



ULUSLARARASI 3B YAZICI TEKNOLOJİLERİ
VE DİJİTAL ENDÜSTRİ DERGİSİ



INTERNATIONAL JOURNAL OF 3D PRINTING
TECHNOLOGIES AND DIGITAL INDUSTRY

ISSN:2602-3350 (Online)

URL: <https://dergipark.org.tr/ij3dptdi>

CSS ÇATILARININ KULLANIMINDA KARŞILAŞILAN SORUNLAR VE ÇÖZÜM ÖNERİLERİ

USER EXPERIENCE AND ORIGINALITY PROBLEMS IN USING CSS FRAMEWORKS

Yazarlar (Authors): Anar Musayev , Zülfikar Sayın 

Bu makaleye şu şekilde atıfta bulunabilirsiniz (To cite to this article): Musayev A., Sayın Z., “Css Çatılarının Kullanımında Karşılaşılan Sorunlar ve Çözüm Önerileri” *Int. J. of 3D Printing Tech. Dig. Ind.*, 7(1): 90-104, (2023).

DOI: 10.46519/ij3dptdi.1243701

Araştırma Makale/ Research Article

Erişim Linki: (To link to this article): <https://dergipark.org.tr/en/pub/ij3dptdi/archive>

CSS ÇATILARININ KULLANIMINDA KARŞILAŞILAN SORUNLAR VE ÇÖZÜM ÖNERİLERİ

Anar Musayev^a , Zülfikar Sayın^a 

^aHacettepe Üniversitesi, Güzel Sanatlar Fakültesi, Grafik Bölümü, TÜRKİYE

* Sorumlu Yazar: anarmusayev@gmail.com

(Received: 28.01.2023; Revised: 15.03.2023; Accepted: 25.04.2023)

ÖZ

Bu makalede, CSS kod kütüphanelerinin bir sonraki adımı olan CSS çatılarının (Framework) ortaya çıkış, kullanım nedenleri ve web geliştirme süreçlerine olan etkileri incelenmektedir. Web tasarımında CSS çatılarının kullanımı birbirinden ayırmayan web tasarımlarının ortaya çıkmasına neden olmaktadır. Bir grafik tasarım ortamı ve ögesi olarak CSS çatıları hem bir bütün olarak hem de öğelerine ayrılarak irdelenmektedir. Öğeler ayrı ayrı özgünlük ve işlev bağlamında incelenmekle birlikte aynı zamanda grafik tasarım ilkeleri bağlamında da irdelenmektedir. Söz konusu ilkeler bütünlük, farklılık, vurgu, denge, düzen, alan, karşıtlık, tartım (ritim), tamamlayıcılık, devamlılık, oran/orantı vb. olarak belirlenmiştir. CSS çatılarını oluşturan öğeler ve birbirleri ile olan ilişkileri ilgili ilkeler bağlamında incelenmektedir. CSS çatılarının öğelere ayrılıp incelenmesinde özellikle CSS çatı öğelerinin tasarımını etkileyen kullanıcı deneyimi (UX) ile kullanılan teknolojilerin etkileri ve sınırlılıkları göz önünde bulundurulmaktadır. İncelemenin başka bir ölçüt olan işlevsellik açısından değerlendirmeler yapılırken özellikle kullanıcı deneyimi ve teknik özelliklerin etkileri üzerinde durulmaktadır. Yapılan değerlendirmeler sonucunda elde edilen verilerden hareketle web tasarımında nitelikli grafik etkiler elde etmek için CSS çatılarının önemi ortaya konmakta ve bu bağlamda önerilerde bulunmaktadır.

Anahtar Sözcükler: Grafik Tasarım, Web Tasarımı, Arayüz Tasarımı, CSS, HTML, CSS Çatısı, Uygulama Çatısı, Kullanıcı Deneyimi.

USER EXPERIENCE AND ORIGINALITY PROBLEMS IN USING CSS FRAMEWORKS

ABSTRACT

This article examines the emergence of CSS frameworks which is the next step of CSS code libraries, the reason for their usage and the impacts on web development processes. The use of CSS frameworks in web design leads to the emergence of inseparable web designs. As a graphic design environment and element, CSS frameworks are examined both as a whole and by separating them into elements. While elements are individually examined for both genuineness and functionality, they are also examined in the context of graphical design principles. The principles are determined as integrity, explanation, emphasis, balance, order, area, contrast, rhythm, completion, continuity, ratio/proportion. The CSS framework elements and their relationships are examined in the context of the principles. In the analysis of CSS framework elements, the user experience (UX) that affect the design of CSS framework elements and the effects and limitations of the technologies used are especially taken into consideration. While making evaluations in terms of functionality, which is another criterion of the examination, especially the effects of user experience and technical features are emphasized. As a result of the analysis, it reveals the importance of CSS frameworks to obtain desired graphic effects in web design and suggestions are provided in this context.

Keywords: Graphic Design, Web Design, Interface Design, CSS, HTML, CSS Framework, Application Framework, User Experience.

1. GİRİŞ

Web tasarımında CSS çatıları kullanıcı deneyimine olan olumlu etkilerinden dolayı tercih edilirken; söz konusu çatıları kullanarak oluşturulmuş web sitelerinin özgünlük sorunları tartışılarda neden olmuştur. Çağdaş endüstri toplumlarında üretim başarısı; ürünün mümkün olan en kısa sürede, en düşük maliyetle ve kabul edilebilir kalite sınırlarında sunulabilmesi ile doğrudan ilgilidir. Bu ilke bir grafik tasarım ortamı ve ögesi olan web sitesi tasarımı için de geçerlidir. Web sayfalarında etkileşim gereksinimleri, farklı boyut ve aygıtlarda görüntülenebilme ve gereksinim duyulan sayfaların üretim hacminin katlanması gibi nedenlerden dolayı “terzi işi” web sayfalarının yerini CSS çatıları ile hızlandırılmış ve yüksek tutarlılıkta seri üretim web sayfalarına bırakmasına neden olmuştur. Bu durum farklı hedef kitlelere hitap eden ve farklı içeriklere sahip web sitelerinin benzer arayüzlere sahip olmaları gibi bir olumsuz sonuç ortaya çıkarmıştır. Shenoy ve Prabhu’ a göre CSS çatısı veya çerçevesi (Framework), web siteleri ve web uygulamaları geliştirme görevini kolaylaştıran önceden tasarlanmış bir dizi kavram, modül ve standartlaştırılmış ölçütlerin tümüne verilen addır (Shenoy ve Prabhu [1]). Bu çalışmada CSS çatılarının web tasarımına olan etkilerinin yanı sıra bu etkilerin sonuçları da değerlendirilmektedir. Bu çalışmada amaçlanan, konu ile ilgili sorunları belirlemek, nedenlerini açıklamak ve söz konusu sorunların çözümü için öneriler sunmaktır. Tasarım dili odaklı bir CSS çatısı geliştirmek web tasarım sürecine olan bakış açısını değiştirmek açısından önemlidir. Bu çalışma ile CSS çatılarının sadece yazılım sürecini iyileştirmek için değil aynı zamanda özgün ve kullanıcı dostu siteler geliştirmek için de önemini ortaya konması amaçlanmaktadır. CSS Çatısı framework teknolojisine alternatif olabilecek seviyede benzer teknolojiler bulunmadığından bu makalede bir karşılaştırmaya yer verilmemiştir.

2. TANIMLAR ve KAVRAMLAR

Bu çalışmada kullanılan ve açıklanmasına gereksinim duyulan terimler ile kavramlar aşağıdaki gibi tanımlanabilir:

Web Sitesi, ortak bir alan adıyla tanımlanan ve en az bir web sunucusunda yayınlanan web sayfaları ve ilgili içeriklerin tümü için kullanılan bir sözcüktür.

UX (User Experience'in / Kullanıcı Deneyimi), kullanıcıların algı ve davranışlarını etkilemek amacı ile onların belli bir kurum ile olan deneyimlerini etkileyecek öğelerin oluşturulması ve senkronize edilmesidir (Unger ve Chandler,[2]).

CSS, Cascading Style Sheets (Basamaklı Stil Şablonları ya da Basamaklı Biçim Sayfaları, bilinen kısa adıyla CSS), HTML’e ek olarak metin ve format biçimlendirme alanında fazladan olanaklar sunan bir işaretleme dilidir.

Web Sayfası ise internette ulaşılabilen; görüntü, komut dosyaları ve genellikle başka sayfalara bağlantıların yer aldığı hiper metin dosyaları için kullanılmaktadır. (The Free Dictionary, [3]).

UI (User Interface / Kullanıcı Arayüzü), kullanıcıların bir makine, cihaz, bilgisayar programı ya da karmaşık aletlerle etkileşimini sağlayan yöntemlerin bileşkesine verilen ad olarak tanımlanabilirken; İnternet Tarayıcısı için web sayfalarının görüntülediği bilgisayar yazılımıdır, denebilir.

CSS Reset Dosyası (veya reset.css), HTML elementlerinin, tarayıcılara özgü stil kodlarını sıfırlayan kod dosyası olarak tanımlanabilirken; Kod Küçültme, işlevini değiştirmeden tüm gereksiz karakterleri, yorumlanmış programlama dillerinin veya biçimlendirme dillerinin kaynak kodundan kaldırma işlemi için kullanılmaktadır.

