

**ANKARA ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BLM 4062-N PROJE RAPORU**

**HerŞey Bulunur E-Ticaret Sitesi**

**Oğuzhan Sezer**

**19290786**

**Ayhan Aydın**

**06.06.2023**

## ÖZET

Bu çalışmada kullanıcılara Online alışverişlerde rahat bir kullanım için kolaylık sağlamak konusunda yardımcı olacak ve ürünlerin müşterilere karşı tanıtımını yapmasını sağlayarak müşterilerin alışverişlerini kısa sürede tamamlayarak güvenilir bir ödeme sistemi sağlayacak bir E-Ticaret sitesi anlatılmaktadır.

Projenin adı HerşeyBulunur.com web sitesidir. Projenin kullanımı herkese açıktır.

HerşeyBurada.com sitesi Asp.Net Core5(MVC) kullanılarak C# Programlama dili ile geliştirilmiştir.

HerşeyBulunur.com sitesi Visual Studio 2022 IDE 'si kullanılarak geliştirilmiştir.

Veri tabanı olarak Microsoft Sql Server Management Studio kullanılmıştır

Projede temel amaç kullanıcılara almak istedikleri ürünü daha etkin daha hızlı bir şekilde bulmalarını sağlamak ve ürünleri kolay ve güvenilir bir şekilde sipariş edilmesini sağlamaktır. Diğer yandan ürünlerin çeşitliliği kullanıcılar için birçok fikir sahibi olmalarını sağlamaktadır. Bu sayede ürünler müşterilere daha çok seçenek sunabilmektedir.

## İÇİNDEKİLER

ÖZET .....	2
İÇİNDEKİLER .....	3
1. GİRİŞ .....	4
2. HerŞeyBulunur.COM E-Ticaret Sitesi .....	5
2.1. <u>Proje Açılması ve Projenin Veri Tabanının Oluşturulması</u> .....	5
2.2. <u>Kategori Modelinin Oluşturulması</u> .....	7
2.3. <u>Ürün Modelinin Oluşturulması</u> .....	10
2.4. <u>Ürünler Sayfası İçin Ekleme,Düzenleme,Silme Sayfalarının Oluşturulması</u>	12
2.5. <u>Kullanıcı Sayfasının Oluşturulması</u> .....	15
2.6. <u>Roller Sayfasının Oluşturulması</u> .....	16
2.7 <u>Sepet İşlemleri İçin Kullanılacak Olan Sayfanın Oluşturulması</u> .....	17
2.8. <u>Ürün Detay Sayfasının Oluşturulması</u> .....	19
2.9. <u>Sepet Sayfasının Oluşturulması</u> .....	21
2.10. <u>Sipariş Verme Sayfası</u> .....	23
2.11. <u>Verilen Siparişleri Görüntüleme ve Listeleme</u> .....	26
2.12. <u>Kategori Listesi Görüntüleme</u> .....	28
3. SONUÇ .....	29
4.KAYNAKLAR .....	30

Bu raporda kullanıcıların online alışveriş yaparken kendilerini rahat ve güvende hissetmeleri için oluşturulan E-ticaret sitesinin geliştirilme aşamaları anlatılmaktadır.

Teknolojinin gelişmesiyle birlikte online alışverişler yaygınlaşmaya başlamış ve zamanla büyük bir gelişme göstermiştir. Özellikle MVC ile geliştirilen sitelerin sayısı oldukça fazladır.

E-ticaret sitesi geliştirmek için MVC yapısı model-view-controller oluşturarak geliştiricilere büyük kolaylıklar sağlamaktadır. Dışarıdan entegre edilen kodlar kolaylıkla bu yapıya uyum sağlamaktadır.

HerŞeyBulunur.com sitesi Visual Studio 2022 kullanılarak C# programlama dili ile geliştirilmiştir.

Proje temelde 6 sayfadan oluşmaktadır. Bunlar; Anasayfa Sayfası, Ürünler Sayfası, Kategori Sayfası, Kullanıcılar Sayfası, Siparişler Sayfası, Admin işlemleri sayfasıdır.

İlerleyen bölümlerde proje geliştirme detaylarından adım adım bahsedilecektir.

## 2. TAMİRCİM.COM FLUTTER UYGULAMASI

### 2.1.Projenin açılması ve Proje Veritabanının Oluşturulması

Projeyi geliştirmek için Visual Studio IDE'si kullanılmıştır.Projeyi açmak için Visual Studio 2022 açılmıştır ve New Project butonuna tıklanarak aşağıdaki işlemler gerçekleştirilmiştir:

The screenshot shows the 'New Project' dialog in Visual Studio 2022. The 'Project name' is 'Eticaret', the 'Location' is 'E:\Eticaret\'. The 'Solution name' is 'Eticaret'. The 'Framework' is '.NET 5.0 (Out of support)'. The 'Authentication type' is 'Individual Accounts'. The 'Configure for HTTPS' checkbox is checked. The 'Docker OS' is 'Linux'. The 'Enable Razor runtime compilation' checkbox is unchecked.

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Windows Cloud Service Web

Project name

Eticaret

Location

E:\Eticaret\

Solution name ⓘ

Eticaret

☐ Place solution and project in the same directory

Additional information

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Windows Cloud Service Web

Framework ⓘ

.NET 5.0 (Out of support)

Authentication type ⓘ

Individual Accounts

☒ Configure for HTTPS ⓘ

☐ Enable Docker ⓘ

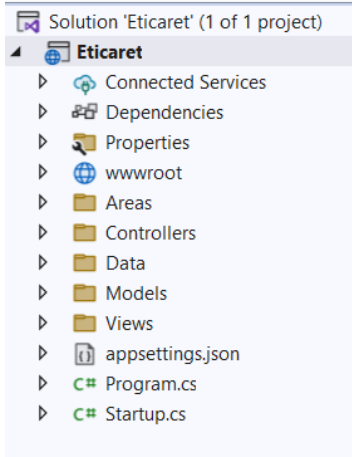
Docker OS ⓘ

Linux

☐ Enable Razor runtime compilation ⓘ

Şekil 2.1.1 Proje Açılması

Proje ilk açıldığında aşağıdaki gibi görülmektedir.

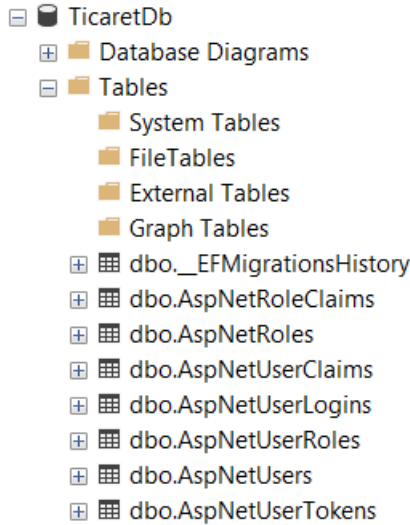


Projede veri tabanı olarak Microsoft SQL Management Studio kullanılmıştır.

