

İSTANBUL MEDİPOL ÜNİVERSİTESİ
MESLEK YÜKSEKOKULU



YÖNLENDİRİLİMİŞ ÇALIŞMA II DERSİ VİZE SUNUMU
BLOG WEB SİTESİ TASARIMI KODLAMA VE DIAGRAM ANALİZİ

KÜBRA AYKOL
(H5210072)

DERS DANIŞMANI: ÖĞR. GRV. BEYZA KOYULMUŞ

İÇİNDEKİLER

Özet.....	1
Veritabanı Diagramları ve Veritabanı Örnek Veri Ekran Görüntüleri.....	2
Ecran Detayları.....	5
Admin Panelinin Oluşturulması.....	18
Admin Sayfası Giriş Ve Şifre Yenileme.....	20
Kodlama ve Prensip Açıklaması	24
Eğitim Sayfası Ekran Detayları.....	26
Deneyim Sayfası Ekran Detayları.....	30
Hobiler Sayfası Ekran Detayları.....	33
Yetenek Sayfası Ekran Detayları.....	35
Sertifika Sayfası Ekran Detayları.....	39
İletişim Sayfası Ekran Detayları.....	43
Sosyal Medya Sayfası Ekran Detayları.....	45
Auzhorized ve Global Ekran Detayları.....	48

ÖZET

Projem bir blog web geliştirme projesi olup, CV niteliğinde kullanılırken aynı zamanda yazılım sektörüne yönelik bilgi ve becerilerimin içeriğini oluşturan bir çalışma olması niteliğini taşımakta. Eğitim bilgilerim, yeteneklerim, sosyal medya ikonlarım ve iletişim gibi bölümler vardır.

Sayfa düzeninde bölümler açıkça görüntülenebilmekte ve tıklamalarla akışı içerisinde yer değiştirmeler ile görsel destekli bir tasarım ile oluşturulmuştur. Bunun yanı sıra yönetici paneli ile de siteye içerik ekleme, çıkarma gibi işlemlerin yapılacağı panel bulunmaktadır.

Teknik özellikler olarak MS MSQQL, C#, MVC teknolojileri ile tasarımda HTML-CSS kodlamaları yapılmıştır.

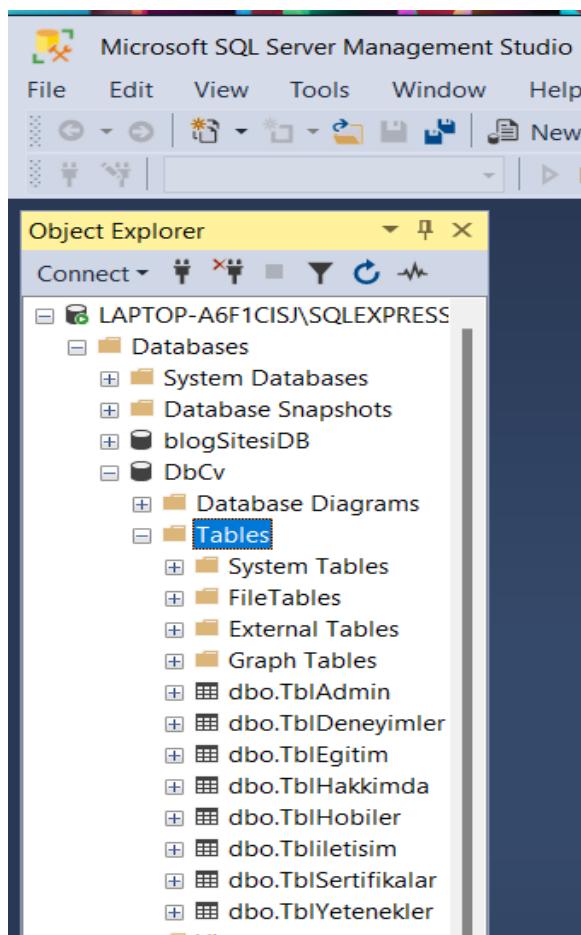
Projenin Bahar Dönemi Vize sunumunda ağırlıklı olarak ön yüz tasarımı, veri tabanı bağlantısı ve veri tabanındaki verilerin ön yüzde görüntüleniş biçimini üzerinde ilerleme kaydedilmiştir. Bunun yanı sıra tarayıcıda çalışır vaziyetteken veri girişleri ve bu giriş neticesinde veri tabanına kaydolan veriler mevcuttu.

Devam eden süreçte mevcuttaki web projesinin verilerinin yönetildiği kısmı olan **“Admin Paneli”** oluşturulmaya başlanıldı. Panelin tasarımı ve panel üzerinden verilerin oluşturulması, düzenlenmesi, silinmesi şeklinde komutlara yönelik kodlamalar yapıldı. Bunlarla beraber butonlar çalışır vaziyete getirildi.

Tamamlanan kısımlara yönelik kodların işlevleri ve teorik açıklanışı, Visual Studio Community’den çalıştırılan bu kodların tarayıcı üzerindeki görüntüleri ise ekran görüntüsü ile desteklenmiştir.

Proje bu aşamanın da tamamlanmasıyla, gereken alan adı ve host işlemleriyle canlıya çıkmaya hazır bir proje haline gelmiştir.

VERİTABANI DIAGRAMLARI VE VERİTABANI ÖRNEK DATA EKRAN GÖRÜNTÜLERİ



Resim1:Blog Web Projesi

Resim1'de görüldüğü üzere SQL Server Management Studio 2019 kullanılarak öncelikle tablolarımızı oluşturuyoruz. Burada tablolarımızı birbirleri ile ilişkili biçimde bağlamadan işlemimize devam ediyoruz. Aşağıda “TblHakkimda” ve “TblDeneyimler” tabloları ve bu tablolara ait içerikler mevcuttur.

LAPTOP-A6F1CISJ\... - dbo.TblHakkimda								
	ID	Ad	Soyad	Adres	Telefon	Mail	Açıklama	Resim
▶	1	Kübra	Aykol	Edirne/İstan...	(546)5809508	kubraaykol...	Merhaba ...	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

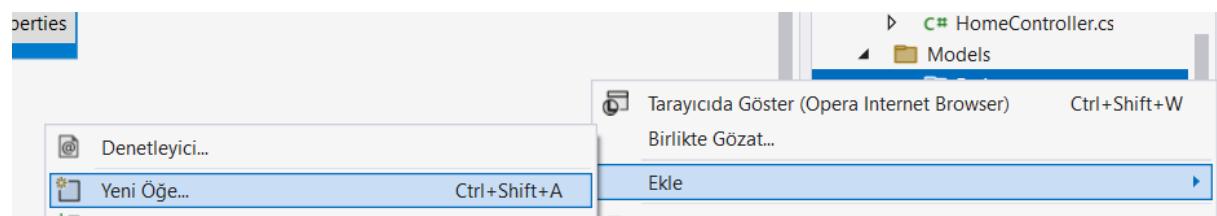
Resim2:”TblHakkimda” tablosu

LAPTOP-A6F1CISJ\...dbo.TblDeneyimler					
	ID	Baslik	AltBaslik	Aclarlama	Tarih
►	1	YTU-TeknoP...	Stajyer	PROJ-E Yazılım...	Temmuz 20...
	2	Turkcell Glo...	TurkishAirlin...	Turcell Glob...	Mayıs 2016- ...
	3	SistemHost ...	Stajyer	Hosting, Do...	Eylül 2012- ...
*	NULL	NULL	NULL	NULL	NULL

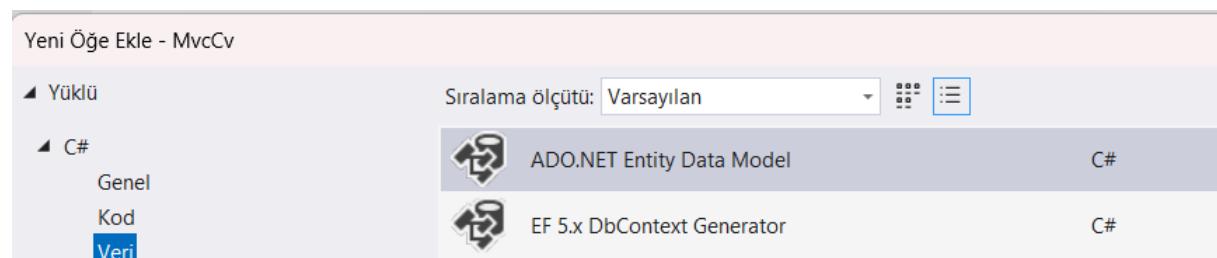
Resim3: "TblDeneyimler" tablosu

Buradan itibaren ise tablolardaki verileri IDE üzerinden ekleyebilmek, güncelleyebilmek gibi dinamik işlemler için veritabanı bağlantımızı Visual Studio'ya gelerek oluşturmaya başlıyoruz. Visual Studio üzerinde yapacağımız geliştirmede MVC teknolojisi ile gelen Models klasörümüzün altına yeni bir Entity kalsörünü oluşturuyoruz.

Resim3 ve Resim4 görsellerinde görüldüğü gibi Ekle>Yeni Öğe>Veri>ADO.NET Entity Data Model direkt öğremizin adını "Model1" olarak seçip ekleme işlemine başlıyoruz

