cankatbinici@gmail.com

Web Tasarım Güz-Final Projesi Rapor Çalışması

Selahattin Cankat Atbinici

2021370133

1 . **Çalışmanın özeti**

Bu çalışmada herhangi bir framwork kullanılmadan salt html, css, javascript kodları kullanılarak basit bir alışveriş sayfası tasarlanmaya çalışılmıştır. Proje kapsamında responsive.html , login.html , style.css, script.js ve login.js isimli 5 dosya oluşturulmuş, sayfanın çalışması için gereken ilgili kodlar bu dosyaların içerisine eklenmiştir. Sayfaların görünümünü ayarlamak adına css kodları mümkün olduğunca external olarak kullanılmış olmasına karşın bazı durumlarda inline olarak kullanımı da tercih edilmiştir. JavaScript dosyaları ise tamamen external olarak html kodlarını manipüle etmektedir.

Proje içerisinde ürünleri ve kullanıcıları tutan iki adet farklı array tanımlanmıştır. Döngüler ve "if-else" yapıları sayesinde bu arrayların ilgili verileri ilgili yerlere ulaştırması sağlanarak sayfanın doğru çalışması amaçlanmıştır. Ayrıca bu çalışma içerisinde eklenebilecek bir register ve ürün ekleme interface'i ile bu arrayların veri tabanı gibi çalışarak yeni ürünlere ve kullanıcılara yönelik geliştirmelere açık olması amaçlanmıştır.

Çalışma içerisine fotograf eklendiğinde sayfa görünümü oldukça kötüleştiğinden projenin geliştiricisi tarafından kendi fotoğrafı eklenmemiştir. Ancak fotoğraf ekleme yöntemlerinin bilindiği sayfada ürünlere ve logolara ait fotoğraflardan sınanabilir olarak görülmüştür.

**2. Proje Dosyaları**

**2.1. Login.html:**

Login.html sayfasında kullanıcı sade ve kullanımı kolay bir inter-face ile karşılaşır. Bu sayfada sisteme üye olarak kayıtlı olduğu varsayılan 10 adet kullanıcının oturum açma bilgileri istenmektedir. Doğru kullanıcı adı ve şifre eşleşmesi saptandığında js kullanılarak kullanıcı responsive.html sayfasına yönlendirilmektedir. Sisteme giriş yapabilecek kullanıcı adı ve şifre eşleşmeleri aşağıdaki gibidir.

1. const userDataSet = [
2. {username: "cankat",
3. password: "cankat"},
4. {username: "canberk",
5. password: "canberk"},
6. {username: "banu",
7. password: "banu"},
8. {username: "yaprak",
9. password: "yaprak"},
10. {username: "hilal",
11. password: "hilal"},
12. {username: "webtasarim",
13. password: "webtasarim"},
14. {username: "nemutluturkumdiyene",
15. password: "nemutluturkumdiyene"},
16. {username: "admin",
17. password: "admin"},

Yukarıdaki görsel login.js dosyasına aittir

Buna göre "cankat" isimli kullanıcı adına sahip kullanıcının oturum açması için gerekli şifre "cankat" dır. Liste bu şekilde aşağıya doğru devam etmektedir. Sisteme kayıtlı kullanıcı isimleri sırası ile "cankat, canberk, banu, yaprak, hilal, webtasarım, nemutluturkumdiyene, admin" şeklindedir. (şifreler ile kullanıcı isimleri aynıdır)

**2.2. Responsive.html:**

responsive .html sayfası websitesinin ana sayfası niteliğindedir. (ismi index.html olarak belirlenmemiştir) sayfanın ismi seçilirken proje kapsamı -responsive bir tasarım olması gerekliliği- ön koşul olarak görülmüştür. Dolayısı ile sayfanın ismi responsive.html olarak belirlenmiştir.

Kullanıcı responsive.html sayfasına geldiğinde, ürünlerin listelendiği bir article side, sepetin rahatça açılabilmesi için ayrılmış "a side" en altta sayfasının logo ve isminin yer aldığı header ve üzerinde herhangi bir bilgi olmasa da en altta bir footer bolümü vardır. Sayfa içerisindeki tüm containerler daha fazla ürün eklenmesi ve sayfasının aşağıya doğru uzaması ihtimali düşünülerek hazırlanmıştır.