Drupal, ücretsiz, açık kaynaklı bir içerik yönetim dizgesi (sistemi) ya da içerik yönetim odaklı bir altyapı yazılımı iken; Duyarlı Web Tasarımı (Responsive Web Design), bir web sitesinin kendisini herhangi bir aygıtta ya da ekran genişliğine uyarlamasını sağlamak için sayfa düzeni (mizanpaj) çalışmalarında uygulanan tekniklerin tümüne verilen ad olarak açıklanabilir (Eva Harb, [4]).

İçerik Yönetim Sistemi: CMS (İng. Content Management System), sayısal (dijital) içeriğin herhangi bir kod yardımına gereksinim duyulmaksızın üretilmesi ve yönetilmesine olanak sağlayan bir çevrimiçi web yazılımıdır. *Animasyonlu GIF*, tek bir dosyada bir dizi görüntü veya çerçeve içeren ve kendi grafik kontrol uzantısı tarafından işlenen, İngilizce

Grafik Değişirme Biçimi anlamına gelen Graphics Interchange Formating (GIF) kısaltması olan kodlanmış bir görüntü türüdür. WYSIWYG, İngilizce'de "What You See Is What You Get" teriminin baş harflerinden oluşan bir bilgisayar terimidir. Türkçesi "Ne Görüyorsan Onu Alırsın" demek olup ekranda görülene çok benzer bir çıktı alınacağı ortamları tanımlamaktadır.

W3C (İng. World Wide web Consortium), üye organizasyonların, tam zamanlı çalışanların ve halkın web standartlarını geliştirmek için birlikte çalıştıkları uluslararası bir topluluktur. Web'in mucidi ve Yöneticisi Tim Berners-Lee ve CEO'su Jefferey Jaff tarafından yönetilmekte olan W3C'nin misyonu, web'in tam potansiyeline ulaşılabilmesidir (W3C, [5]).

3. CSS ÇATILARI

Temel amaçları yazılımcının işini hızlandırmak olan CSS çatıları web sayfaları veya uygulamaları geliştirmek için kullanılacak araçların ve görsel öğelerin bütünü barındıran bir yazılım çatısı türüdür. Bilgisayar programlamasında, yazılım çatısı, genel işlevsellik sağlayan yazılımın, kullanıcı tarafından yazılan ek kodlarla seçici olarak değiştirilebildiği ve üzerine geliştirme yapılabilen temel kod kaynağıdır. CSS çatılarını incelemeyen önce genel olarak yazılım çatılarının ne olduğunu, özellikle nesne yönelimli kullanım nedenlerini anlamak gerekmektedir. Beck ve Johnson'a göre nesne yönelimli yazılım çatısı en genel tanımıyla, dizge ve alt dizgelerin, sınıflar ve bu sınıfların etkileşimi sayesinde tekrar kullanılabilir kod parçacıklarının bir bütünüdür (Beck ve Johnson, [8]). Yazılım çatıları çalışılan alan ve hedef platformlara göre üç ana öbekte (grupta) toplanabilir: i) mobil platformlar için geliştirilmiş çatılar, ii) veri bilimi ve makine öğrenimi için geliştirilmiş çatılar ve iii) CSS framework'lerin de dahil olduğu web uygulama yazılım çatıları.

CSS kullanımının yaygınlaşması, sadece HTML den oluşan web sitelerinin görsel zenginlik ve kullanıcı deneyimi açısından yetersiz kalması nedeniyle olmuştur. CSS sözcüğü Türkçeye Basamaklı Stil Şablonları ya da Basamaklı Biçim Sayfaları olarak çevrilebilir (İng. "Cascading Style Sheets"). İlk ticari web site olarak bilinen *Global Network Navigator* (GNN) web sitesini tasarlayan

Jennifer Niederst, buradaki "cascading" kavramını yukarıdan aşağıya birbirinin üzerine yazarak (bu aynı doküman için veya farklı dosyalar tarafından yapılabilir) oluşturulan ters hiyerarşiyi açıklamak için kullanıldığını belirtmektedir (Niederst, [6]).

CSS framework'lerin temelinde tasarım örüntüleri (İng. design patterns) ve kod kütüphaneleri (kod kitaplıkları) vardır. Bir kitaplık kullanmanın avantajı, kullanılmak istenen özelliğin veya çalıştırılmak istenen algoritmanın kütüphane kullanıcısı tarafından tekrar kodlanması gereksinimini ortadan kaldırmasıdır. Bunun yerine kütüphanenin yapısal tarafı için talimatlara uyularak sonuç elde edilebilmektedir (Meiert, [7]).

Günümüzde yaygın olarak kullanılan CSS Çatılarının birçoğu büyük web siteleri tarafından iç gereksinimler için geliştirilmiş ve daha sonra açık kaynak olarak internette paylaşılan projelerdir. Başlangıçta Twitter Baseline olarak adlandırılan Bootstrap birçok geliştirici için ilk kapsamlı CSS çatısı olmuştur. Mark Otto ve Jacob Thornton tarafından Twitter'da dahili araçlar arasında tutarlılığı teşvik etmek için bir çerçeve oluşturmak amacı ile geliştirilmeye başlanmıştır. Bootstrap'tan önce, kullanıcı arayüzü geliştirme için kullanılan çeşitli kütüphaneler, tutarsızlıklara ve yüksek bakım yüküne neden olmuştur. Tüm bu çalışmalar tek çatı altında toplanarak ilk olarak "Twitter Blueprint" daha sonra da 2011 Ağustos ayında Bootstrap olarak yayınlanmıştır (Otto, [9]).

CSS çatılarının ortaya çıkma nedeni olan büyük web projeleri birden çok ekip tarafından geliştirilen, bazen yüzlerce yazılımcı ve tasarımcının bir arada çalıştığı sürekli güncellenen ve gelişen kapsamlı projelerdir. Bir ekip tarafından oluşturulan bir modülün başka bir ekip tarafından kolaylıkla kullanılabilmesi ve hatta geliştirilebilmesi veya değiştirilebilmesi gerekmektedir. Günümüzde yaygın olarak kullanılan framework'lerin çoğu büyük web projeleri tarafından geliştirildikten sonra açık kaynak kod olarak paylaşılmışlardır. Twitter'ın Bootstrap ve Facebook'un React frameworkü en bilinen site, proje eş örneklerdendir (Çizelge 1). CSS Framework kullanımı yaygınlaştıkça, yazılımcı ve tasarımcılardan oluşan geniş camialar oluşmuştur. Bu da yazılım çatılarının test ve geliştirilmelerine katkı sağlamıştır. Herkese

açık olarak yayınlanan CSS çerçevelerin, onu kullanan, test eden ve geliştiren birçok geliştiricisi bulunmaktadır. Geniş topluluk tarafından geliştirilen ve test edilen kod hem güncel kalmakta hem de daha az hata içermektedir.

Çizelge 1. Farklı CSS çatılarının webde kullanım oranları. (bir web sitesinde birden çok CSS çatısı kullanılabilir). (Web Technology Surveys, [21]).

Bootstrap	% 78.9
Animate	% 39.3
Foundation	% 2.3
UIkit	% 0.7
Skeleton	% 0.6
Tailwind	% 0.5
Materialize	% 0.5
Material Design Lite	% 0.3
Semantic UI	% 0.3
Bulma	% 0.2
MetroUI	% 0.1

Geniş web tasarımcı kitleleri tarafından düzenli olarak framework'lerin kullanılması, iyileştirmek için bir yol bulunduğu veya karşılaşılan sorunlara çözüm üretildiğinde, geri bildirimlerle framework'lerin tasarımlara etkileri özyinelemeli hale gelmektedir (Bradley , [10]).

4. CSS ÇATISI KULLANIMININ AVANTAJLARI

CSS Çatısı kullanmanın en büyük avantajı geliştirme hızını ve üretkenliği artırmasıdır. Herhangi bir tasarıma (projeye) başlamadan önce üzerine geliştirme yapılabilecek temelin hazır olması çalışmaya başlama yolunda kolaylıklar sağlamaktadır. CSS çatısı kullanılmadan geliştirilen web sitelerinde projenin başlangıç süreci uzamakta ve farklı çözünürlüklerde tutarlılık sorunları ortaya çıkmaktadır. Özellikle, kullanılan tarayıcı çeşitliliğinin artması nedeniyle, oluşturulması gereken ilk proje dosyası artık reset.css olmaktadır . Farklı tarayıcılarda girdi alanları (İngilizce; input fields), açılır liste menüleri (İngilizce; drop-down), ipuçları, etiketler, tablolar, sıralı ve sırasız listeler farklı yorumlandığından CSS çatılarının sağladığı sıfırlama özellikleri reset.css dosyası gereksinimini de karşılayabilmektedir. CSS çatılarında uyarı pencereleri, bildirim iletileri, butonlar ve birçok tekrar kullanımına gereksinim duyulan tasarım öğelerinin hazır

bulunması nedeniyle doğrudan ve hızlı bir biçimde web sitesinin tasarımına başlanması kolaylaşmaktadır. Çok özel durumlar dışında web tasarımında ve genel olarak yazılım geliştirmede uygulama çatısı kullanmanın daha avantajlı olduğu söylenebilir. Standartlaşmış kod temelini oluşturulması hem bireysel hem de kurumsal kullanımda, CSS çatı kullanımının tercih edilmesinin en önemli nedenlerinden biridir. Özellikle geçmişe dönük bakım, destek çalışmalarında ve yeni özellikler ile modüllerin eklenmesinde CSS Çatısı kullanımı zamandan ve emekten tasarruf sağlayabilmektedir. Takım olarak çalışmayı kolaylaştırması, yazılım şirketlerinin CSS yazılım çatılarını tercih etmelerinin en önemli nedenlerinden biridir. CSS'e organize yaklaşım sağlaması ve şirket içi eğitim maliyetlerini düşürmesi başka bir nedendir. Yukarıdaki açıklamalar bağlamında denebilir ki; yazılım çatısı kullanmak hem takım olarak hem de bireysel kullanımda doğru tasarım alışkanlıkları geliştirilmesi açısından çokça önemli ve gereklidir.

