.";
$locale['ig603'] = "Bağlantılar ve metinler için 216 renkten oluşan renk paleti kullanılmalıdır.";

$locale['ig702'] = "Tablolarda, tablo sonu etiketleri her zaman kullanılmalıdır.";
$locale['ig703'] = "Tablolarda, tablo arka plan rengi kullanılmamalıdır.";
$locale['ig704'] = "Tablolarda, tablo arka plan resmi kullanılmamalıdır.";

?>

```

EK C.5.2 İngilizce Dil Dosyası (/locale_cri/en.php)

```

<?php

$locale['ic101'] = "Main page should include the institution logo.";
$locale['ic102'] = "Main page should include the institution name.";
$locale['ic103'] = "Website's purpose should be shown at website's main page.";
$locale['ic104'] = "Main page should include main links of the website.";
$locale['ic105'] = "Main page should include news/announcements about the institution.";
$locale['ic106'] = "Main page should include search mechanism.";
$locale['ic107'] = "Main page should include link for help pages.";
$locale['ic108'] = "Main page should include link for faqs pages.";
$locale['ic109'] = "Main page should include links for complaint/feedback pages.";

$locale['ic201'] = "Main page should include communication info of the institution (Telephone, address, e-mail etc).";
$locale['ic202'] = "Main page should include communication info of the institution (Telephone, address, e-mail etc).";
$locale['ic203'] = "Main page should include communication info of the institution (Telephone, address, e-mail etc).";
$locale['ic204'] = "General addresses should be preferred instead of personal addresses for e-mail addresses. <br>Example: info@...gov.tr etc.";

$locale['ia101'] = "Web pages should be designed to view by different web browsers as same form. Web pages should not be designed to view for only one kind of web browsers.";
$locale['ia102'] = "Web pages should be designed to view by different web browsers as same form. Web pages should not be designed to view for only one kind of web browsers.";
$locale['ia103'] = "Web pages should be designed to view by different web browsers as same form. Web pages should not be designed to view for only one kind of web browsers.";
$locale['ia104'] = "Web pages should be designed to view by different web browsers as same form. Web pages should not be designed to view for only one kind of web browsers.";

$locale['ia201'] = "Web pages should be designed to view without problems at different display resolution.";
$locale['ia202'] = "Web pages should be designed to view without problems at different display resolution.";
$locale['ia203'] = "Web pages should be designed to view without problems at different display resolution.";

```

\$locale['it101'] = "Web pages should be passed the W3C HTML validation test. ";
 \$locale['it102'] = "Web pages should be passed the W3C CSS validation test. . ";
 \$locale['it103'] = "TIDY accessibility test result for level 1, level 2 and level 3. ";
 \$locale['it110'] = "Script using should be limited for very necessary situation.
In the situation of script using, there should be a text based alternative version of the web pages.";

\$locale['it120'] = "Java Applets should not be used at web pages.";
 \$locale['it121'] = "Metatags should be used to index easily by the search engines. ";
 \$locale['it122'] = "Cookies should not be used at web pages. When the need of session control, session control of server side should be preferred as possible as instead of using cookies.";

\$locale['it202'] = "Main page size should be max 50 KB along with images and scripts.
 Other information pages' size should be max 120 KB. ";

\$locale['it301'] = "Doctype declaration should be done at web pages.";
 \$locale['it310'] = "At web pages, comment codes should not be used more than necessary. In comments, person or company names should not included. The comments should not be written for thanks.";
 \$locale['it320'] = "Page title should be used at web pages. Page title length should be no more than 60 chars and related to the content of web page.";

\$locale['it330'] = "Coding should be made according to HTML standards.";

\$locale['it340'] = "'Body tag and this tag's 'text', 'link', 'alink' and 'vlink' attributes should be used at web pages.";

\$locale['it353'] = "The forms at web pages should be prepared in HTML format.";
 \$locale['it354'] = " All form element descriptions should be used with label tag. The label tags should be related to form elements.";
 \$locale['it355'] = "Tab indexes sequence of all form element should be checked and this sequence should be correct.";

\$locale['ig101'] = "Main page should contain main menu aimed at web site content.";

\$locale['ig202'] = "All texts on logos should be read easily.";
 \$locale['ig203'] = "Logos should be at GIF format.";
 \$locale['ig204'] = "Animated gifs should not be used for logos. ";

\$locale['ig311'] = "Links should be text based format as far as possible.";
 \$locale['ig312'] = "When links contain image, alt attributes should be used.";

\$locale['ig321'] = "Menus should be text menu based HTML.";

\$locale['ig330'] = "Using tabindex increases the accessibility.";

\$locale['ig340'] = "Using accesskey attributes increases the accessibility.";

\$locale['ig352'] = "Image maps should be used when the areas are part of a whole. For example, the image map is a country and the areas are it's cities.";
 \$locale['ig354'] = "The 'alt' attribute should be used for all the areas of image map.";

\$locale['ig401'] = "Font style, size etc should be declared in CSS instead of font tag ";
 \$locale['ig402'] = "Static texts should be used at web pages.";

\$locale['ig410'] = " Font size values should be xx-small, x-small, small, medium, large, x-large, xx-large.";

\$locale['ig502'] = "Images should be GIF/JPEG format. ";
 \$locale['ig503'] = "Image size should be 30 KB max.";

```

$locale['ig504'] = "Width attribute should be used in img tags.";
$locale['ig505'] = "Height attribute should be used in img tags.";

$locale['ig511'] = "Alt attribute should be used in img tags.";
$locale['ig512'] = "Alt attribute value should be 100 chars max.";
$locale['ig513'] = "The last char of alt attribute value should be a dot/space.";
$locale['ig514'] = "The last chars of alt attribute value should be a dot and space.";

$locale['ig601'] = "Color palette with 216 colors should be used for links, texts and images.";
$locale['ig602'] = "Color palette with 216 colors should be used for links, texts and images.";
$locale['ig603'] = "Color palette with 216 colors should be used for links, texts and images.";