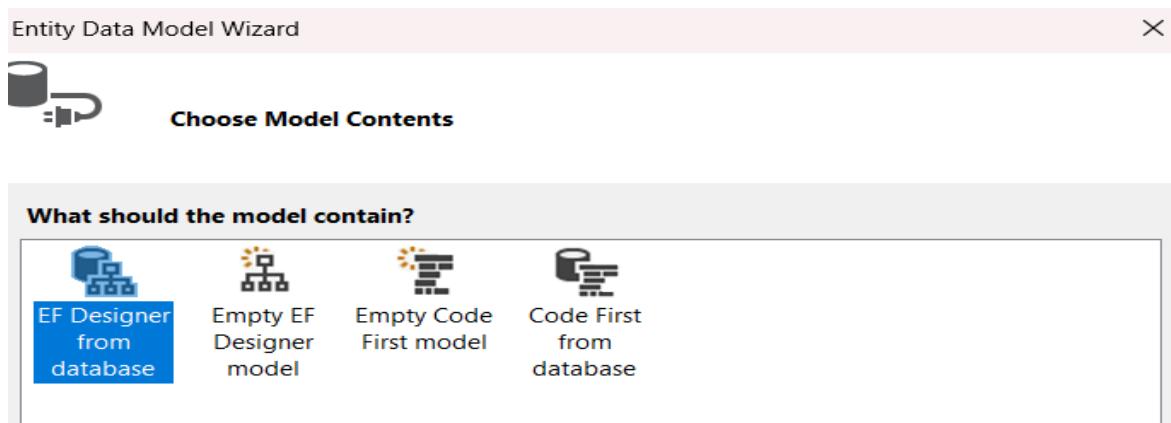


Resim3: Öğe ekleme

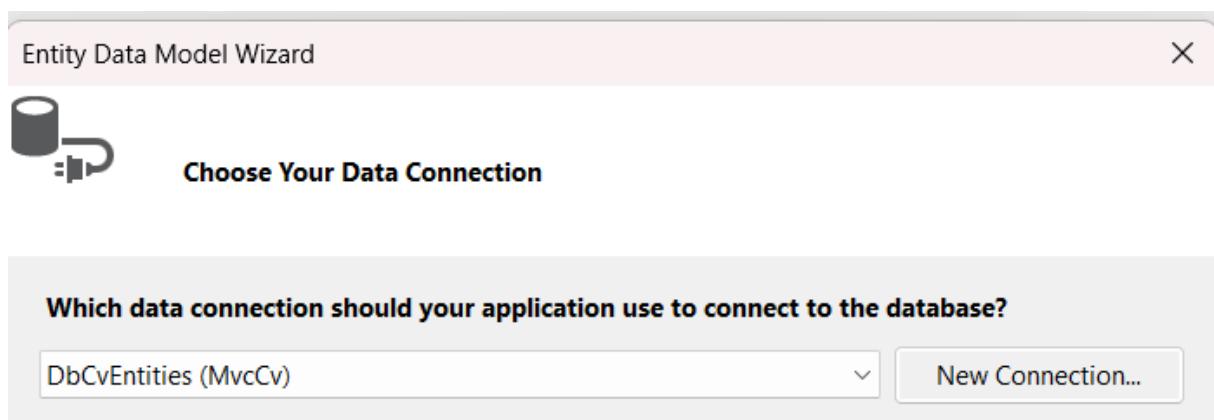


Resim4: ADO.NET Entity Data Model

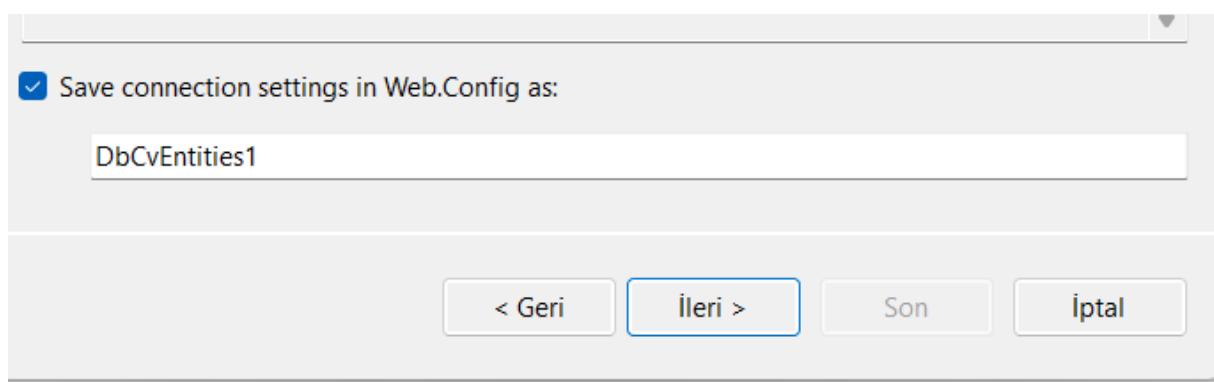
Visual Studio üzerinden SQL bağlantılarının sağlanması için gereken işlem sıralaması aşağıdaki görsellerle de desteklenerek açıklanmıştır.



Resim5:Model içerik tercihi

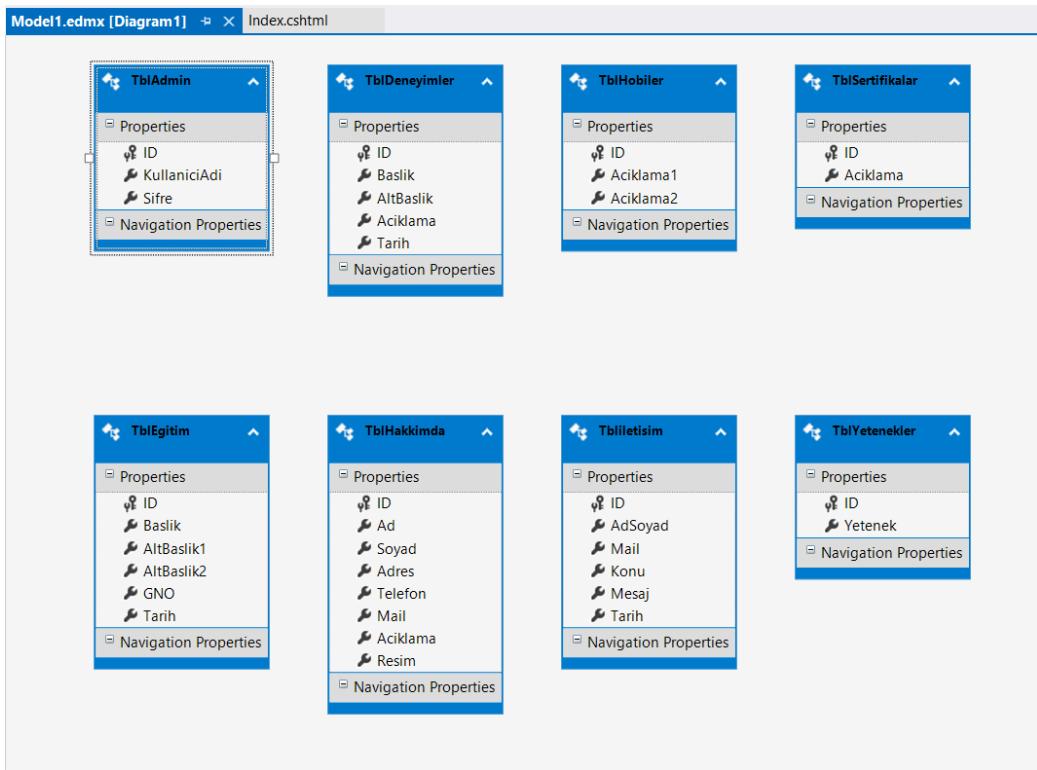


Resim6:SQL üzerindeki tablo seçimi



Resim7:Veritabanı öğelerinin bağlantısı için son işlem

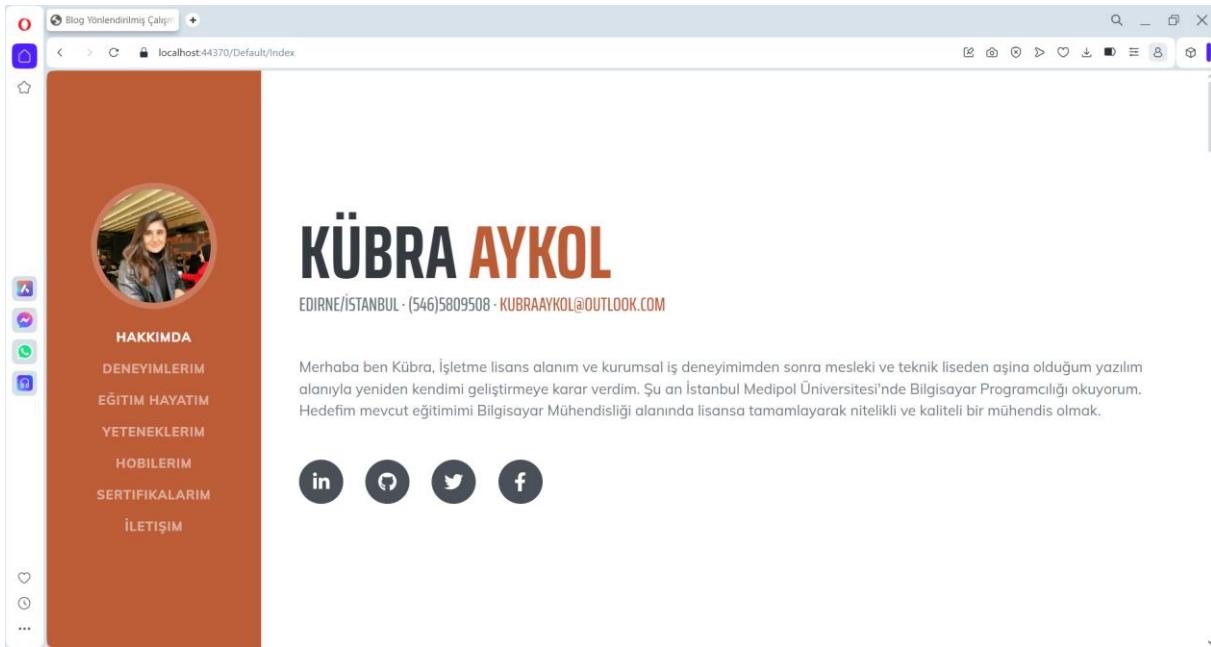
Artık veritabanımız ile bağlantı sağlanarak IDE üzerinden veri ekleme ve güncelleme gibi işlemlerin yapılabileceği dair “Modell” diyagramımız aşağıdaki görsellerdeki gibi işleme hazırlanmıştır.



Resim8:Sql bağlantı ile IDE'ye aktarılan Diyagramın tamamı

E-1 ANASAYFA(NAVBAR) VE HAKKIMDA EKRANI





E1-a.) *DefaultController.cs

Yukarıdaki görselde yer alan kod bloğunda 6. Satırdaki “using MvcCv.Models.Entity;” kodu ile önce Sql üzerinden oluşturduğumuz sonra da IDE’imize yedirdiğimiz bütün tabloların tanımlamasını gerçekleştiriyoruz, klasörü çağrırdık da diyebiliriz. 13. Satırda yazdığımız “DbCvEntities db=new DbCvEntities();” kodundaki “db” isimli türetilen nesne aracılığı ile nesne içerisinde türetilen sınıflara erişim sağlayabiliriz.

16. satırdaki kod ile de “TblHakkimda” tablosunda yer alan verileri de “.ToList();” yöntemi ile bize geri çevirmiş olur. “return View(değerler)” kodu ile de geri döndürmesini istediğimiz verileri tanımlatırız.

E1-b.)*Index.cshtml

```

Index.cshtml ✘ X
1  @using MvcCv.Models.Entity
2  @model List<TblHakkimda>
3
4
5  @{
6      Layout = null;
7  }
8
9  <!DOCTYPE html>
10 <html lang="en">
11 <head>
12     <meta charset="utf-8" />
13     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
14     <meta name="description" content="" />
15     <meta name="author" content="" />
16     <title>Blog Yönlendirilmiş Çözümler</title>
17     <link rel="icon" type="image/x-icon" href="assets/img/favicon.ico" />
18     <!-- Font Awesome icons (free version) -->
19     <script src="https://use.fontawesome.com/releases/v5.15.1/js/all.js" crossorigin="anonymous"></script>
20     <!-- Google fonts-->
21     <link href="https://fonts.googleapis.com/css?family=Saira+Extra+Condensed:500,700" rel="stylesheet" type="text/css" />
22     <link href="https://fonts.googleapis.com/css?family=Muli:400,400i,800,800i" rel="stylesheet" type="text/css" />
23     <!-- Core theme CSS (Includes Bootstrap)-->
24     <link href="startbootstrap-resume-gh-pages/css/styles.css" rel="stylesheet" />
25 </head>
26 <body id="page-top">
27     <!-- Navigation-->
28     <nav class="navbar navbar-expand-lg navbar-dark bg-primary fixed-top" id="sideNav">
29         <a class="navbar-brand js-scroll-trigger" href="#page-top">
30             <span class="d-block d-lg-none">KÜBRA AYKOL</span>
31             <span class="d-none d-lg-block"></span>
32         </a>
33         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span>
34     <div class="collapse navbar-collapse" id="navbarSupportedContent">
35         <ul class="navbar-nav">
36             <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#about">Hakkımda</a></li>
37             <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#experience">Deneyimlerim</a></li>
38             <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#education">Eğitim Hayatım</a></li>
39             <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#skills">Yeteneklerim</a></li>
40             <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#interests">Hobilerim</a></li>
41             <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#awards">Sertifikalarım</a></li>
42             <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#contact">İletişim</a></li>
43         </ul>
44     </div>
45 </nav>

```

Index sayfasına Frontend, Controller sayfasına ise Backend dediğimiz ve iki alan arasında da bağlantı bağlantı sağlayarak prejemizi bir bütün haline getirmeye başlaz. Bunun için de öncelikle 1. Satırda yazan “MvcCv.Models.Entity” kodu ile yaparız. Sql’deki verilerimizi çekememiz için de “@model List<TblHakkimda>” yazılı 2. Satırda yazan kod ile yaparız.