Çizelge 2. meyerweb.com web sitesinden alınmış örnek CSS Reset Dosyası. (CSS Tools: Reset CSS. [22])

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
body {
    line-height: 1;
}
ol, ul {
```

```

        list-style: none;
    }
    blockquote, q {
        quotes: none;
    }
    blockquote:before, blockquote:after,
    q:before, q:after {
        content: "";
        content: none;
    }
    table {
        border-collapse: collapse;
        border-spacing: 0;
    }

```

5. CSS ÇATISI KULLANIMININ DEZAVANTAJLARI

Grafik tasarım açısından CSS Çatısı kullanımının istenmeyen en önemli sonuçlarından biri ortaya çıkan web sitelerinin özgünlükten yoksun olmasıdır denebilir. Bunun nedeni CSS çatısının, geliştirildiği platformun tasarım dilini taşıyor olmasıdır. Büyük web tasarımları için geliştirilen CSS çatılarındaki söz konusu tasarımların kurumsal izlerini silmek için teknik sınırlılıkların bilincinde olan tasarımcılar tarafından yapılan düzenlemeler ve bu düzenlemelerin koda uygulanması ile bu sorun aşılabilmektedir. Bu düzenlemelerin uygulama maliyeti yüksektir. Web tasarımında CSS çatısı kullanmanın getirisinin ötesine geçmesinden dolayı bu tarz özgünleştirme çalışmalarının çoğu CSS çatısının kullanımını sınırlamaktadır. Buna ek olarak belli bir tasarımın gereksinimleri için geliştirilmiş CSS çatıları kod yapısı ve modül içeriği bakımından tam olarak genel kullanıma uygun olmayabilmektedir. Buna karşın herhangi bir çatı kullanmadan yapılan geliştirmeler özgünlüğü kısıtlayaabilecek teknik sınırlılıklardan azat olmaktadır.

CSS çatılarının web tasarımına olan etkileri konusunda farklı bakış açıları söz konusudur. Bu etkiler kimilerine göre olumlu, kimilerine göre ise çeşitli olumsuzluklar bağlamında değerlendirilebilmektedir. CSS Çatılarında uygulanan grafik tasarımlarda karşılaşılan özgünlük sorunları grafik tasarımcılar açısından bir kayıp olarak görülürken, birçok geliştirici ise bu durumun kullanıcı deneyimine olumlu yansımalarını düşünmektedir.

CSS çatılarının yaygın olarak kullanılmasının en büyük etkisi duyarlı web tasarımlarının yaygınlaşması olmuştur. Özbahçe'ye göre Türkçe'ye "duyarlı" olarak çevrilen bu kavram, henüz evrensel kullanımı olan "responsive" kadar yaygınlaşmamıştır (Çelik Özbahçe, [11]). Bootstrap gibi CSS çatıları sayesinde birçok web tasarımcı farklı platformlarda çalışabilen akışkan ve hatasız web sitesi tasarlayabilme olanağına kavuşmuştur denebilir. CSS çatısının parçalarını kullanarak hızlı bir şekilde web sitesi oluşturma olanakları web sitelerinde özgünlük sorunlarına yol açabilmektedir. Öyle ki, bu bağlamda tüm sitelerin birbirinin benzeri haline geldiği eleştirisi sıklıkla ortaya atılmaktadır. CSS çatıları hazır birçok görsel modülü sunarken, bu modüllerin kullanımını zorunlu da kılmamaktadır. Tasarlanan web siteleri için özgün bir kurumsal kimlik oluşturmak yerine doğrudan CSS çatılarının sunduğu öğeleri kullanmak, tasarımcıların sıklıkla tercih ettiği özgünlükten yoksun uygulamalardır. İşe yeni başlayan duyarlı web tasarımcıların veya web sitesi tasarlayan duyarlı yazılım uzmanlarının CSS çatısı kullanması; zaman zaman özgün olmayan web sitelerinin ortaya çıkması sonucunu doğurmakta ise de hem görsel hem de işlevsel anlamda çok da kötü olmayan web sitelerinin tasarlanmasına da yol açabilmiştir. Denebilir ki; CSS çatılarının kullanımı bir kartopu etkisi yaratarak hem CSS çatılarını hem de web tasarım ilkelerini değiştirebilmektedir.

Web sitelerinin kullanılan aygıtlara göre kendini şekillendirebilmeleri genel olarak sayfalardaki arayüz yapılarının sağladığı esneklikler ile sağlanabilmektedir. Çünkü, aygıtlara özellikli davranış gösterme eylemleri, web sitelerini oluşturan sayfalardaki arayüz yapılarının kendilerini aygıtlara uydurması olanağı ile de gerçekleştirilebilmektedir.

Web siteleri, genellikle belirli aygıtların görüntülük (ekran) boyutlarına göre geliştirilebilmektedir. Çünkü tasarımcılar, hangi boyutta uygulama geliştireceklerine, hedef kitlelerinin yoğun olarak kullandığı aygıt ya da dünya genelinde en yoğun olarak kullanılan aygıt boyutuna göre karar vermektedirler. Aygıt boyutuna göre uygulama geliştirme sürecinde birden fazla görüntülük boyutundan söz edildiği zaman ise, işler biraz daha karmaşıklaşabilmektedir. EWT (Esnek Web Tasarımı) tanımı ortaya çıkmadan önce

tasarlanan web siteleri, ziyaretçilere, çoğunlukla masaüstü bilgisayar ortamında hizmet verirken, zamanın popüler görüntülük boyutuna göre tasarlanmışlardır. Ancak, zamanla İnternet erişim özelliği bulunan farklı aygıtların da insan yaşamına girmesi bu durumunun değiştirilmesinin neden olabileceğidir. Bu nedendir ki, farklı aygıtların kullanım oranlarının artması ve özellikle de mobil aygıtların oldukça yoğun bir şekilde kullanımı, EWT'nin gitgide daha fazla önem kazanmaya başlamasına neden olmuştur (Budak ve Gezer, [12]). CSS çatılarının kullanımı duyarlı web sitelerinin tasarlanmasını hem yaygınlaştırmış hem de kolaylaştırmıştır. Esnek tasarımların akışkan yapılarının yetersiz kaldığı durumlarda, yani geniş yelpazede çeşitlenen boyutlara sahip mobil aygıtlar için CSS3 Medya sorguları (İngilizce; Media Queries) kullanılması gerekmektedir.

Medya sorguları, belirli bir tür görünüm için bir öğeye birden fazla CSS atanmasına olanak tanıyan bir sorgu türüdür. Bu sorgular, görüntülük tipine ve çözünürlüğüne göre stilleri yeniden yapılandırarak genişlik, yükseklik, yön ve hatta çözünürlük gibi özellikleri değiştirmeyi kolaylaştırabilmektedir (Turan ve Şahin, [13]). Dolayısıyla, akışkan tasarım ve mobil uyumluluk sorunları CSS çatılarında çözümünü bulmuş sorunlardan olmuştur.

Yazılım kütüphaneleri ve CSS çatılarının eskiden beri en çok dile getirilen bir sorunu da gereğinden fazla kod barındırmaları olmuştur. Bazen kod kütüphanesinin sadece %5'ne gereksinim duyulmasına karşın, bu kütüphanenin tüm uygulamaya dahil edilmesi gerekebilmektedir. Ama bu sorun, günümüzde bilgisayar ve telefonların işlem ve görüntü işleme güçlerinin artması ve internet hızlarının kod temelli dosyaları yüklerken yeterli seviyede olmasından dolayı -özel durumlar hariç- ciddi bir dezavantaj olarak görülmemektedir. Ayrıca CSS çatısı kodları genellikle kod küçültme (İngilizce; minification) işlemi yapılarak dağıtılmaktadır.

6. YAYGIN KULLANILAN CSS ÇATILARININ KARŞILAŞTIRILMASI

CSS çatılarının geliştirilme nedenleri, kullanım amaçları, hedef kitleleri ve ortaya çıkma öyküleri farklı olduğundan birçok CSS çatısının birbirleri ile kategorik karşılaştırılması pek doğru olmayabilmektedir. Geliştirici ve

tasarımcılar framework tercih ederken önceliklerini belirleyip onlara en uygun CSS çatısı ile devam etmeyi seçmektedirler. Bu kararlar bireysel veya kurumsal olmalarından bağımsız olarak gerçekleştirilecek tasarımın kapsamına göre belirlenmektedir. Eğer kullanılan teknoloji güncel ise genellikle en yaygın olan uygulama çatısını seçmek bazı durumlarda özgünlükten ödün vermek anlamına da gelebilmektedir. CSS çatısının yaygın olarak kullanılıyor olması ve özgün web siteleri oluşturabilmeye uygun olması istenen en temel özelliklerdir. Bu iki ekten birbirini sınırladığından dolayı, bu kısır döngü genellikle gelişen teknoloji ile çözülmektedir.