$locale['ig702'] = "Table end tags should be used.";
$locale['ig703'] = "Table background colors should not be used.";
$locale['ig704'] = "Table background images should not be used.";

?>

```

EK C.6 Stil Sayfası Kodu (style.css)

```

body {
font-family:'Trebuchet MS', arial, verdana, tahoma, sans-serif;
}

a,a:visited,a:active {
color:#000;
text-decoration:none;
}

a.act:visited {
color:#90c;
text-decoration:none;
font-weight:700;
}

a:hover {
color:#90c;
}

.info {
text-decoration:none;
}

p {
padding-top:5px;
clear:both;
}

h1 {
font-size:x-large;
}

```

```
h2 {
font-size:large;
padding-top:5px;
clear:both;
}

h3 {
font-size:medium;
font-weight:100;
color:#90c;
}

img {
border:0;
}

.ana {
width:630px;
margin:auto;
}

.ana .hd1 {
width:370px;
height:26px;
float:left;
border-top:1px solid #03f;
border-bottom:1px solid #03f;
background:#fcf;
text-align:left;
padding-left:1px;
padding-top:3px;
font-style:italic;
font-weight:700;
}

.ana .hd2 {
width:125px;
height:26px;
float:left;
border-top:1px solid #03f;
border-bottom:1px solid #03f;
background:#fcf;
text-align:left;
padding-left:1px;
padding-top:3px;
font-style:italic;
}

.ana .d1 {
width:370px;
height:26px;
float:left;
border-bottom:1px solid #03f;
text-align:left;
padding-left:1px;
padding-top:3px;
}
```

```
.ana .d2 {  
width:125px;  
height:26px;  
float:left;  
border-bottom:1px solid #03f;  
text-align:left;  
padding-left:1px;  
padding-top:3px;  
}
```

```
.fra {  
width:800px;  
margin:auto;  
}
```

```
.fra .head1 {  
width:299px;  
height:26px;  
float:left;  
border-top:1px solid #03f;  
border-bottom:1px solid #03f;  
background:#fcf;  
text-align:left;  
padding-left:1px;  
padding-top:3px;  
font-style:italic;  
font-weight:700;  
}
```

```
.fra .head2 {  
width:90px;  
height:26px;  
float:left;  
border-top:1px solid #03f;  
border-bottom:1px solid #03f;  
background:#fcf;  
text-align:center;  
padding-left:1px;  
padding-top:3px;  
font-style:italic;  
}
```

```
.fra .head3 {  
width:120px;  
height:26px;  
float:left;  
border-top:1px solid #03f;  
border-bottom:1px solid #03f;  
background:#fcf;  
text-align:center;  
padding-left:1px;  
padding-top:3px;  
font-style:italic;  
}
```

```
.fra .head4 {  
width:95px;  
height:26px;  
float:left;
```



```
border-top: 1px solid #03f;
border-bottom: 1px solid #03f;
background: #fcf;
text-align: left;
padding-left: 1px;
padding-top: 3px;
font-style: italic;
}
```

```
.fra .head5 {
width: 100px;
height: 26px;
float: left;
border-top: 1px solid #03f;
border-bottom: 1px solid #03f;
background: #fcf;
text-align: center;
padding-left: 1px;
padding-top: 3px;
font-style: italic;
font-weight: 700;
}
```

```
.fra .head6 {
width: 90px;
height: 26px;
float: left;
border-top: 1px solid #03f;
border-bottom: 1px solid #03f;
background: #fcf;
text-align: center;
padding-left: 1px;
padding-top: 3px;
font-style: italic;
font-weight: 700;
}
```

```
.fra .t1 {
width: 299px;
height: 26px;
float: left;
border-bottom: 1px solid #03f;
text-align: left;
padding-left: 1px;
padding-top: 3px;
}
```

```
.fra .t2 {
width: 90px;
height: 26px;
float: left;
border-bottom: 1px solid #03f;
text-align: center;
padding-left: 1px;
padding-top: 3px;
}
```

```
.fra .t3 {
width:120px;
height:26px;
float:left;
border-bottom:1px solid #03f;
text-align:center;
padding-left:1px;
padding-top:3px;
}
```

```
.fra .t4 {
width:95px;
height:26px;
float:left;
border-bottom:1px solid #03f;
text-align:center;
padding-left:1px;
padding-top:3px;
}
```

```
.fra .t5 {
width:100px;
height:26px;
float:left;
border-bottom:1px solid #03f;
background:#ffc;
text-align:center;
padding-left:1px;
padding-top:3px;
}
```

```
.fra .t6 {
width:90px;
height:26px;
float:left;
border-bottom:1px solid #03f;
text-align:center;
font-style:italic;
padding-left:1px;
padding-top:3px;
}
```

```
.fra .det {
float:left;
border:1px solid #03f;
width:800px;
text-align:left;
padding-left:1px;
}
```

```
.fra .suml {
width:300px;
float:left;
padding-left:1px;
padding-top:3px;
text-align:left;
background:#ffc;
}
```

```
.fra .sumr {
width:490px;
float:right;
padding-left:1px;
padding-top:3px;
text-align:left;
}

.fra .expl {
float:right;
border-top:1px solid #03f;
width:125px;
text-align:left;
padding-left:3px;
padding-top:5px;
}

.fra .post {
width:800px;
height:26px;
float:left;
border-bottom:1px solid #03f;
text-align:center;
padding-left:1px;
padding-top:3px;
}

.fraent {
width:377px;
}

.fraent .ent1 {
width:150px;
height:26px;
float:left;
border-bottom:1px solid #03f;
text-align:left;
padding-left:1px;
padding-top:3px;
}

.fraent .ent2 {
width:225px;
height:26px;
float:left;
border-bottom:1px solid #03f;
text-align:left;
padding-left:1px;
padding-top:3px;
}