Veritabanı oluşturmak için appsettings.json dosyasına kendi veritabanı adımız kopyalanmıştır.



Sonrasında kendi Migrations dosyamızı oluşturmak için üst menüden Tools->NuGet Package Manager->Package Manager Console ile gerekli adımlar izlenerek MSSQL üzerinden bir Migration oluşturulur. Artık veritabanımızda TicaretDb isimli bir veritabanı oluşmuştur. Görüntüsü aşağıdaki gibidir.



## 2.2. Kategori Modelinin Oluşturulması


Admin tarafından Kategori işlemlerini yapılabilmesi için bir Kategori Modeli oluşturulur. Bu Model bir ID ve Name içermektedir.

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel.DataAnnotations;
4 using System.Linq;
5 using System.Threading.Tasks;
6
7 namespace OrnekSite.Models
8 {
9     public class Category
10    {
11        [Key]
12        public int Id { get; set; }
13        [Required]
14        public string Name { get; set; }
15    }
16 }
17
```

Şekil 2.2.1 Kategoriler Sayfasının Oluşturulması

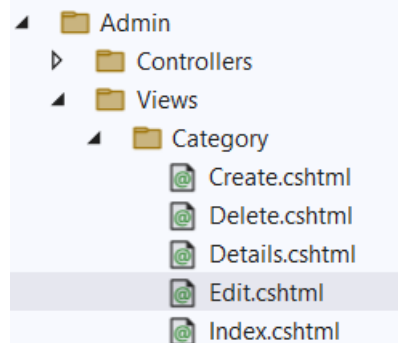
Bu adımdan sonra ApplicationDbContext.cs sayfasına veritabanında kullanmak için Categories isimli bir tablo eklenir. Gerekli migration işlemleri yapıldıktan sonra veritabanında güncelleme yapılır ve tablonun eklendiği gözlemlenir.

```
10 references
public DbSet<Category> Categories { get; set; }
12 references
```

 **dbo.Categories**

Şekil 2.2.2 Kategorilerin Tablosunun Veritabanına Eklenmesi

Daha sonra Kategori sayfasının görünümünün kodları oluşturuldu. Kategori adının girileceği Ekleme, Düzenleme ve Silme butonları da oluşturuldu. Ardından bir DataTable eklenerek Arama yapma kısmı oluşturuldu. Bu sayfadaki veriler Türkçe dilinde desteklenmektedir.



Şekil 2.2.3 Kategori Sayfalarının Genel Görünümü





```

1 @model OrnekSite.Models.Category
2
3 @{
4     ViewData["Title"] = "Edit";
5     Layout = "~/Views/Shared/_Layout.cshtml";
6 }
7
8
9 <div class="row">
10     <div class="col-md-4">
11         <form asp-action="Edit">
12             <div asp-validation-summary="ModelOnly" class="text-danger"></div>
13             <input type="hidden" asp-for="Id" />
14             <div class="form-group">
15                 Kategori Adı
16                 <input asp-for="Name" class="form-control" />
17                 <span asp-validation-for="Name" class="text-danger"></span>
18             </div>
19             <div class="form-group">
20                 <input type="submit" value="Güncelle" class="btn btn-success" />
21                 <a asp-action="Index" class="btn btn-primary">Kayıt Listesi</a>
22             </div>
23         </form>
24     </div>
25 </div>
26
27 <div>
28
29 </div>
30
31 @section Scripts {
32     @await Html.RenderPartialAsync("_ValidationScriptsPartial");
33 }
34

```

Şekil 2.2.4 Kategori Düzenleme Sayfası İçin Yazılan Kodlar

## Bu Kaydı Silmek İstiyor Musunuz?

Kategori Tablet

Adı

Sil

Kayıt Listesi

```

1 @model OrnekSite.Models.Category
2
3 @{
4     ViewData["Title"] = "Delete";
5     Layout = "~/Views/Shared/_Layout.cshtml";
6 }
7
8 <div class="container">
9     <div class="row">
10         <div class="col-md-4">
11
12         </div>
13         <div class="col-md-8">
14             <h3>Bu Kaydı Silmek İstiyor Musunuz? </h3>
15             <div>
16
17                 <dl class="row">
18                     <dt class="col-sm-2">
19                         Kategori Adı
20                     </dt>
21                     <dd class="col-sm-10">
22                         @Html.DisplayFor(model => model.Name)
23                     </dd>
24                 </dl>
25
26                 <form asp-action="Delete">
27                     <input type="hidden" asp-for="Id" />
28                     <input type="submit" value="Sil" class="btn btn-outline-danger" />
29                     <a asp-action="Index" class="btn btn-info">Kayıt Listesi</a>
30                 </form>
31             </div>
32         </div>
33     </div>
34 </div>

```

Şekil 2.2.5 Kategori Silme Sayfası İçin Yazılan Kodlar

## 2.3. Ürün Modelinin Oluşturulması

Models klasörü altında Product isimli bir sayfa oluşturulur. Product sayfasında ürünlerin sergilenebilmesi için gerekli özellikler yer alır. Oluşturulan Product.cs sayfasının özellikleri aşağıdaki gibidir.

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel.DataAnnotations;
4 using System.ComponentModel.DataAnnotations.Schema;
5 using System.Linq;
6 using System.Threading.Tasks;
7
8 namespace OrnekSite.Models
9 {
10     public class Product
11     {
12         [Key]
13         public int Id { get; set; }
14         [Required]
15         public string Title { get; set; }
16         [Required]
17         public string Description { get; set; }
18         public double Price { get; set; }
19         public string Image { get; set; }
20         public bool IsHome { get; set; }
21         public bool IsStock { get; set; }
22         public int CategoryId { get; set; }
23         [ForeignKey("CategoryId")]
24         public Category Category { get; set; }
25     }
26 }
27
28
```