CSS çatısı kullanmada iki temel yaklaşım söz konusudur. Bunlardan birincisi CSS çatısını temel alıp onun mevcut sınıflarının üzerine özelleştirilmiş sınıflar yazarak ilerlemektir. Söz konusu yaklaşım genellikle daha geleneksel CSS çatılarının kullanıcıları için geçerlidir ve sonuçta ortaya çıkan siteler büyük farklılıklar gösterebilmektedirler. Diğer bir yaklaşım ise CSS çatılarındaki sınıfları tasarım genelinde eksiksiz kullanmaktır. Bazı durumlarda söz konusu ön tanımlı çatı sınıfları yetersiz kalabilmektedir. Böylesi durumda ise kullanıcı topluluklarının veya üçüncü tarafların sağladığı eklentiler kullanılmaktadır.

Tüm CSS çatıları sundukları özellikleri farklı şekillerde sınıflandırmaktadırlar. Bunun nedeni bazen CSS çatısının tarihsel gelişimi bazen de çatıyı geliştirenlerin anlamsal (semantik) bakış açısıdır. Bazı CSS çatısı belgelerinde listelenen sınıflandırmalar aşağıdaki gibi sıralanabilir.

Aeon: Sadece ızgara sisteminden oluşmaktadır.

Bootstrap: Yerleşim, İçerik, Formlar, Bileşenler, Yardımcılar, Araçlar.

Bulma: Sütunlar, Elemanlar, Bileşenler, Formlar, Yerleşim, Yardımcılar.

Materialize CSS: Bileşenler, JavaScript Öğeleri, Forumlar, Mobil, Temalar.

Milligram: Tipografi, Blok Alıntılar, Butonlar, Listeler, Formlar, Tablolar, Izgara, Kod, Araçlar.

Pure CSS: Izgara, Formlar, Tuşlar, Tablolar, Menüler, Araçlar, Özelleştirme.

Semantic UI: Elemanlar, Koleksiyonlar, Görüntüleme, Modüller, Davranışlar.

Tailwind CSS: Özelleştirme Araçları, Temel Stillere, Yerleşim, Flexbox ve Izgara, Boşluk, Ölçülendirme, Tipografi, Fonlar, Çerçeveler,

Efektler, Filtreler, Tablolar, Geçiş ve Animasyonlar, Dönüşümler, Etkileşimler, SVG, Erişilebilirlik, Eklentiler.

UIKit: Bileşenler.

YAML: Normalleştirme Bileşenleri, Düzen, Izgara Sistemi, Sütün Yapısı, Form Elemanları, Yüzen Öğeler, Erişilebilirlik.

ZURB Foundation: Araçlar, Tipografi, Kontroller, Navigasyon, Konteynerler, Media, Eklentiler, Kütüphaneler.

Bu listeden de anlaşılacağı gibi, CSS çatılarının web tasarımına yaklaşımları arasında birçok ortak ve ayrışan nokta bulunmaktadır. Bu durum CSS çatılarının işlev ve özgünlük bakımından farklı ve birbirlerinin eşdeğeri olmadıklarını göstermektedir.

CSS Çatılarının gelişimi Izgara sistemlerine ihtiyaç duyulması ile başlamaktadır. Tüm CSS çatılarının temellerini ızgara sistemleri oluşturmaktadır. 960 Grid Sistem, yaygın olarak kullanılmaya başlanan ilk ızgara sistemidir (960 Grid System, 2021). Söz konusu sistemin ilk yayınlandığı dönemde ekran çözünürlükleri günümüzdeki gibi çok yüksek değerlerde olmadığından maksimum genişliği 960 piksel olarak sabitlenmiştir. Bu değer belirlenmesindeki başka bir etken de 960'ın , 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 32, 40, 48, 60, 64, 80, 96, 120 , 160, 192, 240, 320 ve 480 ile bölünebilmesidir. Daha sonra geliştirilen birçok CSS çatısında da benzer bir yol izlenmiştir.

Özellikle hem üçe hem de ikiye bölünebildiği için 12'lik sistem 2000'lerin ortalarında yaygın olarak kullanılan Drupal gibi içerik yönetim sistemlerinde de kullanılmıştır. Bootstrap'ın 12'lik ızgara sisteminin temelinde de benzer bir mantık vardır (Şekil 1). Bu ızgara sistemlerinin kullanımı CSS çatılarının günümüz web sitelerinin plan ve yerleşimini önemli derecede etkilemiştir. Birçok web sitesinin benzer oranlara sahip olmasının temel nedeni Bootstrap ızgara sistemidir. Bu şekilde sayfaların bölünmesi tasarımın bütünlüğüne, ritmine ve öğelerin birbirileri olan oran ilişkilerini olumlu olarak etkilemektedir.

.col-md-8	.col-6.col-md-4	
.col-6.col-md-4	.col-6.col-md-4	.col-6.col-md-4
.col-6	.col-6	

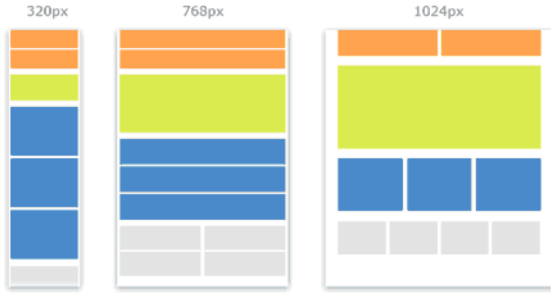
Şekil 1. Bootstrap 5'in farklı genişliklerdeki kolonlarına örnek

(<https://getbootstrap.com/docs/5.0/layout>).

7. CSS ÇATILARININ ÖĞELERİ BAĞLAMINDA İNCELENMESİ

7.1. Platformlar arası uyum

Platformlar arası uyum sorunu CSS Çatılarının çözüm getirdiği sorunların başında gelmektedir. Duyarlı tasarım sadece ana ızgaranın ölçülendirilmesinde değil, buton, menü, görsel gibi diğer tüm öğelerin tasarımında da belirleyici olmuştur. İçerik kolonlara yerleştirilirken içeriğin türüne göre farklı boyutlandırmalar gerekebilmektedir. Örneğin bir metin yerleştirilirken satırların çoğalıp metnin boyunu uzatması sebebiyle metin kesme işlemini ön yüzde yapmak gerekebilmektedir. Görseller boyutlanırken en boy oranını koruyarak boyutlandırmak önemlidir. Önceleri, tasarımların uyum sağlaması gereken aygıt boyutları bu kadar fazla değilken, her boyut için özelleştirilmiş tasarımlar yapılmış ve yazılımlar uyarlanmıştır. CSS medya sorguları böylesi sorunların çözümü için geliştirilmiştir. CSS sorguları tek başına yeterli olmayacağından, ölçü birimi olarak piksel ve punto yerine yüzde kullanılmaktadır. Bu tasarım yaklaşımı duyarlı web tasarımı (İng. Responsive Web Design) olarak adlandırılmaktadır. Duyarlı bir web sitesinde içeriklerin yer aldığı paneller belirlenmiş alanı tam kaplayacak şekilde yerleştirilmektedirler. Eğer içeriği göstermek için yatayda yeterli alan bulunmazsa sağda bulunan öğe soldakinin altına geçerek yine yataydaki genişliği doldurmaktadır (Şekil 2). Bu yerleşim tarzı da sadece mobil cihazlarda değil genel olarak web tasarımında kullanıcı deneyimini büyük ölçüde olumlu etkileyen dikey akışın önemini artırmıştır.



Şekil 2. Farklı çözünürlüklerde web sayfasını oluşturan blokların yerleşiminden görüntüler.

7.2. Dikey Akış

İnsanlar günlük yaşamlarında genellikle yatay düzlemde hareket ettiğinden, ufka baktıklarında gözlerini sağdan sola ya da soldan sağa hareket ettirmektedirler. Dolayısıyla ilk web sayfaları da yatay monitörlere uygun olarak yatay tasarlanmıştır. Mobil aygıtların gelişimi ile bu durum değişmiştir, tek parmak hareketi ile tüm içeriği dikey gezmek zorunluluğu ortaya çıkmıştır. İlk başta mobil aygıtlar için seçenekli tasarımlar geliştirilirken özellikle CSS çatılarının yerleşimi ile standartlaştırma çabalarından dolayı bütünlüklü çözümler ortaya çıkmıştır. Bunun sonucunda *dikey akış* web sitelerinin genelinde yaygınlaşmıştır.

Dikey akışın CSS çatılarında temel yönelim olmasının nedeni sadece mobil aygıtlarla uyumluluğu yakalamak değildir. Konuya psikolojik açıdan bakıldığında da öğelerin görsel olarak nasıl algılandığıyla ilgili en ünlü kuramlardan biri, aslen 19. yüzyılın sonlarında Almanya'da ortaya çıkan psikolojik bir kavram olan Gestalt ilkesidir. "Gestalt'ın Kapanış Yasası" gereği yatay tasarımlarda kapalı bir blok içerisinde içeriğin devamı olmadığı var sayılır ve daha fazla bilgi aranmaz. Sonuç olarak sayfanın geri kalanı gözden kaçırılabilir (Margalit, [14]). Çok geniş ekranlarda CSS çatılarının ızgara sistemleri kullanılarak kullanıcılara içeriğin bitmediği mesajı verilmektedir. Gözün sürekli kesilmiş görselleri tamamlama dürtüsü ile sayfa kullanıcı tarafından kaydırılmaya devam edilmektedir. Böylece yatayda yan yana duran içerikler bile dikey akışta bir devamlılık sağlayabilmektedir. Dikey akışı sağlamak için dikdörtgen kutular ve yatay kurallar, içeriğin sona erdiğinin görsel bir göstergesini ekler ve aksi takdirde sonraki içerik öğesini görüntüleyebilecek dikey pikselleri kullanır. Bu öğeleri ortadan kaldırmak ve sayfanın dikey boyutunu daraltmak, ziyaretçinin daha fazla kaydırma

