**T.C.**

**MERSİN ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**



**EĞİTİM, UYGULAMA VE STAJ**

**RAPOR DOSYASI**

|  |  |
| --- | --- |
| **BÖLÜMÜ** | **BİLGİSAYAR MÜHENDİSLİĞİ** |
| **PROGRAMI** | **STAJ(40 GÜN)** |
| **ADI SOYADI** | **YUSUF GÜLBİN** |
| **ÖĞRENCİ NO** | **16-155-037** |

|  |  |
| --- | --- |
| **ÖĞRENCİNİN;** | |
| **ADI SOYADI** | **Yusuf GÜLBİN** |
| **ÖĞRENCİ NO** | **16-155-037** |
| **BÖLÜMÜ** | **Bilgisayar Mühendisliği** |
| **PROGRAMI** | **Staj** |
| **SINIFI** | **4 (Mezun Durumunda)** |

|  |  |
| --- | --- |
| **STAJ YAPILAN YERİN;** | |
| **ADI** | **Mersin Üniversitesi** |
| **ADRESİ** | **Çiftlikköy, Mersin Ünv.,33110 Yenişehir/MERSİN** |
| **TEL/FAKS NO** | **0324 361 00 01/17040** |
| **E-POSTA** | **bilgisyarmuhendisligi.sek@mersin.edu.tr** |

|  |  |  |
| --- | --- | --- |
| **ÖĞRENCİDEN** | **SORUMLU PERSONELİN;** | |
| **ADI SOYADI** | | **Hamza EROL** |
| **ÜNVANI** | | **Profesör Doktor** |
| **TARİH** | | **15.09.2021** |
| **İMZA** | |  |

**İÇİNDEKİLER**

1. GİRİŞ …………………………………………………………………………………... 1

2. TASARIM ……………………………………………………………………………… 2

2.1. Navbar .……………………………………………………………………..... 2

2.2. Footer …..………………………………………………………………......... 7

2.3. Anasayfa …..…………………………………………………………………. 9

2.4. Kategori Sayfaları …..……………………………………………………… 11

2.5. Kayıt Ol Sayfası …..………………………………………………………... 12

2.6. Giriş Yap Sayfası ….……………………………………………………….. 14

2.7. Ürün Sayfası …..……………………………………………………………. 16

2.8. Sepetim Sayfası …..………………………………………………………… 19

2.9.Siparis Ver Sayfası …..……………………………………………………… 21

2.10. Siparişlerim Sayfası …..…………………………………………………... 24

2.11. Admin Sayfası …..………………………………………………………… 26

2.12. Hesap Bilgileri Sayfası …..………………………………………………... 28

2.13. Şifre Değiştir Sayfası …..…………………………………………………. 30

2.14. Hakkımızda Sayfası …..…………………………………………………... 31

2.15. Kayıt Koşulları …...……………………………………………………….. 33

2.16. Müşteri Hizmetleri Sayfası …..…………………………………………… 35

3.SUNUCU TARAFI ……………………………………………………………………. 37

3.1. Xampp …..………………………………………………………………….. 37

3.2. PhpMyAdmin …..…………………………………………………………... 38

3.3.Veri Tabanı Tabloları …..…………………………………………………… 39

3.4. Veri Tabanı Bağlantısı …..…………………………………………………. 41

3.5. Üye Kaydı …..……………………………………………………………… 42

3.6. Kullanıcı Girişi …..…………………………………………………………. 44

3.7 Çıkış Yap …..……………………………………………………………….. 46

3.8. Sepete Ekleme …..………………………………………………………….. 47

3.9 Sepetim …..………………………………………………………………….. 49

3.10. Sepet sil …..……………………………………………………………….. 51

3.10. Sepet sil …..……………………………………………………………….. 52

3.12. Toplam Fiyat …..………………………………………………………….. 53

3.13. Siparişler …..……………………………………………………………… 55

3.14. Admin …..…………………………………………………………………. 57

3.15. Kargoya Ver …..…………………………………………………………... 58

3.16. Hesap Bilgileri …..………………………………………………………... 60

3.17. Şifre Değiştirme …..………………………………………………………. 61

4. SİTENİN KULLANIMI HAKKINDA BİLGİLER ………………………………….. 63

5. SONUÇ ……………………………………………………………………………….. 67

6. Kaynak Kodlar ………………………………………………………………………... 68

**1.GİRİŞ**

Günümüzde internet üzerinden alışveriş yapmak hayatın vazgeçilmez bir parçası haline geldi. İnternet üzerinden alışveriş yapmanın hem müşteriler hem satıcılar için birçok kolaylıkları bulunmaktadır. Amozong.com buna bir örnek olabilir. Müşteri amazong.com sitesine girerek istediği şekilde alışveriş yapabilir ve satın aldığı ürünler kargo ile istediğiniz yere getirilebilir. Alışverişinizi yorulmadan bir şekilde yapmış olursunuz.

Benim staj ödevim bu kapsamda Elbise, gömlek, pantolon, ayakkabı, satıldığı bir e-ticaret sitesi yapmak. Sitemin adı ise OVAKENT dir. Sitede bir e-ticaret sayfasında olması gereken sayfalar bulunacaktır. Bunlar; anasayfa, ürün sayfaları, kategori sayfaları, giriş-kayıt sayfaları, destek-iletişim sayfaları, gibi sayfalardır. Aşağıda site ile alakalı birkaç detay bulunuyor:

1- Site Türkçe dilinde hazırlanacaktır.

2- Sitedeki ödeme sayfasında para birimi Türk Lirası olacaktır.

3- Sitede gerekli olan yazılımlar (arayüz, veritabanı, sistem vs.) sağlanacaktır.

4- Kullanıcı işlemleri için veritabanı bağlantıları sağlanacaktır.

Kullanılacak Materyaller ve Diller:

Frontend: HTML CSS, JAVASCRIPT

Backend: PHP

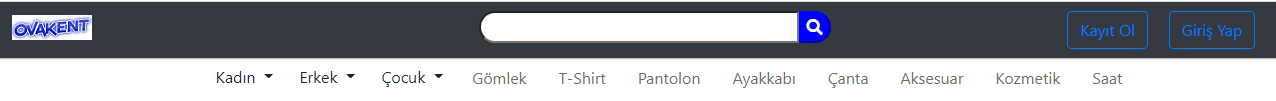
Kütüphaneler: Bootstrap4

Veritabanı: MySQL

**2. TASARIM**

**2.1.Navbar**

Navbar menüsü sayesinde sayfaya giren internet kullanıcıları, internet sitesinin en önemli bölümleri arasında kolayca geçiş yapabilirler.

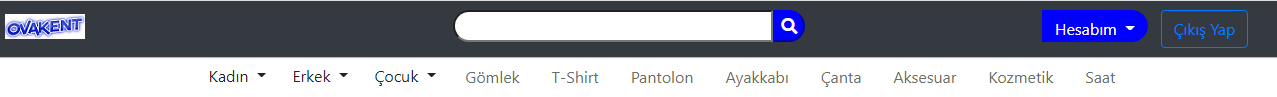


(Resim 2.1.1)

Benim sitemdeki navbar kısmında(Resim 2.1.1) sol kösede Ovakent yazılı sitemin amblem kısmı bulunmaktadır. Bu amblem tıklandığında anasayfa’ya(index.php) yönlendirmektedir. Navbarın orta kısmında ise search yani arama butonu bulunmaktadır.

Navbarın sol tarafında ise Kayıt Ol ve Giriş Yap butonları bulunmaktadır. Kayıt ol butonuna tıklayarak bizi kullanıcı kaydına(kayit.php) yönlendirecektir ve buradan yeni bir kullanıcı oluşturabiliriz. Kullanıcı kaydından sonra Giriş yap butonuna tıklayarak kullanıcı girişini yapılmaktadır.

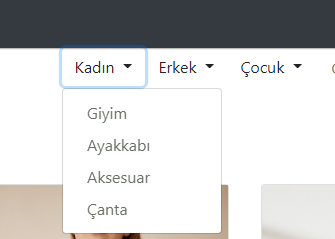
Üye girişi yapıldıktan sonra Navbar kısmında Kayıt Ol ve Giriş Yap kısımları Hesabım ve Çıkış Yap şeklinde değişecektir.(Resim 2.1.2). Hesabım butonun da hesap bilgileri, sepet, sipariş bölümleri bulunmaktadır.



(Resim 2.1.2)

Alt Navbarda ise kategoriler bulunmaktadır. Bu kategoriler Kadın, Erkek, Çocuk, Gömlek, T-Shirt, Pantolon, Ayakkabı, Çanta, Aksesuar, Kozmetik, Saat bulunmaktadır. Bunlardan birine Tıkadığımız zaman bağlı olduğu sayfaya yönlendirilecektir.

Kadın, Erkek, Çocuk kisminda alt kategori bulunmaktadır.(Resim 2.1.3).



(Resim 2.1.3)

**Kod:**

<section *class*="nav-bar navbar-dark bg-dark">

<div *class*="container-fluid px-0">

<nav *class*="navbar navbar-expand-lg navbar-light">

<a *class*="navbar-brand" *href*="index.php">

     <img *src*="images/ova.png" *class*="d-inline block aligntop " *alt*="ovakent"></a>

   <div *class*="collapse navbar-collapse pr-5" *id*="navbarNav">

   <div *class*="mr-auto"></div>

     <ul *class*="navbar-nav">

       <li *class*="nav-item">

         <form *action*="#" *class*="search-nav form-inline">

            <input *type*="text" *class*="" />

                <button> <i *class*="fas fa-search form-control-feedback mb-1"></i></button></form></li>

       <li *class*="nav-item item-2">

         <form *method*="post">

                <?php

                    if(isset($\_SESSION['uye'])){

                        echo '<ul class="navbar-nav ">';

                        echo '<li class="nav-item active">';

                        echo '<div class="dropdown">

                        <button class="btn Secondary link dropdown-toggle" type="" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"> Hesabım </button>

<div class="dropdown-menu" aria-labelledby= "dropdownMenuButton">

<a class="dropdown item" href="sayfa/hesapbilgileri.php">Hesap Bilgileri</a>

                            <a class="dropdown-item" href="sayfa/sepetim.php">Sepetim</a>

                            <a class="dropdown item" href= "sayfa/siparislerim.php">Siparislerim</a>

                            <a class="dropdown-item" href="sayfa/sifre.php">Sifre Degistir</a>

                          </div>

                          </div>';

                        echo '</li>';

                        echo '<li class="nav-item active">';

                        echo '<a class="btn btn-outline-primary" href="php/cıkısyap.php">Çıkış Yap</a>';

                        echo '</li>';

                        echo '</ul>';

                    } else {

                        echo '<ul class="navbar-nav">';

                        echo '<li class="nav-item active">';

                        echo '<a class="btn btn-outline-primary mr-2" href="sayfa/kayit.php">Kayıt Ol</a>';

                        echo '</li>';

                        echo '<li class="nav-item active">';

                        echo '<a class="btn btn-outline-primary" href="sayfa/giris.php">Giriş Yap</a>';

                        echo '</li>';

                        echo '</ul>';

                    } ?>

                </form>

          </li>

              </ul>

            </div>

          </nav>

</div>

      </section>

Başlangıç olarak bir section etiketi açarak navbar ve navbar-dark classlarını uyguluyoruz. Navbar-dark siyah bir görünüm kazandırmaktadır. Bu navbar için bootstrap ile aynı zamanda kendi css kodlarıyla farklı görünümler oluşturdum. Css kodum ise css klasörü içerisinde b.css dosyasındadır. Bu dosyayı Aşağıdaki kullanarak projemize dahil ediyoruz.

  <link *rel*="stylesheet" *href*="css/b.css" /> />

Aynı zamanda bootstrap kütüphanesi kullanıyoruz.

 <link *rel*="stylesheet" *href*="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" *integrity*="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" *crossorigin*="anonymous">

Link etiketi kullanarak bootstrap kütüphanesini projemize dahil ediyoruz.

Ardından da bir container-fluid uygulanmış div açılıp, tüm elemanları bunun içine girecektir.

Bir navbar-brand classıyla a etiketi açarak img etiketi içerisinde amblemimiz olan ova.png dosyamızı yazıyoruz. Bu ambleme tıklandığı zaman ana sayfaya(index.php) yölendirecektir.

Div etiketleri açarak search inputunu ve bul iconlarını yazıyoruz.

Daha sonra bir php etiketi açıyoruz. Çünkü giriş yapılmadığı zaman kayıt ol ve giriş yap butonları görünecek eğer giriş yapıldığı zaman ise hesabım ve çıkış yap butonları görünecektir. Bunu şu şekilde yapıyoruz php etiketi içerisine if(isset($\_SESSION['uye'])) koşulu açıyoruz. Bu koşul ise kullanıcı girişi sağlanıp sağlanmadığını kontrol edecektir.Tüm web programlama dillerinde olduğu gibi Php’de de oturum bilgilerini saklamak ve oturumları yönetmek için Session’lar kullanılır. Sessionlar çerezlerden farklı olarak kullanıcının bilgisayarında değil de sunucu bilgisayarı üzerin de saklanır. Bu nedenle üyelik sistemleri, alış veriş gibi güvenliğin önemli olduğu noktalarda Session kullanımı zorunludur. Session kullanımıyla ilgili bilinmesi gerekenler:

* Session oluşturulduğu anda bu veriler sunucu bilgisayarı üzerinde ki bir text dosyası içerisinde tutulur.
* Kullanıcı oturumu kapattığında veya sayfadan ayrıldığında Session bilgileri silinir.
* Çerezlerde ki gibi süreli Session’lar oluşturulamaz.
* Sayfada Session değişkeni kullanılacaksa sayfa da hiçbir kod yazılmadan önce session\_start() metodu çağrılmalıdır. Aksi takdirde hata mesajı görüntülenir.
* Oturum değişkeni olan $\_SESSION global bir değişkendir.

Bir alt navbarın kodu ise:

<section *class*="category-nav">

        <div *class*="container-fluid">

          <ul *class*="nav justify-content-center">

           <li *class*="nav-item">

             <div *class*="dropdown">

             <button *class*="btn Secondary link dropdowntoggle" *type*="" *id*="dropdownMenuButton" *data-toggle*="dropdown" *aria-haspopup*= "true" *aria-expanded*="false">Kadın</button>

                <div *class*="dropdown-menu" *aria-labelledby*=" dropdownMenuButton">

                 <a *class*="dropdown-item" *href*=" urunsayfa/kadingiyim.php">Giyim</a>

                 <a *class*="dropdown-item" *href*=" urunsayfa/ayakkabi.php">Ayakkabı</a>

                 <a *class*="dropdown-item" *href*=" urunsayfa/aksesuar.php">Aksesuar</a>

                 <a *class*="dropdown-item" *href*=" urunsayfa/canta.php">Çanta</a>

                  </div>

              </div>

          </li>

       <li *class*="nav-item">

              <a *class*="nav-link" *href*="urunsayfa/gomlek.php" >Gömlek</a>

         </li>

          </ul>

        </div>

      </section>

Bir category-nav classıyla bir section etiketi açarak bunun içine container-fluid classıyla bir div etiketi açıyoruz. Buradaki css ise bootstrap kütüphanesi ile kullanıyoruz. Ul ve li etiketleri açarak burada bir liste oluşturuyoruz. Kadın, Erkek, Çocuk 3 tane dropdown oluşturuyoruz. Pantolon, Ayakkabı, Çanta, Aksesuar, Kozmetik içinde yukarıdaki kodda gömlek için yaptığımızın aynısınıda bunlar için yapıyoruz.

**2.2.Footer**

Footer kısmında ise iletişim bilgileri bulunmaktadır. Ayriyeten Hakkımızda, Kayıt Koşulları, Sıkça Sorulan Sorular, Satış sözleşmesi bulunmaktadır. Bunlar ise e-ticaret sitesinin alış veriş ile ilgili bilgiler bulunmaktadır. Her birine tıkladığımız zaman bağlı olduğu sayfaya yönlendirilecektir.



