

WCAG 2.2 DÜZEY AA KRİTERLERİ

Dr. Engin Yılmaz

Engin.yilmaz@thy.com

İçindekiler

Giriş	4
Uluslararası Yasal Mevzuatta WCAG Kriterlerinin Yeri	4
Düzey AA kriterleri	7
1.2.4 Canlı Altyazılar	7
1.2.5 Sesli Betimleme (Önceden Kaydedilmiş)	7
1.3.4 Ekran Yönelimi	8
1.3.5 Girdi Amacını Belirleme	9
1.4.3 Karşıtlık (Minimum)	10
1.4.4 Metni Yeniden Boyutlandırma	12
1.4.5 Metin olarak Görüntü	13
1.4.10Yeniden akış (reflow)	13
1.4.11 Metin dışı Karşıtlık	15
1.4.12 Metin aralığı	16
1.4.13 Odak veya Üzerine Gelindiğinde İçerik (Content on Hover or Focus)	17
2.4.5 Birden Fazla Yol	19
2.4.6 Başlıklar ve Etiketler	20
2.4.7 Odak Görünür	20
2.4.11 Odağın Gizlenmemesi	21
2.5.7 Sürükleme Hareketleri (Dragging Movements)	22
2.5.8 Hedef Boyutu Minimum	23
3.1.2 Bölüm Dili	25
3.2.3 Tutarlı Gezinme	26
3.2.4 Tutarlı Tanımlama	26
3.3.3 Hata Önerisi	27
3.3.4 Hata Önleme (Yasal, finansal, veri)	27
3.3.8 Erişilebilir Kimlik Doğrulama (Asgari)	29

4.1.3 Durum Mesajları

30

Sonuç

32

Giriş

20 Haziran 2025 Cumhurbaşkanlığı genelgesiyle Türkiye'deki kamu ve özel sektör kuruluşlarının web siteleri ve mobil uygulamalarına WCAG 2.2 A düzeyinde erişilebilirlik zorunluluğu getirilmesini takiben, 31 düzey A kriterini detaylı biçimde açıklayarak örnekleriyle anlatan ilk yazıyı yayınlamıştık. Elinizdeki bu yazı bu seriyi devam ettirmek amacıyla uluslararası alanda çok daha yaygın biçimde kabul gören WCAG düzey AA kriterlerini de aynı şekilde inceleyen serinin ikinci parçası. Yazının ilk bölümünde uluslararası mevzuatta WCAG kriterlerinin yeri ve kabul görme sürecine değinildi. İkinci bölümde ise 24 adet AA düzeyindeki erişilebilirlik kriteri dikkatlice ele alındı. Her kriter için iyi ve kötü örneklerin yanında, bu kriterin neden ve hangi engel grubu için önemli olduğu açıklandı. Ardından da kriterin yerine getirilebilmesi için yararlı olabilecek olası teknik çözüm önerilerinde bulunuldu. Serinin birinci yazısında olduğu gibi, her kriter açıklanırken hem en basit düzeydeki kullanıcının hem de teknik açıdan donanımlı geliştirici ve tasarımcıların anlayabilecekleri bir dil kullanımına dikkat edildi. Geliştirici, tasarımcı ve kullanıcılar için yararlı olması umuduyla.

Uluslararası Yasal Mevzuatta WCAG Kriterlerinin Yeri

Bilindiği üzere, ülkemizde 20 Haziran 2025 cumhurbaşkanlığı genelgesiyle kamu kurumları, devlet ve vakıf üniversiteleri, kamu ve özel tüm bankalar, özel hastaneler, ulaşım hizmeti sunan özel kuruluşlar, seyahat acenteleri, internet servis sağlayıcıları ve Telekom operatörleriyle e-ticaret sitelerine WCAG 2.2 Düzey A kriterlerine göre erişilebilir olma zorunluluğu getirilmiştir. Bu genelge daha önceki yazımızda ele aldığımız 31 ilkeyi kapsamaktadır. Öte yandan dünyadaki örneklerle baktığımızda yasal mevzuatın genellikle Düzey AA kriterlerini zorunluluk haline getirdiğini görüyoruz. Yazımızın bu bölümünde Dünyadaki yasal mevzuata kısaca göz atacağız.

Avrupa birliğine baktığımızda web erişilebilirlik alanında ilk kabul edilen yönerge EU 2016/2102 olmuştur. Bu yönerge, kamu sektörü web siteleri ve mobil uygulamaları için WCAG kriterlerini yasal bir zorunluluk haline getirmiştir. Buna göre Eylül 2018 sonrası yayınlanan tüm kamu web sitelerinin WCAG kriterlerine göre erişilebilir

olması, 2020 eylül itibarıyla da eski kamu sektörü web sitelerinin erişilebilir olması bir zorunluluk haline gelmiştir. Haziran 2021 sonrasında da tüm mobil uygulamaların WCAG 2.1 Düzey AA kriterlerine göre erişilebilir olması şart koşulmuştur. Bu yönerge için teknik standart olarak Avrupa standardı EN 301 549 belirlenmiştir ve bu standart da WCAG 2.1 Düzey AA kriterlerini içermektedir.

Kamu sektörü dışındaki dijital erişilebilirlik içinse 2019 yılında Avrupa Erişilebilirlik Yasası (European Accessibility Act) kabul edilerek üye devletlere 2022 yılına dek bunu mevzuatlarına dahil etmeleri istenmiştir. 28 Haziran 2025 tarihinde yürürlüğe giren bu yasayla birçok tüketici hizmetinde dijital erişilebilirlik WCAG 2.1 düzey AA kriterlerine göre zorunlu hale gelmiştir. Söz konusu alanlara e-ticaret hizmetleri, çevrimiçi bankacılık hizmetleri, , görsel işitsel medya hizmetleri, toplu taşıma bilgilendirme hizmetleri, e-kitap hizmetleri ve elektronik haberleşme hizmetleri dahildir. Yaptırım olarak uyarı ve düzeltme süreci ardından para cezalılarının da gündeme geleceği belirtilmektedir.

Birleşik Krallık'ta da AB'ye paralel olarak 2018 yılında kamu sektörü web sitesi ve mobil uygulamalar yönetmeliğiyle tüm merkezi ve yerel kamu siteleri ve uygulamalarının WCAG 2.1 düzey AA düzeyinde erişilebilir olması yasal bir zorunluk haline getirilmiştir. Yasa 2020 itibarıyla tüm web sitelerinin 2021 itibarıyla da mobil uygulamaların erişilebilir olmasını şart koşturmuştur. 2023 yılında WCAG 2.2 sürümünün yayınlanmasıyla beraber, hükümet bundan böyle WCAG 2.2 AA düzeyini esas alacağını belirtmiştir. Özel sektör içinse, 2010 yılında çıkan eşitlik yasasının hükümleri uygulanarak erişilebilir olmayan dijital sistemlerle ilgili davalar açılabilir.

ABD'ye baktığımızda, Web erişilebilirliği genel olarak Amerika Engelliler Yasası (ADA) kapsamında da bir erişilebilirlik meselesi olarak ele alınmış, özgün yasa da doğrudan Web erişilebilirliği ifadesi geçmese de adalet bakanlığı ve mahkemeler web siteleri ve mobil uygulamaların fiziksel mekanlar gibi erişilebilir olması gerektiğini pek çok kez dile getirmiştir. O nedenle sitelerin erişilebilirlik düzenlemeleri yapmamaları dava konusu olmuştur. Bu davalarda WCAG 2.0 veya 2.1 AA düzeyi kriter olarak kabul görmüştür.

Federal kamu siteleri içinse Rehabilitasyon yasasının section 508 başlıklı bölümü 2017 yılında güncellenerek Ocak 2018 sonrasında federal web sitelerinin WCAG 2.0 AA düzeyinde erişilebilir olmasını şart koşturmuştur.

ABD’de hava yolları gibi belirli sektörler için ayrı düzenlemeler de yapılmıştır. 2013 yılındaki yönetmelikle havayolu firmalarının biletleme ve bilgilendirme yapan sayfalarının 2016 aralık ayına kadar WCAG 2.0 AA düzeyine göre erişilebilir olması şartı getirilmiştir. Bu konuda ağır yaptırımlar da uygulanmıştır. Örnek olarak Scandinavian Airlines şirketi web sitelerinin erişilebilir olmaması nedeniyle ulaştırma bakanlığıyla bir anlaşma yaparak 200000 \$ **ödemeyi ve WCAG 2.0 AA düzeyine göre uyumlu hale gelmeyi kabul etmiştir.**

Kanada’da ise Ontario eyaletince 2005 yılında çıkarılan Ontario Engelliler yasası (AODA) 2025 yılına kadar Ontario’yu tamamen engelsiz hale getirmeyi amaçlamaktadır. Bu yasa altında çıkarılan yönetmeliğe göre 50 ve üzerinde çalışanı olan tüm özel sektörler ve kamu kurumları web sitelerine kademeli bir takvime göre erişilebilir hale getirmek zorundadır. 2021 ocak ayıyla birlikte tüm web sitelerinin WCAG 2.0 AA uyumlu olması zorunluluğu getirilmiştir. Yaptırım olarak da Ontario engelliler yasası en somut tanımları yapan yasalardan biridir. Büyük şirketler için günlük 100000 Kanada dolarına kadar para cezaları öngörülmektedir.

Bu ülkelerin yanında Avusturalya, Güney Kore, İrlanda ve İsrail gibi ülkeler de yasal mevzuatlarında WCAG 2.0 ya da 2.1 AA düzeyini zorunluluk olarak tanımlamışlardır.

Görüldüğü gibi, uluslararası mevzuattaki eğilim WCAG Düzey AA kriterleriyle uyumlu olunması yönünde tezahür etmektedir. Ülkemizde başlangıç olarak Düzey A kriterleri ortaya konulsa da yakın gelecekte Düzey AA kriterlerinin esas alınacağı beklenmelidir. O nedenle aşağıda ele alacağımız 24 düzey AA kriterinin geliştirici ve tasarımcılarca dikkatle incelenmesi ve erişilebilirlik geliştirmelerinde bu kriterlerin de sürece dahil edilmesinin sağlanması önemlidir.

Düzey AA kriterleri

Aşağıda verilen her kriterin başındaki numaralar o başarı kriterinin WCAG içerisindeki numarasına karşılık gelmektedir.