olasılığını arttırmaktadır. Web sayfalarının dikey bir akış içerisinde yer alması, ritim, devamlılık ve denge bağlamında incelendiğinde grafik tasarım açısından olumlu sonuçlar doğurmakla beraber tasarımın özgün ve etkileyici görünmesini sağlayan zıtlık ve vurgu ilkelerinden ödün verilebilmektedir.

7.3. Konteynırlar

CSS çatılarının web tasarımında kullanımının yaygınlaşmasına önyak olan başka bir kavram "konteynır"dır. Konteynır kullanılmadığı zaman sayfadaki metin içeriklerinin gazete sayfalarında olduğu gibi görsel içeriklerin etrafını sarması ve boşlukları doldurması genel kabul gören tasarım yaklaşımıdır. CSS çatı kullanımı metin içeriklerinin ayrı bir blok içinde görsellere denk değerlendirilmesini sağlamıştır. Konteynır terimi bazı CSS çatıları tarafından içeriğin bulunduğu tüm öğeler için kullanılırken, diğerleri tarafından sadece içerik ve öğeleri kapsayan en dış çerçeveler için kullanılmaktadır. Dış çerçeveler yeniden boyutlanabilir ana öğeler oldukları için, sitelerin ekran boyutlarına göre yerleşime ve içindeki öğelerin kendi aralarındaki düzenine etki etmektedir. Konteynırların konumları ve görevleri kullanıcılar tarafından iyi bilinmekte bu da sayfaların kullanılabilirliğini artırmaktadır. Kanvası bölerek oluşturulan konteynırların yanı sıra, bir diğer konteynır türü de 'sticky on scroll' konteynırlardır. Bu konteynırlar, sayfanın üstünde, altında veya kenarında bulunan ve referans noktaları sayfanın kendisi olan "kanvas"ın bölünmesiyle oluşan aksine, tarayıcı kenarında bulunan yapışkan konteynırlardır. Özellikle mobil aygıtların yaygınlaşması ile aktif olarak web tasarımında kullanılmaya başlanmış ve CSS çatılarının önemli parçalarından biri haline gelmişlerdir. Bu tür konteynırlara örnek olarak mobil sayfalardaki ana menü konteynırları veya sayfanın bir kenarında bulunan iletişim butonları verilebilir (Şekil 3). Bu ve benzeri standartlaşmalar birbirine benzeyen ama kullanıcıyı hiç zorlamayan web şablonlarının oluşmasına neden olmuştur. Bu düzende bütünlük, denge, tartım (ritim) ve alan kullanımı gibi grafik tasarım ilkelerine uyum sağlanırken, farklılık ve vurgu gibi tasarımın ayırt ediciliğini artıran grafik tasarım ilkelerinden ödün verilmektedir.



Şekil 3. Rollerein.de sitesinin mobil görünümünde sağ altta yer alan telefon simgesi.

7.3. Tipografi

CSS çatıları ve CMS'ler dinamik metin içeriklerinin web sitelerine eklenme yöntemlerini de etkilemiştir. Dinamik sitelerde içerik veri tabanından iki yöntemle gelmektedir. Bunlardan birincisi WYSIWYG aracı kullanılarak yapılan girdilerin gösterilmesidir. Son kullanıcılar tarafından özellikle diğer tasarım kurallarını ezecek şekilde oluşturulan içeriklerin CSS çatıları tarafından formatlanması çok zordur. Genel görünüm için yönetim paneli kullanıcısı tarafından yapılan bazı düzenlemelerin sayfayı şekillendirmesine izin verilmekle beraber özellikle mobil görünüm için birçok tasarım düzenlemelerinin geçersiz kılınması gerekmektedir.

İkinci bir yöntem ise web tasarımı açısından daha sağlıklı olan, içeriklerin veri tabanına saf metin olarak kaydedilmesi ve gösterilirken tümüyle CSS çatılarına ait sınıflar tarafından şekillendirilmesidir. Böylece son kullanıcılar, ellerindeki yönetim paneli araçları ile genel tasarımın bütünlüğünü bozmamış olurlar. Kullanıcıların sınırlandırılması aynı zamanda içeriğe yoğunlaşmalarına yardımcı olduğundan, daha etkin bir yönetim deneyimi sağlamasına katkıda bulunmaktadır. CSS çatısı kullanımı böylece web sitelerinin bütünlüğüne de olumlu katkı sağlamış olmaktadır.

Metin temelli içeriklerin gösterilmesinde Tüm CSS çatıları var olan HTML etiketlerini aktif olarak düzenlemektedirler. Bunlar <h1>, <h2>, <h3>, <h4>, <h5>, <h6> gibi hiyerarşik başlık etiketleri, <p>, <mark>, , <s>, <ins>, <u>, <small>, ve gibi temel düzenleme etiketleri ve , , gibi özel listeleme etiketleridir. Bu etiketler genellikle temel hiyerarşi ve karakter korunarak minimum düzeyde şekillendirilir, zira bu tür ana öğeler sitenin tasarımını temelden etkilediğinden tasarımcı tarafından düzenlenmeye bırakılmadıklarıdır. Tema veya web sitesi tasarımcısı tarafından özelleştirilmeden bırakılan temel etiketler sitelerin özgünlüklerini kaybetmelerinde en önemli etkidir. Temel HTML etiketlerin yanı sıra yardımcı sınıflar da birçok CSS çatısı tarafından kullanılmaktadır. Metin hizalama, özel paragrafların ayrıştırılması, temel listelerin yanı sıra özelleştirilmiş listelerin oluşturulması ve alternatif başlık hiyerarşilerinin belirlenmesi için de yardımcı sınıflar kullanılmaktadır. CSS çatılarının HTML içeriğindeki tipografik öğelerin hiyerarşik yapısıyla uyumlu çözümler sunduğundan dolayı web tipografisinde bütünlüğü, dengeyi, zıtlığı ve vurguyu hem CSS çatısı öğeleri kullanarak hem de çatı üzerine ekleme yaparak sağlamak mümkündür.

7.4. Tablolar

Yine temel HTML öğelerinden olan tablolar, web tasarım tarihinde çok önemli bir yere sahiptir. İnternetteki içeriğin sadece metinlerden oluştuğu dönemde, içeriğin dikey ve yatay yerleşimini HTML tabloları sağlamışlardır. Modern web tasarımında satır ve sütunların oluşturulmasında HTML tablolar kullanılmaktadır. Tablonun ilk sütunu ve ilk satırı diğerlerinden ayrıştırılarak ve satırları tek ve çift olarak renklendirerek temel görünüm sağlanmaktadır. Tabloların yapısı neredeyse tüm CSS çatılarında aynıdır. Farklılıklar genellikle isimlendirme ve renklendirmeler ile sınırlıdır. JavaScript kullanılarak listeleme ve filtreleme özellikleri de eklenmektedir. Tabloların mobil aygıtlarda görünümü en sık karşılaşılan sorunlardan biridir. Bu yerleşim sorunlarının iki temel çözümü vardır. Birincisi aslında sorunu görmezden gelerek, tabloyu yatayda sürüklenerek şekilde yerleştirmektir. Bu çözüm mobil sitelerdeki dikey akışı bozduğundan daha az tercih edilmektedir. Ayrıca tabloların bu şekilde yerleştirilmesi denge, oran/orantı ve alan gibi grafik ilkeleriyle

ters düşmektedir. Diğer bir çözüm tabloyu mobil görünüm için yeniden tasarlamaktır. Sütunları kaldırıp içerikleri kartlar olarak satıra yerleştirerek ya da bazı içerikleri kapatıp açarak yatay bir tabloyu dikey ekrana yerleştirmek mümkündür (Şekil 4).

Kullanıcı No	İsim	Soyad	Yaş
0000001	Anar	Musayev	40
0000002	Ayşe	Kaya	37
0000003	Mehmet	Yılmaz	55
0000004	Fatma	Öztürk	28
0000005	Ahmet	Kara	95

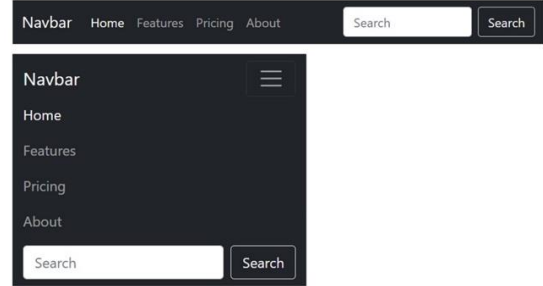
Şekil 4. Farklı ekranlarda tablo görünümü örneği.