(Resim 2.2.1)

**Kod:**

    <footer *class*="bg-dark text-white pt-5 pb-4">

        <div *class*="container">

            <div *class*="row">

                <div *class*="col-6 col-md-3">

                    <h5>İletişim</h5>

                   <ul *class*="list-unstyled">

               <li>

                        <a *class*="text-muted">0535 123 11 11</a>

              </li>

              <li>

                        <a *class*="text-muted">yusuf@gmail.com</a>

                      </li>

              <li>

                        <a *class*="text-muted">Hatay/Antakya</a>

                      </li>

                   </ul>

                </div>

<div *class*="col-6 col-md-3">

                  <ul *class*="list-unstyled">

                    <a *href*="sayfa/hakkimizda.php" >Hakkımızda</a>

                </div>

 <div *class*="col-6 col-md-3">

                   <ul *class*="list-unstyled">

                     <li>

       <a *href*="sayfa/kayitkosullari.php"  >Kayıt Koşulları </a>

                      </li>

                      <li>

                        <a *href*="sayfa/sikcasorulansorular.php">Sıkça Sorulan Sorular</a>

                      </li>

                   </div>

          <div *class*="col-6 col-md-3">

                    <ul *class*="list-unstyled">

                     <li>

                       <a *href*="sayfa/musterihizmetleri.php">Satış Sözleşmesi ve Müşteri Hizmetleri</a>

                      </li>

                 </div>

         </div>

     </footer>

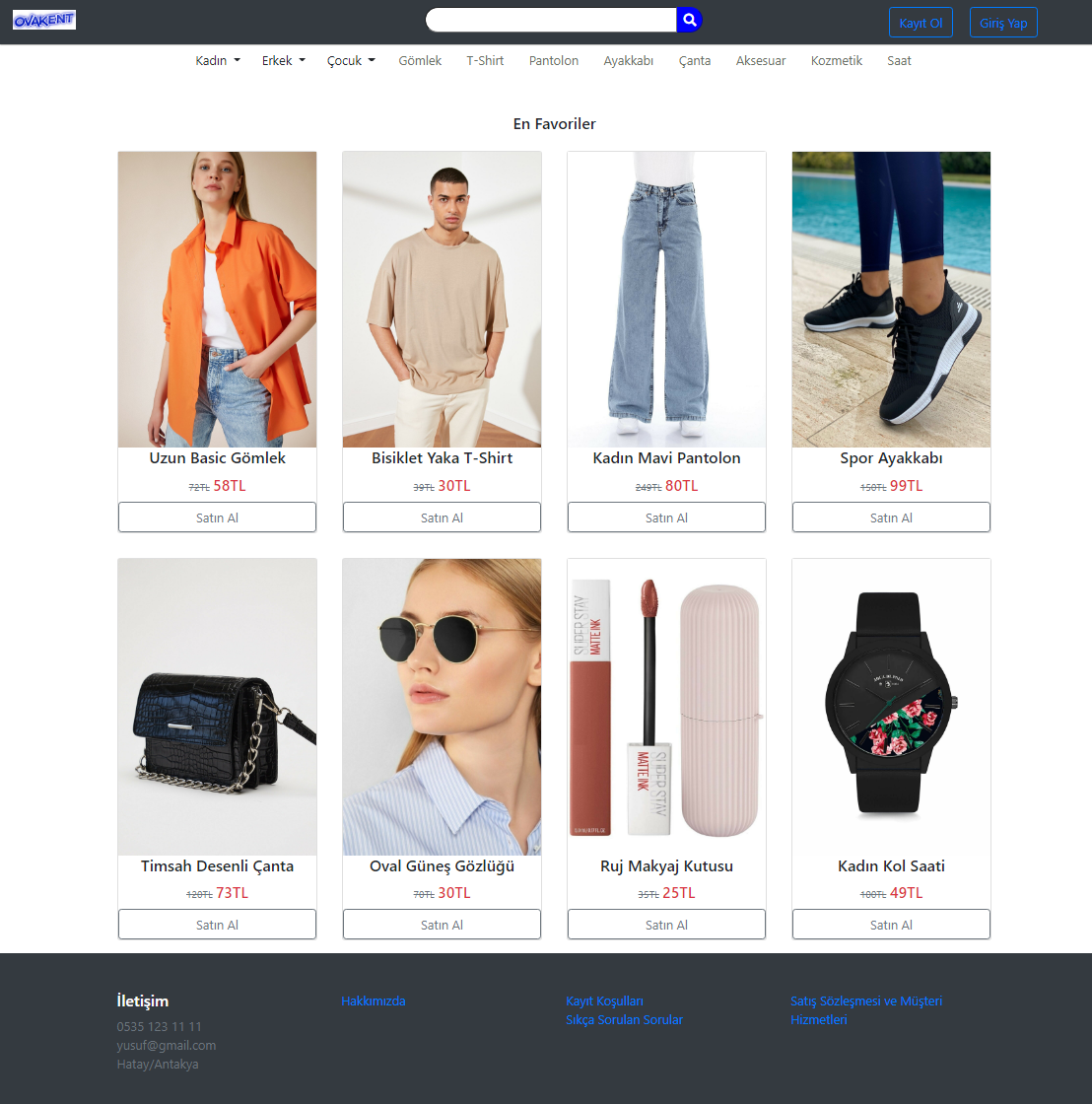
Footer etiketi açarak bu etiketin içine container ve row classlarıyla div etiketlerini açıyoruz. Burada footer alanını oluşturmuş oluyoruz. Daha sonra div etiketleri açarak a etiketi içerisine iletişim, Hakkımızda, Kayıt kosulları, Sıkça sorulan sorular ve satış sözleşmesi yazıyoruz. A etiketine yazmamızın amacı tıklandığı zaman başka sayfaya bizi yönlendirmesi için href elementlerini kullanılmasıdır.

Hakkımızda ya tıkladığımız zaman sayfa klasörünün içindeki hakkımızda.php dosyasına yönlendilecektir. Kayıt Kosulları na tıkladığımız zaman sayfa klasörünün içindeki kayitkosullari.php dosyasına yönlendilecektir. Sıkça sorulan sorulara tıkladığımız zaman ya tıkladığımız zaman sayfa klasörünün içindeki sıkcasorulansorular.php dosyasına yönlendilecektir.

Navbar ve footer hemen hemen butun sayfalarda bu şekildedir.

**2.3.Anasayfa(index.php)**

Ana sayfa kullanıcıların karşısına çıkan ilk sayfadır ve bu sayfada en favori ürünler satılmaktadır (Resim 2.3.1). Ana sayfa index.php dosyasıdır. Ürünler her biri card classlarıyla bir dikdörtgen şeklinde ve her ürünün fotoğrafı bulunmakta ve ürünün fiyatı bulunmaktadır. Bu fiyatın altında satın al butonu bulunmaktadır. Satın al butonuna tıklandığı zaman ürün sayfasındaki klasöründe her ürün için hazır olan php dosyasına yönlendirilmektedir.



(Resim 2.3.1)

Ana sayfadaki bir ürün için kod kısmı aşağıdadır.

**Kod:**

<div *class*="col-md-3 col-sm-6 p-3">

<div *class*="card text-center">

<div *class*="card-block">

<img *src*="https://cdn.dsmcdn.com/mnresize/1200/1800/ty144/product/media/images/20210712/14/109752632/152488873/1/1\_org\_zoom.jpg" *class*="img-fluid">

<div *class*="card-title">

<h5> Uzun Basic Gömlek</h5>

</div>

<h6 *class*="text-center py4">

  <small><span *class*="text-secondary"> <del>72TL</del></span></small>

  <big><span *class*="text-danger">58TL</span></big>

</h6>

<a *style*="margin-top: 10px;" *href*=" urunsayfa/gömlek/uzunbasicgomlek.php" *class*="btn btn-outline-secondary btn-block">Satın Al</a>

</div>

</div>

</div>

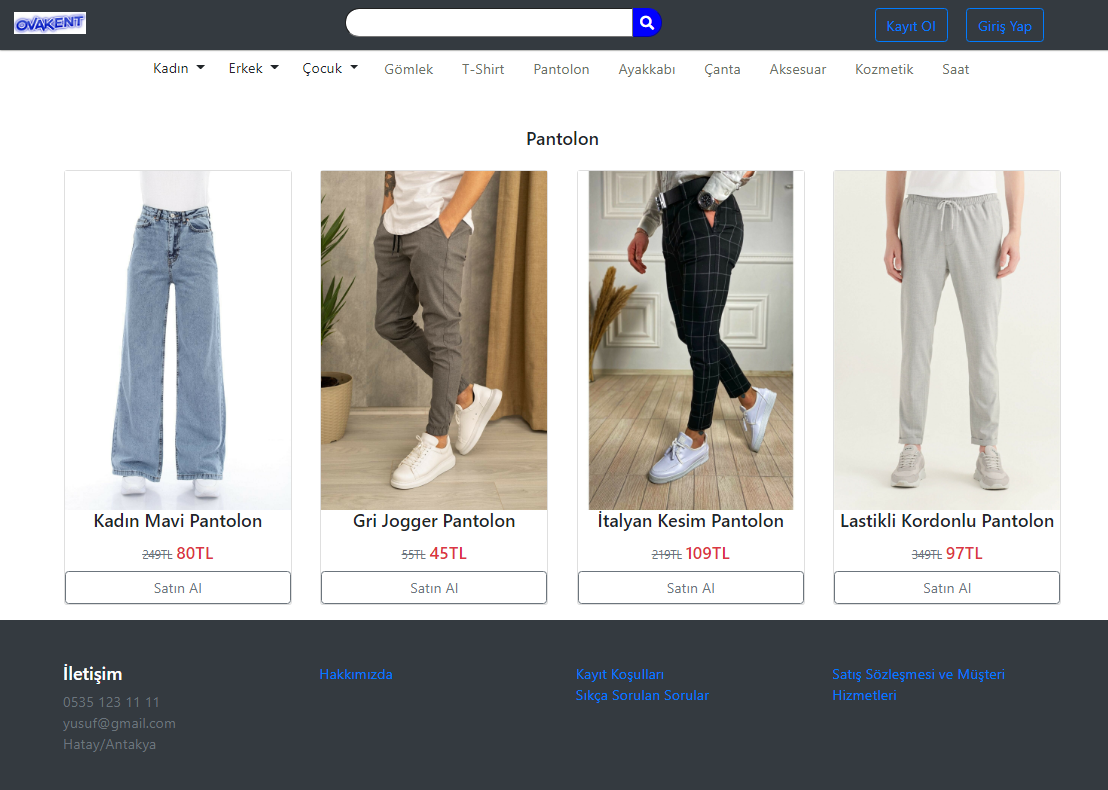
Bir container ile row classları içinde butun ürünleri yazıyoruz. Bootstrap kütüphanesindeki card’ı kullandım. Burada ürün fotoğrafının adresini yapıştırdım. Satın al butonuna tıklandığı zaman ürün sayfası klasörünün içinde gömlek klosörünin içindeki uzunbasikgomlek.php dosyasına yönlendirilecektir. Bu dosyada ürünle ilgili bilgiler, sepete ekleme, ürünün farklı fotoğrafları bulunmaktadır. Diğer ürünler için ayrı dosyalar bulunmaktadır.

Diğer ürünler için aynı şekilde div etiketinin bittiği yerden yeni aynı classları kullanarak aynı div etiketleri açıyoruz.

Bu şekilde Ansayfada(İndex.php) 8 tane ürün bulunmaktadır.

**2.4.Kategori Sayfaları**

Kadın, erkek, çocuk, gömlek, T-shirt, aksesuar, çanta, kozmetik, ayakkabı kategorisindeki sayfalar ana sayfa ile aynıdır. Bu sayfalar ürünsayfa klasörünün içinde her bir kategori için dosyalar bulunmaktadır. Her kategoriyle ilgili ürünler bulunmaktadır.

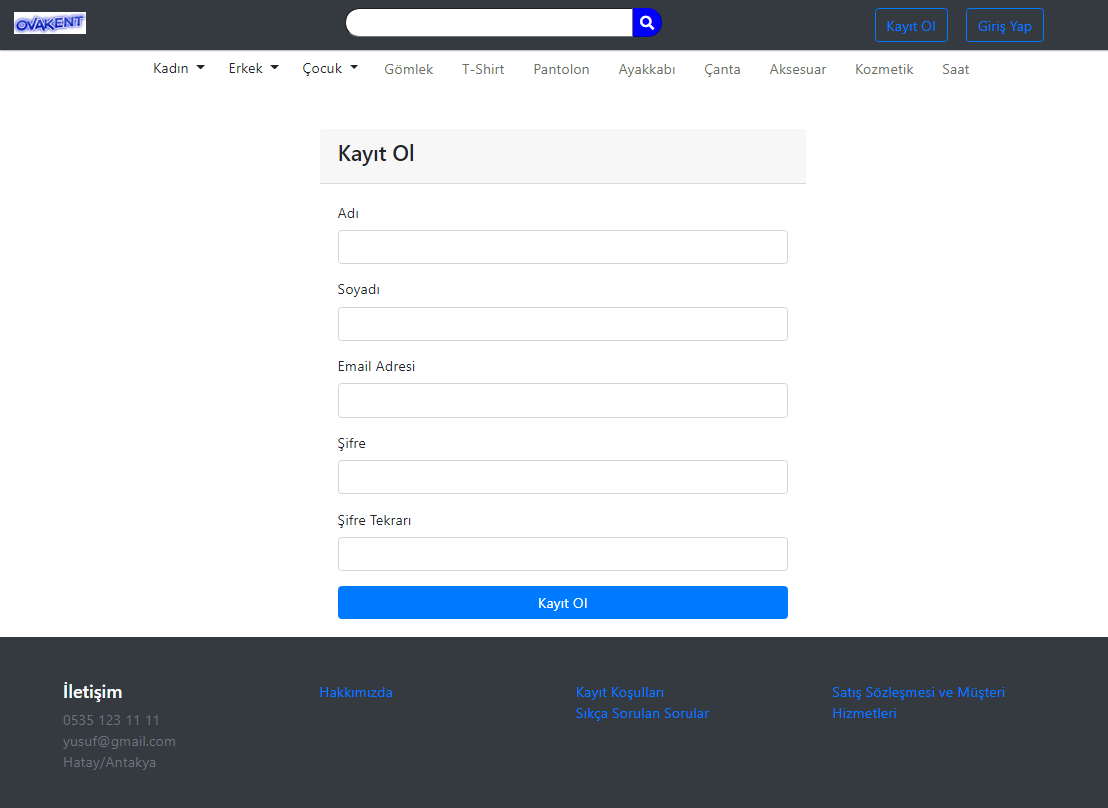
(Resim 2.4.1)

Her bir kategoride dört tane ürün bulunmaktadır. Pantolona tıkladığımızda(Resim 2.4.1) pantolon ile ilgili ürünler bulunmaktadır. Kategorideki herhangi birine tıkladığımız zaman o kategorideki ürünler çıkmaktadır.

Kod kısmı ise anasayfa(index.php) ile aynıdır.

**2.5.Kayıt Ol Sayfası**

Sayfa klasörünün içerisindeki kayıt.php dosyasıdır. Kayıt Ol sayfası kullanıcıların üye olması için oluşturulmuş ve her sayfaya sayfaya eklenmiş bir modal sayfadır. Kayıt ol(1.5.1) sayfasında bir form bulunmaktadır bu forma kullanıcı bilgilerini almaktadır ve Kayıt Ol butonu sayesinde bu bilgiler veri tabanına aktarılmaktadır. Bu veri tabanı ile olan kısmını ayrı bir başlıkta anlatacağım burada ise sadece tasarım kısmıdır.



(Resim 2.5.1)

Kullanıcı kayıt olması için kullanıcının bilgileri boş bırakmayacak şekilde girmesi gerekmektedir. Aksi taktir de kayıt olmaz.