1.2.4 Canlı Altyazılar

Uzun süredir beklediğiniz bir basın açıklamasının bağlantısını gördünüz ve hemen tıkladınız. Ama ortam çok kalabalık ve gürültülü olduğu için sesleri duyamıyorsunuz. Maalesef bir altyazı yok. Basın açıklamasından bihaber kaldınız. Canlı altyazılar kuralı getirilirken elbette öncelikle sağır ve az duyabilen bireylerin ihtiyaçları düşünülmüş. Ancak yukarıda belirttiğimiz senaryo aslında her an yaşanması muhtemel bir durum. Çoğu kişi bir kafede, ya da kalabalık ortamda seslere değil ekrandaki altyazılara bakarak bir şeyleri takip eder. O yüzden özellikle önemli bir basın açıklaması, haber yayını veya internetten yapılan bir derse canlı altyazı eklenmesi WCAG'nin düzey AA kriterlerinden birisi. Bildiğiniz gibi Düzey A kriteri önceden kaydedilmiş medya içeriklerine altyazı koyma zorunluğu getiriyordu. Düzey AA bu süreci bir adım ileri taşıyarak canlı medya içeriklerine de altyazı koyma zorunluğu getiriyor.

Teknik açıdan Cart adı verilen İletişim Erişimi gerçek zamanlı çeviri servisleri gerek toplantılar gerek medya içerikleri için altyazı servisleri sunmaktadır. Bu servislerde eğitimli bir altyazı operatörü hem konuşmaları hem de olası ses efektlerini yazılı halde kullanıcıya sunar. Otomatik altyazı servisleri de bulunmakla birlikte buralardaki olası hataların fazla olma ihtimaline karşı dikkatli olunmalıdır. Özellikle çok dilli bir konuşma içerisinde otomatik altyazıların hataları daha da fazla olabilmektedir. Web sayfalarında track> etiketiyle eşzamanlı metin akışları sağlanabilmektedir.

1.2.5 Sesli Betimleme (Önceden Kaydedilmiş)

Web sayfasında çok merak ettiğiniz bir tanıtım filmine tıkladınız ve tek duyduğunuz şey güzel bir müzik. Videoda hiçbir konuşma yok ve her şeyi kaçırdınız. Görme engelli birinin başına çok sık gelen bir durumdan bahsediyoruz: Sesli betimleme

eksikliği. 1.2.5 düzey AA kriteri önceden kaydedilmiş medya içeriklerine ayrı bir eş zamanlı sesle sesli betimleme eklenmesini zorunlu kılıyor. Sesli betimleme herhangi bir medya içeriğindeki görsel öğelerin dış bir ses tarafından sözel olarak açıklanması anlamına geliyor. Bu görsel öğeler konuşma veya ses efektleriyle anlaşılamayacak karakterlerin hareketleri, sahne ve zaman değişimleri, ekrandaki metinler gibi noktaları kapsıyor. Bilindiği gibi Düzey A kriterlerinde de önceden kaydedilmiş videolar için bir betimleme zorunluluğu bulunuyordu; ama orada bu betimlemenin ayrı bir metin dosyası olarak sağlanması yeterli kabul edilmekteydi. Bu kriter ise betimlemenin medya içeriğiyle birlikte eşzamanlı olarak sunulan bir dış ses kaydıyla yapılmasını zorunlu hale getiriyor. Yani video oynarken, uygun noktalardaki bir dış sesin devreye girerek betimleme yapması gerekiyor. Zaman zaman bu betimlemeler sırasında video duraklatılabiliyor ya da uygun boşluklara sığacak betimlemeler yapılarak video süresi aynı bırakılıyor. Burada önemli olan kullanıcının salt göremediği için içerikteki önemli detayları kaçırmaması ve herkesle aynı anda takibinin sağlanması.

Özellikle eğitim, ya da belgesel, dizi ve filmlerin yayınlandığı platformların bu erişilebilirliği sağlaması görme engellilerin erişimi açısından büyük önem taşıyor.

Teknik açıdan videolar için ikinci bir ses parçası hazırlamak ve betimlemeleri buraya koymak mümkün. HTML 5 ortamında <track kind="descriptions">ile betimlemeler sunulabiliyor. Ya da desteklemiyorsa ayrı bir betimlemeli versiyonu oluşturmanız da mümkün. Her ne kadar Youtube tarzı platformlarda sesli betimlemenin ayrı bir parça olarak videolara eklenmesinin mümkün olduğu belirtilse de bu yazının hazırlandığı sırada her kullanıcı için bu özelliği kullanmanın pek de mümkün olmadığını tespit ettik. Betimlemenin daha da yaygınlaşması için tüm sosyal medya ve video yükleme platformlarının betimleme yüklenmesini teknik olarak daha kolay bir süreç taşımasını umuyoruz.

1.3.4 Ekran Yönelimi

Cihazınızı sadece yatay modda kullanıyorsanız ve bir oyuna girdiğinize şöyle bir uyarı çıktı: “Lütfen uygulamayı kullanabilmek için cihaz yönünü dikey yapın”. Aslında belki de tekerlekli sandalyenize monteli ve tek bir yönde kullanılıyor cihazınız. Ya da başka bir uygulamada cihazı yatay moda getirdiniz ama yazılar ve görseller dönmedi.

İşte yönelim başarı kriteri kullanıcının bir uygulamayı belirli bir ekran yöneliminde kullanmaya zorlamaması gerektiğini düzenliyor.

Pek çok kullanıcı cihazını tek bir yönde kullanır. Az önce belirttiğim gibi cihazın bir yere monteli olması, az gören birinin yazıları daha rahat okumak için cihazı yatay tutması gibi nedenlerle cihazın yönünü değiştirmeyecek kullanıcılar vardır. Elbette doğası gereği yatay olması gereken piyano uygulaması tarzı sistemler bunun istisnası. Ancak bir kullanıcı cihazı yalnızca bir yönde kullanmak zorundaysa uygulamalar onu tek bir yönde kullanıma mecbur bırakmamalıdır. Uygulamadaki kontrol ve metinler kullanılan yönelime göre otomatik ayarlanmalıdır.

Teknik açıdan CSS ile sadece tek bir yönde kaydırma zorunluluğu olmadan tasarım ayarlanabilir. Mobil tarafta da Portrait yanında Landscape seçenekleri de işaretlenerek her iki yönde otomatik kayma sağlanmalıdır. Bunların hiçbirisi mümkün olmuyorsa aynı içeriğin farklı bir yönde de kullanılabileceği alternatif bir sayfa hazırlanmalıdır.

1.3.5 Girdi Amacını Belirleme

Aslında tarayıcınıza önceden kart bilgileriniz ve iletişim bilgilerinizi kaydettiniz. Ama yeni bir alışveriş sitesinde hepsini yeniden soruyor ve daha önce kayıtlı bilgileri getirmiyor tarayıcınız. Siz de hatırlamıyorsunuz kart numaranızı. Kabahat tamamen sizde mi? Girdi amacını belirleme kriterimize göre yanıt hayır. Eğer bu alışveriş formunda istenen girdilerin amacı programatik olarak belirlenebilseydi, tarayıcınız bunu anlayacak ve ona göre size alternatif öneriler sunabilecekti. Sıkça toplanan ad, e-posta adres gibi bilgilerin istendiği alanlarda kod tarafında alanın amacı da belirtilmelidir. Bunun da esasında çok basit bir özelliği var: HTML autocomplete özelliği. Örneğin autocomplete="given-name", autocomplete="family-name", autocomplete="email" özellikleri ad, soyad ve e-posta istenen alanlara verildiğinde tarayıcı o form alanında bu bilgilerin istendiğini programatik olarak anlayacak ve buna göre dilenirse bir otomatik tamamlama kullanacaktır ya da tanıdık bir ikon gösterecektir.

Bu kriter özellikle bilişsel farklılıkları olan veya hafıza güçlüğü yaşayan kullanıcılar için önemli. Bir alanın görünen bir metin etiketi olmasının önemini önceki düzey A

kriterlerinde zaten vurguladık. Ama bu görünen etiket tek başına yeterli olmayabiliyor. Bir kişi yalnızca sokak yazan bir form alanına ne gireceğinden emin olmayabilir, ama burada Autocomplete="address" gibi bir özellik eklendiğinde tarayıcı daha önce girilen adresi otomatik olarak kişiye önerebilecektir. Bu da bilişsel güçlükler yaşayan kullanıcıların hayatını kolaylaştırır.

Teknik açıdan yukarıda da vurguladığımız üzere HTML5 autocomplete özellikleriyle form alanlarını tanımlamak, kullanıcının otomatik doldurma verisiyle buraları tek tıkla doldurmasının yolunu açmış olur.

Mobil tarafta Autofill ve Autofill apilerinin desteklenmesi önerilmektedir. Örneğin iOS tarafında UITextField için textContentType özelliği "emailAddress" olarak ayarlanırsa, iOS otomatik doldurma paneli kullanıcıya e-posta adresini önerecektir.

1.4.3 Karşıtlık (Minimum)

Girdiğiniz bir sayfanın tasarımında zeminin açık sarı yazının beyaz olduğunu hayal edin. Herhangi bir görme güçlüğü çekmeyen kullanıcı için bile metni zeminden ayırt edip okumak çok zor olacaktır. Maalesef yazı ve arka plan arasındaki renk kontrast ihlali en sık karşılaşılan ve düzeltilmeyen erişilebilirlik ihlalleri arasında yer alıyor.

1.4.3 kriterine göre Metin ve metin içeren görüntüler arka planlarına karşı en az 4.5:1 kontrast oranına sahip olmalıdır. Buradaki kontrast oranı görsel algıya dayalı bir matematiksel hesaplamayla belirlenir ve amaç metnin okunabilirliğini arttırmaktır. Hesap, renk tonunu değil parlaklık farkını temel alır. Eğer metin 14 punto kalın şekilde yazıldıysa veya yazı boyutu 18 puntodan büyükse kontrast oranı 3:1 olmalıdır. Eğer bir metin o an devre dışı bir kontrolün parçasıysa, dekoratif amaçlı ve bilgi vermeyen bir elementse ya da bir logoya aitse renk kontrastı aranmaz.