Sepete ekle butonuna bağlı sepet <div> 'i sayfaya ilk girildiğinde display="none" olarak ayarlanmıştır ve kullanıcı tarafından görülmez. İstendiğinde header bölümünün sağında yer alan sepet butonuna tek tıklandığında sepet <div> i açılır ve kullanıcı sepete eklediği ürünleri - ürün adetini- birim fiyatını- toplam fiyatını görüntüleyebilir.

responsive.html sayfasının eksiği, bu projenin geliştiricisi tarafından - sepet butonunun üzerinde "kullanıcıya sepette ürün olduğunu ifade eden" herhangi bir buton- olmayışıdır. Ancak projenin belirlenen sürede tamamlanması şartından dolayı proje web tasarım final projesi olarak bu hali ile teslim edilmektedir.

**2.3. Login.js :**

Login.js sayfası login.html sayfasındaki formdan gelen verilerin. İç içe "if - else" yapısının bir forEach döngüsü içerisinde kullanılarak kontrol edildiği ve eğer gelen username - password bilgileri hem birbirleri ile hem de "userDataSet" isimli array içerisindeki objectler ile eşleşirse responsive.html sayfasına geçişine izin verilen, aksi taktirde ekranda "Hatalı şifre. Tekrar deneyin" uyarısı döndürerek kullanıcının girişine izin vermeyen bir algoritma ile beslenmiştir.

Bu basit sınama işleminin eksikliği ise username - password için ayrı ayrı uyarı vermemektedir. Kullanıcı bilgileri uyuştuğu taktirde "break" ile forEach döngüsü kırılarak kullanıcının girişine imkan tanımaktadır. Ayrı ayrı uyarı verilmek istendiğinde ise kullanıcı giriş bilgileri doğru olsa da object içerisinde tanımlı kullanıcının sırası gelene kadar döngü her döndüğünde ekrana " Hatalı şifre. Tekrar deneyin" uyarısı gelmeye devam etmekte ardından kullanıcının girişine izin vermektedir. Bu sorun projenin geliştiricisi tarafından forEach döngüsünün sonuna kadar dönmeye devam etmesi olarak gözlemlenmiştir. Ancak projenin kısıtlı sürede teslim edilmesi gerektiği için gerekli iyileştirmeler yapılamıştır. Yeterli süre olduğu taktirde yapılması düşünülen işlem " bir for döngüsünün 0 ile userDataSet.lenght değerleri arasında dönerek gerekli noktada "break" ile kırıldığında daha verimli çalışabileceği düşünülmektedir. "

**2.4. script.js:**

script.js dosyası içerisindeki kod blokları, responsive.html dosyası içerisindeki html kodlarını java script'in DOM metodu kullanılarak manipüle etmesi amaçlanarak bu projenin geliştiricisi tarafından yazılmıştır. dosya içerisinde yer alan kodların işlevleri özetle aşağıda anlatıldığı gibidir.

//Descripted Pruducts

var link = document.createElement('link');

link.href = 'style.css';

link.rel = 'stylesheet';

link.type = 'text/css';

let cart = []

var products2 = [

    { id: 0, name: "Ürün 1", price: 10 , image:src = "resimler/cerrahi1.jpg", style:"width:50px"}, { id: 1, name: "Ürün 2", price: 20 , image:src = 'resimler/cerrahi2.jpg'}, { id: 2, name: "Ürün 3", price: 30 , image:src = "resimler/cerrahi3.jpg"} ,

    { id: 3, name: "Ürün 4", price: 40 , image:src = "resimler/cerrahi4.jpg" } , { id: 4, name: "Ürün 5", price: 50, image:src = "resimler/cerrahi5.jpg" } , { id: 5, name: "Ürün 6", price: 60, image:src = "resimler/cerrahi6.jpg" }

]

Ürünler, products2 isimli bir array içerisinde object olarak tanımlanmıştır.Objectler içerisinde id, price, image gibi ürün kartlarının üzerinde bulunması gereken veriler yer almaktadır.

let increaseCountButton;

let decreaseCountButton;

let addToCartButton;

let removeFromCartButton;