**Kod:**

<div *class*=”container mt-5”>

   <div *class*=”row”>

       <div *class*=”col-6 mx-auto”>

         <div *class*=”card-header bg-bremary”>

         <h4>Kayıt Ol</h4>

       </div>

        <div *class*=”card-body”>

              <?php

                include ‘../php/uyekaydi.php’;

               ?>

         <form *method*=”post”>

             <div *class*=”form-group”>

               <label *for*=”adi”>Adı</label>

             <input *type*=”name” *name*=”adi” *class*=”form-control”>

           </div>

                   <div *class*=”form-group”>

               <label *for*=”soadi”>Soyadı</label>

             <input *type*=”surname” *name*=”soyadi” *class*=”form-control”>

           </div>

                   <div *class*=”form-group”>

               <label *for*=”email”>Email Adresi</label>

             <input *type*=”email” *name*=”email” *class*=”form-control”>

           </div>

             <div *class*=”form-group”>

               <label *for*=”Sifre”>Şifre</label>

             <input *type*=”password” *name*=”sifre” *class*=”form-control”>

           </div>

           <div *class*=”form-group”>

               <label *for*=”Sifree”>Şifre Tekrarı</label>

             <input *type*=”password” *name*=”sifree” *class*=”form-control”>

           </div>

           <uton *class*=”btn btn-primary w-100” *type*=”submit”>Kayıt Ol</uton>

         </form>

       </div>

      </div>

    </div>

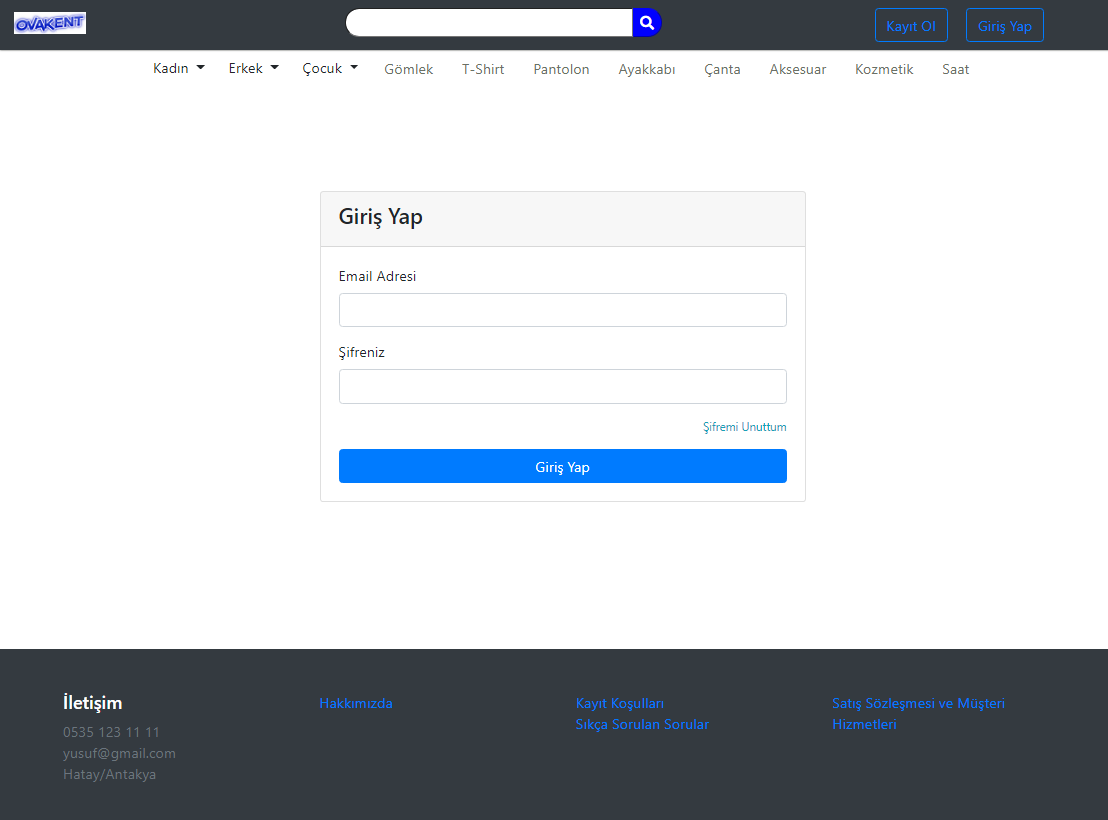
</div>

Bir form içerisinde her bir bilgi için bir input bulunmaktadır.

Php etiketi açarak php sayfasındaki uyekaydi.php dosyasını çağırmaktadır. Bu uyekaydi.php dosyası veritabanı ile olan kısımdır.

**2.6.Giriş Yap Sayfası**

Sayfa klasörünün içerisindeki giriş.php dosyasıdır. Burada kullanıcı girişi sağlanmaktadır. Giriş Yap sayfası üye olan kullanıcıların mail ve şifrelerini girmeleri taktir de siteye giriş yapmalarını sağlayan sayfadır. Bu sayfada yazılan mail adresi ve şifre veri tabanından kontrol edilir. Eğer varsa siteye giriş SESSION’ın başlatılmasıyla sağlanır ve navbar daki kayıt ol ve giriş yap butonları hesabım ve çıkış yap butonlarına dönüşecektir. Yanlış girildiyse girişe izin verilmez.



(Resim 2.6.1)

Giriş yapabilmesi için kullanıcının bilgileri boş bırakmayacak şekilde veritabanına kayıtlı olan bilgilerin girmesi gerekmektedir. Aksi taktir de giriş olmaz.

**Kod:**

<div *class*="container">

   <div *class*="row">

      <div *class*="col-6 mx-auto">

         <div *class*="card">

            <div *class*="card-header bg-bremary">

       <h4>Giriş Yap</h4>

            </div>

       <div *class*="card-body">

         <?php

             include '../php/girisyap.php';

          ?>

          <form *method*="post">

            <div *class*="form-group">

              <label *for*="Email">Email Adresi</label>

            <input *type*="email" *class*="form-control" *name*="email">

          </div>

          <div *class*="form-group">

              <label *for*="Sifre">Şifreniz</label>

            <input *type*="password" *class*="form-control" *name*="sifre">

          </div>

          <div *class*="form-group">

              <a *href*="#" *class*="text-info float-right small mb-3">Şifremi Unuttum</a>

          </div>

          <button *class*="btn btn-primary w-100" *type*="submit">Giriş Yap</button>

          </form>

      </div>

 </div>

 </div>

 </div>

</div>

Bir form içerisinde her bir bilgi için bir input bulunmaktadır.

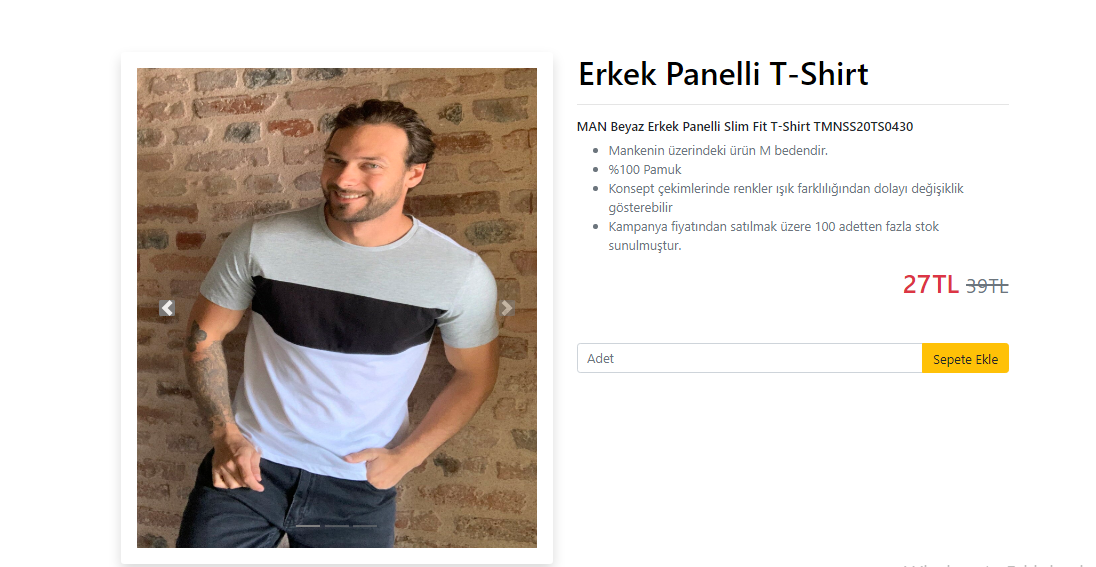
Php etiketi açarak php sayfasındaki girisyap.php dosyasını çağırmaktadır. Bu girisyap.php dosyası veritabanı ile olan kısımdır.

Kullanıcı bilgilerini girildikten sonra giriş sağlandığı zaman bizi ana sayfaya yönlendirecektir.

**2.7.Ürün Sayfası**

Ürün sayfa klasörünün içerisindeki ve kategori klasörlerin içerisindeki dosyalar ürün sayfası dosyalarıdır. Ürün sayfaları ürünlerin 3 farklı resimlerini incelenmek, ürünler hakkında bilgiler ve sepete eklenmek için oluşturulmuş sayfalardır. Her ürün için ayrı ayrı oluşturulmuştur. Giriş yapıldıktan sonra sepete ekleme yapabiliriz.

Her ürününün sepete eklemden önce adet miktarını yazmamız gerekmektedir. Sepete ekle tıkladığımızda ürün bilgileri veri tabanındaki sepetim adlı tabloya kaydedilmektedir.



(Resim 2.7.1)

**Kod:**

    <div *class*="container my-5">

        <div *class*="row">

            <div *class*="col-12 col-md-6">

                <div *class*="card p-0 border-0 shadow">

                    <div *class*="card-body">

                        <div *id*="carousel" *class*="carousel slide" *data-ride*="carousel"> <ol *class*="carousel-indicators">

                  <li *data-target*="#carousel" *data-slide-to*="0" *class*="active" *style*="background-color:gray;"></li>

                  <li *data-target*="#carousel" *data-slide-to*="1" *style*="background-color:gray;"></li>

                  <li *data-target*="#carousel" *data-slide-to*="2" *style*="background-color:gray;"></li>

         </ol><div *class*="carousel-inner">

                       <div *class*="carousel-item active">

                        <img *src*="resim1.jpg " *class*="d-block w-100" *height*="600" *alt*="cimbizaracgerec16"></div>

                        <div *class*="carousel-item">

                        <img *src*="resim2.jpg" *class*="d-block w-100" *height*="600" *alt*="cimbizaracgerec16-1"></div>

                        <div *class*="carousel-item">

                        <img *src*="resim3.jpg " *class*="d-block w-100" *height*="600" *alt*="cimbizaracgerec16-2"></div> </div>

<a *class*="carousel-control-prev" *href*="#carousel" *role*="button" *data-slide*="prev">

<span *class*="carousel-control-prev-icon" *aria-hidden*="true" *style*= "background-color:gray; border-radius: 2px;"></span>

<span *class*="sr-only">Previous</span></a>

<a *class*="carousel-control-next" *href*="#carousel" *role*="button" *data-slide*="next">

<span *class*="carousel-control-next-icon" *aria-hidden*="true" *style*="background-color:gray; border-radius: 2px;"> </span> <span *class*="sr-only">Next</span></a>

                </div></div></div></div>

             <div *class*="col-12 col-md-6">

                <form *method*="POST" *action*="../../php/sepetekle.php">

                    <h1><input *type*="text" *style*="border:none transparent;outline:none;text-align:left;font-weight:500;" *id*="urunAdi" *name*="urunAdi" *value*="Bisiklet Yaka T-Shirt" *readonly*></h1>

<hr>

<h6>Taş Basic Erkek Oversize Bisiklet Yaka Kısa Kollu T-Shirt </h6>

                    <ul *class*="unstyled">

 <li *class*="text-muted">Mankenin üzerindeki ürün M bedendir.</li>

 <li *class*="text-muted">Modelin Ölçüleri: Boy: 1,88 m, Göğüs: 99</li>

 <li *class*="text-muted">%60 Pamuk %40 Polyester</li>

 <li *class*="text-muted">1 Beden küçük alınabilir. </li>

 <li *class*="text-muted"> Konsept çekimlerinde renkler ışık

 farklılığından dolayı değişiklik gösterebilir.</li>

 <li *class*="text-muted"> Bu üründen en fazla 10 adet sipariş

 verilebilir. 10 adetin üzerindeki siparişleri Trendyol iptal etme hakkını saklı tutar</li>

<li *class*="text-muted"> Kampanya fiyatından satılmak üzere 100

 adetten fazla stok sunulmuştur</li>

             </ul>

<h2 *class*="float-right text-danger"><input *type*="text" *class*="text-danger mb-5" *style*="border:none transparent;outline:none;text-align:right;fontweight:500;" *id*="urunFiyat" *name*="urunFiyati" *value*="30" *readonly*>TL  <small><span *class*="text-secondary" *style*="text-align:right"><del>39TL</del></span></small></h2>

                    <div *class*="clearfix"></div>

                    <div *class*="form-group">

                        <div *class*="input-group">

                        <input *class*="form-control" *type*="number" *min*="1" *placeholder*="Adet" *id*="urunAdet" *name*="urunAdet" *required*>

                            <div *class*="input-group-append">

                                <button *class*="btn btn-warning" *name*="sepeteEkle" *id*="sepeteEkle">Sepete Ekle</button>

                            </div>

                        </div>

</div>

                </form>

            </div>

        </div>

</div>

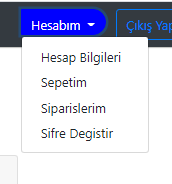
Bootstrap kütüphanesinde olan carousel özelliğini kullandım..Carousel slider yapmamızı sağlıyor. Öncelikle bir div yaratım id atayalım. Bu id slider için etkileşimi için kullanacağız. Bu divin css’ ine carousel slide classlarını veriyoruz. Sliderin otomatik geçiş süresini de data-interval parametresi ile yapıyoruz. Bu şekil de slider hazır. Şimdi İçine 3 alan yerleştiriyoruz. Bunların ikisi,ileri geri hareket için,diğeri ise içerik içindir. Ana divimizin içine diz oluşturup classına carousel-inner verelim. Bu divin içine ise divler açıp classlarına item verelim. Bu itemların içine verileri gireceğiz. İçerik kısmı bu kadar.

Sepete ekle butonuna tıkladığımız zaman php klasöründeki sepetekle.php dosyamızı çağıracaktır. Bu dosya veritabanı ile ilgili kısımdır. Daha sonra bizi sepetim sayfasına yönlendirecektir.

**2.8.Sepetim Sayfası**

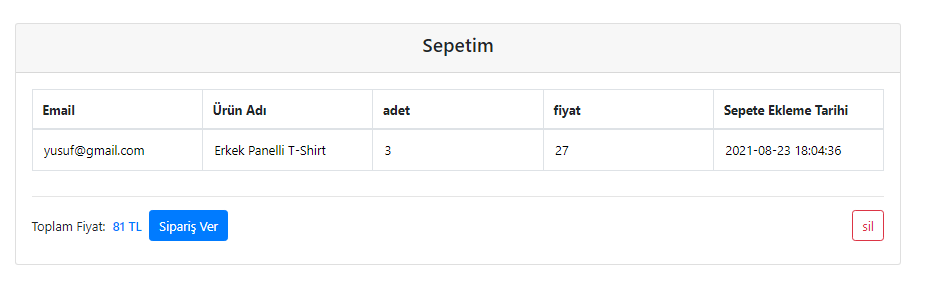
Sayfa Klasörünün içerisindeki sepetim.php dosyasıdır.

Navbar kısmında hesabım butonuna tıkladığımız zaman(Resim 2.8.1) orada çıkan sepetim butonuna tıkladığımız zaman bizi sepetim sayfasına yönlendirecektir.



(Resim 2.8.1)

Sepetim sayfasında veritabanındaki sepetim adlı tablodaki ürün bilgilerini göstermektedir ( veritabanı kısmı başka bir başlıktadır.). Aynı zamanda toplam fiyat, sipariş ver ve sepeti sil kısımları bulunmaktadır.



(Resim 2.8.2)

Sepete eklediğimiz ürünlerin adı, adeti, fiyatı, sepete ekleme tarihi göstermektedir. Bu bilgileri bir tablo halindedir.

**Kod:**

<div *class* ="container">

<div *class*="row">

<div *class*="card mx-auto">

<div *class*="card-header bg-bremary" >

       <h4 *style*="text-align:center">Sepetim</h4>

            </div>

     <div *class*="card-body">

                <form *action*="../php/sepet.php" *method*="get">

                      <?php

                        include "../php/sepet.php";

                       ?>

                 </form>

               <div *class*="form-group">

                <label *for*="toplamFiyat" *class*=" float-left col-form-label">Toplam Fiyat: &nbsp </label>

       <?php include "../php/fiyatopla.php"; ?>

<h6 *class*="float-left text-primary"><label *for*="toplamFiyat" *class*= " float-left col-form-label"><?php echo $toplamFiyat; ?> TL &nbsp </label></h6>

                   </div>

               <form *action*="../php/sepetsil.php" *method*="post">

                <button *class*="btn btn-outline-danger float-right ml-2" *name*="sil" *id*="sil">sil</button>

              </form>

               <button *class*="btn btn-primary float-left" *data-toggle*="modal" *data-target*="#siparisModal">Sipariş Ver</button>

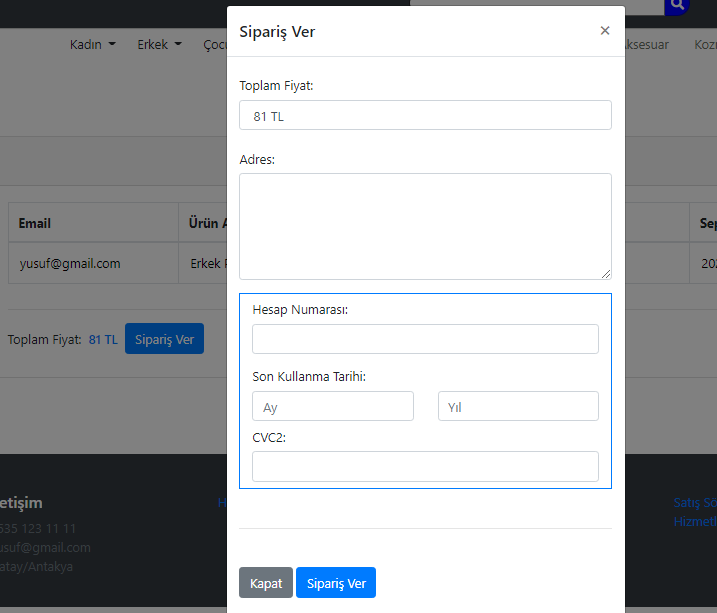
Container ve row classlarıyla div etiketi açıyoruz. Bir form içerisinde php etiketleriyle php klasörünün içerindeki sepet.php dosyasını ekleniyor. Bu dosya ürün ile ilgili Resim 2.8.2 de gördüğümüz gibi bilgileri bir tablo halinde getirecektir.