Düşük görme yetisine sahip kişiler ve yaşa bağlı olarak görme güçlüğü yaşayanlar için yazı ve arka plan arasındaki parlaklık farkı oldukça önemli. Renk kontrast oranları belirlenirken, deneysel çalışmalarda herhangi bir görme sorunu olmayan kişilerin ihtiyaç duydukları oran ve görme oranı 20/40 olan yani belirli derecede düşmüş olan kişilerin ihtiyaçları dikkat alınmış. Herhangi bir görme güçlüğü olmayan kişilerin ihtiyaç duydukları minimum kontrast oranı 3,1 olarak belirlenmiş. Görme oranı 20/40 oranında olan kişiler için ek 1.5 bir kontrast artışı olduğu saptanmış. Bu oran legal

olarak kör kabul edilen kullanıcılardan çok yaşa bağlı görme azalması bulunan kişileri kapsıyor aslında. Özellikle 80 yaş ve üzerindeki kimselerin 20/40 görme oranına sahip oldukları ortaya çıkmış. Buna göre de yazı tipi büyük olmayan metinler için 4.5:1 oranında bir karşıtlık ihtiyacı belirlenmiş. Buradaki hedef kitle herhangi bir yardımcı teknoloji kullanmayan bireyler. WCAG'ye göre yukarıdaki oranlardan daha fazla görme kaybı olanlar yardımcı teknolojilerle renk oranlarını ve yazı boyutlarını ayarlıyor.

WCAG bu oranların bir eşik olduğunu ısrarla vurguluyor. Yani özellikle bir yuvarlama yapılmamasını istiyor. Eğer elinizdeki kontrast 4.49:1 ise bu da kriteri karşılamıyor çünkü belirlenen oranlar gerçekten tam sınır. Mümkünse kontrast oranının bu değerlerin üzerinde olması özellikle tavsiye ediliyor. Yani öğrenci tabiriyle 4 buçuktan 5 ile sınıfı geçemiyorsunuz bu kriterde.

Teknik açıdan renk kontrastını ölçen birçok araç bulunmaktadır. Birkaç örnek verecek olursak, Web aim tarafından geliştirilen WCAG2 contrast checker, TPGI tarafından geliştirilen color contrast Analyzer ve Color Picker tarayıcı eklentileriyle kolayca sayfanızdaki renk kontrast oranları ve ihlallerini görebilirsiniz. Ayrıca Axe DevTools eklentisiyle renk kontrastı ve pek çok WCAG kriterini sayfanızda ölçmeniz mümkün. Bunların yanında Webaim Contrast Checker, Accessible Colors ve Color Contrast Checker Colors gibi web tabanlı araçları da kullanabilirsiniz. Masaüstü uygulaması olarak da TPGI Color Contrast Analyzer ve Adobi color contrast tools gibi araçları kullanabilirsiniz.

Burada en az 4.5:1 karşıtlığı sağlayan renk paletlerinin seçimi önemli. Metnin görüntü olarak verilmesi tavsiye edilmiyor, ama zorunlu hallerde metin görüntü üzerindeyse, görüntünün altına yarı saydam bir koyu örtü eklemek öneriler arasında. Ayrıca CSS ile color ve background değerlerini kullanıcıya göre kilitlememeniz çok önemli. Kullanıcılar kendi yüksek karşıtlık modlarını veya sitillerini uygulayabilmelidir. Kısaca gerçek metin kullanmak, minimum karşıtlık oranını aşan bir arka plan yazı karşıtlığı sağlamak ve mümkün olan her yerde gerçek metin kullanarak kullanıcının değişiklik yapmasına olanak tanımak bu kriter için yapılan öneriler arasında yer alıyor.

1.4.4 Metni Yeniden Boyutlandırma

Bir sayfada metni tarayıcı ayarlarınızdan yakınlaştırdınız veya yazı boyutunu iki katına çıkardınız. O da ne, satırın kalanı nerede? “İse giderken Trafik Çok yoğundu. Çoğu kişi işe geç geldi.” Gibi bir yazının ilk satırı işe giderken deyince kesilmiş alt satırda çoğu kişi diye başlamış. Satırın kalanını ancak farenizi sağa doğru kaydırınca görebildiniz. İşte bu metni yeniden boyutlandırma kriterinin ihlal edildiği anlamına geliyor. İçerikteki metin, yardımcı teknoloji kullanmadan iki katı kadar büyütüldüğünde içerik veya işlev kaybı olmamalıdır.

Burada amaç düşük görme kapasiteli kullanıcıların metni büyüttüklerinde içeriği rahat okuyabilmeleri ve sayfanın bozulmamasının garanti altına alınmasıdır. Daha doğrusu bu kriter, web içerik geliştiricilerinin sayfanın yakınlaştırılmasına engel olmamalarını ve büyütme sonrası sayfanın kullanılabilir kalmasını sağlamalarını gerektiriyor. Neden ise birçok kullanıcının metinleri yalnızca belirli bir büyüklükte okuyabilmeleri. Eğer bu büyüklükte metinler üst üste biniyorsa, satırlar kesiliyorsa veya yatay kaydırma yapmak zorunda kalınıyorsa, kriterimiz ihlal edilmiş demektir. Sabit piksel yüksekliği belirlenmiş bir metin kutusu en ciddi ihlal nedenleri arasındadır; çünkü boyut arttırıldığında içerik kesilecek veya taşacaktır.

İyi bir örnekte sayfa büyütüldüğünde sayfa tek sütuna dönerek kalan kısımlar alt satıra kayar ya da bir menü varsa o menü öğeleri de dikey listelenebilir. Yani kullanıcı yalnızca dikey hareket ederek tüm içeriği okuyabilir ve hiçbir metin kaybı yaşanmaz.

Teknik açıdan, yakınlaştırmaya engel olmamak için mobil için meta viewport etiketinde maximum-scale=1.0 gibi kısıtlamalar koymaktan kaçınılmalıdır. Tasarımda piksel yerine orantılı birimler (em, rem, yüzde) kullanılarak metin boyutlarının kullanıcı ayarlarına uyumlu değişmesi sağlanmalıdır. CSS’de esnek kutu (flex) ve grid düzenlerini kullanarak öğelerin taşma yapmadan alt alta akması sağlanabilir. Sabit yükseklikli metin kutularından uzak durulmalıdır. Bu özellikleri test etmek için sayfanızı %100 oranında yakınlaştırmak her şeyin yolunda olduğunu teyit edebilirsiniz.

1.4.5 Metin olarak Görüntü

Bir seyahat sitesinde üste uçuş kampanyaları gözünüze çarptı, ama meğerse resmin içine gömmüşler metni. Büyütmeye kalktığınızda pikseller oluşuyor ve okunamaz oluyor renk karışıklığını değiştirmek ne mümkün. Kriterimiz tam da böyle senaryolar için ortaya çıkmış. İstenilen sunum normal metin kullanılarak elde edilebiliyorsa bilgi iletimi için normal resim kullanılmamalıdır. Logo gibi özel durumlar bunun bir istisnası.

Düzey A kriterlerini açıklarken metin kullanımının önemine değinmiştik. Bir şeyin metin olarak sunulması esasında kullanıcıya önemli bir anahtar veriyor. Metin olan bir içerik farklı formatlara kolayca dönüştürülebilir. İster ekran okuyucuyla sese veya Braille alfabesine dönüşüyor, ister yeniden boyutlandırılabilir, rengi değiştirilebilir, farklı sitiller uygulanabilir. Yani kullanıcı metin olan bir içeriği kolaylıkla özelleştirebilir. Bu da kör, az gören veya görme güçlüğü çeken kişiler, renk körlüğü, disleksi gibi pek çok kullanıcı profiline içeriği algılaması ve kullanabilmesi anlamına geliyor. Ama siz metni sırf estetik kaygılarla resim içinde sunduğunuzda, hem bu kullanıcıların ihtiyaçlarını göz ardı etmiş oluyorsunuz, hem de hedef kitlenizin bir kısmını kaybetmiş oluyorsunuz. Belki de kampanyadan yararlanmak isteyen bir müşteriye kaçırıyorsunuz. O Nedenle Metni HTML olarak sunmak ve görsel ayarları burada yapmak mümkün ve olmazsa olmaz bir kriter.

Teknik açıdan da bir içeriğin hem estetik olarak güzel görünmesi hem de HTML metin olarak sunulması o kadar zor değil. Örneğin başlıklar CSS kullanılarak istenen yazı karakterleri ve efektlerle stilize edilebilir Metin gerçek yazı olarak kalabilir. Metne özel yazı tipi vermek için @font-face kullanılabilir veya gölge efekti vermek için CSS text-shadow kullanılabilir. Yani mümkün olan her yerde metin HTML kullanın ve CSS ile istenen görünümü verin. Eğer bir sanat eseri gibi özel bir görsel kullanmanız gerekiyorsa, o zaman da alt öz niteliği sağlamanız gerekiyor. Vektörel metin fontları veya svg kullanıyorsanız, bunlara da alternatif metin sağlamalısınız.

1.4.10Yeniden akış (reflow)

Birden fazla sütuna yayılmış bir haber sitesindesiniz ama siz ekranı yakınlaştırarak dikey biçimde dar bir mobil ekranda okumak istiyorsunuz. Yakınlaştırma yaptığınız anda ekranda satırın yalnızca dörtte birini görüyorsunuz. Aşağı indiğinizde cümlelerin

çoğunun orada olmadığını fark ettiniz. Tek yapmanız gereken önce yatay biçimde sayfayı kaydırarak satırın sonuna gitmek. Ama şimdi de yazının başı kayboldu. Yeni satırda tekrar sayfayı en sola kaydırmak zorundasınız. Bu şekilde bir makale okumak ister miydiniz? Çekilir çile değil. Çok kötü bir kullanıcı deneyimi. Yeniden akış kriteri bu tarz durumları önlemek amacıyla ortaya çıkmış. İçeriğin 320 CSS piksel genişliğe (yaklaşık bir mobil ekranın dikey durumdaki genişliği) kadar daraltılması veya 256 CSS piksel yüksekliğe kadar kısaltılması durumunda, bilgi veya işlev kaybı olmadan ve iki yönde (hem yatay hem dikey) kaydırma gerektirmeden sunulabilmesi gerekiyor. Tablo, harita gibi içeriğin sadece iki boyutlu kaydırma zorunluysa bu durumlar istisna olarak kabul edilebilir. Ancak basit bir makalede ve düz yazıda tek boyutlu kaydırma, tercihen dikey kaydırmayla kullanıcı tüm içeriği okuyabilmelidir.

Burada amaç görme engelli kullanıcıların yakınlaştırma yaptıklarında veya herhangi bir kullanıcının içeriği küçük bir mobil ekranda görüntülediğinde, metni satır satır okumak için ileri geri kaydırmak zorunda kalmamasıdır. Bir paragrafı okumak için hem sağa kaydırıp satır bitince tekrar sola kaydırmak hem bilişsel hem fiziksel yükü artıran bir durum. Bu, odak takibini de güçleştirecek ve yeni satırları bulmayı zorlaştıracaktır. Ekran büyütücü kullanan birisi için çift yönlü kaydırma satır takibini epeyce güç hale getirecektir. Bu nedenle içerik dar ekranda ve yüksek yakınlaştırmada yeniden akış yaparak tek sütunda görüntülenebilmelidir.