Navbar içerikleri hakkımda kısmında yer aldığı için sayfa akışındaki detaylı hakkımda bölümü için ise 49. Satırda yazan “@foreach(var x in Model {})” koduyla da @foreach döngüsünü açarak içerisinde ‘var’ cinsinde bir x değeri alır ve bunları Model’in içinden alır.

“Hakkımda” kısmına ait alanda çağrıracagımız, listeleyeceğimiz verileri gereken alanlarda foreach döngüsüne tanımladığımız değişkenlerle beraber bağlantı kurarak çağrırmamız gereklidir. Bunun için de gereken alanlarda @ yaparak Modele bağlı olan properties yani özellikleri çağrırmış oluruz. @x. Ad @x. Soyad @x.Adres @x.Telefon @x.Mail @x.Acıklama şeklinde çağrımları tanımlamış oluruz.

E1-c.)*Index.cshtml

```
47  <div class="container py-5" style="background-color: #f9f9f9; padding: 10px; border-radius: 5px; margin-bottom: 20px;">
48    <!-- About-->
49    @foreach (var x in Model)
50    {
51      <section class="resume-section" id="about">
52        <div class="resume-section-content">
53          <h1 class="mb-0">
54            @x.Ad
55            <span class="text-primary">@x.Soyad</span>
56          </h1>
57          <div class="subheading mb-5">
58            @x.Adres + @x.Tel
59            <a href="@x.Mail">@x.Mail</a>
60          </div>
61          <p class="lead mb-5">@x.Aciklama </p>
62          <div class="social-icons">
63            <a class="social-icon" href="https://www.linkedin.com/in/kubraaykol/" target="_blank"><i class="fab fa-linkedin-in"></i></a>
64            <a class="social-icon" href="https://github.com/kubraaykol" target="_blank"><i class="fab fa-github"></i></a>
65            <a class="social-icon" href="#"><i class="fab fa-twitter"></i></a>
66            <a class="social-icon" href="#"><i class="fab fa-facebook-f"></i></a>
67          </div>
68        </div>
69      </section>
70    }
```

E-2 DENYEYİMLERİM

The screenshot shows a resume page on a local host. The main content is as follows:

YTU-TEKNOPARK STAJYER Temmuz 2022- Eylül 2022

PROJ-E Yazılım AŞ'de E-Ticaret alanında prestijli markaların web projelerinin geliştirilmesi, projelerin canlıya alınması, canlıya olnıktan sonra karşılaşılan hatalar ve bu hataların çözümlenmesine yönelik ürünün bütün süreçlerini görme fırsatı oldu. Aynı zamanda web projesinin geliştirilmesindeki Backend, Frontend ve Proje Yönetimi alanındaki uzman geliştiricilerin deneyim ve tecrübeleri üzerine yaptığım görüşmeler ile önemle önemle yazılım alıyla ilgili araştırmalar yaptım.

TURKCELL GLOBAL BİLGİ Mayıs 2016- Temmuz 2017

Turcell Global Bilgi'nin anlaşmalı olduğu seçkin markalardan biri olan Türk Hava Yolları'nda müşteri temsilcisi olarak ilk görev oldım. Operasyon içerisinde bir ilki başarak o yıl mart ayında tüm yeteneklerden tam puan olarak birinciliği deneyimledim. Miles Smiles Elite biriminde de müşteri temsilcisi olarak görev olarak takım liderliği sürecim ile devam etmeye hazırlıklarken gümüze ilgili yaşadığım ciddi sağlık sorunu sebebi ile istifa edip kariyerime ara vermek zorunda kaldım bu dönemde.

SISTEMHOST LLC. Eylül 2012- Haziran 2013

Hosting.Domain ve Server Hizmeti veren bu şirkette mesleki ve teknik lisedeki eğitime gelen uzun dönem staj programımı tamamladım. Bir web projesinin geliştirilmesi aşamalarının yanı sıra daha çok sunucu hizmetleri, bakımı gibi teknik işleyişle alakalı bir staj deneyimim oldu.

The sidebar on the left contains the following links:

- HAKKIMDA
- DENEYİMLERİM
- EĞİTİM HAYATIM
- YETENEKLERİM
- HOBİLERİM
- SERTİFİKALARIM
- İLETİŞİM

Deneyimler kısmında verilerimizi listeleyebilmemiz için Controller tarafında yeni bir hareket tanımlaması yapmamız gereklidir. Ama MVC'de tek sayfada sadece bir tabloya ait verileri listeleyebiliriz ki bunu da "Hakkımda" tablosu ile yaptık zaten. Bunun için de "PartialViewResult"lar kullanarak sayfayı kısımlara ayırarak Model yapısının her birine ait propertieleri çağırabilelim.

E-2: ANASAYFA BACKEND ALANINDA KODLANMASI

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar includes 'Dosya', 'Düzen', 'Görünüm', 'Git', 'Proje', 'Derle', 'Hata Ayıklama', 'Test', 'Analiz', 'Araçlar', 'Uzantılar', 'Pencere', and 'Yardım'. The toolbar has icons for file operations like Open, Save, and Print. The status bar at the bottom shows '99 %' and 'Sorun bulunmadı'.

The code editor displays the 'DefaultController.cs' file under the 'MvcCv' project. The code defines a controller with methods for index, social media, and experience pages, all annotated with [AllowAnonymous].

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using MvcCv.Models.Entity;
7
8  namespace MvcCv.Controllers
9  {
10     [AllowAnonymous]
11     public class DefaultController : Controller
12     {
13         // GET: Default
14         DbCvEntities db = new DbCvEntities();
15
16         public ActionResult Index()
17         {
18             var degerler = db.TblHakkimda.ToList();
19             return View(degerler);
20         }
21
22         public PartialViewResult SosyalMedya()
23         {
24             var sosyalmedya = db.TblSosyalMedya.Where(x=>x.Durum==true).ToList();
25             return PartialView(sosyalmedya);
26         }
27
28         public PartialViewResult Deneyim()
29         {
30             var deneyimler = db.TblDeneyimler.ToList();
31             return PartialView(deneyimler);
32         }
33     }
34 }
```

E3: 29. SATIRDAN İTİBAREN ADMIN PANEL KODLAMA EKRAN GÖRÜNTÜSÜ

This screenshot continues the visual studio interface from the previous one, showing the 'DefaultController.cs' file. It shows the implementation of several partial view results for the Admin panel, including 'Yeteneklerim', 'Hobilerim', 'Sertifikalar', and 'iletisim' methods, each returning a PartialView.

```
28     }
29     public PartialViewResult Yeteneklerim()
30     {
31         var yetenekler = db.TblYetenekler.ToList();
32         return PartialView(yetenekler);
33     }
34     public PartialViewResult Hobilerim()
35     {
36         var hobiler = db.TblHobiler.ToList();
37         return PartialView(hobiler);
38     }
39     public PartialViewResult Sertifikalar()
40     {
41         var sertifikalar = db.TblSertifikalar.ToList();
42         return PartialView(sertifikalar);
43     }
44     [HttpGet]
45     public PartialViewResult iletisim()
46     {
47         return PartialView();
48     }
49     [HttpPost]
50     public PartialViewResult iletisim(Tbliletisim t)
51     {
52         t.Tarih = DateTime.Parse(DateTime.Now.ToString("yyyy-MM-dd"));
53         db.Tbliletisim.Add(t);
54         db.SaveChanges();
55         return PartialView();
56     }
57 }
58 }
```

E4-A.) NAVBAR DENEYİM SATIRI

```
18      | 0 başvuru
19      | public PartialViewResult Deneyim()
20      |
21      |     var deneyimler = db.TblDeneyimler.ToList();
22      |     return PartialView(deneyimler);
23      | }
```

Controller tarafına gelip “PartialViewResult” tasarım aracını dışarıdan erişilebilir olması için “public” olarak tanımlayıp “Deneyim” isimli view dosyamızı bu araç ile çağrırmış oluruz. ‘var’ cinsinden “deneyimler” değişkenini , “TblDeneyimler” Modelinden listeleme metoduyla beraber db nesnesi getirmiş olacak verilerimizi.

E5-b.) NAVBAR DENEYİM VERİTABANI VE ÖNYÜZ BAĞLANTISI