Toplam fiyat ise php klasörünün içerisindeki fiyatopla.php dosyasını ekleyerek sepetteki ürünlerin toplam fiyatını gösterecektir.

Sepet sil ise php klasörünün içerisindeki sepetsil.php dosyasını ekleyerek sepetteki ürünleri temizleyebiliriz.

**2.9.Siparis Ver Sayfası**

Sepetim sayfasında sipariş ver butonuna tıkladığımız zaman bir form çıkacak ve istenilen bilgileri doldurmamız gerekecek(Resim 2.9.1) .



(Resim 2.9.1)

Bu sayfada gerekli bilgiler forma girildikten sonra Öde butonuna basılıp ödeme işlemi gerçekleşmektedir ve sepet sıfırlanmaktadır. Ödeme işlemi gerçek olmayıp ödeniyormuş gibi gösterilmektedir. Burada ürünü kargoya verdiğimiz zaman göndereceğimiz adres ile toplam fiyat veri tabanına eklenmektedir.

Sipariş ver butonuna tıkladığımız zaman php klasörünün içinde order.php dosyasını çağırmaktadır. Burada veritabanındaki siparişler adlı tabloya kaydedilmektedir. Daha sonra siparişlerim sayfasına yönlendirilecektir.

**Kod:**

<div *class*="modal fade" *id*="siparisModal" *tabindex*="-1" *role*="dialog" *aria-labelledby*="siparisModalLabel" *aria-hidden*="true">

<div *class*="modal-dialog" *role*="document">

<div *class*="modal-content">

   <div *class*="modal-header">

    <h5 *class*="modal-title" *id*="siparisModalLabel">Sipariş Ver</h5>

    <button *type*="button" *class*="close" *data-dismiss*="modal" *aria-label*="Close">

     <span *aria-hidden*="true">&times;</span>

    </button>

    </div>

    <div *class*="modal-body">

     <form *action*="../php/order.php" *method*="POST">

     <div *class*="form-group">

<label *for*="toplamFiyat" *class*="col-form-label">Toplam Fiyat: </label>

<input *type*="text" *class*="form-control bg-transparent" *id*="toplamFiyat" *name*="toplamFiyat" *value*=" <?php echo $toplamFiyat; ?> TL" *READONLY*>

     </div>

     <div *class*="form-group">

       <label *for*="adres" *class*="col-form-label">Adres: </label>

<textarea *class*="form-control" *id*="adres" *name*="adres" *rows*="5"></textarea>

      </div>

      <div *class*="form-group">

      <div *class*="container border border-primary mb-2">

<label *for*="hesapNumara" *class*="col-form-label"> Hesap Numarası: </label>

<input *type*="text" *class*="form-control mb-2" *id*="hesapnumara" *name*="hesapnumara" *maxlength*="16">

       <label *for*="sonKullanmaTarihi" *class*="col-form-label"> Son Kullanma Tarihi: </label>

        <div *class*="row">

        <div *class*="col">

        <input *type*="text" *class*="form-control w-100" *id*=" sktarihi" *name*="sonkullanmatarihiay" *maxlength*="2" *placeholder*="Ay">

        </div>

        <div *class*="col">

        <input *type*="text" *class*="form-control w-100" *id*="sktarihi" *name*="sonkullanmatarihiyil" *maxlength*="2" *placeholder*="Yıl">

        </div>

       </div>

        <label *for*="cvc2" *class*="col-form-label">CVC2: </label>

<input *type*="text" *class*="form-control mb-2" *maxlength*="3" *id*="cvc2" *name*="cvc2">

    </div>

     </div>

     <hr *class*="my-5">

       <div *class*="form-group">

<button *type*="button" *class*="btn btn-secondary" *data-dismiss*="modal"> Kapat</button>

<button *type*="submit" *class*="btn btn-primary" *id*="ver" *name*="ver"> Sipariş Ver</button>

             </div>

           </form>

         </div>

      </div>

 </div>

Form için gerekli olan classlarla etiketleri açıyoruz. Burada <?php echo $toplamFiyat; ?> ile veri tabanından toplam fiyatını alarak yazdırıyoruz.

Gerekli inputları girerek order.php dosyasını çağıracak ve siparis vermiş olacağız. Daha sonra siparişlerim sayfasını yönlendirilecektir.

**2.10.Siparişlerim Sayfası**

Sayfa Klasörünün içerisindeki siparislerim.php dosyasıdır.

Siparişlerim sayfasında veritabanındaki siparisler adlı tablodaki bilgilerini göstermektedir(veritabanı kısmı başka bir başlıktadır).



(Resim 2.10.1)

Sepete eklediğimiz ürünlerin adı, kargo durumu, kargo takip numarası bilgilerini göstermektedir. Bu bilgileri bir tablo halindedir. Kargo durumu ise admin tarafından kargoya verildiğinde veritabanındaki kargo durumunu değiştirerek burada da kargoya verildi diye değişecektir. Kargo takip numarası ise yine admin tarafından verilecektir.



(Resim 2.10.2)

**Kod:**

<div *class* ="container">

<div *class*="row">

<div *class*="card mx-auto">

<div *class*="card-header bg-bremary" >

       <h4 *style*="text-align:center">Siparislerim</h4>

            </div>

     <div *class*="card-body">

               <?php

                 include "../php/siparislerimphp.php";

                ?>

</div>

</div>

</div>

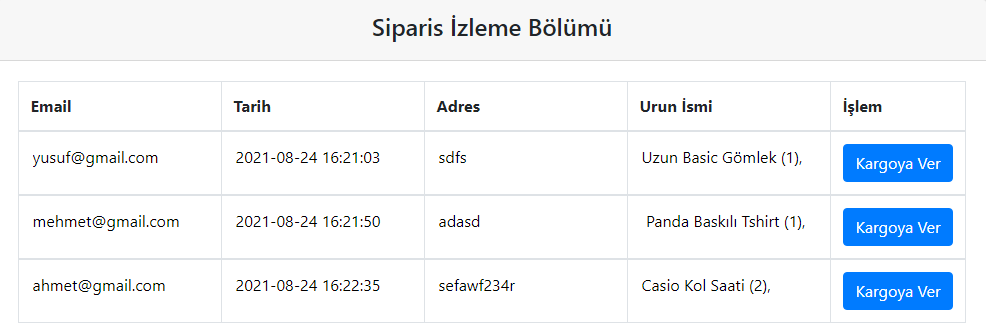
</div>

Burada php klasöründeki siparislerimphp.php dosyasını çağırarak tablo halinde bilgiler gelecektir.

**2.11.Admin Sayfası:**

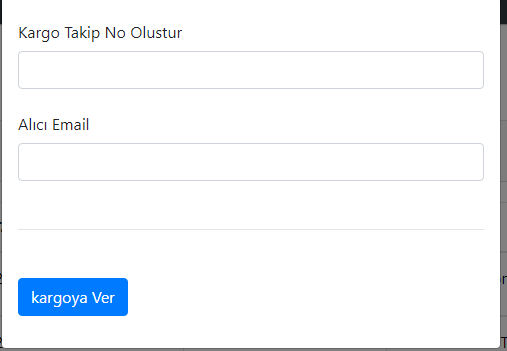
Sayfa Klasörünün içerisindeki admin.php dosyasıdır.

Admin sayfasına giriş yapabilmek için [admin@gmail.com](mailto:admin@gmail.com) email adresi ile admin şifresini girerek admin sayfasına giriş yapmış olursunuz. Admin sayfasında ise her kullanıcının sipariş verdiği ürünlerin listesi halinde gelecektir. Admin bu kullanıcıların siparis verdiği ürünleri kargoya verecektir ve kargo takip numarası oluşturacaktır.



(Resim 2.11.1)

Kargoya ver dediğimizde karşımıza bir form çıkacaktır. Formda ise kargo takip numarası girmemizi isteyecektir(Resim 2.11.2).



(Resim 2.11.2)

Burada kargo takip numarası kendimiz oluşturacağız. Yalnız alıcı email ise siparis verin kişinin email ini yazmamız gerekmektedir.

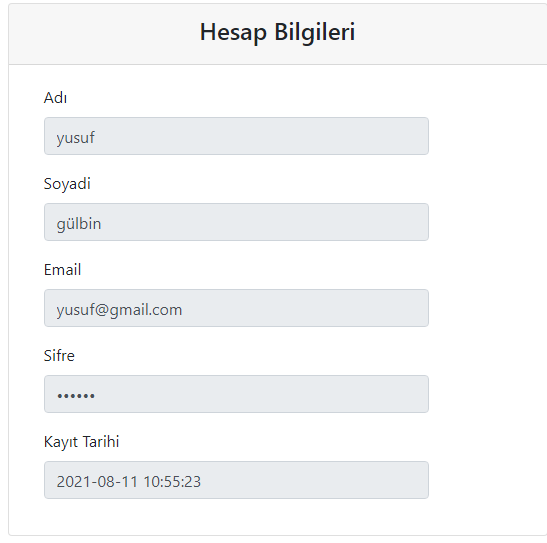
Kargoya ver butonuna tıklandığı zaman veritabanındaki siparişler tablosundaki kargo durumunu “Kargoya verildi” diye değiştirecektir. Aynı zamanda kargo takip numarasını tabloya ekleyecektir.

Kod kısmını başka başlıkta anlatacağım.

**2.12.Hesap Bilgileri Sayfası:**

Sayfa Klasörünün içerisindeki hesapbilgileri.php dosyasıdır.

Burada ise kullanıcının adı, soyadı, email, şifre, kayıt tarihi bilgileri tablo halinde gösterilmektedir. Kullanıcıya ait bilgilerin veritabanındaki uyeler isimli tablodaki hesap bilgilerini çekecektir.



(Resim 2.12.1)

**Kod:**

Kısaca bahsedeyim. İlk önce php klasöründeki hesapbilgilerinicek.php dosyasını ekleyecektir. Oda şöyledir.

 <?php

       include '../php/hesapbilgisicek.php' ;

       ?>

Daha sonra bu dosyadan gelen bilgileri inputlara yazdırılacaktır oda şöyle:

<div *class*="card-body">

  <div *class*="form-group">

   <div *class*="col-sm-10">

    <label *for*="adi">Adı</label>

    <input *type*="text" *class*="form-control" *id*="adi" *placeholder*= Ad" *name*="adi" *value*=" <?php echo $adi;?> " *READONLY*>

   </div></div>

  <div *class*="form-group">

  <div *class*="col-sm-10">

  <label *for*="adi">Soyadi</label>

  <input *type*="text" *class*="form-control" *id*="soyadi" *placeholder*= "Soyad" *name*="soyadi" *value*=" <?php echo $soyadi;?> " *READONLY*>

   </div></div>

  <div *class*="form-group">

   <div *class*="col-sm-10">

  <label *for*="adi">Email</label>

    <input *type*="email" *class*="form-control" *id*="email" *placeholder*= "Email" *name*="email" *value*=" <?php echo $email;?> " *READONLY*>

    </div></div>

    <div *class*="form-group">

     <div *class*="col-sm-10">

      <label *for*="adi">Sifre</label>

      <input *type*="password" *class*="form-control" *id*="sifre" *placeholder*="Şifre" *name*="sifre" *value*="<?php echo $sifre;?>" *READONLY*>

     </div></div>

     <div *class*="form-group">

     <div *class*="col-sm-10">

      <label *for*="adi">Kayıt Tarihi</label>

      <input *type*="text" *class*="form-control" *id*="tarih" *placeholder*="Şifre" *name*="tarih" *value*="<?php echo $tarih;?>" *READONLY*>

        </div> </div>

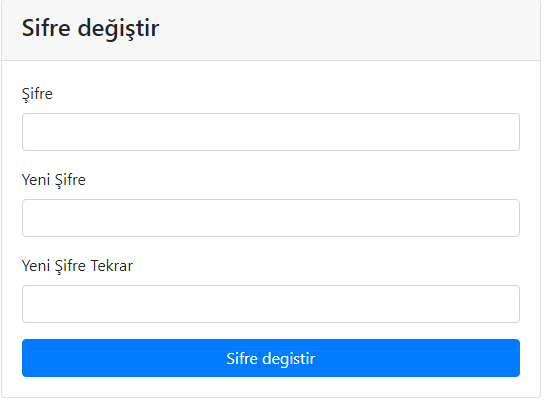
 </div>

Hesapbilgilericek.php den gelen bilgileri "<?php echo $ad;?>", "<?php echo $soyad;?>", "<?php echo $emal;?>", "<?php echo $sifre;?>", "<?php echo $tarih;?>" kodları ile inputların value değerine yazdırdım.

**2.13. Şifre Değiştir Sayfası:**

Sayfa Klasörünün içerisindeki sayfa.php dosyasıdır.

Kullanıcının şifresini değiştirebilmek için hazırladığım sayfadır.



(Resim 2.13.1)

Burada ise mevcut şifreyi daha sonra oluşturacak yeni şifreyi göstermektedir. Yeni şifre oluşturabilmesi için mevcut şifreyi doğru girmesi gerekmektedir aynı zamanda yeni şifre ile yeni şifre tekrar aynısı olması gerekmektedir. Aksi halde şifre değişmeyecektir.

Şifre değiştir butonuna basıldığı zaman kullanıcının veritabanındaki uyeler tablosundaki şifresi değişecektir.

**Kod:**

<div *class*="container">

   <div *class*="row">

      <div *class*="col-6 mx-auto">

         <div *class*="card">

            <div *class*="card-header bg-bremary">

       <h4>Sifre değiştir</h4>

            </div>

       <div *class*="card-body">

         <?php

             include '../php/sifrephp.php';

          ?>

          <form *method*="post">

            <div *class*="form-group">

              <label *for*="Email">Şifre</label>

            <input *type*="password" *class*="form-control" *name*="sifre">

          </div>

          <div *class*="form-group">

              <label *for*="Sifre">Yeni Şifre</label>

 <input *type*="password" *class*="form-control" *name*="yenisifre">

          </div>

          <div *class*="form-group">

              <label *for*="Sifre">Yeni Şifre Tekrar</label>

<input *type*="password" *class*="form-control" *name*="yenisifretekrar">

          </div>

<button *class*="btn btn-primary w-100" *type*="submit"> Sifre degistir</button>

          </form>

      </div>

 </div>

 </div>

</div>

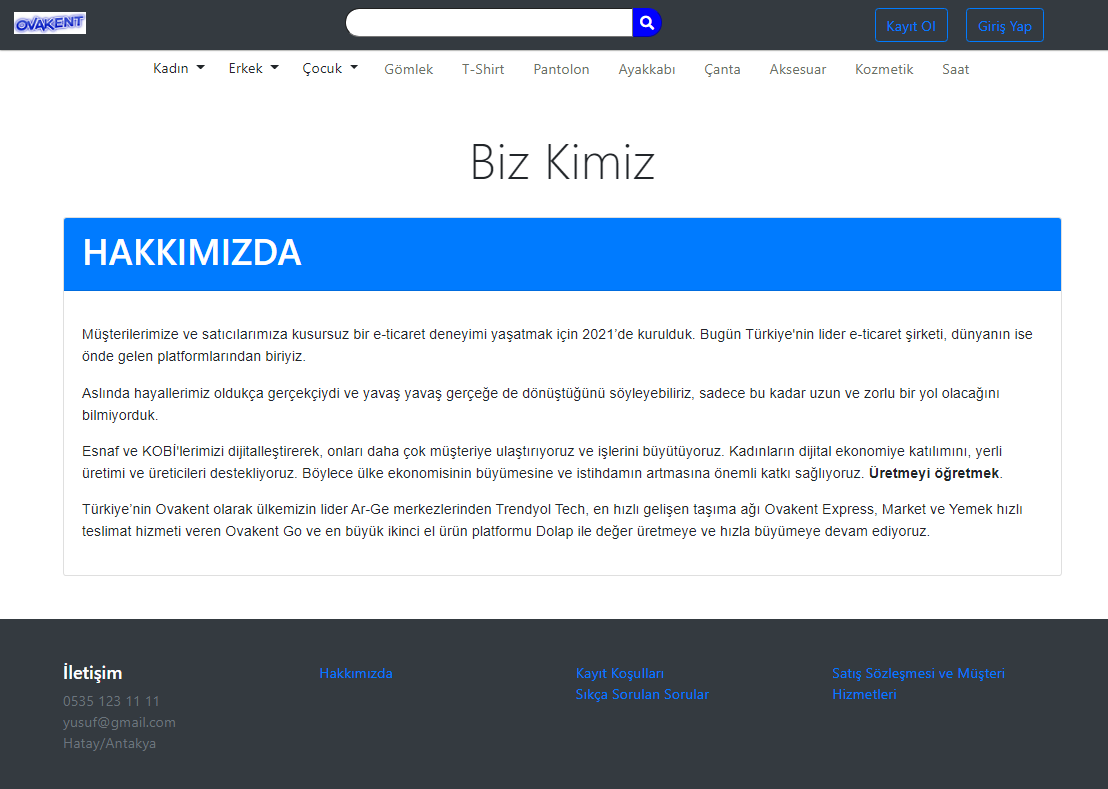
</div>

php klasöründeki şifrephp.php dosyasını ekliyoruz. İnputlara girdiğimiz bilgileri Sifre değiştir butonuna basıldığı zaman şifrephp.php gönderiyor. şifrephp.php koşullarını sağlarsa şifre değişmiş olacaktır.