Teknik açıdan tasarımda duyarlı (responsive) yaklaşımlar benimsenmelidir. Geliştiriciler, CSS medya sorgularıyla farklı ekran boyutlarına uygun düzenleri tanımlayarak geniş içeriklerin dar ekranda tek sütuna yığılmasını sağlayabilir. Örneğin, üç sütunlu bir düzen, 320px genişlikte tek sütun halinde alt alta gelmelidir. Metin kutuları ve resimler yüzdelik genişliklerle veya maksimum genişlik kısıtlarıyla ayarlanarak taşma yapmaları önlenabilir. Ayrıca geliştiriciler, test aşamasında tarayıcı yakınlaştırmalarını kullanarak veya tarayıcı penceresini küçülterek içeriğin tek yönde (tercihen dikey) kaydırma ile gezilebilir olduğunu kontrol etmelidir. Yani burada eğer kullanıcı yalnızca okuma yönünde kaydırma yaparak tüm içeriği görebiliyorsa, yeniden akış kriteri karşılanmış olur.

Burada yeniden akış kriteriyle 1.4.4 yeniden boyutlandırma benzer gibi görünse de bazı farkları bulunuyor. Yeniden boyutlandırma kriteri yalnızca metnin boyutunun büyütülmesiyle ilgilenirken, yeniden akış kriteri Tüm sayfa yapısının yeniden akış ile düzenlenmesini kapsar. Yeniden akışın asıl odak noktası mobil cihazlardır. Aynı

şekilde yeniden boyutlandırma kriteri metin boyutunun tarayıcı ayarlarından büyümesini ele alırken, yeniden akış zoom veya ekran büyütücü gibi araçlarla tüm ekranı büyütme durumunda geçerlidir. Yeniden boyutlandırmada yatay kaydırma yeniden akış kadar kesin bir dille engellenmiş değildir.

1.4.11 Metin dışı Karşıtlık

Bir form doldururken en sonundaki sözleşmeyi kabul ediyorum kutucuğunu bir türlü göremiyorsunuz. Sanki bir şey var orada ama nerede tam olarak? Derken koltuk seçeyim diyorsunuz ve seçilemeyen koltukları belirtecek çarpı işareti zemin rengine o kadar yakın ki belirsiz. Bu tarz durumlar 1.4.11 kriterinin ihlali. Öncesinde metin rengi ve arka plan renginin karşıtlığını konuşmuştuk. Ancak renk karşıtlığı metin dışı düğme, onay kutusu, bağlantı ve ikonlar için de oldukça gerekli kriterimize göre Kullanıcı arayüz bileşenlerinin önemli görsel göstergeleri ve içerikteki anlam ifade eden grafik öğeler, çevrelerindeki renklere karşı en az **3:1** kontrasta sahip olmalı. Burada devre dışı öğe ve kontroller istisnadır.

Düşük karşıtlığı olan düğmeler, ikonlar ya da durum göstergeleri görme yetersizliği olan kişilerce görülemeyebilir. Örneğin açık gri bir düğme kenarlığı beyaz arka planda belirsiz kalabilir. Kullanıcı da orada aslında bir düğme olduğunu fark etmeyebilir. Grafikler için de benzer sorunlar yaşanabilir. Oradaki önemli bir çizgi arka plan rengine gereğinden fazla yakın olduğunda belki de oradaki veri gözden kaçacaktır.

İyi bir örnek olarak bir web formundaki metin giriş alanlarının koyu gri kenarlıkları ve beyaz arka plan gösterilebilir. Böylece form alanları kolayca ayırt edilebilecektir. Yine odakla etkinleştğinde belirli bir mavi parlama ya da kalın kenarlık gösteren bir düğme de iyi bir örnek olarak değerlendirilebilir. Onay kutularındaki tiklerin de arka planla karşıtlık göstermesi oldukça önemlidir. Aslında bu kriterle Düzey A Rol ve değerler kriteri aynı hedefi gerçekleştirmeye yönelik durumlar. Rol ve değerler tamamen kör kullanıcıların belirli kontrolleri algılamasını sağlamayı amaçlarken, Metin dışı kontrast kriteri de görme zorluğu yaşayan kullanıcıların renk karşıtlığıyla düğme, bağlantı, onay kutusu gibi metin dışı kontrolleri ayırt etmelerini amaçlamaktadır.

Teknik açıdan etkin bileşenlerin renk ve kalınlıklarını en az 3.1 karşıtlık sağlayacak şekilde ayarlamak önemli. Açık renkli bir arka planda açık renkli bir kenarlık

kullanmayın. İkonlarla ilgili olarak ikon renginin koyulaştırılması veya arkasına karşıtlık sağlayan bir daire, çerçeve eklemek düşünülebilir. Seçili, işaretli gibi durum göstergelerinde renk değişiminin gerçekten gerekli karşıtlığı sağladığından emin olun. Buralarda opaklık veya ek şekil de kullanılabilir. Mümkün olduğunca odak halkasını tamamen kaldırma gibi düzenlemelere gitmeyin.

Mobil tarafta varsayılan olarak ikonlar karşıtlığı sağlar; ancak özel bir ikon tasarlıyorsanız karşıtlığı mutlaka kontrol etmelisiniz.

1.4.12 Metin aralığı

Bir forumda mesaj okurken satır aralığını arttırdınız. Birden mesajın alt kısmı yok oluverdi. Satır aralığını düşürdünüz geri geldi. Hay Allah ne oldu ki şimdi. Yaşadığınız bu deneyim geliştiricinin metin kutusunu sabit yükseklikte tutarak kurduğu tuzağa düşmenizden kaynaklanıyor ve böyle tuzaklar metin aralığı kriterinin ihlali. Burada tam bir tanım ve değerleri yazarak kriterin net anlaşılmasını sağlamak gerekecek.

İçerik, kullanıcının satır aralığı, paragraf aralığı, harf aralığı ve kelime aralığı gibi metin boşluk ayarlarını artırması halinde bozulmamalıdır. kullanıcı aşağıdaki ayarları uygularsa içerikte hiçbir bilgi veya işlev kaybolmamalıdır:

- Satır yüksekliği (satır aralığı) yazı karakteri boyutunun en az 1,5 katı,
- Paragraf altı boşluğu yazı karakteri boyutunun en az 2 katı,
- Harf aralığı yazı karakteri boyutunun en az %12'si (0.12em),
- Kelime aralığı yazı karakteri boyutunun en az %16'sı (0.16em).

Disleksik bireyler için harf ve satırları birbirine karıştırmadan okumak için aralıkları arttırmak önemli bir ihtiyaç. Aralıklar arttırılarak çok daha rahat bir okuma deneyimi elde edebiliyorlar. Aslında tarayıcı ayarlarından veya kendi sitilleriyle metinlerin satır, paragraf, sözcük veya karakter aralıklarını arttırmak da mümkün. Ancak geliştirici içerik kutusunun alanını sabit genişlik veya overflow:hidden denen özelliklerle kısıtladığında disleksik kullanıcının aralık artışı aleyhine işliyor çünkü içeriğin bir kısmı kayboluyor. Yine görme zorluğu veya diğer bilişsel farklılıklar için de aralıkları kendi ihtiyaçlarına göre arttırmak hayati olabiliyor. O yüzden paragraf, satır, kelime ve karakter aralıklarının artışının içeriği etkilememesi ve içeriğin artışa göre otomatik olarak şekillenmesi gerekiyor.

Teknik açıdan yukarıda da belirttiğimiz sabit yükseklik ya da overflow:hidden gibi kurallardan kaçının. Yüksekliği otomatik bırakmak ya da mantıklı bir maksimum değer vermek faydalı olabilir. Ayrıca, satır aralığı, harf aralığı gibi özellikleri kullanıcı stilini geçersiz kılacak şekilde !important ile tanımlamamak gerekir. Esnekliği test etmek için tarayıcının geliştirici konsolundan HTML elementlerine örnekteki değerler (1.5 line-height, vs.) uygulanarak sayfanın reaksiyonu gözlenebilir. Burada tablo hücre yüksekliklerine özellikle dikkat edilmelidir.

1.4.13 Odak veya Üzerine Gelindiğinde İçerik (Content on Hover or Focus)

Web’de bir navigasyon menüsünde dolaşıyorsunuz. Belli ki her menünün alt menüleri de var. ama sorun şu ki buralar yalnızca fareyi üzerine getirince açılıyor. Siz bir üst menü öğesine odaklanıp tam alt menüye gideceksiniz ki, menü kapanıyor. Bir türlü alt menülere gidemiyorsunuz çünkü hareket eder etmez kayboluyor. Başka bir kötü senaryoda görme engellisiniz ve arkadaşlarınız bağlantı üzerine gelince bir açıklama çıktığından söz ediyor. Neden size hiçbir şey okunmuyor? Çünkü bu açıklama ancak fare ile bağlantı seçildiğinde çıkıyor. Klavyeyle görünmüyor. Can sıkıcı başka bir durumdaki bir kullanıcı tam onaylama düğmesine geliyor ki, altta çıkan bir memnuniyet anketi mesajı var ve kapatmanın hiçbir yolu yok. Formu gönderemiyor bir türlü. Bu da istediği yeri görmesine mâni. Odak ve üzerine gelindiğinde açılan içerik kriteri bu tarz baş ağrıtıcı sorunları adresliyor. Yine tam tanıımı aşağıya koyalım.

Fareyle bir öğenin üzerine gelindiğinde veya klavyeyle bir öğeye odaklanıldığında görünen ekstra içerik (ör. özel bir baloncuk, açılır menü) aşağıdaki koşulları sağlamalıdır:

- **Kapatılabilir (Dismissible):** Ek içerik, fareyi hareket ettirmeden veya odaktan çıkmadan kapatılabilecek bir mekanizma sunmalıdır (Örneğin *Esc* tuşu), *aksi halde* bu ek içerik kullanıcıya hata iletisi veriyorsa veya başka hiçbir içeriği kapatmıyorsa bu şart aranmaz.
- **Üzerinde Durulabilir (Hoverable):** Fareyle tetikleniyorsa, kullanıcı fare imlecini açılan ek içeriğin üzerine getirdiğinde içerik kaybolmamalıdır; yani fare, tetikleyici öğeden yeni açılan içeriğe geçiş yapabilmelidir.