CSS çatılarının bileşen listeleri büyük farklılıklar göstermektedir. Bazı CSS çatıları internette kullanılabilecek tüm modülleri kapsarken diğerleri oldukça sınırlı ana bileşen havuzuna sahiptir. Kullanılan çatının geliştirilecek olan web sayfasının kurumsal kimliği üzerindeki etkisini azaltmak için bileşenlerin tasarım kodlarının küresel dosyalara yazılması önemlidir. Yazı karakteri/boyutu ve renk paleti gibi özelliklerin tek bir yerden yönetilebilir olması CSS çatılarının kullanım kolaylığını artırmakla beraber, düzenlenmesi ek iş yükü getirmeyeceğinden tasarlanan web sitesinin özgünlüğünün korunmasını da kolaylaştırmaktadır.

7.5. Menü ve Bağlantılar

CSS çatı bileşenlerinde hiyerarşik olarak en başta menüler gelmektedir. Menülerin yerleşimi -katı kurallara bağlı olmasa da- web sitelerinde genellikle sol veya en üstedir. Bunun nedeni ekranın alt bölümünün işletim sistemlerinin, yönetim çubukları tarafından sahiplenilmiş olmasıdır. Yatay konumlandırma ise sayfa dilinin alfabetinin yazım yönü ile

alakalıdır; başka bir deyişle menüler genellikle ekranın sol tarafında yer almaktadır. Mobil tasarımda bu yerleşim ergonomik sebeplerden dolayı tümüyle değişmiştir. İşletim sistemi bildirim ekranları tepededir ve kapalıdır. Bu durum butonların bir kısmının sayfanın altında ve yatayda çok yer tutan menü bağlantılarının da hamburger ikonu olarak da bilinen ve üç yatay çizgiden oluşan tek bir simgenin altında gruplanmasına neden olmaktadır. Bu tercihte yazı yönünden ziyade insanın alet kullanmasında önemli rolü olan başparmak etkili olmuştur. Bazı CSS çatıları, geliştiricilere bağlantıların yanı sıra arama alanları ve özel bazı butonlar yerleştirme olanağı da sunmaktadır. Her durumda mobil görünüm için yeniden yerleşim yapılmaktadır (Şekil 5). Bu şekilde menülerin CSS çatıları tarafından standartlaştırılması farklılık, vurgu, alan, zıtlık (karşıtlık), ritim (devamlılık) gibi grafik tasarım ilkeleri bağlamında değerlendirildiklerinde sonucun olumlu olduğu pek söylenememektedir. Ama kullanıcı deneyimi açısından bu yapının vazgeçilmez olduğu kabul edilmektedir.



Şekil 5 . Bootstrap Navbar, bilgisayar ve mobil aygıtta görünümü (Bootstrap, 2018).

Metin ve büyük görsellerin yanı sıra web tasarımında simgelerin özel bir yeri vardır. Birçok simge takımı farklı CSS çatısı ile uyumlu çalışmaktadır. Bunlardan en yaygın olarak kullanılanı Font Awesome'dur. İlk versiyonlarında adından da anlaşılacağı gibi simgeleri font karakteri olarak kullanıma sunan Font Awesome en güncel versiyonunda SVG dosya formatının tüm olanaklarından yararlanmaya olanak tanımaktadır. Sonuç olarak, ister CSS çatıları için tasarlanmış olsun, ister bağımsız proje olarak geliştirilmiş olsun, genellikle simgeler çapraz olarak kullanılabilir durumdadır. Tasarımdaki özgünlüğün korunması adına simgelerin tasarıma özel olarak hazırlanması doğru seçenek olarak görülmektedir.

7.6. Uyarı ve Mesaj Pencereleri

Uyarı ve mesaj pencereleri kullanıcıları bilgilendirmek için kullanılan genellikle metinden ve bazen de bilgilendirici simgelerden oluşan öğelerdir. Söz konusu öğeler genellikle yapılan bir işlem sonucu veya işlem öncesi uyarı ve bilgilendirilme amacı ile yerleştirilmektedir. Bazı özel durumlarda üzerlerinde küçük butonlar da yer alabilmektedir. CSS çatılarının büyük bir bölümünde bulunan bu uyarılar; bilgi (İng. info), dikkat (İng. alert), başarı (ing. success), temel (İng. primary), uyarı (İng. warning), tehlike (İng. danger) gibi anahtar sözcükler ile adlandırılarak kullanım amaçlarına uygun renklerle eşleştirilip sınıflandırılmaktadır. Semantic UI örneğinde olduğu gibi sadece renklerin İngilizcedeki isimleri de kullanılabilir. Bu tür uyarı pencereleri gereksinim duyulan alana yerleştirilmek için <div> etiketleri içerisinde oluşturulmaktadır.

Web 2.0 ile yaygınlaşan etkileşimli web sitelerinin kullanıcıdan girdi almasının en önemli yollarından biri web formlardır. <input>, <textarea>, <select> <option>, <checkbox>, <radio>, <file> form etiketleri ile oluşturulan form elemanları <label> etiketi ile isimlendirilmektedir. Kullanıcıyı yönlendirmek için en basit şekliyle girdi alanlarının üzerine yazılan <label> etiketleri bazen aynı satırda bazen de yüzen etiket olarak tasarlanabilmektedir. Girdi alanları sade metin temelli girdiler dışında aralık, tarih, zaman ve renk gibi seçiciler için özelleştirilmiş tasarımlarla gelmektedir. Form alanları doldurulurken birçok kullanıcının yetersiz yönlendirilmesi kullanıcı deneyimini negatif etkilemektedir. Boş bırakılamaz alanlar için uyarılar, girilen içeriğin türünün uygunluğu gibi ön denetimlerin yapılması gerekmektedir. CSS çatılarında JavaScript ile yazılmış doğrulama araçlarının tespitlerini uyarı olarak kullanıcıya iletmek amaçlı renklendirme ve uyarı/bilgilendirme metinleri kullanılmaktadır.

7.7. Tuşlar veya Butonlar

Tuşlar veya butonlar web sitelerinde hem sayfalar arası yönlendirmeyi hem de zamanlanmış görevler hariç, yürütmeyi sağlayan önemli öğelerdir. CSS Çatılarında farklı görevleri üstlenen ve hem görünüm hem de işlevsellik bakımından farklılık gösteren tuş birim ve grupları bulunmaktadır. Tuşlar genellikle üzerine gelince (hover) renk veya

form değiştiren içerisinde metin ve/veya simge barındırabilen <input>, <a>, <button> etiketleridir. Gelişmiş CSS Çatılarında değişimler canlandırmalar (animasyon), hareketlendirmeler ile gerçekleştirilmektedir. CSS çatılarında semantik veya kurumsal renkler doğrultusunda renklendirme olanakları sunulmaktadır.

7.8 Etiketler

Farklı CSS çatıları tarafından İngilizce “tag”, “label”, “ribbon”, “badge” veya “tooltip” olarak adlandırılan, içeriğe ikincil bir açıklama olarak eklenen çeşitli etiket türleri vardır. Küçük bir görsel ile desteklenen etiket, kullanıcı listeleme ekranında veya herhangi bir listede renklendirmenin de yardımıyla öğeleri ayırtmaya yardımcı olabilmektedir.

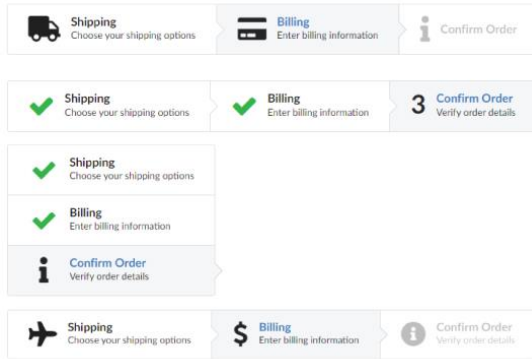
CSS çatıları tarafından animasyonlu GIF veya daha modern bir yaklaşım olan CSS canlandırmaları kullanılarak oluşturulan hareketli ilerleme veya önyükleme göstergeleri (İng. Preloader, Loader, Progress Bar, Spinners) farklı çatılar tarafından farklı isimlendirilmektedir. Özellikle AJAX kullanılarak içerik oluşturulurken, simge ve yüklenme çubukları söz konusu içeriğin yükleniyor olduğu bilgisini vermek için kullanıcı deneyimi göz önünde bulundurularak tasarlanmıştır.

7.9. Sayfalama ve Yönlendirme Bileşenleri

Web tasarımlarında kullanıcıyı yönlendirmede kullanılan bir başka önemli uygulama da sayfalama bileşenleridir. Genellikle tablolarda sayfalara numara koymak için kullanılan sayfa numaralandırmalarında önemli görevler üstlenen CSS çatılarında kartlardan oluşan haber başlıkları gibi içerikler de sayfalanamaktadır. Bir genel geçer olarak, sırasız liste etiketleri içerisine yerleştirilen (,) numaralı sayfa ve yönlendirme bağlantılarından da oluşan CSS çatılarının neredeyse tümünde benzer yapı ve görünümler söz konusudur.