**2.14.Hakkımızda Sayfası:**

Sayfa Klasörünün içerisindeki hakkimizda.php dosyasıdır.

İnternet sitesi ile ilgili “Biz Kimiz” başlıklı bilgiler bulunmaktadır. Hakkımızda sayfası marka hakkında genel bir bilgi vermek amacıyla oluşturulmuş sayfadır.



(Resim 2.14.1)

**Kod:**

<div *class*="container my-5">

<h1 *data-aos*="fade-up" *class*="display-4 text-center mb-2">Biz Kimiz</h1><br>

        <div *class*="card">

            <div *class*="card-header text-white bg-primary mb-3">

                    <h1>HAKKIMIZDA</h1>

            </div>

            <div *class*="card-body">

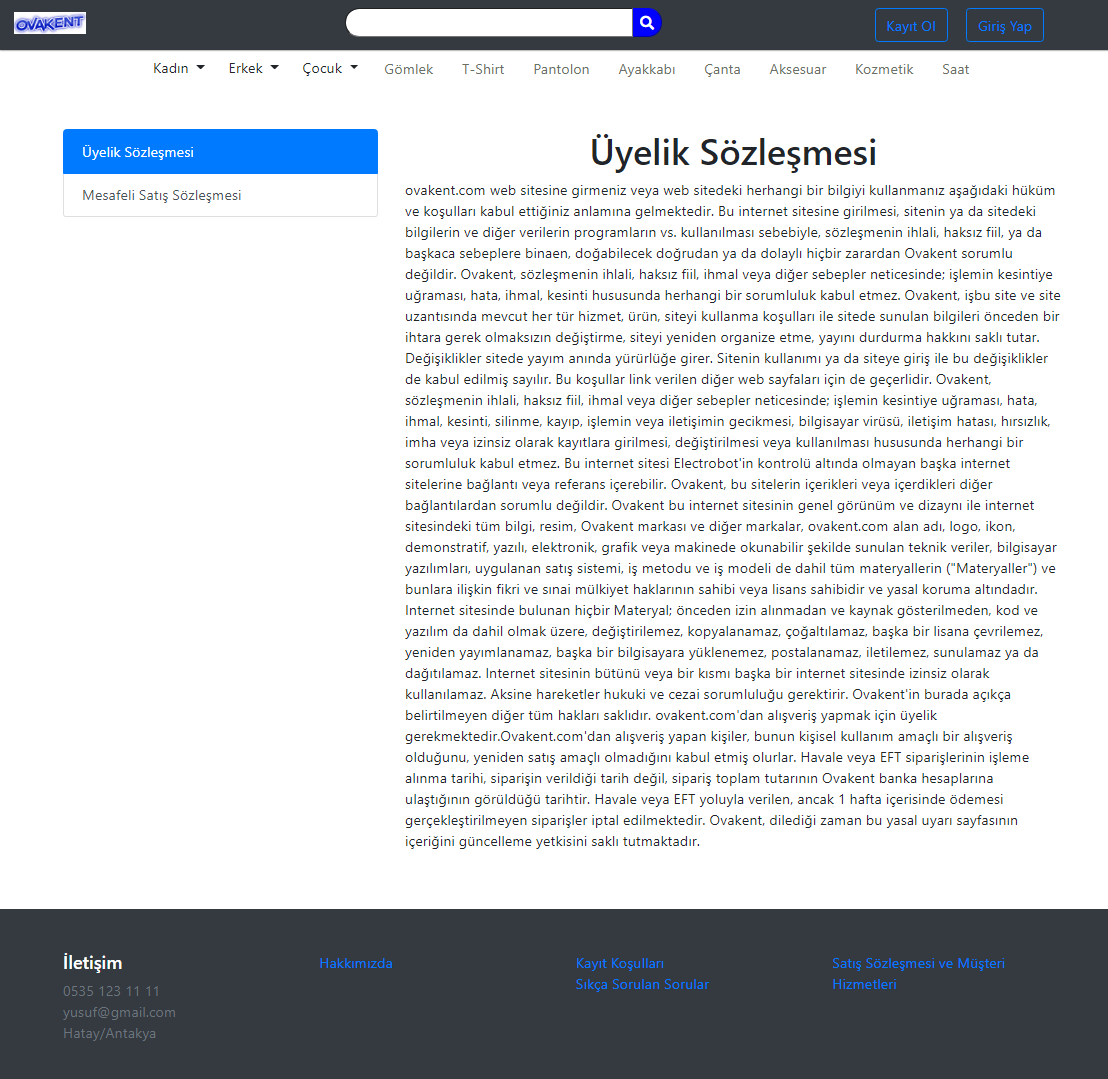
             <p><span *style*="font-family: helvetica; font-size: 12pt;">bilgileri buraya yazıyoruz...<br></span></p>

  </div> </div></div>

**2.15.Kayıt Koşulları:**

Sayfa Klasörünün içerisindeki kayitkosullari.php dosyasıdır.

İnternet sitesi ile ilgili “Kayıt Koşulları” başlıklı bilgiler bulunmaktadır. Kayıt koşulları sayfası kayıt ile ilgili site hakkında genel bir bilgi vermek amacıyla oluşturulmuş sayfadır.

(Resim 2.15.1)

**Kod:**

Önemli kod parçası

 <div *class*="container my-5">

       <h1 *data-aos*="fade-up" *class*="display-4 text-center mb-2"></h1>

            <div *class*="row">

                <div *class*="col-4">

                  <div *class*="list-group" *id*="list-tab" *role*="tablist">

 <a *class*="list-group-item list-group-item-action active" *id*="list-home-list" *data-toggle*="list" *href*="#list-home" *role*="tab" *aria-controls*="home">Üyelik Sözleşmesi</a>

                    <a *class*="list-group-item list-group-item-action" *id*="list-profile-list" *data-toggle*="list" *href*="#list-profile" *role*="tab" *aria-controls*="profile">Mesafeli Satış Sözleşmesi</a>

                </div>

            </div>

            <div *class*="col-8">

                <div *class*="tab-content" *id*="nav-tabContent">

                    <div *class*="tab-pane fade show active" *id*="list-home" *role*="tabpanel" *aria-labelledby*="list-home-list"><center><h1>Üyelik Sözleşmesi</h1></center>

<p> Buraya paragrafı yazıyoruz...</p>

                    </div>

                <div *class*="tab-pane fade" *id*="list-profile" *role*="tabpanel" *aria-labelledby*="list-profile-list"><center><h1>Mesafeli Satış Sözleşmesi</h1></center>

  Burayada paragrafları yazıyoruz...

                        <table *class*="table table-bordered">

                            <tbody>

                            <tr>

                                <td>SATICI</td>

                                <td>ALICI</td>

                            </tr>

                            </tbody>

                        </table>

                    </div>

                </div>

            </div>

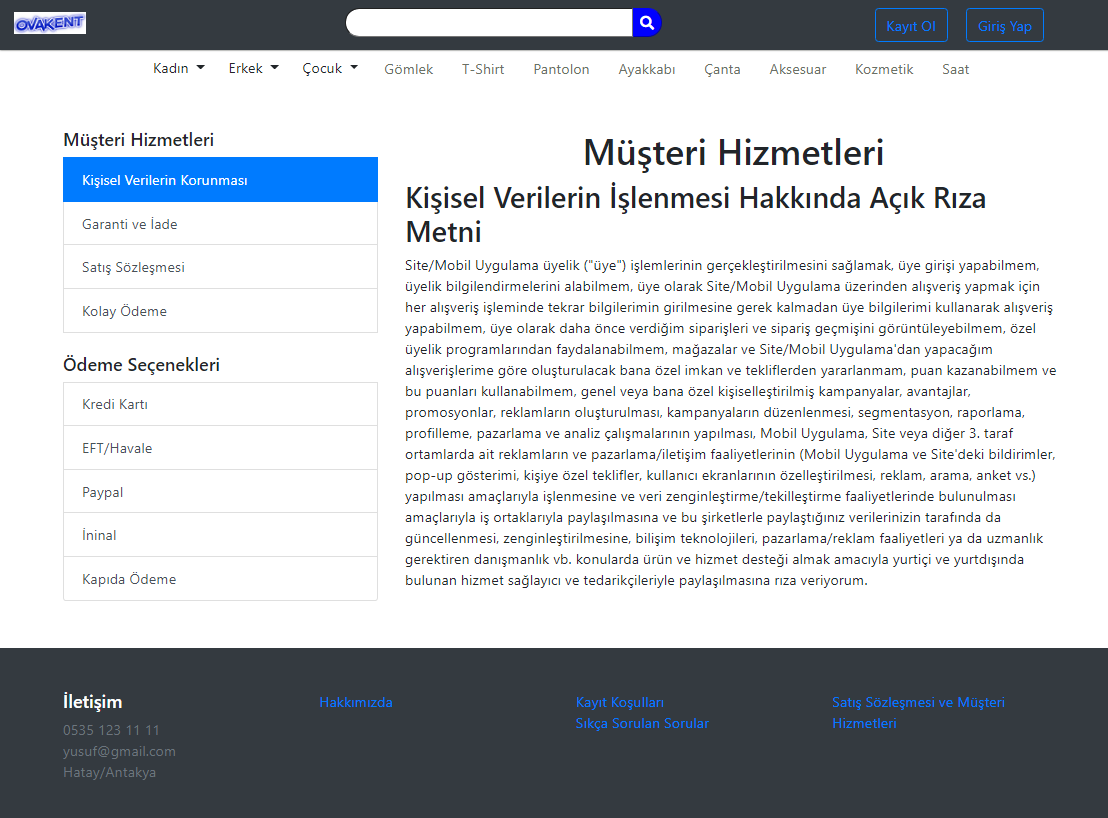
          </div>

        </div>

**2.16.Müşteri Hizmetleri Sayfası:**

Sayfa Klasörünün içerisindeki musterihizmetleri.php dosyasıdır.

İnternet sitesi ile ilgilimüşteri hizmetleri, bize nasıl ulaşacağı bilgiler bulunmaktadır. Site hakkında genel bir bilgi vermek amacıyla oluşturulmuş sayfadır.

(Resim 2.16.1)

**Kod:**

<div *class*="container my-5">

<div *class*="row">

  <div *class*="col-4">

    <div *class*="list-group" *id*="list-tab" *role*="tablist">

      <h5>Müşteri Hizmetleri</h5>

      <a *class*="list-group-item list-group-item-action active" *id*="list-home-list" *data-toggle*="list" *href*="#list-home" *role*="tab" *aria-controls*="home"> Kişisel Verilerin Korunması</a>

      <a *class*="list-group-item list-group-item-action" *id*="list-profile-list" *data-toggle*="list" *href*="#list-profile" *role*="tab" *aria-controls*="profile">Garanti ve İade</a>

      <a *class*="list-group-item list-group-item-action" *id*="list-messages-list" *data-toggle*="list" *href*="#list-messages" *role*="tab" *aria-controls*="messages">Satış Sözleşmesi</a>

      <a *class*="list-group-item list-group-item-action" *id*="list-settings-list" *data-toggle*="list" *href*="#list-settings" *role*="tab" *aria-controls*="settings">Kolay Ödeme</a>

     <br>

    <h5>Ödeme Seçenekleri</h5>

    <a *class*="list-group-item list-group-item-action" *id*="list-a-list" *data-toggle*="list" *href*="#list-a" *role*="tab" *aria-controls*="a">Kredi Kartı</a>

    <a *class*="list-group-item list-group-item-action" *id*="list-b-list" *data-toggle*="list" *href*="#list-b" *role*="tab" *aria-controls*="b">EFT/Havale</a>

    <a *class*="list-group-item list-group-item-action" *id*="list-c-list" *data-toggle*="list" *href*="#list-c" *role*="tab" *aria-controls*="c">Paypal</a>

    <a *class*="list-group-item list-group-item-action" *id*="list-d-list" *data-toggle*="list" *href*="#list-d" *role*="tab" *aria-controls*="d">İninal</a>

    <a *class*="list-group-item list-group-item-action" *id*="list-e-list" *data-toggle*="list" *href*="#list-e" *role*="tab" *aria-controls*="e">Kapıda Ödeme</a>

    </div>

  </div>

  <div *class*="col-8">

    <div *class*="tab-content" *id*="nav-tabContent">

      <div *class*="tab-pane fade show active" *id*="list-home" *role*="tabpanel" *aria-labelledby*="list-home-list"><center><h1>Müşteri Hizmetleri</h1></center>

     <h2>Kişisel Verilerin İşlenmesi Hakkında Açık Rıza Metni</h2>

        <p> Buraya bilgiler yazılır... </p>

    <div *class*="tab-pane fade" *id*="list-c" *role*="tabpanel" *aria-labelledby*="list-c-list"><p>...</p></div>

    <div *class*="tab-pane fade" *id*="list-d" *role*="tabpanel" *aria-labelledby*="list-d-list"><p>...</p></div>

    <div *class*="tab-pane fade" *id*="list-e" *role*="tabpanel" *aria-labelledby*="list-e-list"><p>...</p></div>

    </div>

  </div>

</div>

</div>

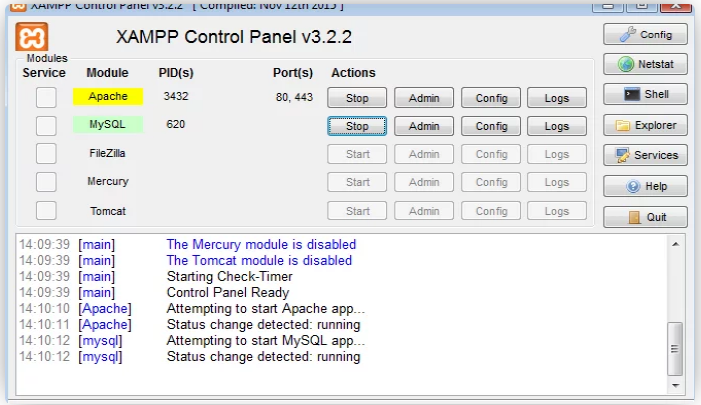
**3.SUNUCU TARAFI**

**3.1.Xampp:**

Xampp web sunucu yazılımıdır. Bilgisayara Php, MariaDB, Perl, Apache, Filezilla, MercuryMail ve tomcat gibi birden çok sistemleri kurarak hazır web sunucusu ortamı oluşturur. xammp aslında paket web sunucu yazılımıdır. xammp içinde phpmyadmin kurulu halde gelmektedir.

Bilgisayarınızda sanal olarak web sunucu ortamı hazırlamanıza olanak tanır. Hosting kiralamanıza gerek kalmadan kişisel bilgisayarınızda ücretsiz hosting ortamını hazırlayabilirsiniz. Bu web sunucu sizin kişisel bilgisayarınızda kurulur ve sadece siz erişebilirsiniz. Dış ortamdan erişim sağlanamaz.

Xampp Control Panel görünümü aşağıdaıdır.