- **Kalıcı (Persistent):** Ek içerik kullanıcı tetiklemeyi kaldırmadıkça, kullanıcı kapatana kadar veya içeriğin kendisi geçerliliğini yitirene kadar ekranda kalmalıdır. Yani kullanıcı içeriği okumayı bitirene dek aniden kaybolmamalıdır.

Bu kriterin yerine getirilmesi özellikle motor becerileri etkilenen ve görme engelli kullanıcılar için önemli. İnce motor becerisi etkilenmiş bir kullanıcı fareyi hareket ettirirken zorlanacağından, bir kontrole geldiğinde açılan içeriği fareyi yeniden hareket ettirmek zorunda kalmadan hızlıca kapatabilme imkânı önem taşıyacaktır. Ya da en azından okumak veya tıklamak için fareyi ana tetikleyiciden yeni içeriğe geçirdiğinde hemen kayboluyorsa, belki de hiç işlemi yerine getiremeyecektir.

Görme engelli biri için de çoğu zaman klavyeyle bir erişim sağlanmadığından bu içeriğe ulaşmak neredeyse imkânsız olabilir. Az gören biri de açılıp hemen kayboluveren bilgi mesajlarından çok mustarip. Bu mesajı okumak için çok daha fazla zamana ihtiyacı olacağından kişi istemedikçe bu içeriğin kaybolmaması da önem taşıyan bir durum.

Teknik açıdan Geliştiriciler, özel açılır içerikler tasarlarlarken aşağıdaki önlemleri almalıdır:

- **Kapatma Mekanizması:** Açılır içerikler için daima bir kapatma yöntemi sağlayın. Klavye kullanıcıları için global bir *Escape* kısayolu eklemek iyi bir çözümdür. Fare kullanıcıları için içerik içinde kapatma butonu (“X”) bulundurmak da yararlıdır.
- **İçerik Konumu:** Mümkünse açılır içeriği, tetiklediği öğenin yanında konumlandırın ancak diğer içerikleri örtmeyecek şekilde ayarlayın. Örneğin, bir tooltip kutusunu butonun hemen altına koymak hem odağı engellemez hem de kullanıcı fark eder.
- **Hover için:** CSS veya JavaScript ile açılan içeriğin mouseenter/mouseleave olaylarını iyi yönetin. mouseleave (fare ayrılması) olayında hemen kapatmak yerine küçük bir gecikme koymak veya fare yeni içeriğe girerse kapatma işlemini iptal etmek gerekir. Bu sayede kullanıcı imleci dikkatlice hareket ettirip içeriğe ulaşabilir.
- **Odak için:** Klavye odaklandığında da aynı içerik görünmeli ve *Odak kaybolmadıkça* açık kalmalıdır. Odak içerik penceresine geçerse (ör. tooltip içinde link varsa), içerik kapatılmamalıdır.

- **Erişilebilirlik:** Açılır içeriğe ARIA role="tooltip" veya role="dialog" vermek ve uygun şekilde odak yönetimi yapmak, ekran okuyucu kullanıcılarının da bu içeriği fark etmesini sağlar. Örneğin, tetikleyici elemente aria-describedby="tooltipID" ekleyerek ekran okuyucunun ipucu içeriğini okumasını sağlayabilirsiniz.

Mobil tarafta da örneğin bir butona uzun basıldığında yeni bir içerik çıkıyorsa Voice Over veya Talk Back kullanıcıları bunu kolayca kapatabilmelidir.

2.4.5 Birden Fazla Yol

Bir arkadaşınız X sitesinden bir şehriye çorbası tarifi aldığını ve çok sevdiğini söyledi. Hemen siteye girip arama alanına şehriye çorbası yazdınız, sonuç yok. Meğerse, sayfada önce başlangıçlar, oradan çorbalar, oradan Türk çorbaları ve oradan da şehriye çorbasını bulmanız gerekiyormuş. Tarife site içinden ulaşmanın başka bir yolu yok. Bu tarz durumlar 2.4.5 kriterinin ihlali anlamına geliyor. Bir sitedeki içeriğe ulaşmanın birden fazla yolu olması gerekiyor. İçeriğe site içindeki menü ve alt menülerden ulaşılabilceği gibi, sitenin arama alanından veya site haritasından da ulaşmak mümkün olmalı.

Bu kriter özellikle bilişsel farklılıklara veya görme engeline sahip bireyler için kritik olabiliyor. Bazı siteler menü yapısı ve sınıflandırma açısından gereğinden kalabalık ve karışık olabiliyor. Bu karışıklık içinde kimi kullanıcılar kaybolabiliyor. Örneğin yukarıdaki yemek tarifi sitesinde eğer arama alanından da istenen tarife ulaşmak mümkün olsaydı, kişi sitenin o karmaşasına girmeden aradığı tarife kolayca ulaşabilecekti. Ya da A'dan Z'ye tüm tariflerin sıralandığı bir liste veya site haritası kişilerin hayatını kolaylaştırabilirdi. Esasında hep aynı noktayı vurgulamamız gerekiyor. Site ve uygulamaları kullananlar tek tip değil, farklı gereksinimleri ve beklentileri var. Tasarımcı ve geliştiricilerin amacı bu farklılıkları elden geldiğince kapsamak olmalı.

Teknik açıdan 5 6 sayfalık küçük sitelerde menüde her içeriğe bağlantı sunmak iyi bir çözüm olacaktır. Büyük sitelerde ise site haritası, arama işlevi ve A'dan Z'ye listeleme gibi seçenekler önerilmektedir. Bazı durumlarda sitenin altbilgi bölümüne de bu tarz

bağlantılar yedirilebilir. Mobil uygulamalarda ise, ürünler kategorilerden seçilebileceği gibi mutlaka bir arama ile de bulunabilmelidir.

2.4.6 Başlıklar ve Etiketler

Uzun bir projenin sayfasına girdiniz ve amacınız proje başlıklarına hızlıca göz atmak. Sayfada pek çok başlık görünce sevindiniz ve ne güzel hazırlamışlar içeriği diye geçirdiniz içinizden. Fakat sevinciniz kısa sürdü. Çünkü başlıklar arasında dolaşırken adlarının bölüm 1, bölüm 2 gibi adlandırıldığını anladınız. İçerikle ilgili hiçbir fikir vermiyor. Derken bir geri bildirim vereyim dediniz, form alanları sözde etiketli ama etiketler, bir değer giriniz, tarih gibi fikir vermeyen metinlerden ibaret. Hangisi isim, hangisi doğum tarihi, hangisi mesaj yazma alanı belirsiz.

2.4.6 kriteri tam böyle yapılmış olmak için yapılmış olan başlıklandırma ve etiketleri hedefliyor. Bir yerin yalnızca etiketlenmesi değil, nitelikli biçimde etiketlenmesi gerekiyor. Yani bir uzun pasajdaki başlıkların yalnızca bölüm 1, 2, şeklinde değil, bölüm 1: Türkiye'nin coğrafi bölgeleri, bölüm 2: geçim kaynakları gibi içeriğe ilişkin fikir verecek şekilde etiketlenmesi gerekiyor. Aynı şekilde form alanı etiketlerinin de açıkça istenen bilgiyi belirtmesi gerekli. Bir değer girin değil ad, soyad girin gibi.

Teknik açıdan yalnızca yer tutuculara güvenilmemesi öğütleniyor. Bunun yanında başlıkların kısa, tutarlı ve açıklayıcı olması isteniyor. Yalnızca sonuçlar değil 2025 anket sonuçları gibi etiketler öneriliyor. Yalnızca ikonlarla gösterilen yerlerde bu ikonların aria-label, Accessibility-label veya ContentDescription özelliklerinin doğru ayarlanması gerekiyor.

2.4.7 Odak Görünür

İnce motor becerilerinizdeki farklılık nedeniyle sayfaları klavyeyle kullanmak durumundasınız. Bir sayfada form alanları arasında tab ile gezerken ekran nereden olduğunuz anlaşılmıyor. Gönder düğmesine basacaksınız ama o an üzerine geldiniz mi bilemiyorsunuz. Bu sorununun nedeni web tasarımcılarının türlü estetik kaygılarla sayfadaki odak halkasını gizlemeleri veya arka plandan ayırt etmesi imkânsız bir renkle belirtmeleri. Yani açıkça 2.4.7 kriterinin ihlali söz konusu. Kriter

klavyeyle işletilebilen tüm arayüz bileşenlerinin klavye odağı aldıklarında görsel olarak belirgin olmalarını gerektiriyor. Tab ile form alanları veya bağlantılar arasında dolaşırken, odak alınan bileşen kenarlık, vurgulu arka plan gibi bir şekilde görülebilmelidir.

Bilgisayarları yalnızca ekran okuyucu kullanıcıları klavyeyle kullanmayabilir. İnce motor becerileri etkilenen veya farklı nedenlerle fareyi kullanamayan ama ekranı görebilen kullanıcılar için klavyeyle odaklanılan noktayı görsel olarak anlamak hayati önemdedir. Görsel odak belirtimi olmadan klavyeyle gezdiklerinde gözü kapalı biçimde gezinmekten farklı bir deneyim yaşanmayacaktır. O nedenle odak halkası mutlaka görünür olmalıdır. Bu kriter dikkat veya hafıza sorunu yaşayan kullanıcıların da daha kolay biçimde sitelerde gezinmelerine destek olacaktır.

Teknik açıdan en iyi çözüm, tarayıcıların varsayılan odak göstergelerini korumak olacaktır. Bu göstergeler genellikle odaklanılan bileşenin etrafına mavi veya siyah halka çeker. CSS'de `:focus { outline: none; }` kuralından kaçınılmalıdır. Tasarım bir özelleştirme gerektiriyorsa, odaklanma için özel bir stil verilmelidir. Örnek: `:focus { outline: 3px solid #005fcc; }`). Bazı geliştiriciler fareyle seçildiğinde gösterilmeyen ve sadece klavyeyle odaklanıldığında gösterilen Focus-Visible seçicisini de kullanabilir. Burada Java Script ile odağı kaldıran anti-pattern gibi kodlar kullanılmamalıdır.

Mobil uygulamalarda da harici klavye desteği varsa bu kurala dikkat edilmeli, özellikle TV uygulamalarında odağın korunmasına özen gösterilmelidir.