```
72      | <!-- Experience-->
73      | <section class="resume-section" id="experience">
74      |     @Html.Action("Deneyim")
75      | </section>
76      | <hr class="m-0" />
```

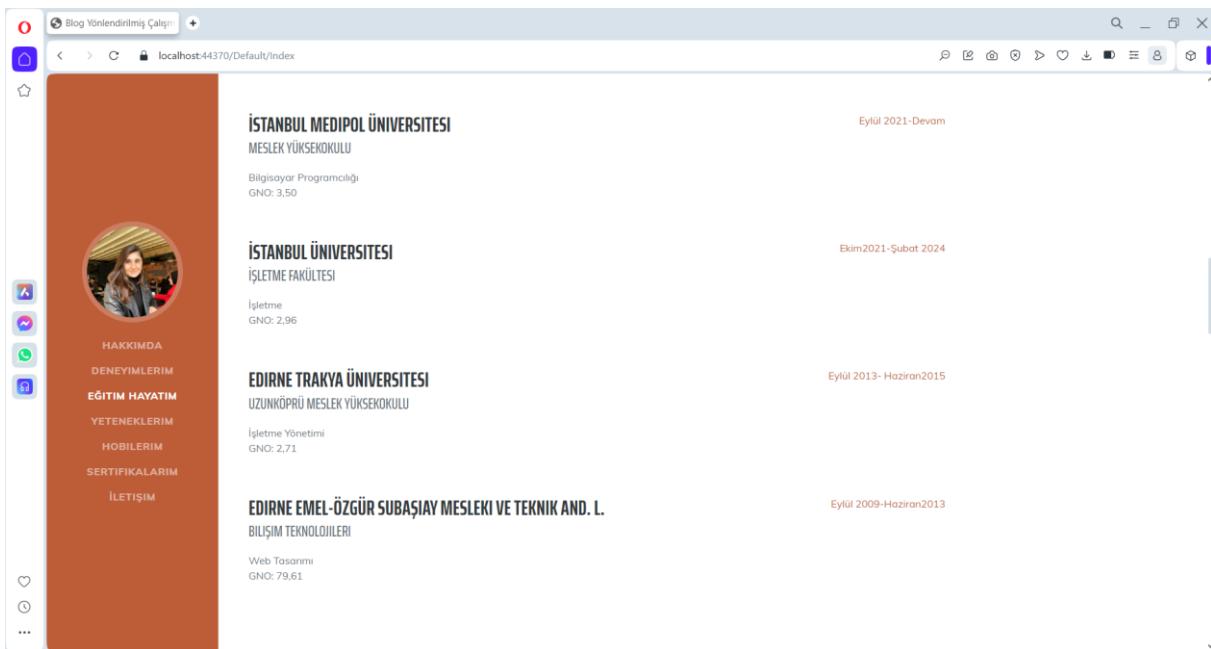
@Html nesnesinin “Action” metodunu kullanarak da ‘(Deneyim)’ verilerimiz artık sayfamızda listelenmeye hazır hale gelir. Aşağıda da “Model”ler aracılığıyla Sql’e kaydedilen ve oradan çekilmesini istediğimiz veriler görülmektedir.

E6-A.) VERİ TABANI VERİ GİRİŞLERİ

LAPTOP-A6F1CIS\S...dbo.TblDeneyimler					
	ID	Baslik	AltBaslik	Aciklama	Tarih
►	1	YTU-TeknoPark	Stajyer	PROJ-E Yazılım AŞ.'de E-Ticaret alanınd...	Temmuz 2022- Eylül 2022
	2	Turkcell Global Bilgi	TurkishAirlines	Turcell Global Bilgi'nin anlaşmalı olduğ...	Mayıs2016- Temmuz2017
*	3	SistemHost LLC.	Stajyer	Hosting,Domain ve Server Hizmeti vere...	Eylül 2012- Haziran2013
*	NULL	NULL	NULL	NULL	NULL

***Buradan sonraki tüm işlemlerimiz aynı mantık ile devam edip yalnızca kısımlardaki isimlendirmeler değişmiştir. Yalnızca “iletişim” bölümüyle alakalı yapılan bir işlem daha vardır ve bunun açıklaması hem raporda hem de sunumda detaylandırılacaktır.

E-7: EĞİTİM HAYATIM



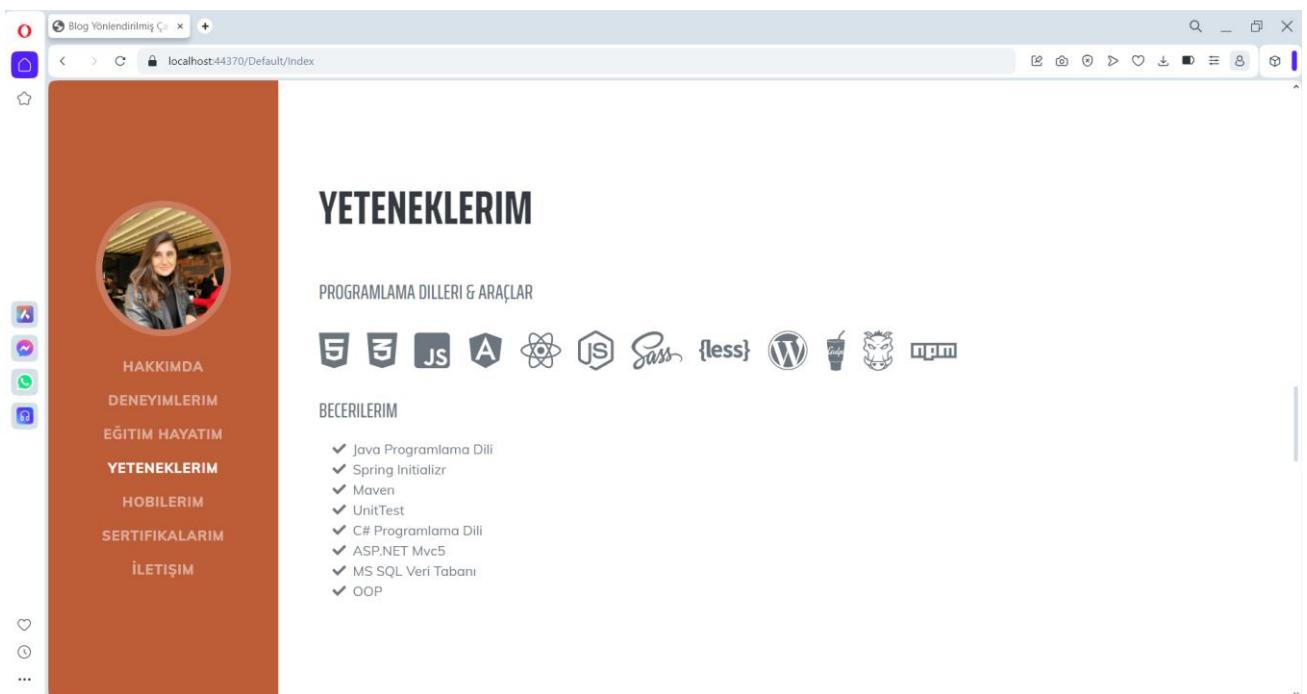
E8- NAVBAR EĞİTİMLERİM SATIRI

```
24      0 başvuru
25      public PartialViewResult Egitimlerim()
26      {
27          var egitimler = db.TblEgitim.ToList();
28          return PartialView(egitimler);
}
```

E9-NAVBAR EĞİTİMLERİM VERİTABANI VE ÖNYÜZ BAĞLANTISI

```
78      <!-- Education-->
79      <section class="resume-section" id="education">
80          @Html.Action("Egitimlerim")
81      </section>
82      <hr class="m-0" />
```

E-10: YETENEKLERİM



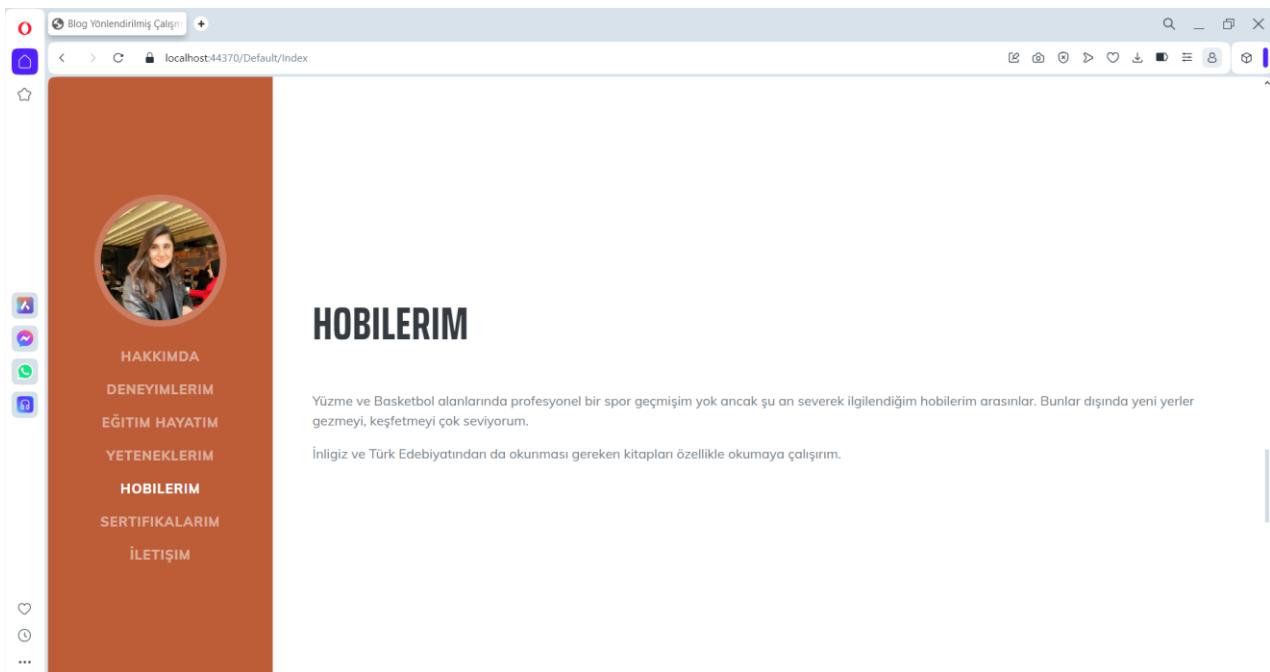
E11- NAVBAR YETENEKLERİM SATIRI

```
29      0 başvuru
30      public PartialViewResult Yeteneklerim()
31      {
32          var yetenekler = db.TblYetenekler.ToList();
33      }
```

E12-NAVBAR YETENEKLERİM VERİTABANI VE ÖNYÜZ BAĞLANTISI

```
83      <!-- Skills-->
84      <section class="resume-section" id="skills">
85          @Html.Action("Yeteneklerim")
86      </section>
87      <hr class="m-0" />
```

E-13 HOBİLERİM



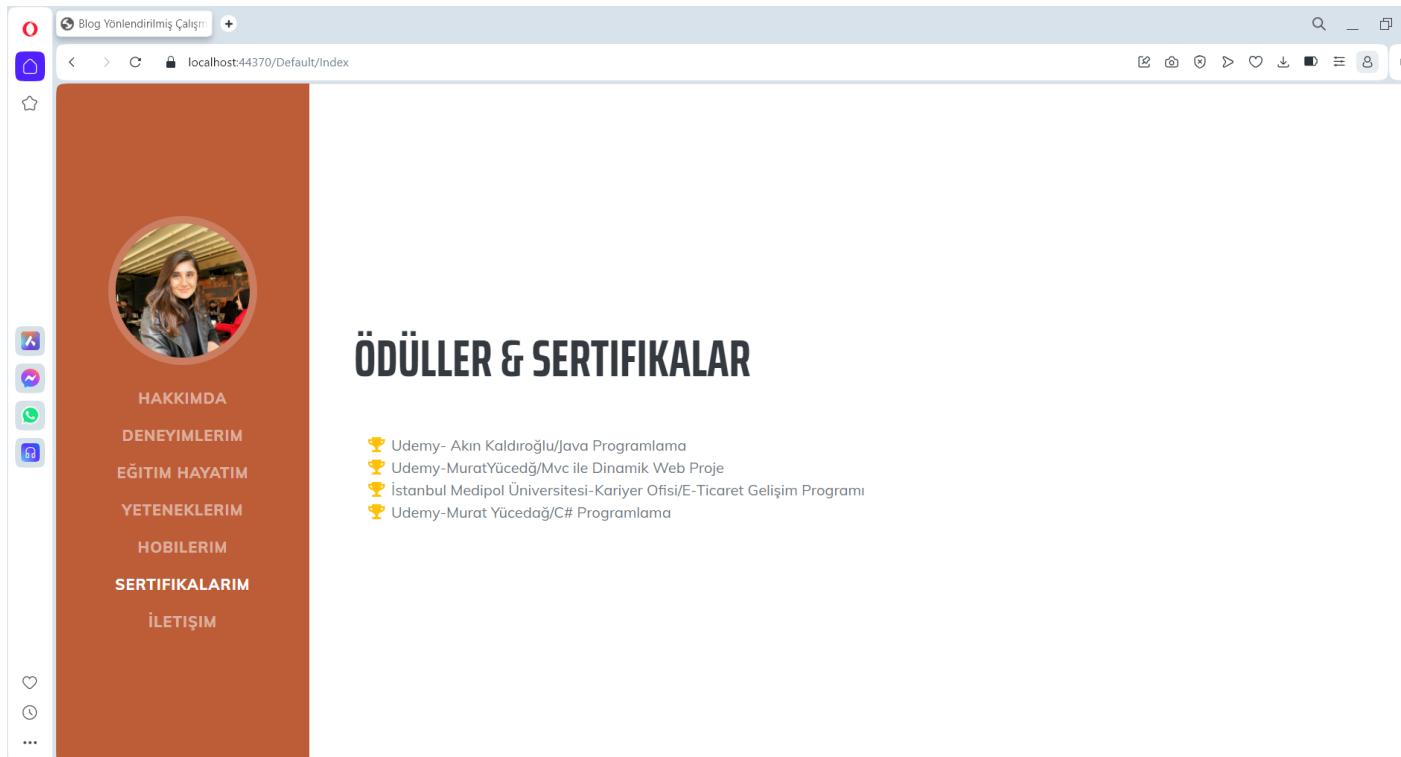
E-14: NAVBAR HOBİLERİM SATIRI