Şekil 2.3.1 Product.cs Sayfası İçin Yazılan Kodlar

Product sayfasının Index sayfası ürünlerin ekranda görüntülemek için ve Ürün Ekleme, Düzenleme, Silme işlemlerinin yapılacağı butonları içermektedir.


```
1 @model IEnumerable<OrnekSite.Models.Product>
2
3 @{
4     ViewData["Title"] = "Index";
5     Layout = "~/Views/Shared/_Layout.cshtml";
6 }
7
8 <div class="container">
9     <p>
10         <a asp-action="Create" class="btn btn-primary form-control">Ürün Ekle</a>
11     </p>
12     <table class="table table-striped table-bordered" id="myTable">
13         <thead>
14             <tr>
15                 <th>Resim</th>
16                 <th>Başlık</th>
17                 <th>Anasayfa</th>
18                 <th>Stok</th>
19                 <th>Kategori</th>
20             </tr>
21         </thead>
22         <tbody>
23             @foreach (var item in Model)
24             {
25                 <tr>
26                     <td>
27                         @if (item.Image == null)
28                         {
29                             
30                         }
31                         else
32                         {
33                             
34                         }
35                     </td>
36                 </tr>
37             }
38         </tbody>
39     </table>
40 </div>
```

```

47     }
48     </td>
49     <td>
50         @Html.DisplayFor(modelItem => item.Title)
51     </td>
52
53     <td>
54         @if (item.IsHome)
55         {
56             <label class="btn btn-success btn-sm"><i class="bi bi-check-lg"></i>Evet</label>
57         }
58         else
59         {
60             <label class="btn btn-danger btn-sm"><i class="bi bi-x-lg"></i>Hayır</label>
61         }
62     </td>
63
64     <td>
65         @if (item.IsStock)
66         {
67             <label class="btn btn-success btn-sm"><i class="bi bi-check-lg"></i>Var</label>
68         }
69         else
70         {
71             <label class="btn btn-danger btn-sm"><i class="bi bi-x-lg"></i>Yok</label>
72         }
73     </td>
74     <td>
75         @Html.DisplayFor(modelItem => item.Category.Name)
76     </td>
77     <td>
78         <a asp-action="Edit" asp-route-id="@item.Id" class="btn btn-success btn-sm"><i class="bi bi-pencil"></i></a>
79         <a asp-action="Delete" asp-route-id="@item.Id" class="btn btn-danger btn-sm"><i class="bi bi-trash"></i></a>
80     </td>
81 </tr>
82 </tbody>
83 </table>
84 </div>
85
86 @section Scripts{
87     <script type="text/javascript">
88     $(document).ready(function () {
89         $('#myTable').DataTable({
90
91             "language": {
92                 "url": "https://cdn.datatables.net/plug-ins/9dcbecd42ad/i18n/Turkish.json"
93             }
94         });
95     });
96 </script>

```

Şekil 2.3.2 Product Index Sayfasının Kodlarının Görünümü

Ürün Ekle					
Sayfada 10 Kayıt Göster		Bul: <input type="text"/>			
Resim	Başlık	Anasayfa	Stok	Kategori	
	Iphone	<span>✓Evet</span>	<span>✓Var</span>	Telefon	<span>✎</span> <span>🗑</span>
1 Kayıttan 1 - 1 Arası Kayıtlar				Önceki	1 Sonraki

Şekil 2.3.3 Product Index Sayfasının Görünümü

## 2.4. Ürünler Sayfası İçin Ekleme,Düzenleme,Silme Sayfalarının Oluşturulması

Ürünler sayfasında admin tarafından gerçekleştirilecek işlemlerde kullanılmak üzere Create.cshtml,Delete.cshtml,Details.cshtml,Edit.cshtml sayfaları oluşturulmuştur.

Create sayfası ürünleri eklemek için kullanılmaktadır Delete sayfası ürünleri silmek için, Edit sayfası ise var olan ürünleri düzenlemek için kullanılmaktadır. Details sayfasının görevi herhangi bir sayfaya gidildiğinde o sayfada yer alan Kayıt Listesi sayfasına geri döndürmek için kullanılır.

```
<div class="row">
  <div class="col-md-4">
  </div>
  <div class="col-md-4">
    <h4 class="text-success">Ürün Ekleme Sayfası</h4>
    <form asp-action="Create" enctype="multipart/form-data">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group">
        Başlık
        <input asp-for="Title" class="form-control" />
        <span asp-validation-for="Title" class="text-danger"></span>
      </div>
      <div class="form-group">
        Açıklama
        <input asp-for="Description" class="form-control" />
        <span asp-validation-for="Description" class="text-danger"></span>
      </div>
      <div class="form-group">
        Fiyat
        <input asp-for="Price" class="form-control" />
        <span asp-validation-for="Price" class="text-danger"></span>
      </div>
      <div class="form-group">
        Resim
        <input asp-for="Image" class="form-control" type="file" />
        <span asp-validation-for="Image" class="text-danger"></span>
      </div>
      <div class="form-group form-check">
        <label class="form-check-label">
          <input class="form-check-input" asp-for="IsHome" /> Anasayfa
        </label>
      </div>
      <div class="form-group form-check">
        <label class="form-check-label">
          <input class="form-check-input" asp-for="IsStock" /> Stok
        </label>
      </div>
      <div class="form-group">
        Kategori
        <select asp-for="CategoryId" class="form-control" asp-items="ViewBag.CategoryId"></select>
      </div>
      <div class="form-group">
        <input type="submit" value="Ekle" class="btn btn-outline-success" />
        <a asp-action="Index" class="btn btn-info">Kayıt Listesi</a>
      </div>
    </form>
  </div>
</div>
```

Şekil 2.4.1 Create.cshtml sayfasının görünümü

```

</div>
<div class="col-md-4">
  <h4 class="text-success">Ürün Düzenleme Sayfası</h4>
  <form asp-action="Edit" enctype="multipart/form-data">
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
    <input type="hidden" asp-for="Id" />
    <div class="form-group">
      Başlık
      <input asp-for="Title" class="form-control" />
      <span asp-validation-for="Title" class="text-danger"></span>
    </div>
    <div class="form-group">
      Açıklama
      <input asp-for="Description" class="form-control" />
      <span asp-validation-for="Description" class="text-danger"></span>
    </div>
    <div class="form-group">
      Fiyat
      <input asp-for="Price" class="form-control" />
      <span asp-validation-for="Price" class="text-danger"></span>
    </div>
    <div class="form-group">
      Resim
      <input asp-for="Image" class="form-control" type="hidden" />
      <input asp-for="Image" class="form-control" type="file" />
      <span asp-validation-for="Image" class="text-danger"></span>
    </div>
    <div class="form-group form-check">
      <label class="form-check-label">
        <input class="form-check-input" asp-for="IsHome" /> Anasayfa
      </label>
    </div>
    <div class="form-group form-check">
      <label class="form-check-label">
        <input class="form-check-input" asp-for="IsStock" /> Stok
      </label>
    </div>
    <div class="form-group">
      Kategori
      <select asp-for="CategoryId" class="form-control" asp-items="ViewBag.CategoryId"></select>
    </div>
    <div class="form-group">
      <input type="submit" value="Düzenle" class="btn btn-success" />
      <a asp-action="Index" class="btn btn-info">Kayıt Listesi</a>
    </div>
  </form>