.t1 .sub1head {
font-style:italic;
padding-left:7px;
}
```

```
.t1 .sub2head {
font-style:italic;
padding-left:22px;
}

.t1 .sub3head {
font-style:italic;
padding-left:37px;
}

.t1 .sub1 {
padding-left:7px;
}
.t1 .sub2 {
padding-left:22px;
}

.t1 .sub3 {
padding-left:37px;
}

.t1 .sub4 {
padding-left:52px;
}

.textbox {
height:20px;
background-color:#ff9;
border:1px solid;
}

hr {
height:1px;
border:1px solid #03f;
}
```

EK C.7 Ekran Görüntüleri ve Özellikleri

EK C.7.1 Giriş Ekranı

“<http://webtest.balikesir.edu.tr>” adresinde yer alan aracın giriş ekranı şu şekildedir:

Türkçe | English

KAMU KURUM VE KURULUŞLARI İNTERNET SİTELERİNİN
2007/4 SAYILI GENELGE KAPSAMINDA DEĞERLENDİRİLMESİNE YÖNELİK
BİR ARAÇ GELİŞTİRİLMESİ

Erkan DURAN
Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü
Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı

(Yüksek Lisans Tezi / Tez Danışmanı: Yrd. Doç. Dr. Taner TANRISEVER)

Balıkesir, 2008

fb633

| | |
|--------------------------------------|--------------------------------------|
| Güvenlik Kodu | <input type="text" value=""/> |
| Web Sayfası | <input type="text" value="http://"/> |
| W3C HTML Testi | <input checked="" type="checkbox"/> |
| W3C CSS Testi | <input checked="" type="checkbox"/> |
| <input type="button" value="Tamam"/> | |

Doğru sonuçlar için yönlendirilen sayfa adresleri yerine sayfanın tam adresi giriniz ve çerçeve içermeyen sayfaları test ediniz.

Toplam Ziyaret: 400

Şekil C.1 Web Sayfası Değerlendirme Aracı Giriş Ekranı

Şekil C.1'den de görüldüğü gibi değerlendirilecek web sayfası için aracın giriş ekranında bir form hazırlanmıştır. Bu formdaki alanların doldurulmasıyla değerlendirme işlemi başlatılabilmektedir. Öte yandan giriş ekranının sağ üst köşesindeki dil seçeneği, hem ara yüz dilini değiştirmekte hem de içerik ile ilgili kriterler için değerlendirme dilini belirlemektedir. Bu nedenle içerik ile ilgili kriterlerde sağlıklı sonuçların alınması için doğru dilin seçilmesi gerekmektedir.

EK C.7.2 Değerlendirme Sonuçları Ekranı

Aracın değerlendirme işlemlerini gerçekleştirmesinin ardından değerlendirme sonuçları kullanıcıya sunulmaktadır. Örnek ekran görüntüsü şu şekildedir:

Değerlendirilen Toplam Site: 428 **1** all all OK
Ortalamaya Dahil Edilen (N): 36 **2** 3 4

http://[redacted].edu.tr Değerlendirme Sonuçları
5
İçerik (Asgari) | Tasarım (Erişebilirlik) | **Tasarım (Teknik Özellikler)** | Genel Görünüm | [Çıkış]

| Genel 6 | Sonuç 7 | Değer 8 | Görüntüle 9 | G. Ortalama 10 | (n) 11 |
|--|-----------------------|-----------------------|---------------------------|------------------------------|----------------------|
| W3C HTML Testi (Hata Sayısı) | ✘ | 4 | | 142.14 | 36 |
| W3C CSS Testi (Hata Sayısı) | ✔ | 0 | | 10.89 | 36 |
| <i>TIDY Erişebilirlik Testi (Uyarı Sayısı)</i> | | | | | |
| Seviye 1 | ✘ | 4 | 🔍 | 304.44 | 36 |
| Seviye 2 | ✘ | 11 | 🔍 | 463.08 | 36 |
| Seviye 3 | ✘ | 13 | 🔍 | 504.31 | 36 |
| <i>Javascript'ler</i> | | | | | |
| Javascript/Noscript (Adet) | | 0/0 | | 7.22/0.33 | 36/36 |
| Script Kullanımı | ■ | | | 25 (%) | 32 |
| Java Applet'ler (Adet) | ✔ | 0 | | 0 | 36 |
| Meta Tag'lar(Adet) | ✔ | 4 | 🔍 | 4.86 | 36 |
| Çerezler | ✘ | | | 36 (%) | 36 |







Şekil C.2 Değerlendirme Sonuçları Örnek Ekran Görüntüsü