```
34     0 başvuru
35     public PartialViewResult Hobilerim()
36     {
37         var hobiler = db.TblHobiler.ToList();
38     }
```

E-15: NAVBAR EĞİTİMLERİM VERİTABANI VE ÖNYÜZ BAĞLANTISI

```
88     <!-- Interests-->
89     <section class="resume-section" id="interests">
90         @Html.Action("Hobilerim")
91     </section>
92     <hr class="m-0" />
```

E-16: SERTİFİKALARIM EKRANI



E-17: NAVBAR SERTİFİKALARIM SATIRI

```
39      0 başvuru
40      public PartialViewResult Sertifikalar()
41      {
42          var sertifikalar = db.TblSertifikalar.ToList();
43          return PartialView(sertifikalar);
}
```

E-18: NAVBAR SERTİFİKALARIM VERİTABANI VE ÖNYÜZ BAĞLANTISI

```
93      <!-- Awards-->
94      <section class="resume-section" id="awards">
95          @Html.Action("Sertifikalar")
96      </section>
97      <hr class="m-0" />
```

E-19: İLETİŞİM EKRANI

BANA ULAŞIN

Ad Soyad

Kübra Aykol

Mail

deneme1@mail.com

Konu

Yönlendirilmiş Çalışma Vize

Mesaj

Vize Sunum Raporu İçerik Denemesi

Mesaj Gönder

E-20: İLETİŞİM EKRANI BAĞINTI OLUŞTURMA KODLARI

```
44 [HttpGet]  
45 0 başvuru  
46 public PartialViewResult iletisim()  
47 {  
48     return PartialView();  
49 }  
50 [HttpPost]  
51 0 başvuru  
52 public PartialViewResult iletisim(Tbliletisim t)  
53 {  
54     db.Tbliletisim.Add(t);  
55     db.SaveChanges();  
56     return PartialView();  
57 }
```

E-21: İLETİŞİM EKRANI BAĞINTI OLUŞTURMA KODLARI

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using MvcCv.Models.Entity;
7  using MvcCv.Repositories;
8
9  namespace MvcCv.Controllers
10 {
11     public class IletisimController : Controller
12     {
13         // GET: Iletisim
14         GenericRepository<Tbliletisim>
15         repo= new GenericRepository<Tbliletisim>();
16         public ActionResult Index()
17         {
18             var mesajlar = repo.List();
19             return View(mesajlar);
20         }
21     }
22 }
```

Kodların çalışma mantıklarını izah edecek olursak, `HttpGet` ve `HttpPost` isimli attribute içerisinde “iletişim” adından iki adet aynı metodу tanımladık. `HttpGet`’e bağlı “iletişim” metodу sayfa tarayıcıda ilk açıldığında düzen ve akış içerisinde tasarıyı tamamlayarak görünüm ve işlev sağlar. `HttpPost`’a bağlı ikinci “iletişim” metodу ile de yine tarayıcı üzerindeki düzen bozulmadan aynı şema içerisinde bulunan kısımdan girilen verinin veri tabanına kaydedilmesi işlevini sağlayan bir yöntem yaratmış oluruz.

Sadece bir tanesini kullanmış olsaydık da aynı veri iki defa kaydedilmiş olunacaktı ve bu da fazla, gereksiz veri tutarak performans düşüklüğü yaratırıldı projede.

`HttpPost` attribute’nda iletişim metodumuza yordamı tanımlamak adına verinin kaydedileceği tablo ismini ve “t” isminde parametre aracılığı ile göndereceğimizi tanımlamış olduk. “`Tbliletisim`” sınıfından “t” parametresi türetmiş olduk.

Db. İle Model aracılığı ile ulaştığımız veri tabanımız anlamına gelmekte. Ve `Add(t)` şeklinde yaptığımız tanımlama ile de “t” parametresi ile gönderilen değerin eklenmesidir. Devam eden kod satırında `db.SaveChanges();` komutu ile değişikler kaydedilir ve son olarak “return” ile geriye tekrardan bir değer dönülür.

E-22: İLETİŞİM EKRANI TASARIM KODLARI

', is highlighted with a light blue background."/>

```
iletisim.cshtml ✘ ✗ DefaultController.cs Index.cshtml
1 <div class="resume-section-content">
2   <form method="post">
3     <h2 class="mb-5">Bana Ulaşın</h2>
4     <label>Ad Soyad</label>
5     <input type="text" class="form-control" name="AdSoyad" />
6
7     <label>Mail</label>
8     <input type="text" class="form-control" name="Mail"/>
9
10    <label>Konu</label>
11    <input type="text" class="form-control" name="Konu"/>
12
13    <label>Mesaj</label>
14    <input type="text" class="form-control" name="Mesaj" />
15    <br />
16    <button class="btn btn-info">Mesaj Gönder</button>
17
18  </form>
19 </div>
```

.cshtml uzantılı sayfada yazılı tasarım kodlarında ise <form> etiketi içerisinde yazılan kodların özellikle de <form method="post"> şeklinde kısım “HttpPost” attribute’ı ile bağlantılı olarak çalıştırılacak kısım olduğu anlamına gelmektedir. Özette sayfada butona bastıktan sonra gerçekleşen post işlemidir.

E-23: İLETİŞİM EKRANI VERİ TABANINA KAYDOLAN MESAJ

	ID	AdSoyad	Mail	Konu	Mesaj	Tarih
▶	1	Mehmet Ay...	NULL	NULL	NULL	NULL
	2	Kübra Aykol	NULL	NULL	NULL	NULL
	3	Kübra Aykol	deneme@m...	Yönlendiril...	Vize Sunum ...	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Execute SQL Ctrl+R

Cut Ctrl+X

Copy Ctrl+C

Paste Ctrl+V

Delete Del

Pane ▾

Clear Results

Properties Alt+Enter

Girilen verinin yansıtılması aşamasında sırasıyla uygulanan adımlardan diğerini güncelleme yapılarak verinin kaydolduğunu görüntülemektir.

E-24: İLETİŞİM EKRANI VERİ TABANINA KAYDOLAN MESAJ

LAPTOP-A6F1CISJ\SQL - dbo.Tbliletisim						
	ID	AdSoyad	Mail	Konu	Mesaj	Tarih
	1	Mehmet Ay...	NULL	NULL	NULL	NULL
	2	Kübra Aykol	NULL	NULL	NULL	NULL
	3	Kübra Aykol	deneme@m...	Yönlendiril...	Vize Sunum ...	NULL
▶	4	Kübra Aykol	deneme1@...	Yönlendiril...	Vize Sunum ...	2024-03-20
*	NULL	NULL	NULL	NULL	NULL	NULL

Web sayfası üzerinden gönderilen mesaj içeriğinin veri tabanına kaydolduğunu “Execute SQL” seçeneğiyle dördüncü sırada görmüş oluyoruz.

FİNAL

ADMİN PANELİNİN OLUŞTURULMASI

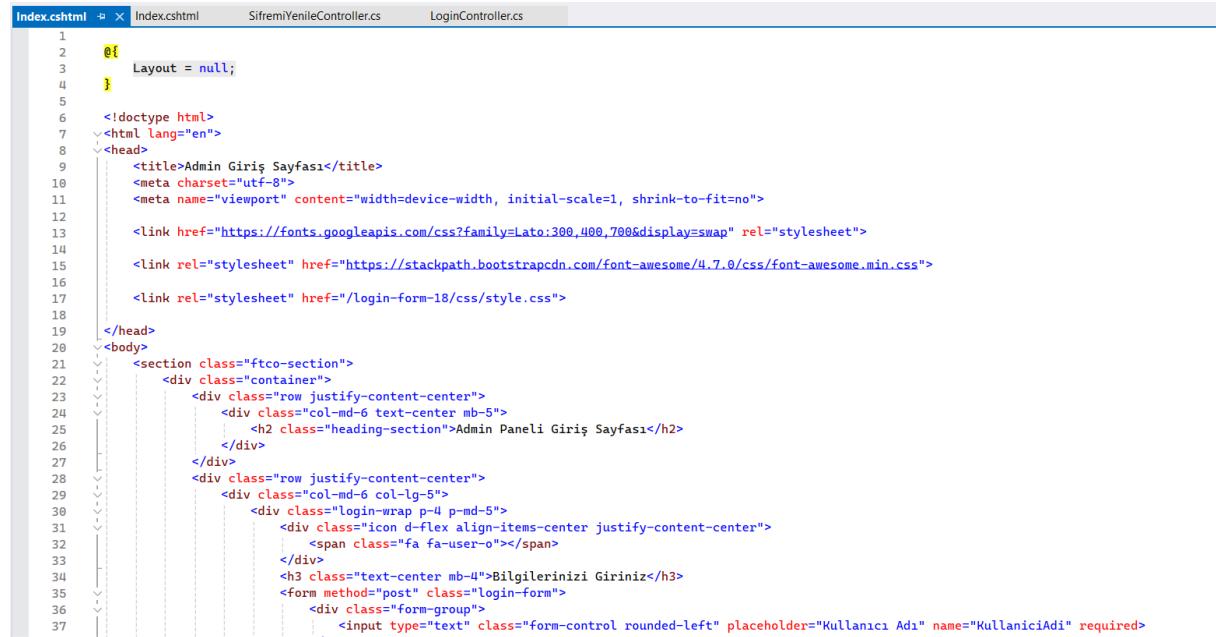
Vize sunumunda daha çok web ön yüz ve arka yüz tasarımları ağırlıklı ilerleyen projelerde “Admin Paneli” ile gerekli bütün Giriş-Çıkış-Şifremi Unuttum, Validasyon ve Authorized işlemleri tamamlanarak canlıya alınacak hale getirilerek halini almıştır.

E-25: ADMİN PANELİ GİRİŞ SAYFASI

The screenshot shows a web browser window titled "Admin Giriş Sayfası". The URL in the address bar is "localhost:44370/Login/Index". The page content is titled "Admin Paneli Giriş Sayfası". It features a central login form with a purple header containing a user icon and the text "Bilgilerinizi Giriniz". Below this are two input fields: "Kullanıcı Adı" (Username) and "Şifre" (Password). To the right of the password field is a link "Şifremi Yenile" (Reset Password). At the bottom left of the form is a checkbox "Beni Hatırla" (Remember Me). A large purple button at the bottom right of the form is labeled "Giriş Yap" (Log In).

Öncelikle Web Önyüzde panele giriş sağlanacak sayfa tasarımlı ve kullanıcı adı, şifre gibi bilgilerin ise veri tabanından çekilerek güvenli bir şekilde işlem yapılacağı sayfa tamamlandı.