2.4.11 Odağın Gizlenmemesi

Az gören bir kullanıcı olarak sayfayı klavyeyle geziyorsunuz ve odak da görünür durumda. Her şey yolunda derken bir kez daha taba bastığınızda tek gördüğünüz sayfanın altındaki sabit sohbet simgesi. Acaba gönder düğmesine geldiniz mi gelediniz mi? Bazı sayfalarda odak görünür olsa bile, sayfanın üstünde veya altında bulunan sabit bir içerik gelinen odağı tamamen örtebilir. Böyle olunca da kriterimiz ihlal edilmiş olacaktır. Klavye odağı almış herhangi bir kullanıcı arayüz bileşeni sayfadaki başka bir içerik tarafından tamamen gizlenmemelidir. Kısmi olarak örtülebilir ama en azından odağın bir kısmı görünür olmalıdır. Bu tür tamamen örtülmeler genellikle sabit bir başlık veya açılır panel gibi öğelerle ona en yakın

elementin bulunduğu yerlerde yaşanmaktadır. Tab ile ilerlenirken odak birden sabit bileşenin altına kayabilir. Burada en azından odağın kısmen görünür olması gerekecektir.

Düşük görme seviyesinde olup ekran yakınlaştırma yapan kişiler odaklarının nerede olduğunu görmek zorundadır. Sabit bir içerik nedeniyle odak görünmez olduğunda, o an nerede bulunduğu, bir şeyin seçili olup olmadığı anlaşılmayacaktır. Burada durum odak göstergesinden bağımsızdır. Sabit bir içeriğin altına kayıp odak göstergesi kaybolduğunda anlamlı olmayacaktır. Yüksek yakınlaştırma kullanan veya küçük ekranlarda klavye kullanan kişilerde de sıkça yaşanabilmektedir. Kriterimiz geliştiricilerin bunu ön görerek sayfa kaydırma davranışlarını değiştirmelerini veya ekrana sığmayan içeriğe tampon boşluklar eklemelerini gerektirmektedir.

Teknik açıdan sabit öğeler kullanıldığında, bunların odak davranışını engellememesi için farklı önlemler alınması gerekir. Örneğin CSS scroll-margin-top (eski adıyla scroll-padding) özelliği, bir elemente odaklanınca tarayıcının onu en üste hizalarken kenar boşluğu bırakmasını sağlar. Böylece sabit üst menü yüksekliği kadar bir boşluk bırakarak odaklanan öğenin menü altında kalmamasını sağlayacaktır. Sabit alt öğeler için de scroll-margin-bottom kullanılabilir.

Mobil tarafta modallar/tooltipler vs. odak almamalı veya kapatılmalıdır. Böylece Voice Over odağı bunlara takılmayacaktır. Bu kriteri test etmek için %400 yakınlaştırma yapın ve klavye ile gezinin. Odak tamamen veya kısmen halen görünüyorsa, bu kriter sağlanmış olacaktır.

2.5.7 Sürükleme Hareketleri (Dragging Movements)

Bir yapılacaklar listesinde görevlerin yerini değiştirmek istiyorsunuz. Size sunulan tek seçenek sürükleyip bırak. Bir görevin üstüne basmanız ve yukarı veya aşağı taşımanız gerekli. İnce motor beceriler nedeniyle bu hareketi yapamadığınız için listeyi kişiselleştiremiyorsunuz. Sürükleyip bırakın engellilere açtığı binlerce dertten yalnızca minik bir günlük örnek. Buradaki hedef kitle klavye kullanıcıları değil. Bununla ilgili zaten düzey A seviyesinde kriterler mevcut. Ancak ekranı parmağıyla dokunarak veya fareyle kullanan kişiler için de sürükleyip bırakın bir alternatifi olmaması ciddi bir

sorun. Herkes bir şeyi tutup ekranda başka bir yere sürükleyebilme becerisine sahip değil. O nedenle de kullanıcı arayüzünde sürükle bırak gerektiren tüm işlevler, sürükleme yapmadan, tek bir işaretçi hareketiyle de gerçekleştirilebilmelidir.

Sürükle Bırak, ince motor becerilerinde farklılık olanların yanında, tremoru olanlar, tek el cihaz kullananlar, gözle veya sesle denetim sistemi kullanıcıları için de zor ve zahmetli bir süreç. O nedenle kriterimiz geliştiricilere sürüklemekten yapma seçeneği de sunmayı şart koşuyor.

Teknik açıdan örnek bir seçenek bir liste veya resim galerisindeki her öğenin yanında yukarı ve aşağı taşı düğmelerinin de yer almasıdır. Veya bir haritada sağa sola ilerlemek için haritayı tutup çekmenin yanında sağ ve sol taraftaki oklara basılarak da kaydırma yapılabilmesidir. Yani sürükleme gerektiren her işlevin analiz edilip alternatif bir tek dokunma yönteminin de eklenmesi istenmektedir. Öge taşımak için sürüklemenin yanında her öğeye taşı menü seçeneği eklenerek istenen hedefin belirlenmesi sağlanabilir. Örneğin iOS cihazlarında bir programı taşımak için önce düzenle menüsüne sonra da sürükle düğmesine basılabilmekte, sonrasında dilenen yere gelinip istenen öğenin üzerine, öncesine veya sonrasına bırak seçenekleri bulunmaktadır.

Bir Slider (kaydırıcı) içinde de değeri arttırıp azaltmak için + veya – simgeleri kullanılabilir veya değerin yazılabileceği bir yazım kutusu eklenebilir. Ses düzeyleri bu şekilde de değiştirilebilir. Yakın zamana kadar Youtube oynatıcısının hızını arttırıp azaltmak böyle bir erişilebilirlik ihlali barındırıyordu. Sonrasında hızı arttır ve azalt düğmeleri eklenerek bu sorun çözüldü.

Çizim ve grafik uygulamalarında şekilleri sürüklemenin yanında seçili şeklin klavye ile taşınması veya koordinatlarının girilebilmesi gibi seçenekler düşünülebilir.

2.5.8 Hedef Boyutu Minimum

Bir alışverişin son aşamasında sözleşmeyi kabul ediyorum onay kutusuna tıklamak istiyorsunuz. Ama tıklama alanı o kadar minik ki, birden iptale tıkladınız ve tüm süreç baştan başladı. Bu tür durumlar az gören, motor becerileri etkilenen pek çok

kullanıcıyı mağdur eden bir kriter ihlali. Kriterimize göre tıklanabilir/dokunulabilir hedeflerin (buton, bağlantı, ikon vb. etkileşimli öğeler) boyutu en az **24x24 CSS piksel** olmalıdır. Bu ölçü yarım santimetrelik bir alana karşılık gelmektedir. Eğer bu boyut tutturulamıyorsa, hedefler arasında yeterli boşluk bulunmalıdır. Bu boşluk da yine 24 CSS piksellik çapla çizilen bir daireye karşılık gelmektedir. Yani bir tıklanabilir hedef küçük olsa bile iki komşu hedef arasında en az yarım santimetrelik çapta tıklanabilecek bir boş alan da bulunmalıdır. Böylece kişi tam hedefe değil de biraz yanına bile tıklasa işlemini gerçekleştirebilmiş olacaktır.

Gereğinden küçük hedefler ve hedef aralıkları birçok farklı kullanıcıyı zora sokan bir sorun. Titreşim, spastisite, kas zayıflığı gibi motor becerileri etkilenenler, baş çubukları, göz takip sistemleri gibi bebek parmak kontrolü olan cihazlar kullananlar, görme güçlüğü yaşayanlar ve yaşlılar için hedef ne kadar küçükse isabet ettirmesi de o kadar zor olabilir. Son dönem testlerimiz çok daha ilginç bir yapının da küçük hedeflerden etkilendiğini gösterdi: yapay zekâ ajanları. Yapay zekâ ajanlarına alışverişi tamamlama gibi çeşitli görevler verdik. Bu görevlerde ajanların en çok çeşitli sitelerdeki küçük sözleşmeyi kabul ediyorum onay kutularına tıklamakta güçlük yaşadıklarını ve vakit kaybettiklerini gözlemledik. Bu sistemler ekran görüntüleri alıp ona göre tıklama yaptıklarından küçük hedeflerden doğrudan etkilendikleri ortaya çıktı.

Teknik açıdan tıklanabilir öğelerin boyutunu minimum 24 css piksel yani yarım CM belirlemeniz öneriliyor. Belirleyemiyorsanız, etrafına padding ekleyerek boyutu büyütebilirsiniz. Küçük bir çöp kutusu ikonu için `.icon-button { padding: 8px; }` tarzı kural eklediğinizde, çöp kutusu ikonunun etrafında görünmez bir tıklama bölgesi oluşturmuş olursunuz. Birden çok küçük hedef yan yana ise, her birinin etrafındaki boşlukları artırın; böylece kullanıcı birine basmaya çalışırken diğeriyle arasında mesafe olur. Burada birbirine bitişik iki link koymak yerine bunları liste halinde ayrı satırlara almak veya aralarına bir ikon koymak faydalı olabilir. Teknik olarak, `line-height` veya `min-width/min-height` CSS kuralları ile link ve butonların asgari boyutunu güvenceye alabilirsiniz. Örneğin:

```
button, .touch-target {  
  
    min-width: 24px;  
  
    min-height: 24px;
```

}.

Bir hedef cümlelerin ortasındaki bir kelime gibi veya bir resmi evrakın elektronik ortamdaki aynı şekli gibi olmak zorundaysa, tıklanabilecek öğenin yanına görünür bir tıklama boşluğu eklemek ve çevresine başka tıklanacak bir öğe koymamak gerekecektir. Son olarak Mobil platformlarda iOS ve Android geliştiricilere çok daha büyük hedef boyutları önermektedir (44 PX gibi)88888

3.1.2 Bölüm Dili

Birden fazla dilde açıklama ve paragrafların olduğu bir makale okuyorsunuz ve yardımcı teknolojinizin otomatik dil saptaması açık. Metnin içindeki İngilizce paragraflara geldiniz ama halen Türkçe seslendirme yapmaya devam ediyor. Bu durumun nedeni bölüm dili kriterinin ihlal edilmesi. Bildiğimiz üzere, düzey A kriteri sayfanın varsayılan dilinin doğru belirtilmesini yeterli görüyordu. Düzey AA olan 3.1.2 kriteri ise bunu bir adım ileri taşıyarak yalnızca sayfanın varsayılan dilinin değil çok dilli sayfalarda bölümlerin de dil etiketlerinin belirtilmesini şart koşuyor.