```

Şekil 2.4.2 Edit.cshtml sayfasının görünümü

```

<dl class="row">
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Title)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Title)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Description)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Description)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Price)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Price)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Image)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Image)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.IsHome)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.IsHome)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.IsStock)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.IsStock)
  </dd>
  <dt class="col-sm-2">
    @Html.DisplayNameFor(model => model.Category)
  </dt>
  <dd class="col-sm-10">
    @Html.DisplayFor(model => model.Category.Name)
  </dd>
</dl>
</div>
<div>
  <a asp-action="Edit" asp-route-id="@Model.Id">Edit</a> |
  <a asp-action="Index">Back to List</a>
</div>

```

Şekil 2.4.3 Details.cshtml sayfasının görünümü

```

<h3>Bu Kaydı Silmek İstiyor Musunuz?</h3>
<div>
    <dl class="row">
        <dt class="col-sm-2">
            Başlık
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Title)
        </dd>
        <dt class="col-sm-2">
            Açıklama
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Description)
        </dd>
        <dt class="col-sm-2">
            Fiyat
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Price)
        </dd>
        <dt class="col-sm-2">
            Resim
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Image)
        </dd>
        <dt class="col-sm-2">
            Anasayfa
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.IsHome)
        </dd>
        <dt class="col-sm-2">
            Stok
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.IsStock)
        </dd>
        <dt class="col-sm-2">
            Kategori
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Category.Name)
        </dd>
    </dl>

```

Şekil 2.4.4 Delete.cshtml sayfasının görünümü

### Ürün Ekleme Sayfası

Başlık

Açıklama

Fiyat

Resim

Dosya seçilmedi

☐ Anasayfa

☐ Stok

Kategori

Telefon

### Ürün Düzenleme Sayfası

Başlık

Açıklama

Fiyat

Resim

Dosya seçilmedi

☒ Anasayfa

☒ Stok

Kategori

Telefon



Şekil 2.4.5 Ekleme ve Düzenleme sayfalarının görünümü

## 2.5. Kullanıcı Sayfasının Oluşturulması

Bu sayfada kullanıcıların sisteme giriş yapabilmeleri için gerekli özellikleri girerek siteye giriş yapabilecekleri bir ortam oluşturulmuştur. Aynı zamanda kullanıcıların veri tabanındaki özellikleri de verilmiştir.


Yazılan kodlar ve sayfanın görünümü aşağıda gösterilmiştir:

```
1 using Microsoft.AspNetCore.Identity;
2 using System;
3 using System.Collections.Generic;
4 using System.ComponentModel.DataAnnotations;
5 using System.ComponentModel.DataAnnotations.Schema;
6 using System.Linq;
7 using System.Threading.Tasks;
8
9 namespace OrnekSite.Models
10 {
11     11 references
12     public class ApplicationUser:IdentityUser
13     {
14         [Required]
15         3 references
16         public string Name { get; set; }
17         [Required]
18         3 references
19         public string Surname { get; set; }
20         2 references
21         public string Adres { get; set; }
22         2 references
23         public string Sehir { get; set; }
24         2 references
25         public string Semt { get; set; }
26         2 references
27         public string PostaKodu { get; set; }
28         [NotMapped]
29         6 references
30         public string Role { get; set; }
31     }
32 }
```

Adres	Discriminat...	Name	PostaKodu	Sehir	Semt	Surname
Ankara Üniv...	Application...	Oğuzhan	12322	Ankara	Ankara/Göl...	Sezer

Kayıt olma işlemini başarıyla gerçekleştirilmesini sağlayan Register.html.cs sayfasının görünümü de aşağıdaki gibi verilmiştir. Bu sayfada kullanıcıdan kendine ait birçok kişisel bilgi istenerek sisteme sorunsuz bir şekilde kayıt olması amaçlanmıştır.

### Kayıt Formu

Adı
Soyadı
Email
Telefon
Adres
Semt
Şehir
Posta Kodu
Şifre
Şifre Tekrar
 Üye Ol

## 2.6. Roller Sayfasının Oluşturulması

Bu sayfada sisteme giriş yapan kullanıcılara sistemi kullanabilmeleri için farklı roller verilmesi amaçlanmıştır. Örneğin admin olan bir kullanıcı site içinde istediği değişiklikleri yapma hakkına sahip olacaktır.

Yazılan kodlar ve sayfanın görünümü ve veri tabanına yansımaları aşağıda gösterilmiştir:

```
public const string Role_Birey = "Birey";
public const string Role_Admin = "Admin";
public const string Role_User = "User";



private readonly RoleManager<IdentityRole> _roleManager;
private readonly ApplicationDbContext _db;

var user = new ApplicationUser
{
    UserName = Input.Email,
    Email = Input.Email,
    Adres = Input.Adres,
    Sehir = Input.Sehir,
    Semt = Input.Semt,
    Name = Input.Name,
    Surname = Input.Surname,
    PhoneNumber = Input.TelefonNo,
    PostaKodu = Input.PostaKodu,
    Role = Input.Role
};
var result = await _userManager.CreateAsync(user, Input.Password);
if (result.Succeeded)
{
    _logger.LogInformation("User created a new account with password.");
    if (!await _roleManager.RoleExistsAsync(Diger.Role_Admin))
    {
        await _roleManager.CreateAsync(new IdentityRole(Diger.Role_Admin));
    }
    if (!await _roleManager.RoleExistsAsync(Diger.Role_User))
    {
        await _roleManager.CreateAsync(new IdentityRole(Diger.Role_User));
    }
    if (!await _roleManager.RoleExistsAsync(Diger.Role_Birey))
    {
        await _roleManager.CreateAsync(new IdentityRole(Diger.Role_Birey));
    }
    if (user.Role == null)
    {
        await _userManager.AddToRoleAsync(user, Diger.Role_User);
    }
    else
    {
        await _userManager.AddToRoleAsync(user, user.Role);
    }
    var code = await _userManager.GenerateEmailConfirmationTokenAsync(user);
}
```

Şekil 2.6.1 Roller Sayfası İçin Yazılan Kodlar

Kullanıcı Listesi Kullanıcı Ekle

Sayfada 10 Kayıt Göster Bul:

Adı	Soyadı	Email	Rol	
Oğuzhan	Sezer	oguzhansezer14@gmail.com	Admin	
Oğuzhan	Sezer	oguzhansezer@gmail.com	User	

2 Kayıttan 1 - 2 Arası Kayıtlar Önceki 1 Sonraki

Şekil 2.6.2 Kullanıcı Listesinin ve Rollerin Görünümü



## 2.7 Sepet İşlemleri İçin Kullanılacak Olan Sayfanın Oluşturulması

Bu sayfada Kullanıcının alışveriş yapmak için belirli bir süre kendi üstünde tutmasına ve almak istediği ürünleri toplu bir şekilde sergilemesine yardımcı olmak için bir E-Ticaret sitesinde bulunması gereken sepet işlemi yapılmıştır.