E-26: GİRİŞ EKRANI TASARIM KODLAMASI(1)



```
Index.cshtml  Index.cshtml  SifremiYenileController.cs  LoginController.cs
1  @{
2      Layout = null;
3  }
4
5  <!doctype html>
6  <html lang="en">
7      <head>
8          <title>Admin Giriş Sayfası</title>
9          <meta charset="utf-8">
10         <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
11
12         <link href="https://fonts.googleapis.com/css?family=Lato:300,400,700&display=swap" rel="stylesheet">
13
14         <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
15
16         <link rel="stylesheet" href="/login-form-18/css/style.css">
17
18     </head>
19
20     <body>
21         <section class="ftco-section">
22             <div class="container">
23                 <div class="row justify-content-center">
24                     <div class="col-md-6 text-center mb-5">
25                         <h2 class="heading-section">Admin Paneli Giriş Sayfası</h2>
26
27                     </div>
28                 </div>
29                 <div class="row justify-content-center">
30                     <div class="col-md-6 col-lg-5">
31                         <div class="login-wrap p-4 p-md-5">
32                             <div class="icon d-flex align-items-center justify-content-center">
33                                 <span class="fa fa-user-o"></span>
34                             </div>
35
36                             <h3 class="text-center mb-4">Bilgilerinizi Giriniz</h3>
37                             <form method="post" class="login-form">
38                                 <div class="form-group">
39                                     <input type="text" class="form-control rounded-left" placeholder="Kullanıcı Adı" name="KullaniciAdi" required>
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
```

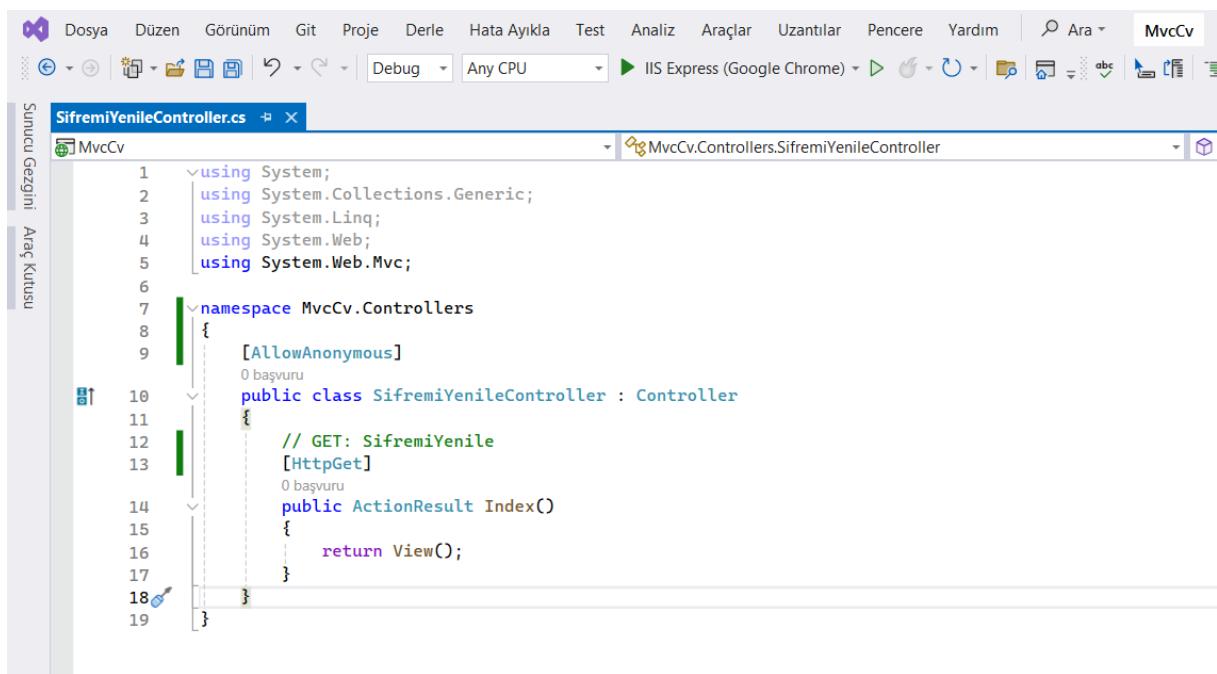
E-27: GİRİŞ EKRANI TASARIM KODLAMASI(2)



```
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
    <div class="form-group">
        <input type="text" class="form-control rounded-left" placeholder="Kullanıcı Adı" name="KullaniciAdi" required>
    </div>
    <div class="form-group d-flex">
        <input type="password" name="Sifre" class="form-control rounded-left" placeholder="Şifre" required>
    </div>
    <div class="form-group d-md-flex">
        <div class="w-50">
            <label class="checkbox-wrap checkbox-primary">
                Beni Hatırla
                <input type="checkbox" checked>
                <span class="checkmark"></span>
            </label>
        </div>
        <div class="w-50 text-md-right">
            <a href="/SifremiYenile/Index/">Şifremi Yenile</a>
        </div>
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-primary rounded submit p-3 px-5">Giriş Yap</button>
    </div>
</div>
</div>
</div>
</div>
</div>
```

Tasarım aşamasında gereken HTML ve CSS kodları yukarıda iki parça olarak toplamda 70 satırda tamamlandı. Aynı tasarım üzerinden “Şifremi Unuttum” giriş ekranı da tasarlandı ve gereken işlevler kazandırıldı.

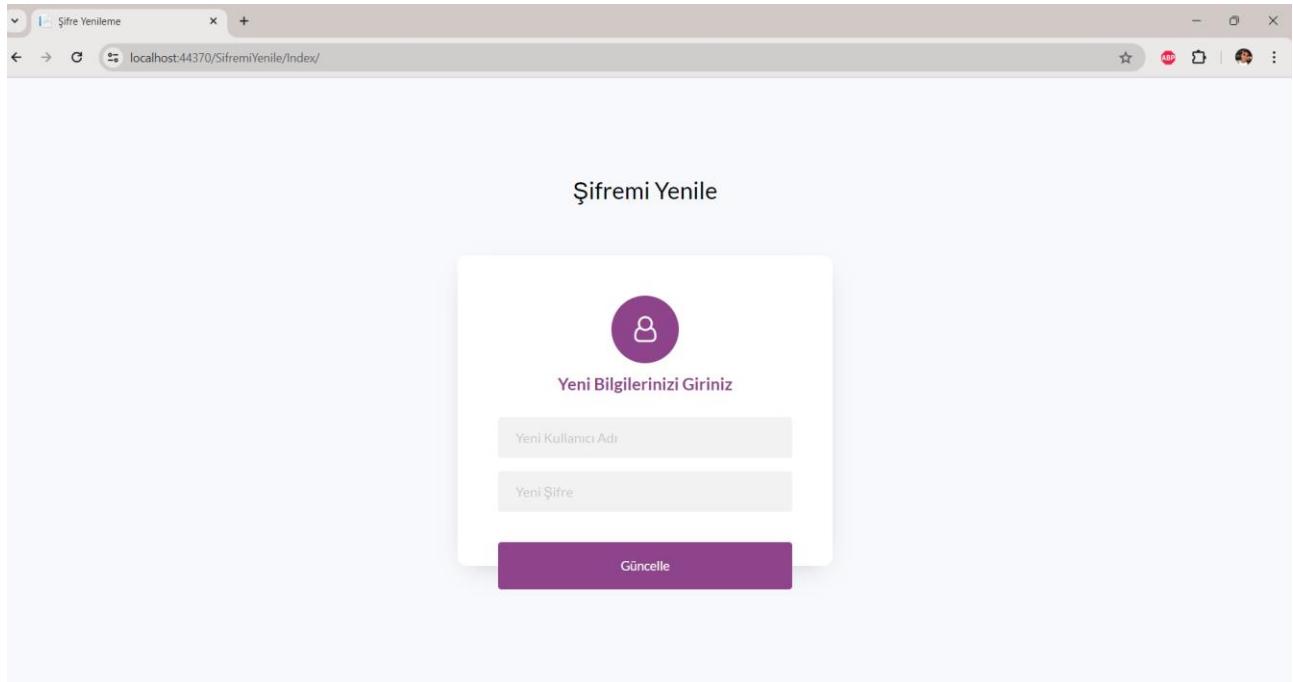
E-28: GİRİŞ EKRANI METHOD KODLAMASI



The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "MvcCv". The menu bar includes "Dosya", "Düzen", "Görünüm", "Git", "Proje", "Derle", "Hata Ayıkla", "Test", "Analiz", "Araçlar", "Uzantılar", "Pencere", "Yardım". The toolbar has icons for file operations like Open, Save, Print, and a search icon. The status bar shows "Debug Any CPU IIS Express (Google Chrome)". The code editor window is titled "SifremiYenileController.cs" and contains the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6
7  namespace MvcCv.Controllers
8  {
9      [AllowAnonymous]
10     public class SifremiYenileController : Controller
11     {
12         // GET: SifremiYenile
13         [HttpGet]
14         public ActionResult Index()
15         {
16             return View();
17         }
18     }
19 }
```

E-30: ŞİFRE YENİLEME



E-31: ŞİFRE YENİLEME TASARIM KODLAMASI(1)

The screenshot shows the Visual Studio code editor with the file 'Index.cshtml' open. The code is a .NET MVC view for a password reset page. It includes meta tags for character encoding and viewport, links to Google fonts and Bootstrap CSS, and a main section with a heading and a login form.

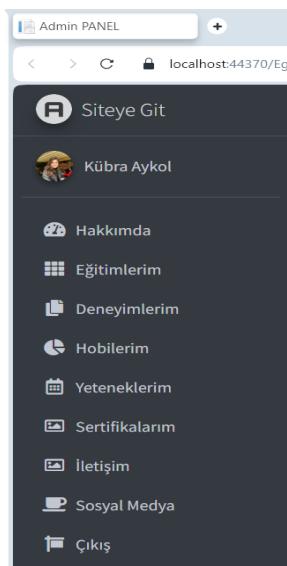
```
1<!DOCTYPE html>
2<html lang="en">
3<head>
4    <title>Şifre Yenileme</title>
5    <meta charset="utf-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7    <link href="https://fonts.googleapis.com/css?family=Lato:300,400,700&display=swap" rel="stylesheet">
8    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
9
10   <link rel="stylesheet" href="/login-form-18/css/style.css">
11</head>
12<body>
13    <section class="ftco-section">
14        <div class="container">
15            <div class="row justify-content-center">
16                <div class="col-md-6 text-center mb-5">
17                    <h2 class="heading-section">Şifremi Yenile</h2>
18                </div>
19            </div>
20            <div class="row justify-content-center">
21                <div class="col-md-6 col-lg-5">
22                    <div class="login-wrap p-4 p-md-5">
23                        <div class="d-flex align-items-center justify-content-center">
24                            <span class="fa fa-user-o"></span>
25                        </div>
26                    </div>
27                </div>
28            </div>
29        </div>
30    </section>
31</body>
32
```

E-32: ŞİFRE YENİLEME TASARIM KODLAMASI(2)

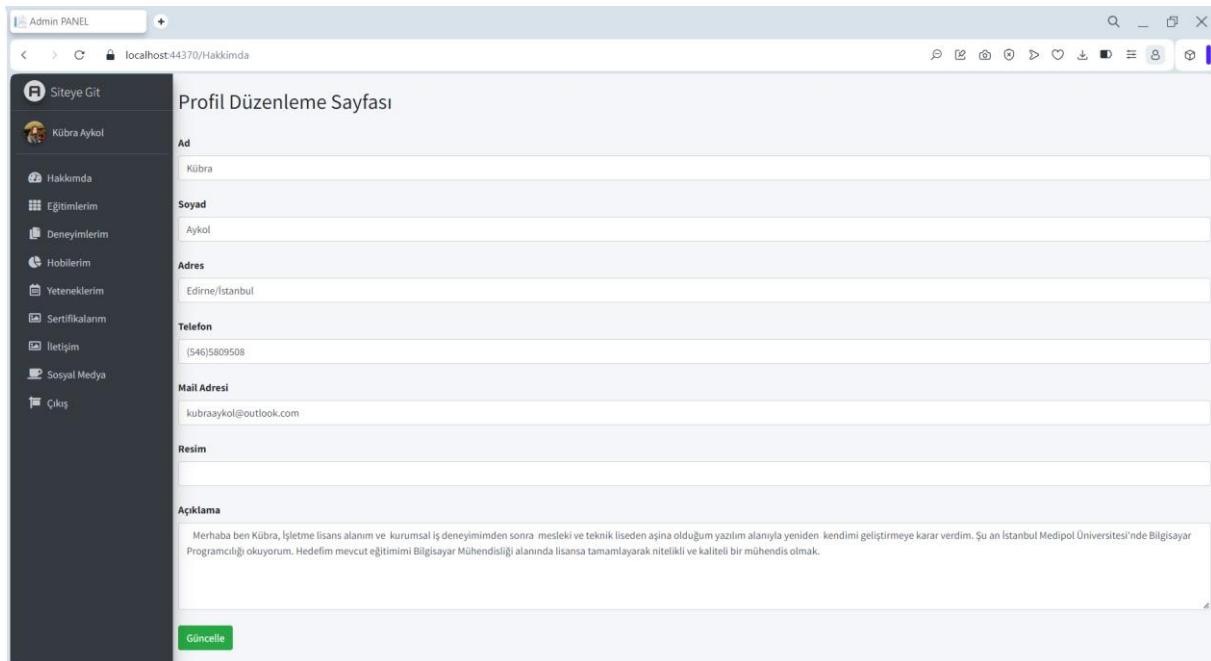
The screenshot continues the Visual Studio code editor showing the 'Index.cshtml' file. It shows a form for entering new user information, including fields for username and password, and a submit button. The page also includes a script section at the bottom.