Şekil C.2’de değerlendirilmiş bir web sayfasının “Tasarım (Teknik Özellikler)” ana başlığının “Genel” alt başlığı altındaki kriterlere ilişkin değerlendirme sonuçları yer almaktadır. Verilen bu örnek ekran görüntüsü diğer ana başlık ve alt başlıklarda da aynı şekildedir. Numaralandırılmış olan alanlar ve bu alanların açıklamaları kısaca şu şekildedir:

1. *Değerlendirilen toplam site sayısı*
2. *Ortalamaya dahil edilen site sayısı*: Değerlendirilen tüm sitelerden en yakın tarihte değerlendirilmiş olanların sayısı
3. *Alan adı uzantısı filtreleme seçeneği (gov, edu vb)*: Ortalamaya dahil edilen siteleri seçime göre filtreler. Böylece seçilen alan adı uzantısına sahip değerlendirilmiş toplam site sayısı ve bu sitelerin kriterlere ilişkin genel ortalamaları görülebilir.
4. *Ülke kodu filtreleme seçeneği*: Seçilen ülke koduna göre ortalama dahil edilen siteleri filtreler. Alan adı uzantısı filtreleme seçeneğine benzer şekilde çalışır.
5. *Değerlendirme sonuçları ana başlıkları*: İçerik (Asgari), Tasarım (Erişilebilirlik), Tasarım (Teknik Özellikler) ve Genel Görünüm.
6. *Ana başlık alt başlığı ve kriterleri*
7. *Kriter değerlendirme sonucu*: Kriterin özelliğine göre “Başarılı”, “Başarısız”, “Olabilir”, “Olmayabilir”, “Yok/Değerlendirme Dışı” veya “Belirlenemedi” olmak üzere 6 çeşittir (bkz. Şekil C.3).
8. *Kritere ilişkin değer*: Kriterin özelliğine göre boş veya yüzdesel de olabilir.
9. *Değerlendirmeye ilişkin veriler*: Web sayfasında kriterle ve/veya değerlendirme sonucu ile ilişkili verilerin olup olmadığını bildirir. Belirtilen simgenin tıklanmasıyla bu verilere ulaşılabilir.

10. *Genel Ortalama*: Ortalamaya dahil edilen sitelerin ilgili kritere ilişkin genel ortalamaları. Yüzdesele bir deęer de olabilir.
11. *(n)*: Genel ortalaması hesaplanan kayıt sayısı

Şekil C.3’de deęerlendirme sonuç simgeleri ve bu simgelerin anlamları yer almaktadır. Bunlar, deęerlendirme sonuçlarının bitiminde her sayfanın altında verilmiştir.














| | | |
|---|---|--|
|  Başarılı |  Olabilir |  Belirlenemedi |
|  Başarısız |  Olmayabilir |  Yok/Deę. dıőı |

Şekil C.3 Deęerlendirme Sonuçları Simgeleri

Öte yandan deęerlendirme kriterlerinin kısa açıklamalarına kriterlere verilmiş olan bağlantıların tıklanmasıyla ulaşılabilmektedir. Örneęin Şekil C.2’de görülen “W3C HTML Testi (Hata Sayısı)”na tıkladığında, ana başlıkların hemen altına “Web sayfaları, HTML kullanımı açısından W3C’nin HTML geçerlilik testinden geçebilmelidir” şeklinde kısa bir açıklama gelecektir.

EK C.7.3 Yönetici Modu Ekranı

Web sayfası deęerlendirmesi yönetici olarak yapıldığında Şekil C.4’ten de görüldüğü gibi araç üzerinden daha ayrıntılı işlemler yapılabilmektedir.

| Deęerlendirilen Toplam Site: 430 | <input type="text" value="1"/> | <input type="text" value="2"/> | all | all | <input type="text" value="3"/> | all | <input type="text" value="4"/> | OK |
|--|---|--------------------------------|-----------------------------------|---|--------------------------------|-----|--------------------------------|----|
| Ortalamaya Dahil Edilen (N): 36 | [Bu siteyi sil] | [Eski siteleri sil] | [Geçici dosyaları/klasörleri sil] | | | | | |
| http://...org.tr Deęerlendirme Sonuçları | | | | | | | | |
| [Siteler] İçerik (Asgari) Tasarım (Erişebilirlik) Tasarım (Teknik Özellikler) Genel Görünüm [Çıkış] | | | | | | | | |
| Genel | Sonuç | Deęer | Görüntüle | G. Ortalama | (n) | | | |
| Kurum Logosu |  | <input type="text" value="1"/> | 9 |  | 92 (%) | 36 | | |
| Kurumun Açık Adı |  | <input type="text" value="1"/> | |  | 100 (%) | 36 | | |