Grafiksel kontrol ögesi olan Breadcrumbs (Türkçe: Ekmek Kırıntısı), bir sayfanın ara yüzünde kolay gezinmeyi sağlayan bir çeşit gezinti menüsüdür. Bu nitelikle Hansel ve Gretel masalında iki kardeşin eve dönüş yolunu hatırlamak için yerlere serptiği ekmek kırıntılarında faydalanmalarına bir göndermedir (Levene, [15]). Breadcrumb'lar

Sayfalama aracında olduğu gibi neredeyse tüm CSS çatılarında benzer yapı ve tasarıma sahiptir. Ya “/” ya da “>” simgesi ile ileri sayfa hiyerarşisi gösterilmektedir. Örneğin *Anasayfa / Hakkımızda / Vizyonumuz* vb. Burada en sağda yer alan ve genellikle diğerlerinden yazı karakteri ağırlığı veya renk bakımından farklı olan öge aktif sayfanın adıdır. Genellikle <a> etiketleri içerisine yerleştirilen bağlantılar aktiftir. Modern interaktif sitelerde kullanımı yaygınlaşan başka bir sayfalama ögesi de adımlardır. Adımlar aşamalı işlemlerin sayfanmasında kullanılmaktadır. CSS Çatılarının birçoğu bu alanda yeterli olmadığından genellikle framework dışı çözümler aranmaktadır. Başarılı örneklerden biri Semantic Framework’ü tarafından sağlanan Step bileşenidir. Sırasız liste etiketleri içerisine yerleştirilen (,) bileşenleri duyarlı tasarım (İng. responsive) özelliğindedir (Şekil 6).



Şekil 6. *Semantic UI adımlar bileşeni (Semantic UI Steps, [16]).*

Yukarıda söz edilen bileşenler (uyarılar/mesajlar, form bileşenleri, tuş/tuş grupları, etiketler/rozetler, sayfalama bileşenleri) CSS çatılarının çoğu tarafından büyük ölçüde sağlanmaktadır. Bu bileşenlerin kullanılması kullanıcı deneyimini olumsuz etkilese de doğru tasarlanmış özgün bileşenlerle değiştirilmeleri web tasarımlarını daha özgün ve bütünlük, farklılık, vurgu, karşıtlık, ritim, devamlılık ve oran/orantı gibi grafik tasarım ilkeleri ile de uyumluluk göstermektedir.

7.10. Açılır Pencereleler

Uyarı, yönlendirme, vurgulama ve sayfalama gibi bileşenlerin yanı sıra birçok CSS çatısı seçenekli içerik gösterimi için açılır pencereler, genişleyen ve kayan öğeler ve kartlar gibi bileşenler sunmaktadır. Aktif sayfayı terk etmeden kullanıcıya daha fazla bilgi vermenin yaygın bir yolu farklı açılır pencereler

kullanmaktır. Web tasarımının gelişimi sürecinin ilk yıllarında tarayıcılar tarafından sağlanan açılır uyarı pencereleri kullanılmıştır. Genellikle otomatik açılan ve reklam gibi istenmeyen içerikler içeren bu pencereler pop-up olarak adlandırılmıştır ve kısa sürede yine tarayıcı ile antivirüs yazılımları tarafından engellenmeye başlanmıştır. Bu dönemde modal pencerelerin kullanımı yaygınlaşmıştır. Bootstrap tarafından yaygınlaştırılan bu biçim birçok diğer CSS çatısı tarafından da benimsenmiştir. Modallar genellikle tüm içeriği kapatacak şekilde açılan bir fonla gelmektedir ve modal içeriği sayfanın odak noktasına yerleşmektedir. Daha küçük (mütevazi) açılır pencerelere gereksinim duyulduğunda ise Bootstrap kullanıcılara Toast olarak adlandırılan sayfanın farklı köşelerinde açılan, gerektiğinde saydam (transparan) olabilen seçenekli bir bileşeni sunmaktadır. Benzer bileşenler diğer CSS çatıları tarafından Rail, FeatureDiscovery gibi isimlerle farklı ve özgün olarak tasarlanmıştır. Semantic UI tarafından tasarlanan Railer sadece pencerenin değil aynı zamanda sayfayı oluşturan konteynerlerin içinde veya dışında konumlanabilmektedir.

7.11. Diğer Bileşenler

Üste ve yana açılan pencereler dışında buldukları yerde genişleyen bileşenler, akordeon ya da İngilizce ‘collapse’ pencere olarak adlandırılmaktadırlar. Akordeon, etiketler veya küçük resimler gibi dikey olarak yığılmış öğeler listesi içeren bir grafik kontrol ögesidir. Her öğe, o öğeyle ilişkili içeriği ortaya çıkarmak için "genişletilebilir" veya "daraltılabilir" özelliğindedir. Akordeonlar *Sıkça Sorulan Sorular* gibi içeriklerin gösteriminde özellikle tercih edilmektedir. Arayüz tasarımında, sekmeli belge arabirimi (İngilizce; tabbed document interface, kısaca TDI) veya sekme olarak da adlandırılan bu bileşenler aslında kâğıt dosyalara veya kart dizinlerine eklenen geleneksel kart sekmelerinden esinlenerek tasarlanmışlardır. Tablar veya Sekmeler, belge kümeleri arasında geçiş yapmak için bir gezinme aracı olarak kullanılarak birden çok belgenin veya panelin tek bir pencerede yer almasına izin veren grafiksel kontrol öğelerindedir.

Açılan öğelerin yanı sıra carouseller gibi kayan öğeler de birçok CSS çatısı tarafından sunulmaktadır. “Carouseller web tasarımında

genellikle iki şekilde kullanılır. Bunlardan birincisi sayfada gezinti anında kullanıcılara sadece tek bir görüntünün sunulduğu düzendir. Genellikle ileri geri ve sayfalama tuşları ile geçişler yönlendirilmektedir. Çok fazla mesaj, hiç mesaj anlamına geldiğinden, özenle tasarlanmış Carousel, afiş körlüğünün kurbanı olur ve içeriği çoğu kullanıcının dikkatinden kaçırır” (What is a carousel in web design, [17]). Bu nedenle web sitelerinin ana sayfalarında değişken fon görüntüsü olarak resmi olarak kullanılırlar. Carousel’ların ikinci kullanımı başka bir web tasarım bileşeni olan kartların sergilenmesi amaçlıdır. Bu seçenek özellikle sayfada çok fazla bilgi yer aldığı, söz konusu bilgiyi daha kompakt bir şekilde gruplamaya yardımcı olmaktadır. Atlı karınca içerisine resimler eklemek, otomatik düzeni etkinleştirmek, devre dışı bırakmak, ayarlayabilmek ve tasarımı özelleştirmek mümkündür. İdeal olarak, her kart, kullanıcının soldan ve sağdan göz atabileceği farklı bir öge içerir. UIKit (Slider) ve Bootstrap (Carousel) en CSS çatılarında sunulan en kapsamlı atlı karınca bileşenleridir.

Bazı CSS çatıları tarafından geleneksel web bileşenlerine ek olarak “Parallax” veya “İrakklik açısı” gibi modern web bileşenleri de sunulmaktadır. “Paralaks terimi ilk olarak, derinlik yanılsaması yaratmak amacıyla farklı hareket hızlarındaki fon görüntüsü kullanan, iki boyutlu ve yandan kaydırmalı video oyunlarının görsel etkisinden / efektinden gelmektedir. Bu etki, genellikle oyunun arka planının daha uzakta görünmesi için ön plana göre daha yavaş hareket ettirilmesiyle sağlanmaktadır” (Brown, [18]). Bir paralaks web sitesi, arka planda yerinde tutulan sabit görüntüler içermektedir ve kullanıcı, görüntünün farklı bölümlerini görmek için sayfayı aşağı kaydırabilmektedir. Görselin farklı bölümlerinin yanı sıra görsel üzerine yerleştirilmiş içerik üstünde de sayfa kaydırıldıkça değişiklikler olabilmektedir. UIkit tarafından sunulan bir Paralaks örneğinde sayfa aşağı kaydırıldıkça görselin renk tonu değişmekte ve üzerindeki yazılı metin aşamalı olarak kaybolmaktadır. Bu gibi farklı uygulamalar web sayfaları farklılık ve ritim bakımından ayrışmasını sağlamaktadır.

HTML 5 ve CSS 3 sayesinde CSS çatılarına eklenen canlandırmalar ve geçiş etkileri en son özellikler olarak sıralanabilir. Sözde sınıf olan