(function (){

    renderProductCards()

})()

function renderProductCards() {

// resmi objecte ekledik ama cart içerisinde yok.

    products2.forEach((product, index) => {

        document.getElementById("productCartsContainer").innerHTML += `

            <div class="ProductCards">

            <div id="${product.id}">

             <div class="productImageStyles">

             <img src = "${product.image}" style="width: 214px; height: 148px;">

               </div>

             </div>

            <div>  </div>

                <div class= "ProductCardsDetail">

                    <p id="${product.id}"> ${product.name}<span> ${product.price} </span> TL <br>

                    </p>

                 </div>

                <input type="button" name="${product.id}" value="Sepete Ekle +" class="AddtoCardButton">

                <input type="button" name="${product.id}" value="Sepetten Çıkar -"    class="removeFromCartButton" >

          </div> `

    })

    addToCartButton = document.getElementsByClassName("AddtoCardButton");

    removeFromCartButton = document.getElementsByClassName("removeFromCartButton");

}

function renderChart(){

    console.log("cart in render chart: " , cart)

    document.getElementById("listitem").innerHTML = ""

    cart.forEach((product, index) => {

        const selectedChartItem = getProductFromChart(product.id)

        document.getElementById("listitem").innerHTML += `

            <div class="ProductCardsBasket">

                <div class= "ProductCardsDetailBasket">

                    <p style="border-color: aliceblue; border-style: groove;

                    background-color: white;" id="${product.id}"> ${product.name}<span> ${product.price} </span> TL <br>

                    </p>

                    <p style="border-color: aliceblue; border-style: groove;

                    background-color: white;"> Ürün Adeti : ${selectedChartItem.count}

                    <p style="border-color: aliceblue; border-style: groove;

                    background-color: white;"> Toplam Tutar: ${selectedChartItem.count \* selectedChartItem.price }

                 </div>

                <input type="button" name="${product.id}" value="Sepete Ekle +" class="decreaseCount">

                <input type="button" name="${product.id}" value="Sepetten Çıkar -"    class="increaseCount" >

          </div> `

    })

render productcards functionu ile yukarıda tanımlanmış products2 isimli arrayin içerisinde yer alan ürün kartlarına ait veriler, bir döngü yardımı ile alınarak her defasında aynı algoritmden geçerek sayfanın ürün kartlarına ayrılmış ilgili containerına - print edilmektedir. Böyle product2 isimli array bir veritabanı gibi kullanılarak yeni ürün kartları eklenmek istendiğinde tek yapılması gereken id, name, price ve image değerlerini girmek olacaktır. Bu yöntem veri tabanı olmaksızın sınırsız sayıda veri responsive.html sayfasında görüntülenmek üzere eklenebilir.

renderProductCharts functionu ise gerekli veriler yardımı ile html kodlarının manipülasyonunu sağlayarak sepete eklenecek ürünlerin değerlerinin aktarımı için kullanılmaktadır.

script kodlarında aşağıya doğru inmeye devam ettikçe, butonlar yardımı ile sepete ekle çıkar işlemleri yapmak ve basit toplama çıkarma işlemlerini doğru şekilde yapmak aynı zamanda sepetteki ürünün adet sayısı <1 olduğu taktirde sepet içerisindeki görünürlüğünü kaldırmaya yarayan algoritmalar ile beslenmiştir.

**2.5. Sytle.Css:**

style.css dosyası ilgili diğer dosyalarda yer alan kod bloklarının kullanıcıya print edilen sayfalarındaki style özelliklerini external olarak düzenlemek için kullanılmaktadır. Bu sayfalara erişmek isteyen .html ve .js sayfalarına ilgili kodlar yazılarak style.css sayfası ile uyum içerisinde çalışmaları sağlanmaktadır.

**SONUÇ**

Proje kapsamında responsive bir tasarım, herhangi framework kullanmaksızın CSS ile biçimlendirilmiş, istenilen hesaplamalar JavaScript dili kullanılarak yaptırılmıştır. Çalışma bir ödev projesi olsa dahi geliştirilebilirliği göz önünde bulundurularak kodlar yazılması amaçlanmıştır.