Yukarıda da belirttiğimiz üzere metni TTS ile okurken, dil saptaması açıksa, motorlar koda bakarak metin içindeki farklı dilleri o dilin sentezleyicisiyle telaffuz edebiliyor. Ama dil belirtilmediyse, bu mümkün olmuyor. Dil etiketlerinden yararlanan bir kitle daha var: Disleksik bireyler. Onlar da zaman zaman otomatik dil çeviri araçları kullanabiliyorlar ve bu araçlar da farklı dilleri algılamak için lang= ile başlayan dil kodlarına bağlılar. O nedenle özellikle çok dilin yaygın olarak bulunduğu sayfalar için bu dillerin belirtilmesi halen önemli bir kriter.

Teknik açıdan bunu belirtmenin yolu lang özniteliğini kullanmak. Örneğin: This is an English sentence.. Bu tarz bir yapı span içinde kalan tüm bölümün İngilizce olarak algılanmasını sağlar. Özellikle sizler de Word PDF gibi bir içerik üretirken İngilizce paragrafları seçip yazım dilini İngilizce olarak ayarlayıp kaydettiğinizde, bu belgeyi bir web editörüne yüklediğinizde de bu bilgi kalıcı olacaktır. Mobil tarafta bu sorun çok yaşanmasa da iOS için AccessibilityLanguage özelliğiyle farklı diller verilebilir.

3.2.3 Tutarlı Gezinme

Her eve geldiğinizde eşyaların yerini değiştiren anneler misali, bazı sayfalardaki üst menülerin sıralaması girdiğiniz her sayfada farklı olabilmekte. Örneğin ana sayfada üs menü duyurular, hakkımızda, iletişim diye giderken bir başka sayfada iletişim, hakkımızda, duyurular diye gidiyorsa, bu hata tutarlı gezinme kriterinin ihlalidir. Yani alt sayfalarda tekrar eden menü yapısının sıralaması her sayfada aynı olmalıdır. Eğer bir alt sayfada menüye yeni bir şey ekleniyorsa, bu sırayı bozmadan menünün altına eklenmelidir. Bir öge bir sayfada üçüncü sıradaysa, diğer sayfada da üçüncü sırada olmalıdır.

Birçok engelli için altın kurallardan en önemlisi rutinlerdir. Özellikle otizm DEHB gibi nöroçeşitli kişiler için sayfaların alıştıkları bir sırada ve düzende olması, kafa rahatlığıyla işlem yapabilmeleri için kritik olabilir. Bu nedenle tekrar eden bölümlerin tutarlı olmasına dikkat edilmelidir.

Teknik açıdan sayfada tekrar eden navigasyon elemanlarını üst menü, yan menü gibi bir şablon içinde tanımlanması önerilmektedir.

3.2.4 Tutarlı Tanımlama

Bir site veya uygulamanın bir noktasında çıkış, diğer bir yerinde oturumu kapat düğmesini gördünüz. İkinci bir örnek olarak aynı düğme bir noktada ara, başka bir yerde bul diye etiketlenmiş. Tüm bunlar tutarlı tanımlama kriterini ihlal eden durumlar.

Özellikle bilişsel farklılıkları olan veya hafıza güçlüğü yaşayan kişiler için tanımların aynı kelimelerle olması önemli. Burada bir tek etiketler değil ikonların ve sembollerin de site ve uygulamalarda aynı olmaları gerekiyor.

Teknik açıdan sitenin genel bir envanterinin çıkartılması ve benzer öğelerin etiket, ikon ve aria-label durumlarının aynı olmasının kontrol edilmesi öneriliyor.

3.3.3 Hata Önerisi

Uzun bir üyelik formu doldururken sonraki düğmesine bastınız ama karşınıza bir hata çıktı: “Lütfen doğum tarihinizi girin”. Halbuki aslında siz doğru biçimde girdiğinizi düşünüyorsunuz aralardaki / (bölü) işareti . (nokta) yaparak tekrar denediniz, yine olmadı. Meğerse deneme yanılmalarda anladınız ki burada istenen format 4 hane yıl – (tire) ay – (tire) gün şeklindeymiş. Eğer size doğru bir hata önerisi yapılsaydı zaman kaybetmeyecektiniz.

Düzy AA seviyesindeki hata önerisi kriteri yalnızca bir alanda hata yapıldığının belirtilmesiyle yetinmiyor. Evet hangi alanın hatalı girildiğinin gösterimi bir ön koşul ve düzy A seviyesindeki bir kriter. Ancak yukarıdaki örnekte olduğu gibi zaman zaman yalnızca bu bilgiyi hatalı girdiniz demek yetmiyor. Doğru bilginin nasıl girileceğinin de belirtilmesi isteniyor. Bu durum özellikle tarih ve parola alanları için kritik. Örneğin parolada hem harf hem rakam hem de bir noktalama işareti isteniyorsa ve en az 8 karakter uzunluk gerekiyorsa, bunlardan yerine getirilmeyenlerin açıkça kullanıcıya belirtilmesi isteniyor. Bu kriter özellikle disleksik ve görme engelli bireyler için önem taşıyor.

Teknik açıdan form validasyonu kurgularken yalnızca required kontrolüyle yetinmeyin ve yaygın hata senaryoları için özel mesajlar hazırlayın. E-postanın @ (at) içermemesi, tarihin beklenenden farklı girilme senaryoları, şifre politikaları bu noktada hazırlanabilecek senaryolardan bazıları. aria-describedby kullanarak hata mesajını formla ilişkilendirirseniz, ekran okuyucular mesajı kolaylıkla algılayacaktır. Pratik olarak daha formu doldururken, kullanıcı tab tuşuna bastığı anda eğer hatalı bir giriş yaptıysa uyarı vermek yararlı olabilir. E-posta adresi kısmında @ (at) işareti yok, telefon numarası alanına harf girilemez gibi. Mobil tarafta AccessibilityHint özelliğiyle form alanlarıyla ilgili ipuçları sağlamak faydalı olabilir.

3.3.4 Hata Önleme (Yasal, finansal, veri)

Bir tatil sitesinden 5 günlük rezervasyon adımlarını tamamlayıp onayla dediğiniz anda birden yanlış tarihleri seçtiğinizi fark ettiniz. Anladınız ki, iş işten geçmiş çünkü sizden bir onay istemededen bankadan paranız çekilmiş bile. Artık bu tarz geri döndürülemez durumlar pek yaşanmasa da yine de hata önleme kriteri bu tarz durumlar için bir onay mekanizmasını garanti altına almak amacıyla konulmuş.

Kullanıcıdan yasal bir onay, finansal işlem veya veri silme/değiştirme gibi önemli işlemler yapmasını isteyen sayfalar, en azından birini sağlamalıdır:

- İşlemi geri alabilme (geri alma/iptal imkânı),
- Kullanıcı girdilerini hatalara karşı kontrol edip düzeltme fırsatı verme,
- İşlem sonunda kullanıcıya girdiği verileri gözden geçirip onaylama adımı sunma.

Burada amaç, kullanıcıyı yüksek riskli işlemlerde (ödeme yapma, form gönderme veya veri silme gibi) korumak, yanlış yapmasını önlemek ve gerekirse işlemini geri alma, telafi etme fırsatını ona sunmaktır. Bu tarz onay telafi mekanizmaları her kullanıcı için çok önemli. Görme engelli ve motor farklılıkları olan bireyler içinse, risk daha yüksek olabilir. Görme engelli kişi yanlış bir yere giriş yapabilir veya motor becerileri etkilenen kişiler yanlışlıkla onay/gönder butonuna tıklamış olabilir. Eğer bu tıklama sonrası bilgilerin özetlendiği bir onay sayfası olursa, kullanıcı tüm girdilerini rahatça inceler ve sonra isterse devam eder, isterse geri dönüp yanlış girdiği yerleri düzeltebilir. Bu yüzden böyle kritik işlemlerde iki adımlı bir mekanizma çok önemlidir. Aynı biçimde bir sayfada silme işlemi yaptığımızda, bir emin misiniz sorusu ya da silinen öğelerin bir süre arşiv veya çöp kutusunda kalması gibi önlemler bu amaçla alınmıştır.

Özellikle çevrimiçi bankacılık, alışveriş gibi sayfa ve uygulamalarda onay ve geri alma mekanizmalarının özenle kurgulanması gerekir.

Teknik açıdan, formlarda şunlardan birini uygulayın: işlem öncesi özet ve onay, geri al (undo) mekanizması, onay isteme (emin misiniz), ek doğrulama (iki adımlı doğrulama, telefona mesaj gönderip onun girilmesi vs.), veri yedekleme.

Kullanıcı testlerinde bu kritik adımların özellikle engelli kullanıcılarla test edilmesi, onay sayfalarının otomatik olarak okunup okunmadığının kontrol edilmesi çok önemli olacaktır.

3.3.8 Erişilebilir Kimlik Doğrulama (Asgari)

Hiçbir sitede oturum açarken sürekli parolanızı sıfırlamak zorunda kaldınız mı? O kadar çok farklı sisteme giriş yapıyoruz ki, bazen parolalar karışıyor. Bazen de sürekli parola değiştirmemiz istendiğinden eskisi, yenisi hepsi bir karmaşa haline gelebiliyor. Hele bir de hafıza sorunu yaşıyorsanız, bilişsel farklılıklarınız varsa veya okur yazar değilseniz, bu parola ve Captcha işlemleri iyice içinden çıkılmaz hale gelebiliyor. WCAG eko sistemine yeni eklenen erişilebilir kimlik doğrulama kriteri, böyle sorunlar yaşayan kullanıcıların sorunlarını adresliyor. Tanım çok açık:

Bir sisteme giriş (kimlik doğrulama) sırasında, kullanıcıdan salt zihinsel bir test (ör. bir şeyi hatırlama, bulmaca çözme) yapması istenmemelidir veya bunun en az bir alternatifi sunulmalıdır. Özellikle, parolalar gibi tamamen hafızaya dayalı doğrulamalar yerine, parola yöneticisi kullanımı, otomatik doldurma veya farklı kimlik doğrulama yöntemleri (SMS, biyometri, e-posta bağlantısı vb.) desteklenmelidir.