| Sitenin Amacı |  | <input type="text" value="0"/> | | | 3 (%) | 36 | | |
| Siteiçi Bağlantılar |  | <input type="text" value="1"/> | |  | 100 (%) | 36 | | |
| Duyurular/Haberler |  | <input type="text" value="1"/> | |  | 100 (%) | 36 | | |
| Arama Mekanizması |  | <input type="text" value="1"/> | | | 64 (%) | 36 | | |
| Yardım Sayfalarına Bağlantı |  | <input type="text" value="0"/> | | | 3 (%) | 36 | | |
| S.S.S. Sayfalarına Bağlantı |  | <input type="text" value="0"/> | | | 19 (%) | 36 | | |
| Şikayet/Geri Bildirim Say. Bağ. |  | <input type="text" value="1"/> | | | 83 (%) | 36 | | |

Şekil C.4 Yönetici Modu Ekran Görüntüsü

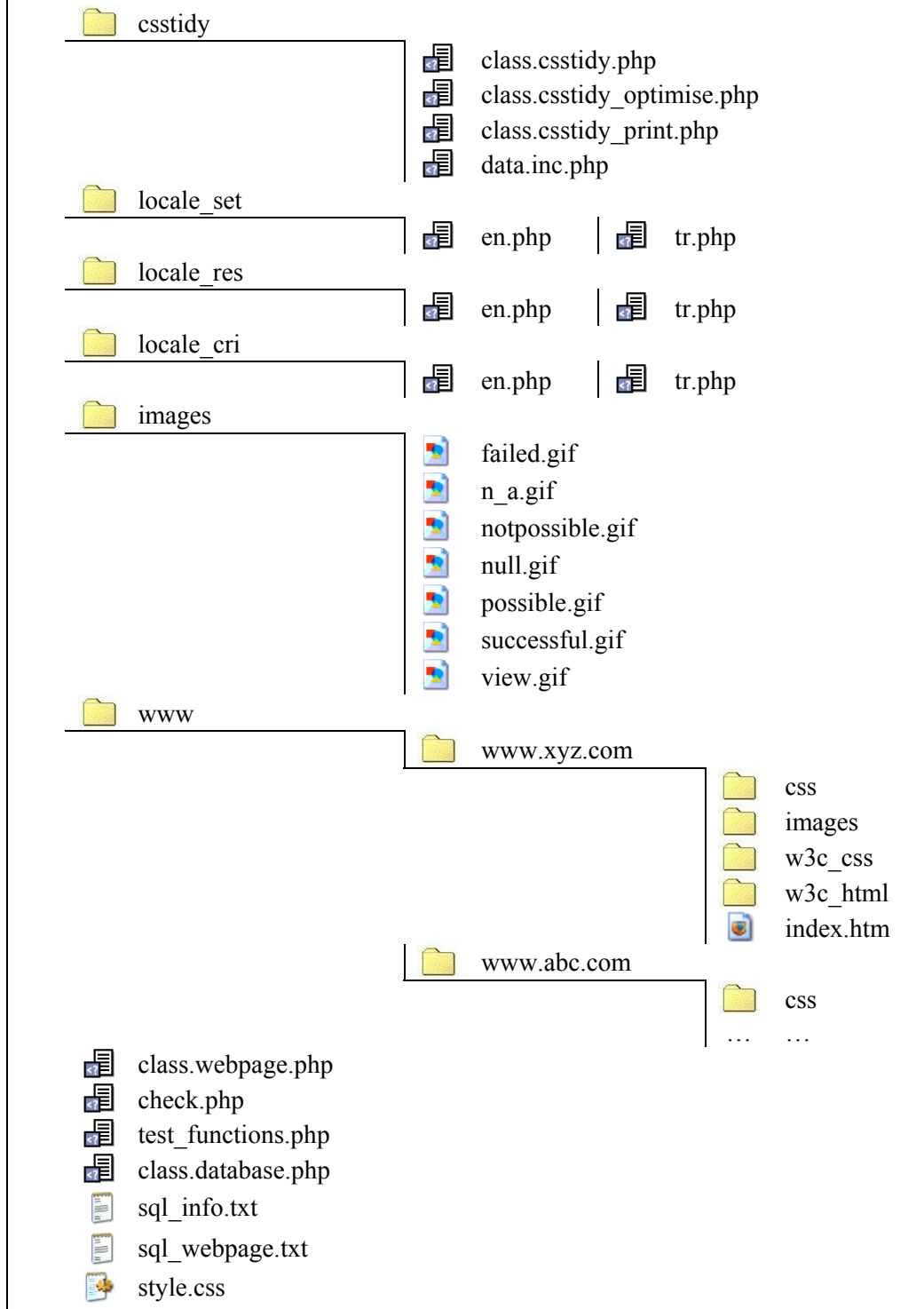
Şekil C.4'teki numaralandırılmış olan alanlar ve bu alanların açıklamaları kısaca şu şekildedir:

1. *Subdomain filtreleme alanı*
2. *Domain filtreleme alanı*
3. *Path filtreleme alanı*

Kullanıcılar alan adı uzantısına ve ülke koduna göre filtreleme işlemleri yaparak çeşitli genel ortalamaları görebilmekte iken, yöneticiler 1., 2., ve 3. alanları da kullanarak daha ayrıntılı filtreleme işlemleri yapabilirler.

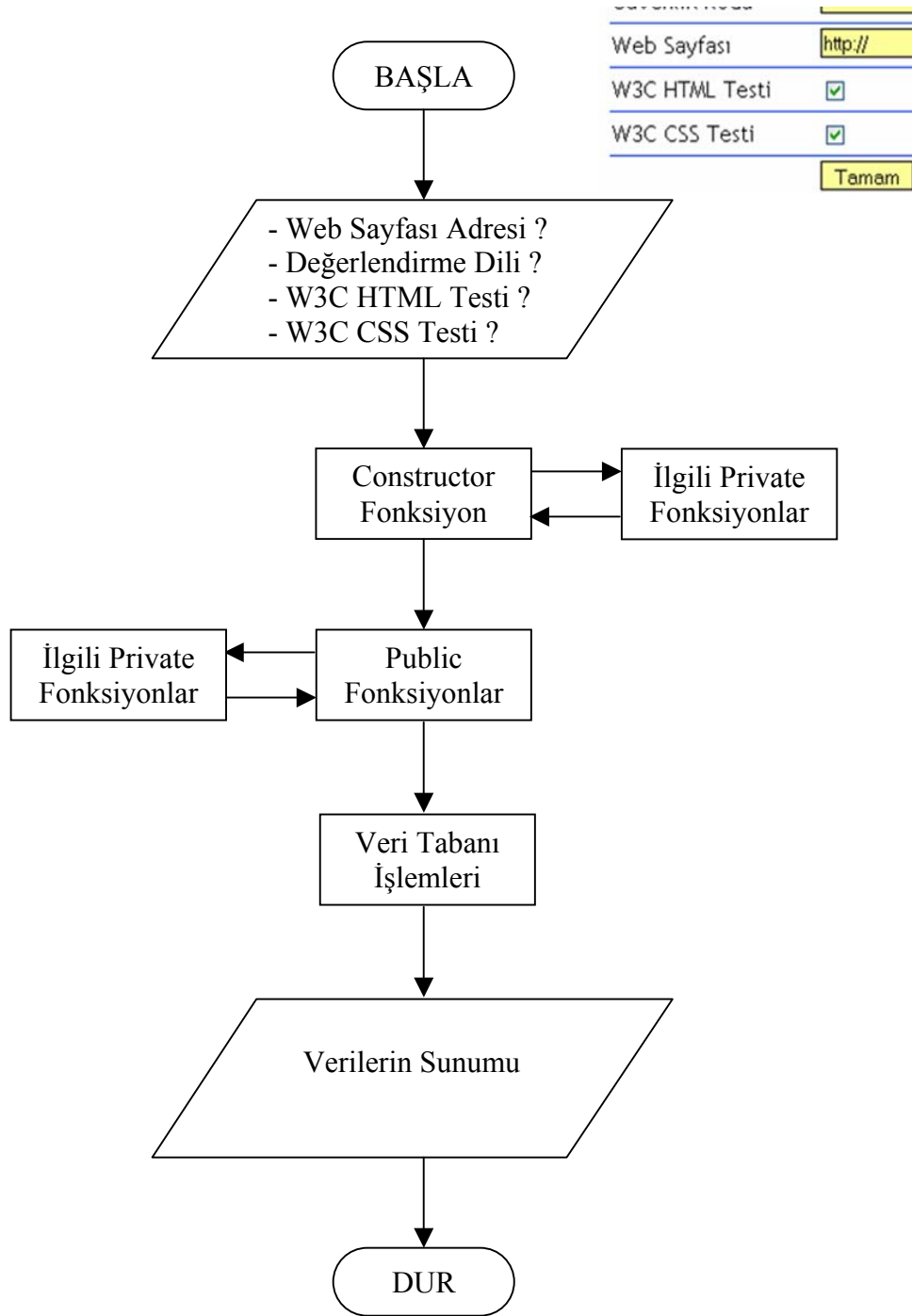
4. *Değerlendirmeyi kimin yaptığı:* Yönetici veya kullanıcılar tarafından değerlendirilen siteleri filtrelemek için kullanılır.
5. *Bu siteyi sil:* Değerlendirilme sonuçları gösterilen siteyi siler.
6. *Eski siteleri sil:* Ortalamaya dahil edilmeyen siteleri siler.
7. *Geçici dosyaları/klasörleri sil:* Değerlendirme işlemleri sırasında sitelerle ilgili kaydedilen dosya ve klasörlerin tamamını siler.
8. *Siteler:* Ortalamaya dahil edilen sitelerin listesini görüntüler. Bu listede; değerlendirilen sitelerin adı, hangi tarihte ve kimin tarafından (yönetici veya kullanıcı) değerlendirildiği ile ilgili bilgiler yer almaktadır. Daha önceden değerlendirilen sitelerin sonuçlarını görüntülemek için bu liste kullanılmaktadır.
9. *El yordamı sonuçları için giriş alanları:* El yordamı gerektiren kriterler için değerlendirme yapıldıktan sonra belirlenen sonucun aracın veritabanına girilmesine olanak sağlar.

EK D ARACIN DİZİN VE DOSYA YAPISI



Şekil D.1 Aracın Dizin ve Dosya Yapısı

EK E ARACIN AKIŞ DİYAGRAMI



Şekil E.1 Aracın Akış Diyagramı

EK F DEĞERLENDİRİLEN WEB SAYFALARI

Tablo F.1 Değerlendirilen Web Sayfaları

| Kurum/Kuruluş | Web Sayfası |
|-------------------------------------|--|
| Milli Eğitim Bakanlığı | www.meb.gov.tr |
| İçişleri Bakanlığı | w3.icisleri.gov.tr |
| Kültür ve Turizm Bakanlığı | www.kultur.gov.tr/TR |
| Çalışma ve Sos. Güv. Bakanlığı | www.calisma.gov.tr |
| Bayındırlık ve İskan Bakanlığı | www.bayindirlik.gov.tr/turkce/index.php |
| Balıkesir Üniversitesi | www.balikesir.edu.tr |
| Boğaziçi Üniversitesi | www.bogazici.edu.tr |
| Ortadoğu Teknik Üniversitesi | www.odtu.edu.tr |
| Ege Üniversitesi | www.ege.edu.tr |
| Çanakkale Onsekiz Mart Üniv. | www.comu.edu.tr |
| Deniz Kuvvetleri Komutanlığı | www.dzkk.tsk.mil.tr/TURKCE/anasayfa.asp |
| Hava Kuvvetleri Komutanlığı | www.hvkk.tsk.mil.tr |
| Harp Akademileri Komutanlığı | www.harpak.tsk.mil.tr |
| Dz. K.K.Envanter Kont Mrk. Kom. | www.ekm.tsk.mil.tr |
| MEBS Okulu ve Eğt. Mrk. Kom. | www.mebsookulu.kkk.tsk.mil.tr |
| Aydın Emniyet Müdürlüğü | www.aydin.pol.tr |
| Yalova Emniyet Müdürlüğü | www.yalova.pol.tr/yalovapolis |
| Bursa Emniyet Müdürlüğü | www.bursa.pol.tr |
| Bolu Emniyet Müdürlüğü | www.bolu.pol.tr |