Yazılan kodlar ve sayfanın görünümü aşağıda gösterilmiştir:

```
12 references
public class ShoppingCart
{
    2 references
    public ShoppingCart()
    {
        Count = 1;
    }
    [Key]
    9 references
    public int Id { get; set; }
    15 references
    public string ApplicationUserId { get; set; }
    [ForeignKey("ApplicationUserId")]
    0 references
    public ApplicationUser ApplicationUser { get; set; }
    6 references
    public int ProductId { get; set; }
    [ForeignKey("ProductId")]
    26 references
    public Product Product { get; set; }
    17 references
    public int Count { get; set; }
    [NotMapped]
    4 references
    public double Price { get; set; }
}
```

Şekil 2.7.1 ShoppingCart.cs Sayfası İçin Yazılan Kodlar

Sipariş verme ekranı için OrderHeader.cs ve bu siparişin detayını içermesi için OrderDetails.cs adlı iki sayfa oluşturduk.

```

24 references
public class OrderHeader
{
    [Key]
    16 references
    public int Id { get; set; }
    5 references
    public string ApplicationUserId { get; set; }
    [ForeignKey("ApplicationUserId")]
    6 references
    public ApplicationUser ApplicationUser { get; set; }
    2 references
    public DateTime OrderDate { get; set; }
    10 references
    public double OrderTotal { get; set; }
    16 references
    public string OrderStatus { get; set; }
    [Required]
    8 references
    public string Name { get; set; }
    [Required]
    8 references
    public string Surname { get; set; }
    [Required]
    8 references
    public string PhoneNumber { get; set; }
    [Required]
    4 references
    public string Adres { get; set; }
    [Required]
    3 references
    public string Semt { get; set; }
    [Required]
    4 references
    public string Sehir { get; set; }
    [Required]
    4 references
    public string PostaKodu { get; set; }
    [Required]
    3 references
    public string CartName { get; set; }
    [Required]
    3 references
    public string CartNumber { get; set; }
    [Required]

```

Şekil 2.7.2 OrderHeader.cs kodları

```

namespace OrnekSite.Models
{
    4 references
    public class OrderDetails
    {
        [Key]
        0 references
        public int Id { get; set; }
        [Required]
        2 references
        public int OrderId { get; set; }
        [ForeignKey("OrderId")]
        0 references
        public OrderHeader OrderHeader { get; set; }
        1 reference
        public int ProductId { get; set; }
        [ForeignKey("ProductId")]
        3 references
        public Product Product { get; set; }
        3 references
        public int Count { get; set; }
        1 reference
        public double Price { get; set; }
    }
}

```

Şekil 2.7.3 OrderDetails.cs kodları

## 2.8. Ürün Detay Sayfasının Oluşturulması

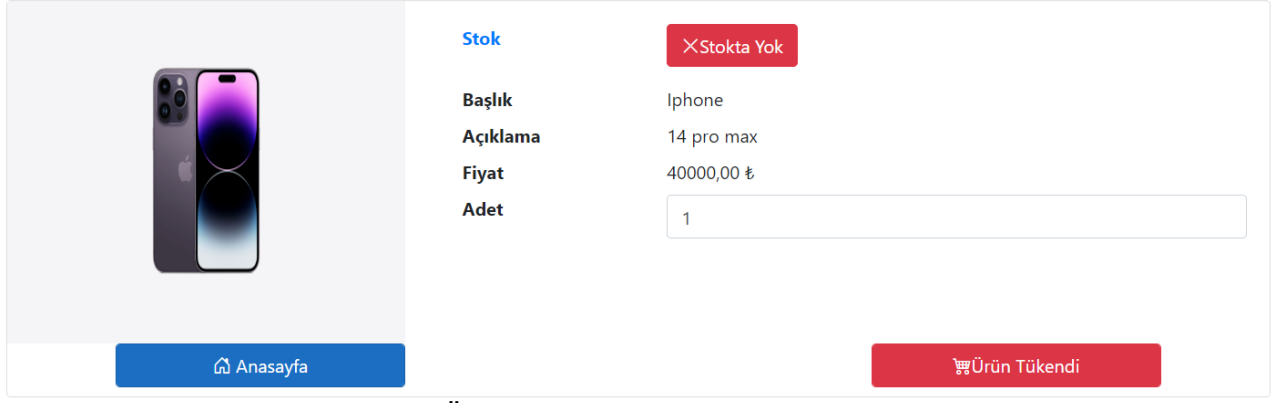
Bu sayfa ürünlerin detayının gösterilmesi için oluşturulmuştur.

Sayfaya yazılan kodlar ve görünümü aşağıdaki gibidir:

```
public IActionResult Details(int id)
{
    var product = _db.Products.FirstOrDefault(i => i.Id == id);
    ShoppingCart cart = new ShoppingCart()
    {
        Product = product,
        ProductId = product.Id
    };
    return View(cart);
}

<div class="container">
    <br />
    <div class="row">
        <form method="post">
            <input hidden asp-for="ProductId" />
            <div class="card mb-12">
                <div class="row g-0">
                    <div class="col-md-4">
                        
                    </div>
                    <div class="col-md-8">
                        <div class="card-body">
                            <dl class="row">
                                <dt class="col-sm-3 text-primary">
                                    Stok
                                </dt>
                                <dd class="col-sm-9">
                                    @if (Model.Product.IsStock)
                                    {
                                        <label class="btn btn-success "><i class="bi bi-check-lg"></i>Stokta Var</label>
                                    }
                                    else
                                    {
                                        <label class="btn btn-danger "><i class="bi bi-x-lg"></i>Stokta Yok</label>
                                    }
                                </dd>
                                <dt class="col-sm-3">
                                    Başlık
                                </dt>
                                <dd class="col-sm-9">
                                    @Model.Product.Title
                                </dd>
                                <dt class="col-sm-3">
                                    Açıklama
                                </dt>
                                <dd class="col-sm-9">
                                    @Model.Product.Description
                                </dd>
                                <dt class="col-sm-3">
                                    Adet
                                </dt>
                                <dd class="col-sm-9">
                                    <input asp-for="Count" class="form-control" />
                                </dd>
                            </dl>
                        </div>
                    </div>
                </div>
                <br />
                <div class="col-md-1">
                </div>
                <div class="col-md-3">
                    <a asp-action="Index" asp-controller="Home" class="btn btn-primary form-control"><i class="bi bi-house-door"></i> Anasayfa</a>
                </div>
                <div class="col-md-4">
                </div>
                <div class="col-md-3">
                    @if (Model.Product.IsStock)
                    {
                        <button type="submit" class="btn btn-success form-control"><i class="bi bi-cart4"></i>Sepete Ekle</button>
                    }
                    else
                    {
                        <label class="btn btn-danger form-control"><i class="bi bi-cart4"></i>Ürün Tükendi</label>
                    }
                </div>
                <div class="col-md-1">
                </div>
            </div>
        </form>
    </div>
</div>
```