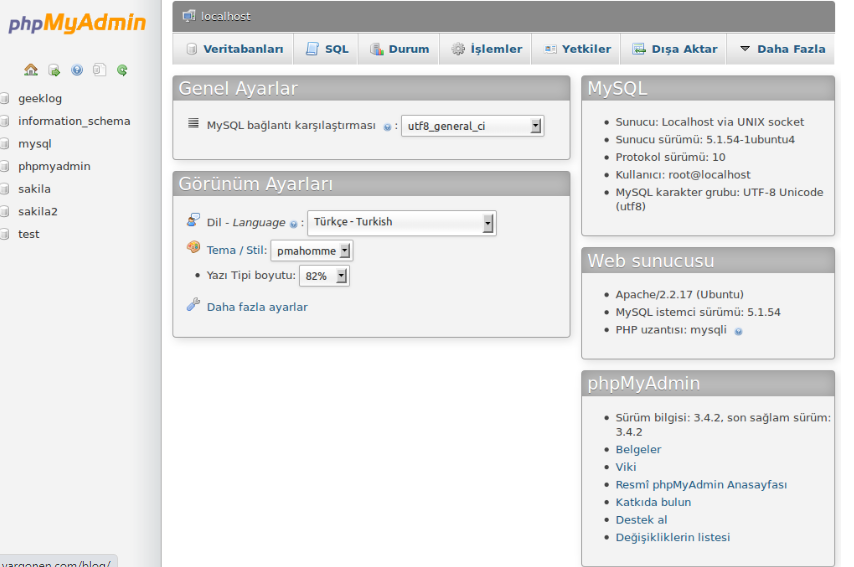
(Resim 3.1.1)

**Tam kapsamlı anlatım bu linktedir.** [**https://www.webhostuzmani.com/xampp-nedir-ne-ise-yarar-kurulumu-resimli/**](https://www.webhostuzmani.com/xampp-nedir-ne-ise-yarar-kurulumu-resimli/)

**3.2. PhpMyAdmin:**

PhpMyAdmin, bir veri tabanı yönetim yazılımıdır. PHP programlama diliyle yazılan sistem, web tabanlı bir altyapıya sahiptir. Başlıca kullanım alanı MySQL veri tabanı yönetimi olan phpMyAdmin, 80 farklı programlama dilini destekler. Veri tabanı oluşturma, değiştirme, silme işlemlerini pratik ve hızlı bir şekilde yapabilme imkânı sunan yazılımdır.

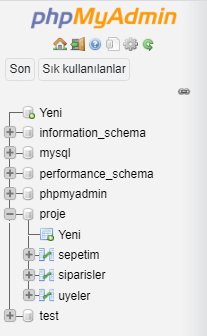
Ayrıca veri tabanına tablo ekleme, değiştirme, onarma, silme ve ön ek ekleme olanağı tanıyan yazılım, SQL sorgularını da çalıştırarak alan anahtarlarını yönetebilmenizi sağlar. Türkçe bir arayüze sahip olan program, çoklu sunucuları da denetleyebilme imkânı sunar. Ayrıca sunucu bakım işlemlerinizi kolaylaştıran phpMyAdmin, MyISAM tablolarında veri tutarlılığını kontrol edebilmenize yardımcı olur.

(Resim 3.2.1)

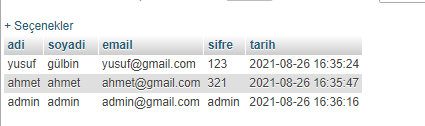
**Tam kapsamlı anlatım bu linktedir.** [**https://www.vargonen.com/blog/phpmyadmin-nedir/**](https://www.vargonen.com/blog/phpmyadmin-nedir/)

**3.3.Veri Tabanı Tabloları:**

Veri tabanı tabloları phpMyAdmin de MySQL ile oluşturulmuştur. Sitede kullanılan proje adlı veritabanında 3 farklı tablo bulunmaktadır.(Resim 3.3.1).



(Resim 3.3.1)



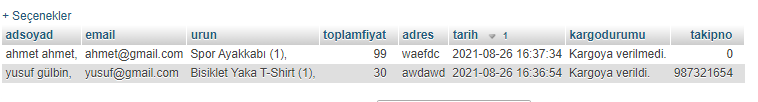
(Resim 3.3.2)

Kullanıcı kaydı için kullanılan uyeler tablosudur. Bu tablo 5 sütundan oluşmaktadır. Bu sütunlar kullanıcının adi, soyadi, email, şifre ve tarih(kullanıcın kayıt olduğu zamanı) bilgilerini tutmaktadır.



(Resim 3.3.3)

Sepete eklenen ürünler için sepetim tablosu kullanılmıştır. Bu tablo 5 adet sütundan oluşmaktadır. Kullanıcının sepete ürün eklediği zaman sepetim tablosunda email, urunismi, adet(ürün adeti), fiyat, tarih(ürünü sepete ekleme tarihi) bilgilerini kaydeder.



(Resim 3.3.4)

Siparisler tablosu kullanıcı sepetteki ürünler sipariş verdiği zaman sepetim tablosundan silinerek siparisler tablosuna kaydedilir. Siparisler tablosu 8 sütundan oluşmaktadır. Bu sütunlar adsoyad, email, urun(ürünün ismi), toplamfiyat, adres, tarih, kargodurumu, takipno bilgilerini tutmaktadır.

Kullanıcı sipariş verdiği zaman kargodurumu “kargoya verilmedi” ve takipno 0 olarak siparisler tablosuna kaydedilir. Admin tarafından ürün kargoya verildiği zaman kargodurumu “kargoya verildi” olarak değişir ve adminin vermiş olduğu kargo takipno kaydedilir.

**3.4. Veri Tabanı Bağlantısı:**

PHP uygulamalarındaki veritabanlarına erişmek ve bunları yönetmek için hızlı ve tutarlı bir arayüz sağlayan bir veritabanı erişim katmanı olan PDO(Php Data Object) yu kullandım. PDO, veritabanlarıyla bağlantı kurma ve çalışma sürecini büyük ölçüde kolaylaştıran veri erişim katmanıdır.

PDO İle Aşağıdaki şekilde bağlantı yapabilmek için bir bağlantı dizgesi oluşturmamız gerekir. Bu bağlantı dizgesinde temel olarak aşağıdaki parametreleri verebiliriz.

* Veri tabanı host (default olarak localhost)
* Veri tabanı kullancıı adı (default root)
* Veri tabanı kullanıcı şifre (default olarak boş bırakılır. Siz mysql kullanıcısı için bir şifre belirlediyseniz o şifreyi girmeniz gerekiyor)
* Veri tabanı adı(proje)

Php klasörünün içindeki baglan.php olan dosyada veri tabanı bağlantısı sağlanmaktadır. Bu bağlantıyı veritabanı ile ilgili olan tüm dosyaya eklenmelidir.

**baglan.php**

<?php

try{

    $DBConnection = new PDO("mysql:host=localhost;dbname=proje;charset=utf8;",'root','');

}catch(PDOException $e){

    echo $e->getMessage();

}

?>

Öncelikle try bloğu çalıştırılır eğer bir hata varsa catch kısmına bir hata fırlatılır. Bu hatayı catch kısmı yakalar ve bize hatayı gösterir.

Kodu çalıştırdığımızda bilgiler doğruysa bağlantı başarılı yanlış ise de neyin hatalı olduğu ile alakalı bir bilgi mesajını ekranda göreceğiz.

**3.5. Üye Kaydı:**

Php klasörünü içerisinde uyekaydi.php dosyasıdır.

Burada kullanıcı kayıt ol sayfasındaki(Resim 2.5.1) forma veriler girdiğinde kayıt ol butonuna basıldığında kayıt başarılı dediği zaman uyekaydi.php çalışır. Verileri veri tabanındaki uyeler tablosuna ekler.

**uyekaydi.php**

<?php

include 'baglan.php';

if($\_POST){

$Gelenadi = $\_POST['adi'];

$Gelensoyadi = $\_POST['soyadi'];

$Gelenemail=$\_POST['email'];

$Gelensifre=$\_POST['sifre'];

$Gelensifree=$\_POST['sifree'];

if ($Gelenadi != "" and $Gelensoyadi != "" and $Gelenemail != "" and  $Gelensifre != "" and $Gelensifree != ""){

   if ($Gelensifree == $Gelensifre){

      $emailsorgu=$DBConnection->prepare("select \* from uyeler where email = ? LIMIT 1");

          $emailsorgu->execute([$Gelenemail]);

          $emailsorgusayi=$emailsorgu->rowCount();

          if($emailsorgusayi > 0){

              echo '<div class="alert alert-danger">Bu email adresi kullanımda, başka adres kullanın!!!</div>';

          }else{

           $sorgu = $DBConnection -> prepare ("INSERT INTO uyeler (adi, soyadi, email, sifre) VALUES(?,?,?,?)");

          $sorgu -> execute([$Gelenadi,$Gelensoyadi,$Gelenemail, $Gelensifre]);

          $sorgusayi = $sorgu -> rowCount();

            if($sorgusayi > 0) {

              echo '<div class="alert alert-success">Kayıt Başarılı</div>';

            } else { echo '<div class="alert alert-danger">Kayıt Hatalı(Başarısız), Tekrar Deneyiniz</div>';

         }}} else {  echo '<div class="alert alert-danger">Şifreler uyuşmuyor</div>';

     }} else{ echo '<div class="alert alert-danger">Lütfen Boş Bırakmayınız</div>';  }} ?>

İlk önce veri tabanı bağlantısı için baglan.php eklememiz lazım. Kayıt ol sayfasındaki(Resim 2.5.1) doldurulan formdaki verileri $\_POST ile alıp bir değişkene atarız. Mesela formdan gelen adi verisini $Gelenadi = $\_POST['adi']; ile değişkene atamış oluruz.

Daha sonra bu değişkenleri kontrol ederiz. Eğer formdaki(Resim 2.5.1) veriler boş bırakılmışsa “lütfen boş bırakmayınız” uyarısı gönderilecektir. Boş değilse yazdığımız kod çalışacaktır.

Daha sonra şifre ile şifre tekrarı kontrol edilecektir. Eğer şifre birbirine eşit değilse “şifreler uyuşmuyor” uyarısı gönderilecektir. Eşit ise yazdığımız kod çalışacaktır.

Daha sonra prapere kalıbını kullanacağız. Prepare kalıpları benzer SQL komutlarını tekrarlayarak çalıştırmamızı sağlayan bir yapıdır. Prepare kalıplarının kullanımı prepare() fonksiyonu ile başlar. Bu komut, tabloya veri eklemek için kullanılan INSERT INTO komutu için hazırlanan MySQL sorgusu içinde sadece sütun adlarını tanımlar, değerler alanında yer alan her bir sütun değeri için bir soru işareti koyar. Veritabanına gönderilen bu sorguya gerekli işlem yapılır ve sonuçlar kullanılmak üzere bekletilir. Daha sonra, sütun değerleri için tanımlanan değişkenlere atanan değerler execute() fonksiyonu ile, tabloya kaydedilir. Bu işlem değişkenlere yeni değerler atanarak istenilen sayıda tekrar edilebilir. İlk önce email için prepare ve execute kalıbını kullandık. Çünkü aynı email üzerinden 2. kez kayıt olmaması içindir. MySQL sorgusu “select \* from uyeler where email = ? LIMIT 1" bu şekildedir. Uyeler tablosundaki email leri getirecek. Daha sonra execute ederek formdan gelen email(Resim 2.6.1) aynısı uyeler tablosunda sorgu yapmış olacak. Daha sonra rowcount kalıbını kullandık. Bu yapı ise veritabanında uyeler tablosunda istenilen email’i bulduğunda sıfırdan büyük bir değeri gösterecektir ve “bu email kullanımda başka email kullanın” uyarısı döndürecektir. RowCount yapısından sıfır değer döndüğünde yazdığımız kod çalışacaktır.

Daha sonra tekrar prepare ve execute yapısını kullanarak formdaki( Resim 2.5.1) verileri veri tabanına kaydeder. Bu seferki mysql sorgısunda INSERT INTO kullanacağız. Bu sorgo ise "INSERT INTO uyeler (adi, soyadi, email, sifre) VALUES(?,?,?,?)" bu sekildedir.Bunu prepare edecez. Daha sonra formdan(Resim 2.6.1) gelen verileri execute yapısını kullanarak $Gelenadi,$Gelensoyadi,$Gelenemail, $Gelensifre execute nin içine yazarak uyeler tablosuna kaydetmiş oluruz. Kaydetmeden önce rowcount kullanarak eğer sıfırdan büyükse “kayıt başarılı ” diyecektir. Sıfırdan küçükse “hata oluştu tekrar deneyiniz” diyecektir.

**3.6. Kullanıcı Girişi:**

Php klasörünü içerisinde girisyap.php dosyasıdır.

Üye kaydı olduktan sonra kullanıcıların giriş yap sayfasına giderek form(Resim 2.6.1) üzerindeki kayıtlı olduğu email adresi ile şifresini girerek kullanıcı siteye girişini yapmış olur. Giriş yaptığında anasayfaya(index.php) yönlendirilecektir.

**girisyap.php**

<?php

include 'baglan.php';

if($\_POST){

    $email = $\_POST['email'];

    $sifre = $\_POST['sifre'];

    if($email != "" and $sifre != ""){

        $uyesorgusu = $DBConnection -> prepare("SELECT \*FROM uyeler WHERE email = ? and sifre = ? ");

        $uyesorgusu -> execute([$email, $sifre]);

        $uyesorgususayi = $uyesorgusu -> rowCount();

        if($uyesorgususayi > 0){

            $\_SESSION['uye']=$email;

        }else{

            echo '<div class="alert alert-danger">email adresi veya sifre yanlış!!!</div>';

        }

        if(isset($\_SESSION["uye"]) && $email == "admin@gmail.com" && $sifre == "admin") {

        echo "<script>window.open('../sayfa/admin.php','\_self')</script>";

        }else{

            if(isset($\_SESSION["uye"])){

                echo "<script>window.open('../index.php','\_self')</script>";

            }

        }

    }else{

        echo '<div class="alert alert-danger">Lütfen Boş Bırakmayınız</div>';

    }

}

?>

İlk önce veri tabanı bağlantısı için baglan.php eklememiz lazım. Giriş yap sayfasındaki(Resim 2.6.1) doldurulan formdaki verileri $\_POST ile alıp $email ve $sifre değişkenlerine atarız. Form üzerindeki veriler boş olup olmadığını kontrol etmek için koşul oluşturuyoruz. Eğer boş ise “”lütfen boş bırakmayınız” uyarısını döndürecektir. Boş değilse yazdığımız kodu çalışacaktır.

Daha sonra prepare, execute ve rowcount yapısını kullanacağız. Burada ise form üzerinde(Resim 2.6.1) girilen email ile şifrenin veri tabanındaki uyeler tablosunda email ile şifrenin aynı olup olmadığına bakılacaktır. Mysql sorgusu "SELECT \*FROM uyeler WHERE email = ? and sifre = ? " bu şekildedir. Formdaki(Resim 2.6.1) girilen veriler ile veritabanındaki uyeler tablosundaki veriler ile aynı ise rowcount tan dönen değer sıfırdan büyük olacaktır ve $\_SESSION['uye']=$email kullanarak bütün sayfalara bu email kullanıcı adresi ile açılacaktır. Eğer rowcount tan dönen değer sıfıra eşit ise “email adresi veya şifre yanlış” uyarısı dönecektir.

Daha sonra yeni bir koşul açarak email adresinin [admin@gmail.com](mailto:admin@gmail.com) ile sifre admin eşit ise admin sayfası açılacaktır. script>window.open('../sayfa/admin.php','\_self')</script>" javascript komutunu kullanarak admin sayfasına geçiş sağlayacaz.

Eşit değilsede kullanıcı girişi sağlanarak anasayfaya(index.php) sayfasına yönlendirilecektir. Onuda javascript "<script>window.open('../index.php','\_self')</script>" kodu ile geçiş sağlayacaz.

**3.7 Çıkış Yap:**

Php klasörünü içerisinde cıkısyap.php dosyasıdır.

Kullanıcı girişi veya admin girişi yapıldıktan sonra navbar kısmında çıkış yap butonu bulunmaktadır. Bu butonu tıklandığı zaman hesaptan çıkış yapılacaktır ve anasayfaya(index.php) yönlendirilecektir.

**cıkısyap.php:**

<?php

session\_start();

session\_destroy();

header("Location: ../index.php");

?>

$\_SESSION['uye'] yapısından çıkış yapmış olacak ve anasayfa(index.php) yölendirecektir.

**3.8.Sepete Ekleme**

Php klasörünü içerisinde sepetekle.php dosyasıdır.