| Edirne Emniyet Müdürlüğü | www.edirne.pol.tr |
| İzmir Büyükşehir Belediyesi | www.izmir.bel.tr |
| Ankara Büyükşehir Belediyesi | www.ankara.bel.tr/TasarimOgeleri/default.aspx |
| Rize Belediyesi | www.rize.bel.tr |
| Kuşadası Belediyesi | www.kusadasi.bel.tr/belediye/default.asp |
| Bodrum Belediyesi | www.bodrum.bel.tr |
| Radyo ve Televizyon Üst Kurulu | www.rtuk.org.tr/sayfalar/default.aspx |
| Bankacılık Düz. ve Denet. Kurumu | www.bddk.org.tr |
| Türkiye Kızılay Derneği | www.kizilay.org.tr/yeni/index.php |
| Türkiye Satranç Federasyonu | www.tsf.org.tr |
| Türk Standartları Enstitüsü | www.tse.org.tr |
| Ankara Fen Lisesi | www.ankarafenlisesi.k12.tr |
| Eskişehir Anadolu Lisesi | www.eskisehiranadolulisesi.k12.tr |
| Üsküdar Çağrıbey Anadolu Lisesi | www.cagribey.k12.tr |
| Balıkesir Ziraat Bankası Fen Lisesi | www.zbfl.k12.tr/portal |
| İzmir Anadolu Öğretmen Lisesi | www.izmiraol.k12.tr/website |

KAYNAKLAR

- [1] Gazi BİTEM Web Sitesi, “21. Yüzyılda Öncelikli AR-GE Alanları ve Stratejik Teknolojiler Çalışma Grubunun Raporu”, <<http://www.bitem.gazi.edu.tr/yirmibir.htm>>, Erişim Tarihi: 02.05.2008
- [2] Ergün, M. ve Ergün, E., “Web Sitelerinin Çeşitli Özellikleri ve Eğitim Kurumları Web Sitelerine Yansımaları”, *Kuramsal Eğitim Bilim Dergisi*, 1, 1, (2008), 2
- [3] T.C. Sayıştay Başkanlığı, “E-Devlete Geçişte Kamu Kurumları İnternet Siteleri Performans Denetim Raporu”, Ankara, (2006)
- [4] Yanık, L., “Değişimin Rotası e-Türkiye”, (18.03.2002), <http://www.edevlet.net/eTurkiye/Degisimin_Rotasi_eTurkiye.pdf>, Erişim Tarihi: 02.05.2008
- [5] T.C. Başbakanlık, 2003/12 Sayılı Başbakanlık Genelgesi, Ankara, (2003)
- [6] ODTÜ Alan Adı Yönetimi Web Sitesi, “Yıllık İstatistikler”, (May. 2008), <<https://www.nic.tr/index.php?USRACTN=YEARSTAT&PHPSESSID=121087309488246152107632178>>, Erişim Tarihi: 05.05.2008
- [7] Gençer, M., “Türkiye İnternet Sitelerinde Standartlara Uyumluluk: Karşılaştırmalı Ölçümler ve Stratejik Açılımlar”, XI. Türkiye’de İnternet Konferansı, Ankara, 22.12.2006
- [8] Devlet Planlama Teşkilatı Müsteşarlığı, e-Dönüşüm Türkiye Projesi 2003-2004 KDEP Uygulama Sonuçları ve 2005 Eylem Planı, 2005
- [9] T.C. Başbakanlık, 2007/4 Sayılı Başbakanlık Genelgesi, Ankara, (2007)
- [10] Türkiye Bilimsel ve Teknolojik Araştırma Kurumu-Marmara Araştırma Merkezi, Kamu Kurumları İnternet Sitesi Kılavuzu, (2006).
- [11] Johansson, R., “Accessibility myths and misconceptions”, (25.05.2005), <http://www.456bereastreet.com/archive/200505/accessibility_myths_and_misconceptions/>, Erişim Tarihi: 05.05.2008

[12] Johansson, R., “Evaluating website accessibility part 1, Background and Preparation”, (06.03.2006),
<http://www.456bereastreet.com/archive/200603/evaluating_website_accessibility_part_1_background_and_preparation/>, Eriřim Tarihi: 05.05.2008

[13] Nyman, R., “What is accessibility?”, (01.03.2006),
<<http://www.robertnyman.com/2006/03/01/what-is-accessibility/>>, Eriřim Tarihi: 05.05.2008

[14] Hayriođlu, F., “CSS ile Eriřilebilir Formlar Yapmak - I”, (25.02.2007),
<<http://www.fatihhayrioglu.com/?p=276>>, Eriřim Tarihi: 15.12.2007

[15] Yıldırım, C., “Web tarayıcıları”, (Ekim 2003),
<<http://cism.odtu.edu.tr/2003-9/browsers.php>>, Eriřim Tarihi: 06.05.2008

[16] Awio Web Services LLC., “Global Web Stats”, (Nisan 2008),
<<http://www.w3counter.com/globalstats.php>>, Eriřim Tarihi: 24.04.2008

[17] W3Schools Web Sitesi, “Browser Statistics”, (Mart 2008),