Şekil 2.8.1 Ürün Detay Sayfasına Yazılan Kodlar



Şekil 2.8.2 Ürün Detay Sayfası Görünümü

Yukarıdaki şekilde de görüldüğü üzere ürünlerin veri tabanında yer alan tüm bilgiler sayfada listelenmektedir.

## 2.9. Sepet Sayfasının Oluşturulması

Bu sayfada sepete eklenen ürünlerin adedi ve tutarı sipariş vermeden önce sergilenmektedir.

Sayfaya yazılan kodlar ve görünümü aşağıda gösterilmiştir:

```
<form method="post">
  @if (Model.ListCart.Count() > 0)
  {
    <div class="container">
      <div class="row">
        <div class="col-md-12">
          <br />
          <table class="table table-striped table-bordered" id="myTable">
            <thead>
              <tr>
                <th>Resim</th>
                <th>Başlık</th>
                <th>Adet</th>
                <th>Fiyat</th>
                <th>Tutar</th>
              </tr>
            </thead>
            <tbody>
              @foreach (var item in Model.ListCart)
              {
                <tr>
                  <td>
                    
                  </td>
                  <td>
                    @item.Product.Title
                  </td>
                  <td>
                    <div class="btn-toolbar" role="toolbar" aria-label="Toolbar with button groups">
                      <div class="btn-group me-2" role="group" aria-label="First group">
                        <input type="text" style="max-width:45px;" class="form-control" value="@item.Count" />
                        <button asp-action="Decrease" asp-route-cartId="@item.Id" class="btn btn-danger"><i class="bi bi-dash-lg"></i></button>
                        <button asp-action="Add" asp-route-cartId="@item.Id" class="btn btn-primary"><i class="bi bi-plus-lg"></i></button>
                        <button asp-action="Remove" asp-route-cartId="@item.Id" class="btn btn-danger"><i class="bi bi-trash"></i></button>
                      </div>
                    </div>
                  </td>
                  <td>
                    @item.Product.Price.ToString("0.00") ₺
                  </td>
                  <td>
                    @((item.Count*item.Product.Price).ToString("0.00")) ₺
                  </td>
                </tr>
              }
            </tbody>
          </table>
          <div class="row">
            <div class="col-md-12">
              <div class="text-align: right">
                <strong>TOPLAMA TUTAR=</strong>
                <div style="background-color: darkred; color: white; padding: 5px; display: inline-block;>
                  @Model.OrderHeader.OrderTotal.ToString("0.00") ₺
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  }
</form>
```

```

        <td style="background-color: #darkred">
            <span style="font-size:large;color:white">@Model.OrderHeader.OrderTotal.ToString("0.00") €</span>
        </td>
    </tr>
</tbody>
</table>
<div class="row">
    <div class="col-md-4">
        <a asp-area="Customer" asp-controller="Home" asp-action="Index" class="btn btn-primary form-control">Alışverişe Devam Et</a>
    </div>
    <div class="col-md-4">
    </div>
    <div class="col-md-4">
        @if (!Model.OrderHeader.ApplicationUser.EmailConfirmed)
        {
            <div>
                <button type="submit" class="btn btn-warning form-control text-white">Lütfen Email Onaylamak için tıklayınız...</button>
            </div>
        }
        else
        {
            <a asp-area="Customer" asp-controller="Cart" asp-action="Summary" class="btn btn-success form-control">Sipariş Ver</a>
        }
    </div>
</div>
</div>
</div>
}



else
{
    <div>
        Sepetinizde ürün bulunmamaktadır...
    </div>
}

</form>

```

### Şekil 2.9.1 Sepet Görünümü Sayfası Yazılan Kodlar

Sayfanın görünümü aşağıda gösterilmiştir:

Resim	Başlık	Adet	Fiyat	Tutar
	Iphone	<div>5<div>-</div><div>+</div><div></div></div>	40000,00 ₺	200000,00 ₺
TOPLAMA TUTAR=				200000,00 ₺
Alışverişe Devam Et			Sipariş Ver	