Ürün sayfasındaki(Resim 2.7.1) sepete ekle butonuna basıldığında sepetekle dosyası çalışacaktır. Ürün bilgilerini alarak veri tabanındaki sepetim tablosuna ekleyecektir.

**sepetekle.php**

<?php

session\_start();

if(isset($\_SESSION['uye'])){

include 'baglan.php';

if($\_POST){

      $email = $\_SESSION['uye'];

      $urunismi = $\_POST['urunAdi'];

      $fiyat = $\_POST['urunFiyati'];

      $adet = $\_POST['urunAdet'];

            $ekle = $DBConnection -> prepare("INSERT INTO sepetim (email, urunismi, adet, fiyat) VALUES(?,?,?,?)");

            $ekle -> execute([$email,$urunismi,$adet,$fiyat]);

            $eklesayi = $ekle -> rowCount();

            if($eklesayi > 0) {

              echo "<script>window.open('../sayfa/sepetim.php','\_self')</script>";

            } else {

              echo 'hata';

            }

}

}

?>

SESSION['uye'] dediği eğer kullanıcı girişi yapılmışsa o kullanıcının email adresidir. if(isset($\_SESSION['uye'])) kodu ise eğer kullanıcı girişi sağlanmışsa koşuludur. Koşul sağlanmamışsa hata döndürecektir.

Ürün sayfasındaki ürün verileri $\_POST ile alıp $urunismi, $fiyat, $adet değişkenine değişkenlerine atarız. SESSION['uye'] ‘de $email değişkenine atarız.

Daha sonra prepare ve execute yapısını kullanarak bu değişkenleri veritabanındaki sepetim tablosuna kaydederiz. Mysql sorgusu “INSERT INTO sepetim (email, urunismi, adet, fiyat) VALUES(?,?,?,?)" bu şekildedir.

Daha sonra rowcount yapısı kullanarak bir koşul açarız. Eğer kaydedilmişse sıfırdan büyük bir sayı dönecektir ve "<script>window.open('../sayfa/sepetim.php','\_self')</script>" kodunu yazarak sepetim sayfasına geçiş yapmış olacağız. Kaydedilmemişse hata döndürecektir.

**3.9 Sepetim**

Php klasörünün içerisindeki sepet.php dosyasıdır.

sepet.php dosyası sepetim sayfasındaki(Resim 2.8.2) bilgiler veri tabanındaki sepetim tablosundaki kendisine ait verileri tablo halinde gelmesini sağlayan dosyadır.

**sepet.php**

<?php

include 'baglan.php';

$ara = $\_SESSION["uye"];

$sorgu = "select \* from sepetim where email like '%$ara%'";

$kayitlar = $DBConnection -> prepare($sorgu);

$kayitlar -> execute();

if($kayitlar->rowCount()){

        echo '<div class="table-responsive">';

        echo '<table class="table table-bordered border-primary">';

        echo '<thead>';

        echo '<tr>';

        echo '<th scope="col">Email</th>';

        echo '<th scope="col">Ürün Adı</th>';

        echo '<th scope="col">adet</th>';

        echo '<th scope="col">fiyat</th>';

        echo '<th scope="col">Sepete Ekleme Tarihi</th>';

        echo '</tr>';

        echo '</thead>';

        while($satir= $kayitlar->fetch(PDO::FETCH\_ASSOC)){

            echo '<tbody>';

            echo '<tr>';

            echo '<td>';

            echo '<input style="border:none transparent;outline:none;" name="email" value="'.$satir['email'].'">';

            echo '</td>';

            echo '<td>';

            echo '<input style="border:none transparent;outline:none;" name="urunismi" value="'.$satir['urunismi'].'">';

            echo '</td>';

            echo '<td>';

            echo '<input style="border:none transparent;outline:none;" name="adet" value="'.$satir['adet'].'">';

            echo '</td>';

            echo '<td>';

            echo '<input style="border:none transparent;outline:none;" name="fiyat" value="'.$satir['fiyat'].'">';

            echo '</td>';

            echo '<td>';

            echo '<input style="border:none transparent;outline:none;" name="tarih" value="'.$satir['tarih'].'">';

            echo '</td>';

            echo '</tr>';

            echo '</tbody>';

            }

            echo '</table>';

            echo '</div>';

            echo '<hr class="my-3">';

}

?>

$ara değişkenine kullanıcın email adresini $\_SESSION["uye"]; atarız. Bir sorgu yaparak kullanıcıya ait email’in aynısını veritabanında sepetim tablosundaki email sütununda aranarak aynı email leri getirmesidir. Sorgumuz "select \* from sepetim where email like '%$ara%'" bu şekildeidr. Daha sonra prepare ve execute yapısını kullandım. Daha sonra rowcount kullanarak kullanıcıya ait kayıt varsa sıfırdan büyük dönecektir ve table yapısı çalısacaktır.

Veri tabanından veri çekmek için fetch() yapısı vardır. Veri tabanındaki sepetim tablosundaki email sütununda kullanıcıya ait email in satırları gelmesi için fetch() kullandım. While içerisinde kullandım çünkü sepetim tablosunda email sütununda kullanıcıya ait emailin 1 den fazla olabilir. Tablodaki sütun isimlerine göre indisli bir dizi dönmesi için PDO::FETCH\_ASSOC yapısını kullandım.

Daha sonra satır halinde çektiğimiz verilerin .$satir['email']. kodu ile email adresini tabloya yazdık. Aynı şekilde ‘.$satir['urunismi'].’, ‘.$satir['urunismi'].’, ‘.$satir['urunismi'].’, ‘.$satir['tarih'].’ kodlarını kullanarak ürünismi, fiyat, adet ve tarihleri sepetim sayfasındaki(Resim 2.8.2) tabloya yazılacaktır.

**3.10.Sepet sil**

Php klasörünün içerisindeki sepetsil.php dosyasıdır.

Sepete eklediğimiz ürünleri sepetten çıkarmak içindir. Sepetim sayfasındaki(Resim 2.8.2) sil butonuna tıklandığı zaman bu dosya çalışacaktır.

**sepetsil.php**

<?php

session\_start();

include 'baglan.php';

if(isset($\_SESSION['uye'])){

    if(isset($\_POST['sil'])){

            $ara = $\_SESSION['uye'];

            $sorgu = "DELETE from sepetim where email like '%$ara%'";

            $kayitlar = $DBConnection -> prepare($sorgu);

            $kayitlar -> execute();

            echo "<script>window.open('../sayfa/sepetim.php','\_self')</script>";

    }

}

?>

Veri tabanı bağlantısı için baglan.php ekledik. $ara değişkenine kullanıcın email adresini $\_SESSION["uye"] atarız. Daha sonra mysql sorgusu ile "DELETE from sepetim where email like '%$ara%'", kullanıcı ait email adresine aynısını veritabanındaki sepetim tablosundan bularak sileriz. Prepare ve ececute yapısını kullanılır. Daha sonra sayfayı yenilemek için "<script>window.open('../sayfa/sepetim.php','\_self')</script>" javascript kodu çalıştırılır.

**3.11.Sipariş Ver**

Php klasörünün içerisindeki order.php dosyasıdır.

Sepetim sayfasında(Resim 2.8.2)ürünleri sipariş vermek için siparis ver butonuna tıklanarak karşımıza çıkan formu(Resim 2.9.1) doldurarak ödeme işlemleri için bilgileri girerek sipariş ver butonuna tıklandığı zaman order.php dosyası çalışacaktır. Bu dosya ise sepetim sayfasındaki ürünü silerek siparislerim sayfasına(Resim 2.10.1) eklenecektir. Ödeme işlemleri ödenmiş gibi gösterilmektedir.

**order.php**

<?php

session\_start();

include 'baglan.php';

if(isset($\_POST['ver'])){

        $ara = $\_SESSION['uye'];

        $bul = "select \* from uyeler where email like '%$ara%'";

        $adbul = $DBConnection -> prepare($bul);

        $adbul -> execute();

        $adbulsayi=$adbul->rowCount();

        if($adbulsayi>0){

            while($adlar= $adbul->fetch(PDO::FETCH\_ASSOC)){

                $ad = "".$adlar['adi']." ".$adlar['soyadi'].", ";

            }

        }

        $sorgu = "select \* from sepetim where email like '%$ara%'";

        $kayitlar = $DBConnection -> prepare($sorgu);

        $kayitlar -> execute();

        $toplamFiyat = 0;

        $kayitlarsayi=$kayitlar->rowCount();

        if($kayitlarsayi>0){

            $email = $ara;

            $urun = "";

            while($satir= $kayitlar->fetch(PDO::FETCH\_ASSOC)){

              $urun .= "".$satir['urunismi']." (".$satir['adet']."), ";

            }

            $toplamFiyat = $\_POST['toplamFiyat'];

            $adres = $\_POST['adres'];

            $kargoDurumu = "Kargoya verilmedi.";

            if(empty($\_POST['adres']) || empty($\_POST['hesapnumara']) || empty($\_POST['sonkullanmatarihiay']) || empty($\_POST['sonkullanmatarihiyil']) || empty($\_POST['cvc2'])) {

   echo "<script>window.open('../sayfa/sepetim.php','\_self')</script>";

            }

            else {

                $sorguu = $DBConnection -> prepare("INSERT INTO siparisler (adsoyad, email, urun, toplamfiyat, adres, kargodurumu) VALUES(?,?,?,?,?,?)");

                $sorguu -> execute([$ad,$email,$urun,$toplamFiyat,$adres,$kargoDurumu]);

                $sorgusayii = $sorguu -> rowCount();

        if($sorgusayii>0){

           $sepetbul = "DELETE from sepetim where email like '%$ara%'";

                    $sepetsil = $DBConnection -> prepare($sepetbul);

                    $sepetsil -> execute();

echo "<script>window.open('../sayfa/siparislerim.php','\_self')</script>";               }           }

        }

}

?>

İlk sorgumuz "select \* from uyeler where email like '%$ara%' sorgusudur. Burada kullanıcı ait email adresi ile veritabanındaki uyeler tablosundan bu emaile ait satırı bulacağız. Daha sonra prepare ve execute kalıbını kullanacağız. Bunu yapmamızın amacı Kullanıcının adını ve soyadını bulmaktır. Daha sonra rowcount kalıbını kullanarak sorguda kullanıcı emaile ait bilgi varsa sıfırdan büyük bir değer dönecektir vedöngü içinde fetch yapısını kullanarak verileri çekeceğiz. Daha sonra $ad değişkenine adi ve soyadını atayacağız. bunu ise $ad = "".$adlar['adi']." ".$adlar['soyadi'].", " kodu ile yapılacaktır. İkinci sorgumuz ise select \* from sepetim where email like '%$ara%'" sorgusudur. Burada ise kullanıcı emaile ait sepetim tablosundaki veriler gelecektir. Prepare ve execute kalıbını kullandıktan sonra rowcount kalıbını kullanarak istenilen satır bulunmuşsa dönen değer sıfırdan büyük olacaktır ve koşula girmiş olacağız. Fetch kalıbını kullanarak verileri çekerek ürünün ismi ve adetini $urun değişkenine atayacağız. Formda girilen( Resim 2.9.1) adres ve toplam fiyatı $adres ve $toplamfiyat değişkenlerine atayacağız. Kargo durumuna değişkenine “kargoya verilmedi” atayacağız. Daha sonra yeni bir sorgu "INSERT INTO siparisler (adsoyad, email, urun, toplamfiyat, adres, kargodurumu) VALUES(?,?,?,?,?,?)") ile prapere ve execute yapısı ile değişkenleri siparişler tablosuna kaydedilecektir. Daha sonra gene bir sorgu "DELETE from sepetim where email like '%$ara%'" ile prepare ve execute yapısı kullanarak veritabanında sepetim tablosundaki kullanıcıya ait ürün verileri silinecektir.

**3.12.Toplam Fiyat**

Php klasörünü içerisinde fiyatopla.php dosyasıdır.

Sepete eklenen ürünlerin toplam fiyatını bulan dosyadır.

**fiyatopla.php**

<?php

include 'baglan.php';

$ara = $\_SESSION["uye"];

$sorgu = "select \* from sepetim where email like '%$ara%'";

$kayitlar = $DBConnection -> prepare($sorgu);

$kayitlar -> execute();

$toplamFiyat = 0;

if($kayitlar->rowCount()){

    while($satir= $kayitlar->fetch(PDO::FETCH\_ASSOC)){

            $urunFiyati = $satir['fiyat'];

            $urunFiyati = floatval($urunFiyati);

            $urunFiyati = $urunFiyati \* $satir['adet'];

            $toplamFiyat += $urunFiyati;

    }

}

?>

$ara değişkenine $\_SESSION["uye"] diyerek kullanıcı email adresini atıyoruz. Veri tabanı sorgumuz "select \* from sepetim where email like '%$ara%'" bu şekildedir. Veri tabanındaki sepetim tablosundan emaile ait verileri sorguluyoruz. Daha sonra prepare ve execute yapılarını kullanıyoruz.

Daha sonra verileri çekmek için döngü içinde fetch işlemi yapıyoruz. Sepetim tablosunda kullanıcıya ait ürünlerin fiyatlarını topluyoruz.

**3.13.Siparişler**

Php klasörünün içerisinde siparislerimphp.php dosyasıdır.

Sepetim sayfasındaki(Resim 2.8.2) gibi veri tabanındaki siparisler tablosundaki kendisine ait verileri tablo halinde gelmesini sağlayan dosyadır. Kullanıcı ürünü sipariş verdiği zaman siparisler sayfasına “kargoya verilmedi” olarak kaydedilir(Resim 2.10.1). Admin ürünleri kargoya verdiği zaman “kargoya verildi” olarak değişecektir(Resim 2.10.2).

**Siparislerimphp.php**

<?php

include 'baglan.php';

$ara = $\_SESSION["uye"];

$sorgu = "select \* from siparisler where email like '%$ara%'";

$kayitlar = $DBConnection -> prepare($sorgu);

$kayitlar -> execute();

if($kayitlar->rowCount()){

        echo '<div class="table-responsive">';

        echo '<table class="table table-bordered border-primary">';

        echo '<thead>';

        echo '<tr>';

        echo '<th scope="col">Ürün Adı</th>';

        echo '<th scope="col">Kargo Durumu</th>';

        echo '<th scope="col">Kargo Takip No</th>';

        echo '</tr>';

        echo '</thead>';

        while($satir= $kayitlar->fetch(PDO::FETCH\_ASSOC)){

            echo '<tbody>';

            echo '<tr>';

            echo '<td>';

            echo '<input style="border:none transparent;outline:none;" name="urunismi" value="'.$satir['urun'].'">';

            echo '</td>';

              if($satir['kargodurumu'] == "Kargoya verilmedi.") {

                echo '<td>';

                echo '<input type="button" class="btn btn-danger" style="border:none transparent;outline:none;cursor:default;" name="adet" value="'.$satir['kargodurumu'].'">';

                echo '</td>';

                echo '<td>';

                echo '<input style="border:none transparent;outline:none;" name="takipno" value="'.$satir['takipno'].'">';

                echo '</td>';

            } else {

                echo '<td>';

                echo '<input type="button" class="btn btn-success" style="border:none transparent;outline:none;cursor:default;" name="adet" value="'.$satir['kargodurumu'].'">';

                echo '</td>';

                echo '<td>';

                echo '<input style="border:none transparent;outline:none;" name="takipno" value="'.$satir['takipno'].'">';

                echo '</td>';

            }

            echo '</tr>';

            echo '</tbody>';

            }

            echo '</table>';

            echo '</div>';

            echo '<hr class="my-3">';

}

?>

$ara değişkenine kullanıcın email adresini $\_SESSION["uye"]; atarız. Bir sorgu yaparak kullanıcıya ait email’in aynısını veritabanında siparisler tablosundaki email sütununda aranarak aynı email leri getirmesidir. Sorgumuz "select \* from siparisler where email like '%$ara%'" bu şekildedir. Daha sonra prepare ve execute yapısını kullandım. Daha sonra rowcount kullanarak kullanıcıya ait kayıt varsa sıfırdan büyük dönecektir ve table yapısı çalısacaktır. Veri tabanından veri çekmek için fetch() yapısı vardır. Veri tabanındaki sepetim tablosundaki email sütununda kullanıcıya ait email in satırları gelmesi için fetch() kullandım. While içerisinde kullandım çünkü siparisler tablosunda email sütununda kullanıcıya ait emailin 1 den fazla olabilir. Tablodaki sütun isimlerine göre indisli bir dizi dönmesi için PDO::FETCH\_ASSOC yapısını kullandım.