<http://www.w3schools.com/browsers/browsers_stats.asp>, Eriřim Tarihi: 14.03.2008

[18] W3Schools Web Sitesi, “OS Platform Statistics”, (Mart 2008),
<http://www.w3schools.com/browsers/browsers_os.asp>, Eriřim Tarihi: 14.03.2008

[19] Indiana University Information Technology Services, “Why does my web page look different when I use different computers and web browsers?”, (27.11.2007), <<http://kb.iu.edu/data/agaa.html>>, Eriřim Tarihi: 10.05.2008

[20] Küçüktunç, O., “Web Tasarımı Rehberi #3”, (16.04.2006),
<<http://www.acikbilgi.com/2006/04/16/web-tasarimi-rehberi-3/>>, Eriřim Tarihi: 16.05.2008

[21] Yılmaz, C., “Web Sitesi Düzenlemesi ve Tasarımı”, (09.02.2000),
<http://www.belgeler.org/howto/web-organizasyonu_uyum.html>, Eriřim Tarihi: 16.05.2008

[22] Dođan, M., “Biliřimi Kavrayamayan Biliřimci”, (17.02.2005),
<http://www.altiustutasarim.com/arsiv/2005/02/bilisimi_kavray.php>, Eriřim Tarihi: 04.01.2007

[23] Dođan, M., “Eriřilebilirlik Kimin İçin ve Ne İçin?”, (30.12.2004),
<http://www.altiustutasarim.com/arsiv/2004/12/bayindirlik_eri.php>, Eriřim Tarihi: 04.01.2007

- [24] Türkiye Sakatlar Derneği Web Sitesi, “Web İçeriği Erişilebilirlik Kılavuzu”, <<http://www.tsd.org.tr/content/view/668/2997/>>, Erişim Tarihi: 03.03.2008
- [25] ULAKBİM Web Sitesi, “HTML Düzenleyicileri”, <<http://www.ulakbim.gov.tr/dokumanlar/webtasarim/htmlduzenleyiciler.uhtml>>, Erişim Tarihi: 03.03.2008
- [26] World Health Organization, “Disability and Rehabilitation”, <<http://www.who.int/disabilities/en/index.html>>, Erişim Tarihi: 15.05.2008
- [27] Devlet İstatistik Enstitüsü ve Özürlüler İdaresi Başkanlığı, “2002 Türkiye Özürlüler Araştırması”, 2002
- [28] Doğan, M., “Engelli Siteler”, (13.01.2005), <http://www.altiustutasarim.com/arsiv/2005/01/engelli_siteler.php>, Erişim Tarihi: 05.01.2007
- [29] Chip Kitap, A’dan Z’ye Web Tasarımı, VogelBurda, İstanbul, (2006).
- [30] Doğan, M., “Web Standartları 1”, (14.12.2004), <http://www.altiustutasarim.com/arsiv/2004/12/web_standartlar.php>, Erişim Tarihi: 05.01.2007
- [31] Doğan, M., “Web Standartları 2”, (16.12.2004), <http://www.altiustutasarim.com/arsiv/2004/12/web_standartlar_1.php>, Erişim Tarihi: 05.01.2007
- [32] Alkış, Z. ve Şişman, A., “E-Devlet Ana Giriş Kapısı ve Alt Giriş Kapıları”, 10. Türkiye Harita Bilimsel ve Teknik Kurultayı, 2005
- [33] Durmaz, R., “e-Devlet”, *GLOKAL ODTÜ İşletme Topluluğu Dergisi*, 14, (2003), 84
- [34] Türkiye Bilişim Şurası, “e-Devlet Çalışma Grubu Raporu”, 2002
- [35] Alır, G., Soydal, İ. ve Öztürk, Ö., “Türkiye’de E-devlet Uygulamaları Kapsamında Kamu Kurumlarına Ait Web Sayfalarının Değerlendirilmesi”, Değişen Dünyada Bilgi Yönetimi Sempozyumu, Hacettepe Üniversitesi-Ankara, 2007
- [36] Çağıltay, K. ve Akıncı, D., “E-devlet Web Sitelerini Kullanmak ya da Kullanmamak: Vatandaş Açısından Kullanılabilirlik Sorunları ve Öneriler”, 21. Ulusal Bilişim Kurultayı, Ankara, 2004

[37] New Zealand Government, “New Zealand Government Web Guidelines”, (13.02.2004), <<http://www.e.govt.nz/standards/web-guidelines/web-guidelines-v-2-1/web-guidelines.pdf>>, Erişim Tarihi: 18.02.2007

[38] Avustralian Government Information Management Office, “Web Publishing Guide”, “<http://webpublishing.agimo.gov.au/>”, Erişim Tarihi: 20.02.2007

[39] UK e-Government Unit, “Illustrated Handbook for Web Management Teams- Guidelines for Government Websites”, (05.04.2007), <<http://archive.cabinetoffice.gov.uk/e-government/resources/handbook/html/htmlindex.asp>>, Erişim Tarihi: 11.05.2008