### Şekil 2.9.2 Sepet Sayfasının Sitedeki Görünümü

## 2.10. Sipariş Verme Sayfası

Bu sayfa kullanıcının siparişleri verme sayfası olarak tasarlanmıştır. Bu sayfada ödeme işlemleri yapılabilmekte ve kullanıcı hakkında bilgiler istenmektedir.

```
<form method="post">
  <div class="container">
    <div class="card">
      <div class="card-body">
        <div class="row">
          <div class="col-md-6">
            <div class="mb-3 row">
              <label for="inputPassword" class="col-sm-3 col-form-label">Adı</label>
              <div class="col-sm-9">
                <input type="text" asp-for="OrderHeader.Name" class="form-control">
                <span asp-validation-for="OrderHeader.Name" class="text-danger"></span>
              </div>
            </div>
            <div class="mb-3 row">
              <label for="inputPassword" class="col-sm-3 col-form-label">Soyadı</label>
              <div class="col-sm-9">
                <input type="text" asp-for="OrderHeader.Surname" class="form-control">
                <span asp-validation-for="OrderHeader.Surname" class="text-danger"></span>
              </div>
            </div>
            <div class="mb-3 row">
              <label for="inputPassword" class="col-sm-3 col-form-label">Adres</label>
              <div class="col-sm-9">
                <input type="text" asp-for="OrderHeader.Adres" class="form-control">
                <span asp-validation-for="OrderHeader.Adres" class="text-danger"></span>
              </div>
            </div>
            <div class="mb-3 row">
              <label for="inputPassword" class="col-sm-3 col-form-label">Semt</label>
              <div class="col-sm-9">
                <input type="text" asp-for="OrderHeader.Semt" class="form-control">
                <span asp-validation-for="OrderHeader.Semt" class="text-danger"></span>
              </div>
            </div>
            <div class="mb-3 row">
              <label for="inputPassword" class="col-sm-3 col-form-label">Şehir</label>
              <div class="col-sm-9">
                <input type="text" asp-for="OrderHeader.Sehir" class="form-control">
                <span asp-validation-for="OrderHeader.Sehir" class="text-danger"></span>
              </div>
            </div>
            <div class="mb-3 row">
              <label for="inputPassword" class="col-sm-3 col-form-label">Telefon</label>
              <div class="col-sm-9">
                <input type="text" asp-for="OrderHeader.PhoneNumber" class="form-control">
                <span asp-validation-for="OrderHeader.PhoneNumber" class="text-danger"></span>
              </div>
            </div>
            <div class="mb-3 row">
              <label for="inputPassword" class="col-sm-3 col-form-label">Posta Kodu</label>
              <div class="col-sm-9">
                <input type="text" asp-for="OrderHeader.PostaKodu" class="form-control">
                <span asp-validation-for="OrderHeader.PostaKodu" class="text-danger"></span>
              </div>
            </div>
          </div>
          <div class="col-md-6">
            <div class="col-md-12">
              <ul class="list-group mb-3">
                <li>
                  <div class="list-group-item d-flex justify-content-between lh-sm">
                    <div>
                      <h6 class="my-0">@item.Product.Title</h6>
                      <small class="text-muted">Adet : @item.Count</small>
                    </div>
                    <span class="text-muted">@((item.Count*item.Product.Price).ToString("0.00")) ₺</span>
                  </li>
                </ul>
                <li class="list-group-item d-flex justify-content-between bg-light">
                  <div class="text-success">
                    <h6 class="my-0">Toplam Tutar :</h6>
                  </div>
                  <span class="text-success">@((Model.OrderHeader.OrderTotal.ToString("0.00")) ₺</span>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</form>
```

```

</div>
<div class="mb-3 row">
  <label for="inputPassword" class="col-sm-4 col-form-label">Kart Adı</label>
  <div class="col-sm-8">
    <input type="text" asp-for="OrderHeader.CartName" class="form-control">
    <span asp-validation-for="OrderHeader.CartName" class="text-danger"></span>
  </div>
</div>
<div class="mb-3 row">
  <label for="inputPassword" class="col-sm-4 col-form-label">Kart Numarası</label>
  <div class="col-sm-8">
    <input type="text" asp-for="OrderHeader.CartNumber" class="form-control">
    <span asp-validation-for="OrderHeader.CartNumber" class="text-danger"></span>
  </div>
</div>
<div class="mb-3 row">
  <div class="col-sm-4">
    <input type="text" asp-for="OrderHeader.ExpirationMonth" class="form-control" placeholder="Ay">
    <span asp-validation-for="OrderHeader.ExpirationMonth" class="text-danger"></span>
  </div>
  <div class="col-sm-4">
    <input type="text" asp-for="OrderHeader.ExpirationYear" class="form-control" placeholder="Yıl">
    <span asp-validation-for="OrderHeader.ExpirationYear" class="text-danger"></span>
  </div>
  <div class="col-sm-4">
    <input type="text" asp-for="OrderHeader.Cvc" class="form-control" placeholder="Güvenlik Kodu">
    <span asp-validation-for="OrderHeader.Cvc" class="text-danger"></span>
  </div>
</div>
</div>
</div>
</div>
<div class="card-footer">
  <div class="row">
    <div class="col-md-6">
      </div>
    </div>
  </div>
  <div class="card-footer">
    <div class="row">
      <div class="col-md-6">
        </div>
      <div class="col-md-6">
        <button type="submit" class="btn btn-success form-control">Sipariş Tamamla</button>
      </div>
    </div>
  </div>
</div>
</div>
</form>

@section Scripts{
<partial name="_ValidationScriptsPartial" />
}

```

Adı	Oğuzhan	Iphone	40000,00 ₺
Soyadı	Sezer	Adet :1	
Adres	Ankara Gölbaşı	<b>Toplam Tutar :</b>	40000,00 ₺
Semt	ankara	Kart Adı	Oğuzhan
Şehir	ankara	Kart Numarası	8794894859389543
Telefon	05511639107	Ay	23 345
Posta Kodu	06790		

Sipariş Tamamla

Şekil 2.10.1 Sipariş Sayfasına Yazılan Kodlar ve Görünümü



All Transactions

06.05.2023 ▶ 06.06.2023 ▼ Filter ▼ Add Detail ▼

Date Range 06.05.2023 - 06.06.2023 ⓘ

Show 20 ▼ Total 1

EXPORT

Conversation ID	Buyer Name	Buyer Surname	Payment Number	Created Date	Paid Price	Payment Status	Cancel / Refund Status
6557 📄	Oğuzhan	Sezer	19653764	06.06.2023 23:23:30	TRY 40,000.00	Success	Not Refunded

Previous Page

Next Page

Şekil 2.10.2 İyzico Sayfasında Siparişin Görünümü

## 2.11. Verilen Siparişleri Görüntüleme ve Listeleme

Bu sayfada kullanıcı ya da admin verilen siparişleri görüntüleyebilmektedir.

```
@model IEnumerable<OrnekSite.Models.OrderHeader>
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="btn-group d-grid gap-2 d-md-flex justify-content-md-end" role="group" aria-label="Basic outlined example">
    <a asp-action="Beklenen" asp-area="Admin" asp-controller="Order" type="button" class="btn btn-outline-primary">Beklenen Siparişler</a>
    <a asp-action="Onaylanan" asp-area="Admin" asp-controller="Order" type="button" class="btn btn-outline-primary">Onaylanan Siparişler</a>
    <a asp-action="Kargolanan" asp-area="Admin" asp-controller="Order" type="button" class="btn btn-outline-primary">Kargolanan Siparişler</a>
    <a asp-action="Index" asp-area="Admin" asp-controller="Order" type="button" class="btn btn-outline-primary active">Tüm Siparişler</a>
</div>
<br />
<div class="container">
    <table class="table table-striped table-bordered" id="myTable">
        <thead>
            <tr>
                <th>Adı</th>
                <th>Soyadı</th>
                <th>Telefon</th>
                <th>Durum</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var item in Model)
            {
                <tr>
                    <td>@item.Name</td>
                    <td>@item.Name</td>
                    <td>@item.Surname</td>
                    <td>@item.PhoneNumber</td>
                    <td>@item.OrderStatus</td>
                    <td>
                        @if (User.IsInRole(Diger.Role_Admin))
                        {
                            <a asp-action="Details" asp-route-id="@item.Id" class="btn btn-success"><i class="bi bi-pencil"></i> Düzenle</a>
                        }
                        else
                        {
                            <a asp-action="Details" asp-route-id="@item.Id" class="btn btn-success"><i class="bi bi-pencil"></i> Detay</a>
                        }
                    </td>
                </tr>
            }
        </tbody>
    </table>
</div>

<section Scripts>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#myTable').DataTable({
                "language": {
                    "url": "https://cdn.datatables.net/plug-ins/9dcbecd42ad/i18n/Turkish.json"
                }
            });
        });
    </script>
</section>
```