Eğer veri tabanındaki siparisler tablosunda kargodurumu “kargoya verilmedi” şeklindeyse Resim 1.10.1 deki gibi olacaktır. Kargodurumu “kargoya verildi” şeklindeyse Resim 2.10.2 deki gibi olacaktır.

Daha sonra satır halinde çektiğimiz verilerin ‘.$satir['urun'].’, ‘.$satir['kargodurumu'].’, ‘.$satir['takipno'].’ kodlarını kullanarak ürünler, kargodurumu, kargo takip no siparisler sayfasındaki(Resim 2.10.1) tabloya yazılacaktır.

**3.14.Admin**

Admin sayfasını oluştururken Veri tabanından siparisler tablosundaki bütün verileri bir tablo halinde admin sayfasına(Resim 2.11.1) gösterilmektedir. Her bir satırda kargoya ver butonu bulunmaktadır.

Kod kısmı ise sepetim(3.9) ve siparislerdeki(3.13) kod kısmı ile hemen hemen aynıdır. Farklı olan kod kısmı ise aşağıdadır ve kargoya ver butonuna basıldığında yeni bir form oluşması(Resim 2.11.2) ve bu forma verileri girdikten sonra tekrar kargoya ver butonuna tıklanmasıdır.

**Kod:**

      echo '<td>';

            echo '<button class="btn btn-primary float-left" data-toggle="modal" data-target="#siparisModal">Kargoya Ver</button>

<div class="modal fade" id="siparisModal" tabindex="-1" role= "dialog" aria-labelledby="siparisModalLabel" aria-hidden="true">

        <div class="modal-dialog" role="document">

        <div class="modal-content">

          <div class="modal-body">

           <form action="../php/kargo.php" method="POST">

            <div class="form-group">

<label for="kargotakip" class="col-form-label"> Kargo Takip No Olustur</label>

             <input type="text" name="kargotakip" class="form-control">

            </div>

            <div class="form-group">

       <label for="adres" class="col-form-label">Alıcı Email</label>

          <input type="name" name="email" class="form-control">

            </div>

            <hr class="my-5">

            <div class="form-group">

             <button type="submit" class="btn btn-primary" id= "kargover" name="kargover">kargoya Ver</button>

            </div></form></div> </div></div></div>';

            echo '</td>';

Resim 2.11.2 deki formdur. Burada kargo takip numarası ile alıcı email girdikten sonra kargoya ver dediğimizde php klasörünün içindeki kargo.php dosyası çalısacaktır.

**3.15.Kargoya Ver**

php klasörünün içindeki kargo.php dosyasıdır.

Admin tarafından kargoya ver dediğnde bu dosya çalısacaktır. Burada ise veritabanındaki siparisler tablosundaki kargo durumunu ve takipno sütunlarını günceller.

**kargo.php**

<?php

include 'baglan.php';

if(isset($\_POST['kargover'])){

$durum="Kargoya verildi.";

$takipno=$\_POST['kargotakip'];

$email = $\_POST['email'];

$sorgu = "select \* from siparisler where email like '%$email%'";

$kayitlar = $DBConnection -> prepare($sorgu);

$kayitlar -> execute();

$kayitlarsayi=$kayitlar->rowCount();

      if($kayitlarsayi>0){

            while($satir= $kayitlar->fetch(PDO::FETCH\_ASSOC)){

                 if($satir['kargodurumu'] == "Kargoya verilmedi."){

                    if($satir['takipno'] == 0){

                    $sorguu = "UPDATE siparisler set kargodurumu='" .$durum. "' WHERE email='" .$email. "'";

                    $kayitlarr = $DBConnection -> prepare($sorguu);

                    $kayitlarr -> execute([$durum,$email]);

                    $sorguuu = "UPDATE siparisler set takipno='" .$takipno. "' WHERE email='" .$email. "'";

                    $kayitlarrr = $DBConnection -> prepare($sorguuu);

                    $kayitlarrr -> execute([$takipno,$email]);

                                                                               echo "<script>window.open('../sayfa/admin.php','\_self')</script>";

                    }

                }

            }

        }else {

            echo "<script>window.open('../sayfa/admin.php','\_self')</script>";

     }

}

?>

Resim 2.11.2 formdaki kargotaki no ile alıcı email verilerini değişkenlere atıyoruz. Mysql sorgusu ile istenilen satırı buluyoruz. Prepare, execute ve rowcount yapılarını kullanıyoruz. Verileri çekmek için fetch yapısını kullanıyoruz.

Koşul içerisinde kargodurumu “kargoya verildmedi” durumlarında "UPDATE siparisler set kargodurumu='" .$durum. "' WHERE email='" .$email. "'"; sorgusu yaparak ve prepare ve execute yapısı kullanarak kargo durumunu “kargoya verildi” olarak güncelliyoruz. Aynı şekilde takipno’yuda güncelliyoruz. Daha sonra "<script> window.open ('../sayfa/admin.php','\_self') </script>" javascript kodu ile sayfayı yeniliyoruz.

**3.16.Hesap Bilgileri**

Php klasörünün içerisindeki hesapbilgisicek.php dosyasıdır.

Hesap bilgileri sayfasındaki(Resim 2.12.1) bilgileri veritabanındaki uyeler tablosundan verileri çekmektedir.

**hesapbilgisicek.php**

<?php

include 'baglan.php';

$ara = $\_SESSION["uye"];

$sorgu = "select \* from uyeler where email like '%$ara%'";

$kayitlar = $DBConnection -> prepare($sorgu);

$kayitlar -> execute();

if($kayitlar->rowCount()){

    while($satir= $kayitlar->fetch(PDO::FETCH\_ASSOC)){

                    $adi = $satir['adi'];

                    $soyadi= $satir['soyadi'];

                    $email = $satir['email'];

                    $sifre =$satir['sifre'];

                    $tarih =$satir['tarih'];

                }

}

?>

Bu kodda $ara değişkenine $\_SESSION["uye"] ile kullanıcı emailini atadık. Veri tabaındaki uyeler tablosundan sorgu yaparak prepare ve execute yapısını kullandım. Mysql sorgusu "select \* from uyeler where email like '%$ara%'"; bu şekildedir. Daha sonra rowcount yapısından dönen değer sıfırdan büyükse döngü içinde fetch yapısını kullanarak verileri çektim. Bu verileri $ad, $soyad, $email, $sifre, $tarih değişkenlerine atatım.

Hesap bilgileri sayfasında(Resim 2.12.1) bu verileri gösterebilmek için hesapbilgileri kodunda <?php echo $adi;?>, <?php echo $soyadi;?>, <?php echo $email;?>, <?php echo $sifre;?>, <?php echo $tarih;?> kodlarını yazdım.

**3.17. Şifre Değiştirme**

Php klasörünün içerisindeki sifrephp.php dosyasıdır.

Şifre değiştir sayfasında(Resim 2.13.1) şifre değiştir butonuna basıldığı zaman bu kod çalışacaktır.

**sifrephp.php**

<?php

include 'baglan.php';

if($\_POST){

    $sifre = $\_POST["sifre"];

    $yenisifre = $\_POST["yenisifre"];

    $yenisifretekrar = $\_POST["yenisifretekrar"];

    $ara = $\_SESSION["uye"];

    $sorgu = "select \* from uyeler where email like '%$ara%'";

    $kayitlar = $DBConnection -> prepare($sorgu);

    $kayitlar -> execute();

    if($kayitlar->rowCount()){

            $satir= $kayitlar->fetch(PDO::FETCH\_ASSOC);

            if ($sifre== $satir["sifre"]){

                if($yenisifre !== $yenisifretekrar){

                    echo '<div class="alert alert-danger">Yeni şifreniz birbirinden farklı. Yeniden deneyiniz!!!</div>';

                }else{

                    $sorguu = "UPDATE uyeler set sifre='" .$yenisifre. "' WHERE email='" .$ara. "'";

                    $kayitlarr = $DBConnection -> prepare($sorguu);

                    $kayitlarr -> execute([$yenisifre,$ara]);

                    echo '<div class="alert alert-success">Sifre Değiştirildi</div>';

                }

            }else{

                echo '<div class="alert alert-danger">Yanliş Şifre Girdiniz, Lütfen Tekrar Deneyininiz!!!</div>';

            }

        }

Bu kodda 2 koşul vardır. İlk koşul mevcut şifre yanlış girildiğinde “Yanlış şifre Girdiniz, Lütfen tekrar deneyiniz” uyarısı gönderecektir. 2. Koşul ise yeni şifre ile yeni şifre tekrarı birbirinin aynısı değilse “şifreler uyuşmuyor” uyarası gönderecektir.

Koşullar doğru ise veritabanındaki uyeler tablosunda sifre güncellenecektir. Mysql sorgusu yaparak prepare ve execute yapısını kullanarak güncelleme yapılır. Sorgu = "UPDATE uyeler set sifre='" .$yenisifre. "' WHERE email='" .$ara. "'"; şeklindedir. $ara değişkenine kullanıcı emailini atamıştık. Bu şekilde şifre değiştirilmiştir.

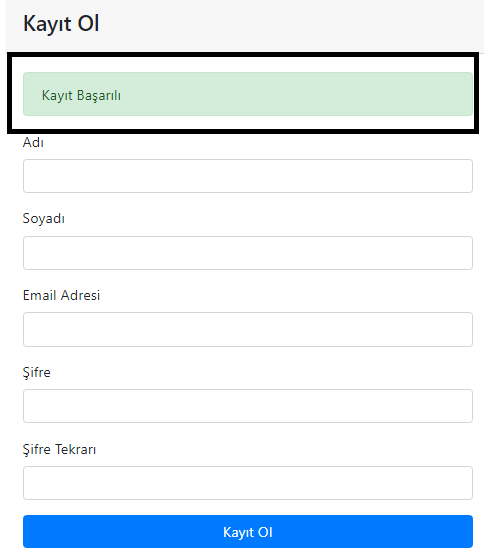
**4.SİTENİN KULLANIMI HAKKINDA BİLGİLER**

Siteye ilk girildiğinde anasayfa görünecektir. Sitemizden alış veriş yapılabilmesi için siteye üye olma koşulu vardır. Üye olabilmek için anasayfanın navbar kısmında sağ tarafta kayıt ol butonuna basılmalıdır.



(Resim 4.1)

Sonra karşımıza çıkan sayfada formdaki bilgileri eksiksiz bir şekilde girilmesi lazımdır. Eğer kayıt başaralı dediyse üye kaydı gerçekleşmiş olur.

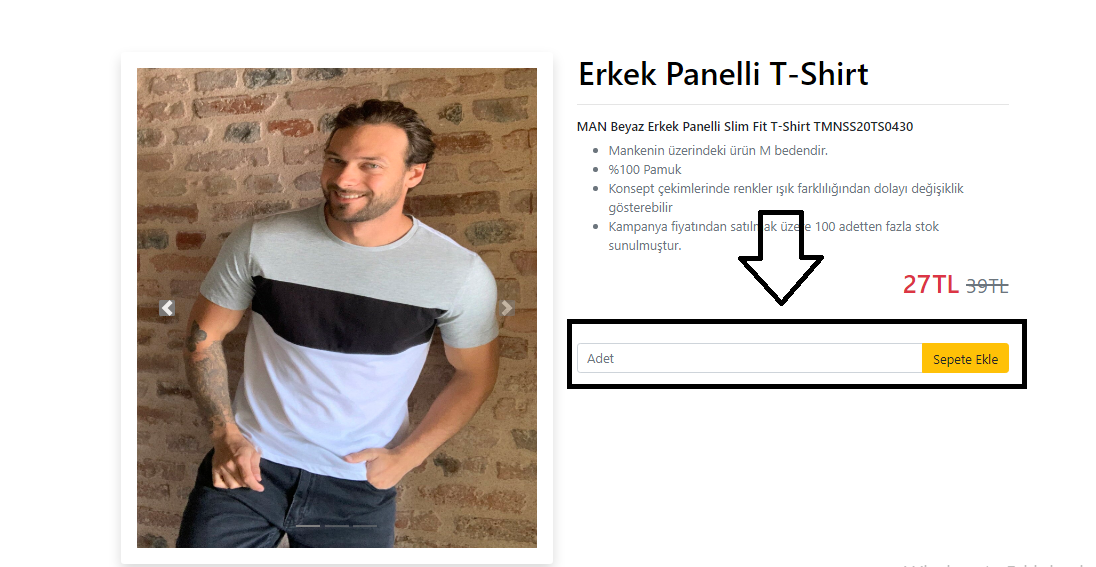


(Resim 4.2)

Şimdi giriş yap butonuna tıklayarak sisteme giriş yapılması gerek. Giriş yap sayfasında bilgileri doldurduğunda kullanıcı girişi olarak anasayfaya giriş sağlamış olursunuz(Resim 2.6.1).

Beğendiğiniz ürünleri satın al butonuna tıklayarak ürün sayfasına gitmiş olursunuz.

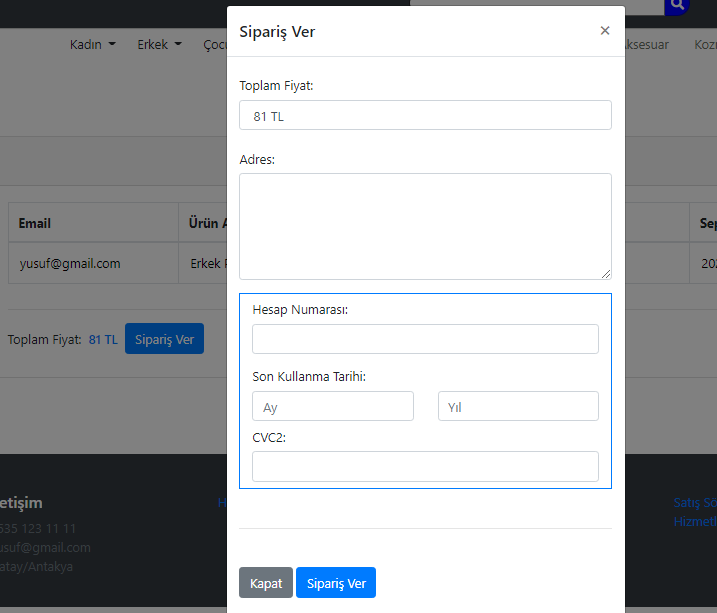
Ürün sayfasında ürün hakkında bilgileri okuyarak bu üründen kaç adet almak için adet kısmına yazmamız gerekmektedir ve sepete ekleme butonu bulunmaktadır.

(Resim 4.3)

Sepete ekleme butonuna tıkladığımız zaman ürün sepete eklenecektir ve sepetim sayfasına yönlendirilecektir.

Sepetim sayfasında ürünleri satın almak istediğinizde sepetim sayfasındaki satın al butonuna tıklanacaktır daha sonra karşımıza çıkan form’u eksiksiz bir şekilde doldurmamız gerecektir.

(Resim 4.4)



Bu formu doldurduktan sonra sipariş ver dediğimizde ürününüz sipariş verilmiş olacaktır.

Ürününüzün kargoya verilip verilmediğini veya kargo takip numarası için navbar kısmında hesabım butonundan siparişler butonuna tıkladığımız zaman sipariş ile ilgili bilgileri gözükmektedir.

(Resim 4.5)



**5.SONUÇ**

Bu projede, projenin aşamaları boyunca yapılan çalışmalar açıklanmıştır. Staj süresi içerisinde yaptığım çalışma sayesinde HTML, CSS, PHP, dillerini ve Bootstrap kütüphanesini kullanma becerilerim arttı. Oluşturduğum siteyi dinamik olacak şekilde tasarlayıp kullanıcının ara yüz ile veri girişi yapması sağladım. Siteyi Türkçe dilinde hazırladım ve para birimi TL olarak ayarladım. Ödeme işlemini farklı bir sayfada sadece görüntüden ibaret şekilde yaptım çünkü gerçek bir ödeme sağlamam olanaksızdı. Search kısmında oldukça zorlandım ve bir sorunla karşılaştım ancak sorunumu çözmeyi başaramadım. Bu yüzdende projeme eklemedim.

Sonuç olarak verimli bir çalışma oldu. Staj süresi boyunca bana yardımcı olan Profesör Doktor Hamza EROL hocama teşekkür ederim.

**6. Kaynak Kodlar**

Kaynak kodları Google drive üzerinden paylaştım. Google drive linki aşağıdadır.

Link: <https://drive.google.com/drive/folders/1cNWsAuulYiUZ_CB4Fwzu8-eDcygl7vZn?usp=sharing>

Projemi video şeklinde anlattım. Bu video’yu Google drive üzerinden paylaştım. Google drive aşağıdadır.

Link: <https://drive.google.com/file/d/1l5KXPi0AzCkgSWEe_MXCgq-hdOG_E7VZ/view?usp=sharing>