[40] Access to Electronic Legal Information Committee, “General Website Evaluation Criteria”, (May 2005), <<http://www.aallnet.org/committee/aelic/criteria.html>>, Erişim Tarihi: 09.05.2008

[41] Henriksson, A., Yi, Y., Frost, B. and Middleton, M., “Evaluation instrument for e-government websites”, (September 2006), <<http://eprints.qut.edu.au/archive/00003113/01/3113.pdf>>, Erişim Tarihi: 10.02.2007

[42] Choudrie, J., Ghinea, G. and Weerakkody, V., “Evaluating Global e-Government Sites: A View using Web Diagnostic Tools”, *Electronic Journal of e-Government*, 2, 2, (2004), 105

[43] Ellison, J., “Assessing the accessibility of fifty United States government Web pages: Using Bobby to check on Uncle Sam”, *First Monday Peer-Reviewed Journal on The Internet*, 9, 7, (2004)

[44] Kaiser, S., “Web Site Accessibility”, (15.11.2007), <<http://websitetips.com/accessibility/tools/>>, Erişim Tarihi: 05.05.2008

[45] The World Wide Web Consortium, “About The W3C Markup Validation Service”, <<http://validator.w3.org/about.html>>, Erişim Tarihi: 28.01.2007

[46] The World Wide Web Consortium, “About The CSS Validator”, <<http://jigsaw.w3.org/css-validator/about.html>>, Erişim Tarihi: 28.01.2007

[47] Vision Australia Web Sitesi, “Toolbar Functions”, <<http://www.visionaustralia.org.au/info.aspx?page=619>>, Erişim Tarihi: 03.03.2007

[48] Pederick, C., “Web Developer History”, (31.01.2007), <<http://chrispederick.com/work/web-developer/history/>>, Erişim Tarihi: 03.03.2007

[49] Johansson, R., “Evaluating website accessibility part 2, Basic Checkpoints”, (16.03.2006),
<http://www.456bereastreet.com/archive/200603/evaluating_website_accessibility_part_2_basic_checkpoints/>, Eriřim Tarihi: 08.05.2008

[50] Karabulut, M., “PHP ve Nesneye Yönelik Programlama”, (16.04.2003),
<http://www.ceviz.net/php-ve-nesneye-yonelik-programlama_s1_a77.html>, Eriřim Tarihi: 10.12.2006

[51] Akçaya, V., Eđitimciler İin Yeni Bir Web Aracı, Yüksek Lisans Tezi, Balıkesir Üniversitesi Fen Bilimleri Enstitüsü, BÖTE Ana Bilim Dalı, Balıkesir, (2006).

[52] PHP.Net Sitesi, “Tidy Introduction”, (30.05.2008),
<<http://tr2.php.net/manual/en/intro.tidy.php>>, Eriřim Tarihi: 27.04.2007

[53] Lyubinskiy, I.S., “HTML Parser”, <<http://www.php-development.ru/abandonware/html-parser.txt>>, Eriřim Tarihi: 13.04.2007

[54] Schmitz, F., “CSSTidy Introduction”, (2005),
<<http://csstidy.sourceforge.net/>>, Eriřim Tarihi: 28.04.2007

[55] Ceviz.net Sitesi, “PHP 5 - Yeni Nesne Modeli I, Private and Protected Üyeler”, (16.11.2003), <http://www.ceviz.net/php-5-yeni-nesne-modeli-i-private-and-protected-uyeler_a254.html>, Eriřim Tarihi: 10.12.2006

[56] Kyrmin, J., “Using @import in Cascading Style Sheets”,
<<http://webdesign.about.com/cs/css/qt/tipcssatimport.htm>>, Eriřim Tarihi: 14.10.2007

[57] W3Schools Web Sitesi, “Multiple Style Sheets”,
<http://www.w3schools.com/css/css_howto.asp>, Eriřim Tarihi: 14.10.2007

[58] W3Schools Web Sitesi, “Cross-browsers Color Values”,
<http://www.w3schools.com/html/html_colors.asp>, Eriřim Tarihi: 13.08.2007

[59] Wagener, S., “database.class.php”,
<<http://www.phpclasses.org/browse/file/2989.html>>, Eriřim Tarihi: 14.03.2008

[60] Hayriođlu, F., “Dođru DOCTYPE Kullanımı”, (15.09.2006),
<<http://www.fatihhayrioglu.com/?p=164>>, Eriřim Tarihi: 19.05.2007