Şekil 2.11.1 Verilen Siparişleri listelemek için yazılan kodlar

Beklenen Siparişler

Onaylanan Siparişler

Kargolanan Siparişler

Tüm Siparişler

Sayfada10Kayıt Göster

Bul:

Adı	Soyadı	Telefon	Durum	
Oğuzhan	Sezer	05511639107	Beklemede	<div>Detay</div>
Oğuzhan	Sezer	05511639107	Beklemede	<div>Detay</div>
Oğuzhan	Sezer	05511639107	Beklemede	<div>Detay</div>

3 Kayıttan 1 - 3 Arası Kayıtlar

Önceki

1

Sonraki

Şekil 2.11.2 Verilen Siparişlerin Sitedeki Görünümü

Beklenen Siparişler

Onaylanan Siparişler

Kargolanan Siparişler

Tüm Siparişler

Sayfada10Kayıt Göster

Bul:

Adı	Soyadı	Telefon	Durum	
Oğuzhan	Sezer	05511639107	Onaylandı	<div>Düzenle</div>

1 Kayıttan 1 - 1 Arası Kayıtlar

Önceki1Sonraki

Şekil 2.11.3 Onaylanan Siparişlerin Görünümü

Beklenen Siparişler

Onaylanan Siparişler

Kargolanan Siparişler

Tüm Siparişler

Sayfada10Kayıt Göster

Bul:

Adı	Soyadı	Telefon	Durum	
Oğuzhan	Sezer	05511639107	Kargoya Verildi	<div>Düzenle</div>

1 Kayıttan 1 - 1 Arası Kayıtlar

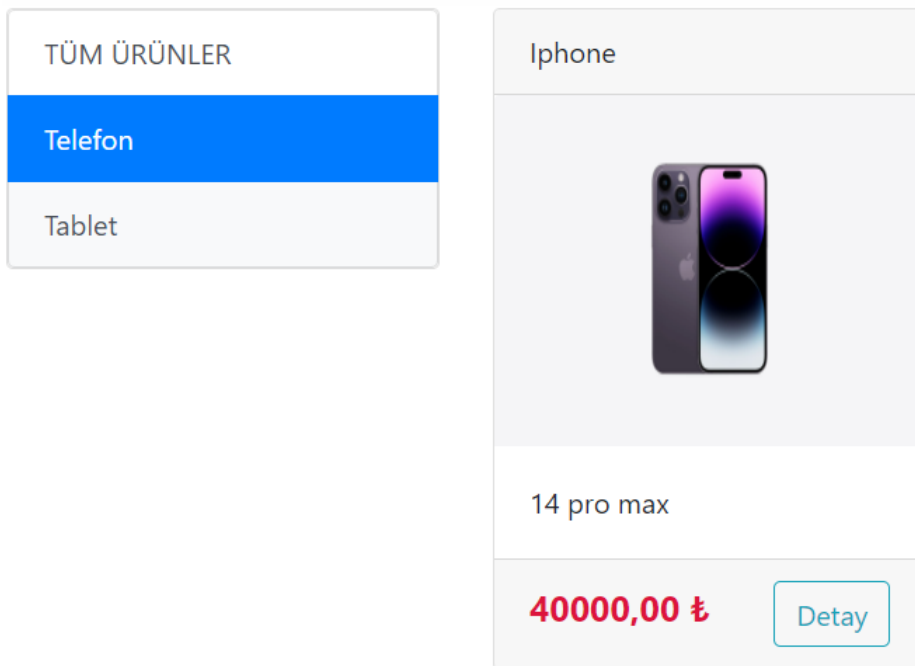
Önceki1Sonraki

Şekil 2.11.4 Kargolanan Siparişlerin Görünümü

## 2.12. Kategori Listesi Görüntüleme

```
@model IEnumerable<OrnekSite.Models.Product>
@{
    ViewData["Title"] = "CategoryDetails";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="container">
    <div class="row">
        @await Component.InvokeAsync("CategoryList")
        @if (Model.Count() > 0)
        {
            @foreach (var item in Model)
            {
                <div class="col-md-3">
                    <div class="card">
                        <div class="card-header">
                            @item.Title
                        </div>
                        
                        <div class="card-body">
                            <p class="card-text">@item.Description</p>
                        </div>
                        <div class="card-footer">
                            <label style="font-size:20px;color:crimson"><b>@item.Price.ToString("0.00") t</b></label>
                            <a asp-action="Details" asp-controller="Home" asp-route-id="@item.Id" class="btn btn-outline-info" style="float:right">Detay</a>
                        </div>
                    </div>
                </div>
            }
        }
        else
        {
            <div class="alert alert-warning text-center">
                Herhangi bir ürün bulunmamaktadır...
            </div>
        }
    </div>
</div>
```



Şekil 2.12.1 Anasayfaya Kategorilere Göre Gelen Ürünleri Listeleme

Sonuç olarak, önceki bölümlerde açıklandığı gibi, bilgiler elde edilmiş ve ASP.NET MVC yapısı kullanılarak bir proje geliştirilmiştir. MVC, SendGrid, Iyzico, DataTables gibi yapıların nasıl kullanıldığı öğrenilmiş ve bilgi birikimine sahip olunmuştur.

Proje geliştirilirken alınan çeşitli hatalarda nasıl yollar izlenmesi gerektiği ve ne gibi araştırılması gerektiği hakkında bilgi sahibi olunmuştur.

Proje değişikliklerini yönetmek ve takip etmek için kullanılan GİT sistemi hakkında tecrübe edinilmiştir.

Anlatılan tüm süreçlerin sonunda her yaştan her cinsiyetten tüm insanlara hitap eden, kolay ve güvenli kullanım sağlayan responsive bir online ticaret projesi oluşturulmuş ve bu proje sayesinde kullanıcıların en iyi ürünlere ulaşması kolaylaştırılmış aynı zamanda ürünlerin tanıtımı açıklayıcı olarak anlaşılır bir şekilde ürünlere ulaşılması sağlanmıştır.

#### 4. KAYNAKLAR

<https://www.yusufsezer.com.tr/asp-net-mvc/>

<https://stackoverflow.com/>

<https://sendgrid.com/>

<https://github.com/iyzico>

<https://www.iyzico.com/>

<https://datatables.net/>

<https://www.w3schools.com/>

<https://icons.getbootstrap.com/>