Oturum açarken sürekli parola hatırlamayabilirsiniz, ama eğer girdiğiniz sistem tarayıcı veya mobil sisteminizin parola yöneticisini destekliyorsa, daha önce girdiğiniz parola saklanacak ve siz tekrar girmek zorunda kalmadan otomatik olarak doldurulabilecektir. Veya e-posta ya da telefonunuza gönderilecek bir onay koduyla oturum açma imkânı sayesinde, her an parolanızı aklınızda tutmak zorunda kalmazsınız. E-posta adresine gönderilebilecek bir tek oturum açma bağlantısı da iyi bir çözüm olabilir. Burada Önemli olan kullanıcıyı sadece bir şeyi aklında tutmak zorunda bırakmamak. Burada gönderilen onay kodlarının da ilgili alanlara yapıştırılmasının engellenmemesi çok kritik bir durum. Mobil sistemlerde sms olarak gönderilen kodların doğrudan yazma alanı açıldığında Mesajlardan düğmesine tıklanarak otomatik olarak alana yazılması, özellikle görme engelli Voice Over kullanıcıları için oldukça yararlı.

Captcha ile ilgili sorunları Düzey A kriterlerinden detaylıca ele almıştık, ama bu kriter yine Captcha konusuna eğilmiş. Tek tip bir Captcha örneğin bulmaca gibi olan bir yapıya mutlaka alternatif sağlanması isteniyor. Örneğin sesli bir Captcha veya ben robot Değilim onay kutusu ve bunun ekran okuyucu kullanımını algılaması önemli. Veya Captcha yerine e-posta veya sms doğrulanması burada da öneriliyor.

Disleksik kişiler, demanslı kullanıcılar, görme engelliler bu kriterin hedef kitlesi arasında.

Teknik açıdan parola alanlarında kopyalama ve yapıştırmayı engellemeyin. yöneticilerinin çalışmasına izin verin (input autocomplete="current-password" gibi niteliği olmalı).

İki adımlı girişi destekleyerek e-posta, sms, sosyal medya hesabıyla giriş gibi yöntemler kullanabilirsiniz.

Mobil uygulamalarda biyometrik girişi destekleyerek parmak izi veya yüz tanımayla giriş gibi seçenekleri de kullanabilirsiniz.

Captcha için reCAPTCHA v3 tarzı kullanıcı etkileşimi istemeyen sistemleri kullanmanız ya da görsel Captcha varsa mutlaka sesli alternatifinin olması gerekiyor.

Güvenlik sorularının da mümkün olduğunca kullanılmaması öneriliyor.

Bu kriteri test etmek için mümkünse bilişsel farklılıkları olan kişilere giriş yaptırarak sorun yaşayıp yaşamadıklarının tespit edilmesi öneriliyor.

4.1.3 Durum Mesajları

Bir alışveriş sitesinde ürünü sepete eklediniz ama hiçbir şey olmadı. Eklendi mi acaba? Ya da sıkça yaşadığım bir şey: koltuk seçtim ama seçip seçmediğimi öğrenmek için ekranın üst taraflarında bir yerdeki durum mesajına bakmam gerekiyor. Bu da zaman kaybı. Düzey AA olan durum mesajları kriteri tam da ekran okuyucu kullanıcılarının bu ihtiyaçlarına yönelik olarak hazırlanmış bir kriter. Odağımı değiştirmeden çıkan otomatik durum mesajlarının ekran okuyucularca otomatik okunmasını şart koşuyor. Tanımımız şöyle:

Web sayfasında ya da mobil uygulamalarda ortaya çıkan durum mesajları (örn. “Sepete eklendi”, “Kaydediliyor...”, “Form başarıyla gönderildi” gibi anlık bildirimler), odağı almaksızın yardımcı teknolojiler tarafından algılanabilir olmalıdır. Yani bu tür mesajlar görünür olduğu anda ekran okuyucular tarafından da otomatik olarak okunabilmelidir; kullanıcı, odaklanmadan da bu değişikliği duyar.

Buradaki anahtar noktalara dikkat. Aslında bu mesajlar zaten ekranda var. Önemli olan bunların üzerinde bulunduğumuz odak değişmeden otomatik olarak belirtilmesi. Anlaşıldığı üzere bu kriter ekran okuyucu kullanan görme engellileri ilgilendiriyor. Çoğunuzun başına gelmiştir. Bir formu doldurup göndere basarsınız, tam bir sessizlik olur. Meğerse üste bir yerde kırmızı renkte lütfen hatalı alanları doldurun uyarısı çıkmıştır. Ancak bu uyarı mesajı ARIA role="status" veya role="alert" şeklinde tanımlanmadığında ekran okuyucu bize bir bildirim yapmaz. Çünkü ekran okuyucular genellikle odaklandıkları elementi okur. Başka bir yerde çıkan mesajın otomatik okunması için ekran okuyucuya programatik olarak bunun bir uyarı olduğunun gösterilmesi gerekir.

Teknik açıdan önce sayfada anlık mesaj veren yerlerin tespiti gerekecektir. Bunlar genellikle form hata/başarı mesajları, yükleniyor, başarıyla kaydedildi gibi Ajax geri bildirimleri, sepete eklendi, favorilere eklendi, kopyalandı gibi uyarılar ile live region sayılabilecek sohbet alanları olabilir. İlk olarak bu elemanları html'de aria live region olarak tanımlamalısınız. Farklı aria rolleri bulunmaktadır role="status" mesajlarını genellikle bilgi verici veya başarılı işlemlerde kullanın (sepete eklendi, kaydedildi gibi). Role="alert" ise daha çok acil ve çoğunlukla hatalı durumlarda kullanılır ve bu zamanlarda odağın da oraya gitmesi gerekecektir.

Örnek kod:

```
<div id="message" role="status" aria-live="polite"></div>
```

Böyle bir kod otomatik olarak başarılı olan bir işlemin belirtilmesi için yararlıdır ve aria-live="polite" kullanıldığı için ekran okuyucu önce diğer okumasını bitirir ve sonra bu mesajı otomatik okur. Eğer Aria-live="assertive" kullanılırsa, ekran okuyucu okumasını yarıda kesip gelen mesajı okuyacaktır.

Eğer durum mesajı yalnızca yeşil tik gibi bir ikonsa, bu ikona bir alt metin ekleyerek okunmasını sağlamak gerekecektir.

Mobil tarafta da durum mesajlarının otomatik okunması için iOS'ta `UIAccessibilityPostNotification(notifyVoiceOver, "metin")`. Kullanılabilir. Android için ise `AccessibilityManager` ile `announce` event kullanılmalıdır.

Yani web'de role = "status" ve role="alert" gibi aria teknikleriyle, mobilde de `UIAccessibilityPostNotification` ve `AccessibilityManager AnounceEvent` yöntemleriyle

odağı deęiřtirmeden durum mesajlarının otomatik okunması mümkündür. Ancak test için mutlaka ekran okuyucuları kullanıp bir formu gönderdiğinizde, bir ürünü sepete eklediğinizde ne olduğunu kontrol etmelisiniz.

Sonuç

Gerek Avrupa bu gerekse ABD, İngiltere, Kanada ve Avusturalya gibi ülkelerin mevzuatı incelendiğinde, bu ülke ve örgütlerin web erişilebilirliğini sağlamak amacıyla WCAG kriterlerini teknik standart olarak kabul ettikleri anlaşıyor. Yasaların çıkış tarihlerine baęlı olarak WCAG 2.0, 2.1 v2.2 sürümlerini kabul eden tüm bu ülkelerin düzey olarak da AA seviyesini temel aldıkları görülüyor. Serinin ilk yazısında ele aldığımız düzey A kriterleri esasında en temel ve olmazsa olmaz erişilebilirlik ilkeleri. Burada çoęu kriter için tasarım ve görünümünden çok kod altyapısını yardımcı teknolojilerin algılayabileceęi şekilde düzenlemek yeterli. Görsellere alt etiket eklenmesi, sayfa diziliminin semantik olarak doęru biçimde kurgulanması, tüm tıklanabilir alanlara klavye ve mobil tarafta fiske hareketleriyle erişilebilmesi, enter, boşluk çubuęu veya çift dokunularak etkinleştirilebilmesi, form alanlarının ve çıkan hataların yardımcı teknolojilerce anlaşılacak Aria etiketleri kullanılarak tanımlanması ve gezinim ve tıklama açısından doęru rol ve deęerlerin verilmesi düzey A kriterlerinin temel bir özeti olarak nitelenebilir.

Düzey AA kriterleri ise bu temel altyapının üzerine çıkılan bir katman. Bu katmanda daha çok ön yüzü doğrudan etkileyen gereksinimler mevcut. Öyle ki, ele alına 24 kriterin en az 17 adedi doğrudan site ve uygulama tasarım ve görünümünü etkiliyor. Minimum renk kontrastının saęlanması, metinlerin resim olarak sunulmaması, ekran yöneliminin deęişimine izin verilmesi, metin boyutu ve aralıklarının ayarlanabilmesi, Tıklanabilecek kontrollerin boyut ve birbirine yakınlıklarının asgari ölçüleri, odaęın her zaman görünür olması ve tamamen gizlenmemesi, tasarımı ilgilendiren düzey AA kriterleri arasında. Bunlara ek olarak, canlı altyazı, sesi betimleme, anlaşılır form ve başlık etiketleri, doęru hata önerileri, kimlik doęrulama kolaylıkları ve durum mesajlarının otomatik okunabilir biçimde tanımlanması gibi ilkeler de Düzey AA kriterleri arasında yer alıyor.

Özetle bu kriterler de web siteleri ve mobil uygulamalarda erişilebilirlięi saęlamanın vazgeçilmez unsurları. AA seviyesi, Daha çok ekran okuyucu kullanıcılarının

gereksinimlerini karřılayan Düzey A kriterlerini tamamlayıcı nitelikte. az gören, ADHD, disleksi, otizm ve hafıza güçlüğü gibi nöro çeřitli bireylerin yanında ince motor becerileri etkilenen kiřilerin ihtiyaıları da Düzey AA kriterleriyle daha fazla adreslenmiř durumda. Aslına bakıldıęında, WCAG açıklamalarında da belirtildięi gibi tüm kriterler yerine getirilse, hatta düzey aaa kriterleri karřılansa bile, herkes için %100 bir erişilebilirlikten söz edilemez. Çünkü ihtiyaılar ve insan çeřitlilięi her an yeni sorun ve beklentileri beraberinde getirme potansiyeli taşıyor. Bu kriterleri karřılamak, çok daha kapsayıcı biçimde hizmet sunmak ve farklılıkları dahil etmek için atılacak ilk adım olarak düşünölmeli. Sonrası bizzat farklı kullanıcılarla birlikte en iyi deneyime ulaşma yolculuęu. Kriterleri yerine getirmek bu yolculukta herkese yer var mesajını vermesi bakımından anlamlı bir başlangıı.