‘hover’ yıllardır ilkel canlandırmalar oluşturmak için kullanılmıştır. 2009 yılının mart ayında W3C tarafından resmen duyurulmasıyla sadece CSS kullanılarak sınırlı da olsa HTML etiketleri içerisine yerleştirilen görsel ve metinler bu sınıf kullanılarak hareketlendirilmeye başlanmıştır. İlk versiyonu 2012 yılında yayımlanan “CSS ANIMATIONS LEVEL 1” projesi W3C tarafından yönetilmektedir (World Wide Web Consortium (W3C), [19]). Günümüzde tarayıcı desteği büyük ölçüde sağlanmış olmakla beraber, bazı sorunların tümüyle çözülmemiş olmasından dolayı CSS çatılarında CSS canlandırma kullanımı hala gelişim aşamasındadır. TailwindCSS canlandırmalar, UIkit ise geçişler adı ile söz konusu özellikleri kullanıcılara sunmaktadır. Bunun yanı sıra Bootstrap gibi yaygın kullanılan birçok CSS çatısının da JavaScript tabanlı animasyon bileşenleri mevcuttur.

Paralaks ve canlandırma / hareketlendirme kullanımı vurgu, zıtlık ve denge gibi grafik tasarım ilkeleri bağlamında değerlendirildiğinde tasarıma çokça olumlu etkiler katmaktadır. Çünkü, web ortamlarının görünür olmasını sağlayan temel öğeler olarak grafik tasarım yapıtları ve uygulamaları ile karşılaşmaktadır. “*Grafik tasarım yapıtlarının* da gücü, niteliği, etkisi vd. özelliklerinin bilimsel irdelenmesi ancak kendine özgü tasarım ilkeleriyle gerçekleştirilebilmektedir” (Sayın [20]). Dolayısıyla Paralaks daha çok teknik bir özellik olmasından kaynaklı olarak web sayfalarının özgünlüklerine negatif bir etki göstermemekte, hatta tam tersi yaratıcı grafik tasarım çözümleri için olanaklar sunmaktadır.

CSS çatıları tarafından sunulan bir başka özellik de temalardır. Temalar geleneksel CSS çatılarının kapsamının dışında olmakla beraber, çatının görünen yüzü olduğundan genellikle temanın örnek uygulamaları hazırlanmaktadır. CSS Çatısını temel alacak bir projenin renk paleti, yazı tipi ölçeği, yazı tipleri, kesme noktaları ve köşe yarıçap değerleri tema dosyalarında tanımlanmaktadır. Özelleştirilmiş temaların kullanımı web tasarımında özgünlükten yoksun kalınmasının temel nedenlerindedir. Çatı kullanıcıları ürün veya kurumsal kimliğe en yakın temayı tercih edecekleri için temanın kullanıldığı her yerde aynı butonlar ve uyarı metinleri ile karşılaşabilmektedir. CSS çatıları genellikle

açık kaynaklı ve ücretsizdirler. Ama CSS çatılarında kullanılmak üzere üçüncü taraflarca hazırlanan şablonlar templatemonster.com gibi sitelerde satışa sunulmaktadır. CSS çatılarına bütünleşmiş temaların da daha küçük ölçekli bir pazarı mevcuttur. MaterializeCSS gibi bazı CSS çatıları, temaları ayrı bir ücret karşılığı resmi sitelerinde sunmaktadır. Temaların hem kullanıcı deneyimi hem sürdürülebilirlik hem de grafik tasarım ilkeleri açısından profesyonel anlamda kullanımından kaçınılmaktadır.

8. SONUÇ

CSS Çatılarında uygulanan grafik tasarımlarda karşılaşılan özgünlük sorunları grafik tasarımcılar açısından bir kayıp olarak görülmekte iken, birçok geliştirici ise bu durumun kullanıcı deneyimine olumlu yansıdığını düşünmektedir. Web tasarımında CSS çatısı kullanımı büyük ölçüde kullanıcı deneyimini olumlu etkilemektedir. Yazılım maliyetlerini düşürmesinin yanı sıra bazı projelerde kullanıcı deneyimi açısından tercih edilmektedir. Web sitesi tasarımlarında özgünlüğe olan olumsuz etkileri en aza indirmek teknik ve tasarım çözümleri ile olanaklıdır. Bu makalede söz konusu çözümler üzerinde durulmuş var olan çözümler incelenmiş ve yeni öneriler sunulmuştur. Bu makalede uzman yazılım ve tasarımcılar tarafından CSS çatısının sunduğu bileşenlere yapılan modifikasyonlar ile özgünlük ve kullanılabilirlik dengesini sağlamanın önemi anlaşılmıştır. CSS çatılarının sadece kullanıcı deneyimi ve teknik açılarından avantajlı olduğunu söylemek yanlış olur. Çünkü bütünlük, oran/orantı, denge ve ritim gibi grafik tasarım ilkeleri bağlamında web tasarımına olumlu etkileri söz konusudur. Grafik tasarım eğitimi veren yükseköğretim kurumlarındaki ilgili bölümlerde web yazılımları ile ilgili bir dersin ya da derslerin okutulması, söz konusu CSS çatılarından yararlanan grafik tasarımcıların özgünlük ve estetik açısından çok daha nitelikli yapıtlar ortaya koymasına için önemli katkılar yapacaktır.

KAYNAKLAR

1. Shenoy, A., & Prabhu, A. "CSS Framework Alternatives: Explore Five Lightweight Alternatives to Bootstrap and Foundation with Project Examples", 25-26, Apress, Mumbai, 2018.
2. Unger, R., & Chandler, C. "A Project Guide to UX Design: For User Experience

Designers in the Field or in The Making (Second Edition)", New Riders, Berkeley, CA. 35-36, 2012.

3. "The Free Dictionary" www.thefreedictionary.com <https://www.thefreedictionary.com/web+page> Ağustos 20, 2021.
4. Eva Harb, Paul Kapellari, Steven Luong, Norbert Spot, "Responsive Web Design", iaweb, ws201, 18-17, Eylül 10, 2021.
5. "W3C, Consortium", <https://www.w3.org/Consortium/>, Eylül 13, 2021.
6. Niederst, J. "Web Design In A Nurshell", O'Reilly, Sebastopol. CA, 2021.
7. Meiert, J. O., "The Little Book of HTML/CSS Frameworks", O'Reilly, Sebastopol, CA, 2015.
8. Beck K., Johnson R., "Patterns generate architectures. 8th European Conference, ECOOP '94 July 4-8, Proceedings", 139-149. Bologna, 1994.
9. Otto, M., "Building Twitter Bootstrap", <https://alistapart.com/article/building-twitter-bootstrap/>, Ocak 17, 2021.
10. Bradley, "The Pros And Cons Of CSS Frameworks", <https://vanseodesign.com/css/css-frameworks-pros-cons/>, Mayıs 14, 2021.
11. Çelik Özbahçe, G., "Responsive Css Frameworkleri İle Yapılan Web Sayfalarında Ortaya Çıkan Özgünlük Problemi Ve Örnek Bir Responsive Web Sayfası Çalışması", Gazi Üniversitesi, Ankara, 2019.
12. Turan, O. B., & Şahin, K., "Responsive Web Design And Comparative Analysis Of Development Frameworks", The Turkish Online Journal of Design, Art and Communication, Ocak s., 14-16, TOJDAC, 2017.
13. Budak, V. O., Gezer, M., "Farklı Ekran Çözünürlükleri İçin Esnek Web Arayüz Yapıları Tasarlanması", Ejoboc (Electronic Journal of Vocational Colleges), 10-24, 2016.
14. Margalit, L., "Is Your New Website Layout Killing Your Engagement?" Neuroscience Marketing: <https://www.neurosciencemarketing.com/blog/articles/horizontal-website-layouts.htm>, Temmuz 11, 2021.

15. Levene, M., "An Introduction to Search Engines and Web Navigation." Wiley, London, 2010.
16. "Semantic UI Steps." semantic-ui.com: <https://semantic-ui.com/elements/step.html>, Eylül 07, 2021.
17. "What is a carousel in web design." <https://elevationballoonacademy.com/xxkvm/what-is-a-carousel-in-web-design>, Temmuz 11, 2021.
18. Brown, J., "What Is Parallax Web Design? Definitions, Tips & Considerations", <https://www.unleashed-technologies.com/blog/what-parallax-web-design-definitions-tips-considerations>, Eylül 10, 2021.
19. "World Wide Web Consortium CSS ANIMATIONS", <https://www.w3.org/standards/history/css-animations-1>, Temmuz 11, 2021
20. Sayın, Z., "Grafik Tasarımda Etki. 2. Baskı." Pegem Akademi, Ankara.
21. "Web Technology Surveys", https://w3techs.com/technologies/overview/css_framework, Mayıs 04, 2023.
22. "CSS Tools: Reset CSS", tarihinde <https://meyerweb.com/eric/tools/css/reset>, Mayıs 04, 2023.