```
33<h3 class="text-center mb-4">Yeni Bilgilerinizi Giriniz</h3>
34<form method="post" class="login-form">
35    <div class="form-group">
36        <input type="text" class="form-control rounded-left" placeholder="Yeni Kullanıcı Adı" name="KullaniciAdi" required>
37    </div>
38    <div class="form-group d-flex">
39        <input type="password" name="Sifre" class="form-control rounded-left" placeholder="Yeni Şifre" required>
40    </div>
41    <div class="form-group">
42        <button type="submit" class="btn btn-primary rounded submit p-3 px-5">Güncelle</button>
43    </div>
44</form>
45</div>
46</div>
47</div>
48</div>
49</section>
50<script src="/login-form-18/js/jquery.min.js"></script>
51<script src="/login-form-18/js/popper.js"></script>
52<script src="/login-form-18/js/bootstrap.min.js"></script>
53<script src="/login-form-18/js/main.js"></script>
54</body>
55</html>
56
57
58
```

E-33 : ADMİN PANELİNE GİRİŞ YAPILMASI NAVBAR EKRAN GÖRÜNTÜSÜ



E-33: ADMİN PANELİNE GİRİŞ YAPILMASI TAM EKRAN GÖRÜNTÜSÜ



Gerekli şifre kombinasyonu sağlandıktan sonra navbarda bulunan ilk birim olan “Hakkımda” alanının “Profil Düzenleme Sayfasına” yönlendirilmiş olunuyor.

Controller’lar üzerinden Repository çağrıma işlemi yapılmaktadır. “repo” isminde bir nesne türeterek de işlemi kod olarak tamamlamış oluyoruz.

Index tarafından verileri listelemek için bir değişken tanımlayıp “repo.” Yazdıktan sonra GenericRepository sınıfına ait olan bütün kullanılabilir metodu görebiliriz ve buradan ihtiyacımız olan “repo.List()” metoduyla listeleme işlemimizi tamamlamaya başlayacağız.

Buradaki önemli nokta GenericRepository sınıfına ait olan bütün methodları çağrımadan kullanabiliyoruz. Sebebi ise “x Repository” üzerinden tanımladığımız “using MvcCv Repository” seçeneğini aktifleştirerek GenericRepository sınıfındaki değerleri OOP, Nesne Yönelimli Programlama’nın temel prensiplerinden biri olan “inheritance” miras, kalıtım alarak kullanmış olmamızdır. Son olarak 17. satırdaki “return” ifadesi ile de geri dönüş yapılacak değerlerin kod ile talebini oluşturmuş oluyoruz.

“Index” üzerine sağ tıklayarak “view” ekleyerek AdminLayout ekliyoruz. Bunun amacı web sayfasından sık kullanılan merkezi/ortak öğeleri tek bir yapıda tanımlamayı ve her yeni sayfa ihtiyacı ortaya çıktığında ortak yapıların tekrar tekrar kodlamamaktır. Örneğin aşağıdaki Admin Paneli ve bu paneldeki her bir bölüm için tekrar tekrar kodlama yapmadan zaman ve performans tasarrufu sağlamaktadır.

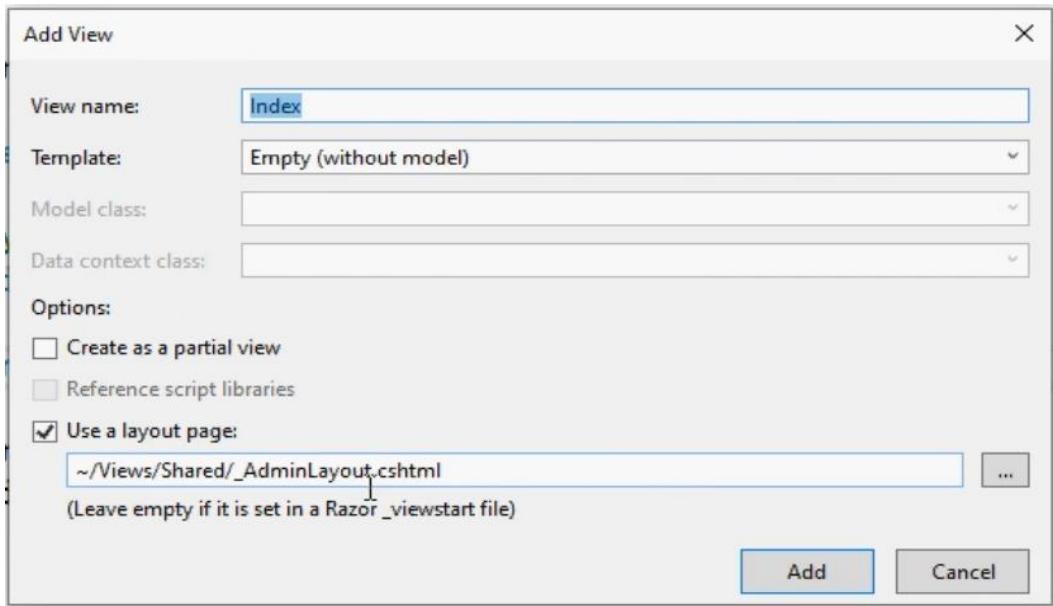
“index.cshtml” isimli görünüm oluşmuş oldu. Burada öncelikle tarayıcıda çalıştırduğumda boş bir sayfa gelmiş oluyor. Kontrolden sonra hata, eksiklik vs. olmaması durumunda listeleme işlemimiz için gerekli kodlamaları yapacağız.

Admin panelimiz üzerinden yapılması gereken talimatlar için başladığımız sıralı işlemlerde tarayıcıda çalıştırıldığında hata veya herhangi bir eksiklik olmadığı görülmektedir.

Buradan sonraki yapacağımız kodlamalar sayesinde veri tabanımızdaki veriler üzerinden ya da önyüzden yapacağımız işlemlerin veri tabanımıza düşmesiyle ilgili işlemler veya admin paneli üzerinden yapılacak işlemlerin web sayfamız üzerinde alacağı görünümleri alması için yapacağımız kodlama işlemlerine geçebiliriz.

Özetle tüm bunları “Controller” üzerinden açtığımız “View”de yapacağımız kodlamalarla yapmış olacağız.

E-34: INDEX İŞLEMLERİ İÇİN KULLANILACAK VIEW SAYFASI EKLENMESİ



“@using MvcCv.Models.Entity” ve “@model List<TblDeneyimler>” yapacağımız bu işlemler için gereken en önemli kodlardan ikisi.

Sırasıyla tablo oluşturup “ID” ile veri tabanından çekeceğimiz işlemler ve tasarım için kodlamalarımıza başlıyoruz ve 13.satırda itibaren yaptığımız “@foreach” döngüsü ile veri listeleme işlemi için gereken talimatların başlangıcı yapıyoruz.

Bu kısımdan itibaren ise “Yeni Deneyim Ekle” butonun çalışmasına yönelik işlemler “ActionResult” fonksiyonu ile kodlamaya başlanmıştır.

Controller bölümünden “ActionResult”, [HttpGet] ve [HttpPost] methodları aracılığı ile gereken komutlar tamamlanmıştır. “View” eklenerek butonun yönlendirileceği sayfa da eklenmiştir. Devamında ise Index.cshtml tarafından buton yönlendirme komutu kullanılarak buton çalıştırılmıştır.

E-35: EĞİTİM LİSTESİ EKRAN GÖRÜNTÜSÜ

ID	Okul Adı	Fakülte	Bölüm	Tarih	GNO	Sil	Düzenle
1	İstanbul Medipol Üniversitesi	Meslek Yüksekokulu	Bilgisayar Programcılığı	Eylül 2021-Devam	3,50		
2	İstanbul Üniversitesi	İşletme Fakültesi	İşletme	Ekim 2021-Şubat 2024	2,96		
3	Edirne Trakya Üniversitesi	Uzunköprü Meslek Yüksekokulu	İşletme Yönetimi	Eylül 2013-Haziran 2015	2,71		
4	Edirne Emel-Özgür Subaşıy Mesleki ve Teknik An. L.	Bilişim Teknolojileri Alanı	Web Tasarımı Dalı	Eylül 2009-Haziran 2013	79,61		

Eğitim ekleme, düzenleme ve silme işlemlerinin yapılabacağı sayfa düzenlenmesi için gereken kodlamalar aşağıdaki görseller ile desteklenmiştir. “Yeni Eğitim Bilgisi Ekle” butonu ile de sayfa üzerine yeni bilgi eklenebilecek ve mevcut bilgiler silinebilir ya da düzenlenebilecektir.

E-36: YENİ EĞİTİM BİLGİSİ EKLEME EKRAN GÖRÜNTÜSÜ

Yeni Eğitim Bilgisi Ekleme Sayfası

Okul Adı:

Fakülte:

Bölüm:

Tarih:

GNO:

Kaydet

E-37: EĞİTİM BİLGİSİ EKLEME CONTROLLER EKRAN GÖRÜNTÜSÜ

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "EgitimController.cs" and "Index.cshtml". The main area displays the C# code for the EgitimController class:

```
1  using MvcCv.Models.Entity;
2  using MvcCv.Repositories;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Web;
7  using System.Web.Mvc;
8
9  namespace MvcCv.Controllers
10 {
11     public class EgitimController : Controller
12     {
13         // GET: Egitim
14         GenericRepository<TblEgitim> repo = new
15             GenericRepository<TblEgitim>();
16
17         public ActionResult Index()
18         {
19             var egitim = repo.List();
20             return View(egitim);
21         }
22
23         [HttpPost]
24         public ActionResult EgitimEkle()
25         {
26             return View();
27         }
28
29         [HttpPost]
30         public ActionResult EgitimEkle(TblEgitim p)
31         {
32             if (!ModelState.IsValid)
33             {
34                 return View("EgitimEkle");
35             }
36             repo.Tadd(p);
37             return RedirectToAction("Index");
38         }
39
40         public ActionResult EgitimSil(int id)
41         {
42             var egitim=repo.Find(x=> x.ID== id);
43             repo.TDelete(egitim);
44             return RedirectToAction("Index");
45         }
46
47         [HttpGet]
48         public ActionResult EgitimDuzenle(int id)
49         {
50             var egitim = repo.Find(x => x.ID == id);
51             return View(egitim);
52         }
53
54         [HttpPost]
55         public ActionResult EgitimDuzenle(TblEgitim t)
56         {
57
58             var egitim = repo.Find(x => x.ID == t.ID);
59             egitim.Baslik = t.Baslik;
60             egitim.AltBaslik1 = t.AltBaslik1;
61             egitim.AltBaslik2 = t.AltBaslik2;
62             egitim.Tarih=t.Tarih;
63             egitim.GNO = t.GNO;
64             repo.TUpdate(egitim);
65             return RedirectToAction("Index");
66         }
67     }
68 }
```

E-38: EĞİTİM BİLGİSİ EKLEME CONTROLLER EKRAN GÖRÜNTÜSÜ (2)

The screenshot shows the continuation of the Microsoft Visual Studio IDE interface. The title bar still says "EgitimController.cs" and "Index.cshtml". The main area displays the remaining C# code for the EgitimController class:

```
32         }
33         repo.Tadd(p);
34         return RedirectToAction("Index");
35     }
36
37     public ActionResult EgitimSil(int id)
38     {
39         var egitim=repo.Find(x=> x.ID== id);
40         repo.TDelete(egitim);
41         return RedirectToAction("Index");
42     }
43
44     [HttpGet]
45     public ActionResult EgitimDuzenle(int id)
46     {
47         var egitim = repo.Find(x => x.ID == id);
48         return View(egitim);
49     }
50
51     [HttpPost]
52     public ActionResult EgitimDuzenle(TblEgitim t)
53     {
54
55         var egitim = repo.Find(x => x.ID == t.ID);
56         egitim.Baslik = t.Baslik;
57         egitim.AltBaslik1 = t.AltBaslik1;
58         egitim.AltBaslik2 = t.AltBaslik2;
59         egitim.Tarih=t.Tarih;
60         egitim.GNO = t.GNO;
61         repo.TUpdate(egitim);
62         return RedirectToAction("Index");
63     }
64 }
```

E-39: EĞİTİM DÜZENLEME VERİTABANI BAĞINTISI EKRAN GÖRÜNTÜSÜ

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "MvcCv". The left sidebar has "Sunucu Gezgini" and "Araç Kütüsü". The main editor window displays the code for "EgitimDuzenle.cshtml". The code is as follows:

```
1 @model MvcCv.Models.Entity.TblEgitim
2 @{
3     ViewBag.Title = "EgitimDuzenle";
4     Layout = "~/Views/Shared/_AdminLayout.cshtml";
5 }
6
7 <br />
8 <h4>Eğitim Bilgisi Düzenleme Sayfası</h4>
9 <br />
10 <form method="post">
11     @Html.Label("Okul Adı")
12     @Html.TextBoxFor(x => x.Baslik, new { @class = "form-control" })
13     @Html.ValidationMessageFor(x => x.Baslik, "", new { style = "color:red" })
14     <br />
15     @Html.Label("Fakülte")
16     @Html.TextBoxFor(x => x.AltBaslik1, new { @class = "form-control" })
17     @Html.ValidationMessageFor(x => x.AltBaslik1, "", new { style = "color:red" })
18     <br />
19     @Html.Label("Bölüm")
20     @Html.TextBoxFor(x => x.AltBaslik2, new { @class = "form-control" })
21     @Html.ValidationMessageFor(x => x.AltBaslik2, "", new { style = "color:red" })
22     <br />
23     @Html.Label("Tarih")
24     @Html.TextBoxFor(x => x.Tarih, new { @class = "form-control" })
25     @Html.ValidationMessageFor(x => x.Tarih, "", new { style = "color:red" })
26     <br />
27     @Html.Label("GNO")
28     @Html.TextBoxFor(x => x.GNO, new { @class = "form-control" })
29     @Html.ValidationMessageFor(x => x.GNO, "", new { style = "color:blue" })
30     <br />
31     <button class="btn btn-info">Kaydet</button>
32 </form>
```

The status bar at the bottom shows "100 %", "Sorun bulunamadı", "Sat: 1 Krkt: 1 BŞL CRLF".

E-40: EĞİTİM EKLEME VERİTABANI BAĞINTISI EKRAN GÖRÜNTÜSÜ

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "MvcCv". The left sidebar has "Sunucu Gezgini" and "Araç Kütüsü". The main editor window displays the code for "EgitimEkle.cshtml". The code is as follows:

```
1 @model MvcCv.Models.Entity.TblEgitim
2 @{
3     ViewBag.Title = "EgitimEkle";
4     Layout = "~/Views/Shared/_AdminLayout.cshtml";
5 }
6
7 <br />
8 <h4>Yeni Eğitim Bilgisi Ekleme Sayfası</h4>
9 <br />
10 <form method="post">
11     @Html.Label("Okul Adı")
12     @Html.TextBoxFor(x => x.Baslik, new { @class = "form-control" })
13     @Html.ValidationMessageFor(x => x.Baslik, "", new { style = "color:red" })
14     <br />
15     @Html.Label("Fakülte")
16     @Html.TextBoxFor(x => x.AltBaslik1, new { @class = "form-control" })
17     @Html.ValidationMessageFor(x => x.AltBaslik1, "", new { style = "color:red" })
18     <br />
19     @Html.Label("Bölüm")
20     @Html.TextBoxFor(x => x.AltBaslik2, new { @class = "form-control" })
21     @Html.ValidationMessageFor(x => x.AltBaslik2, "", new { style = "color:red" })
22     <br />
23     @Html.Label("Tarih")
24     @Html.TextBoxFor(x => x.Tarih, new { @class = "form-control" })
25     @Html.ValidationMessageFor(x => x.Tarih, "", new { style = "color:red" })
26     <br />
27     @Html.Label("GNO")
28     @Html.TextBoxFor(x => x.GNO, new { @class = "form-control" })
29     @Html.ValidationMessageFor(x => x.GNO, "", new { style = "color:blue" })
30     <br />
31     <button class="btn btn-info">Kaydet</button>
32 </form>
```

The status bar at the bottom shows "100 %", "Sorun bulunamadı", "Sat: 1 Krkt: 1 BŞL CRLF".

E-41: EĞİTİM EKLEME VE DÜZENLEME TASARIMI EKRAN GÖRÜNTÜSÜ (1)

```
1  @using MvcCv.Models.Entity
2  @model List<TblEgitim>
3  @{
4      ViewBag.Title = "Index";
5      Layout = "~/Views/Shared/_AdminLayout.cshtml";
6  }
7  @{
8      ViewBag.Title = "Index";
9      Layout = "~/Views/Shared/_AdminLayout.cshtml";
10 }
11
12 <br />
13 <h2>Eğitim Listesi</h2>
14 <table class="table table-bordered">
15     <tr>
16         <th>ID</th>
17         <th>Okul Adı</th>
18         <th>Fakülte</th>
19         <th>Bölüm</th>
20         <th>Tarih</th>
21         <th>GNO</th>
22         <th>Sil</th>
23         <th>Düzenle</th>
24     </tr>
25     <foreach (var item in Model)>
26     {
27         <tr>
28             <td>@item.ID</td>
29             <td>@item.Baslik</td>
30             <td>@item.AltBaslik1</td>
31             <td>@item.AltBaslik2</td>
32             <td>@item.Tarih</td>
33             <td>@item.GNO</td>
34             <td>
35                 <a class="btn btn-danger btn-sm" href="/Egitim/EgitimSil/@item.ID">
36                     <i class="fas fa-trash">
37                 </i>
38             Sil
39             </a>
40             <a class="btn btn-info btn-sm" href="/Egitim/EgitimDuzenle/@item.ID">
41                 <i class="fas fa-pencil-alt">
42             </i>
43             Düzenle
44             </a>
45         </td>
46     </tr>
47 }
48 </table>
49 <a href="/Egitim/EgitimEkle/" class="btn btn-primary"> Yeni Eğitim Bilgisi Ekle </a>
50
51
52
53
54
55
56
57
```

E-42: EĞİTİM EKLEME VE DÜZENLEME TASARIMI EKRAN GÖRÜNTÜSÜ (2)

```
25     </tr>
26     <foreach (var item in Model)>
27     {
28         <tr>
29             <td>@item.ID</td>
30             <td>@item.Baslik</td>
31             <td>@item.AltBaslik1</td>
32             <td>@item.AltBaslik2</td>
33             <td>@item.Tarih</td>
34             <td>@item.GNO</td>
35             <td>
36                 <a class="btn btn-danger btn-sm" href="/Egitim/EgitimSil/@item.ID">
37                     <i class="fas fa-trash">
38                 </i>
39             Sil
40             </a>
41             <a class="btn btn-info btn-sm" href="/Egitim/EgitimDuzenle/@item.ID">
42                 <i class="fas fa-pencil-alt">
43             </i>
44             Düzenle
45             </a>
46         </td>
47     </tr>
48 }
49 </table>
50 <a href="/Egitim/EgitimEkle/" class="btn btn-primary"> Yeni Eğitim Bilgisi Ekle </a>
51
52
53
54
55
56
57
```

E-43: DENEYİM LİSTESİ SAYFASI EKRAN GÖRÜNTÜSÜ

ID	Başlık	Alt Başlık	Tarih	Sil	Düzenle
1	YTU-TeknoPark	Stajyer	Temmuz 2022- Eylül 2022	Sil	Düzenle
2	Turkcell Global Bilgi	TurkishAirlines	Mayıs 2016- Temmuz 2017	Sil	Düzenle
3	SistemHost LLC.	Stajyer	Eylül 2012- Haziran 2013	Sil	Düzenle

Butona tıklanıldıktan sonraki ekran görüntüsü aşağıdaki gibidir.

E-44: DENEYİM EKLEME SAYFASI EKRAN GÖRÜNTÜSÜ

Yeni Deneyim Ekleme Sayfası

Başlığı yazınız

Alt Başlık

Çalışma tarihi yazınız

Açıklama giriniz

Kaydet

Deneyim düzenleme sayfasında düzenleme ve silme işlemlerinin yapılacağı sayfa düzenlenmesi için gereken kodlamalar aşağıdaki görseller ile desteklenmiştir. “Yeni Deneyim Ekle” butonu ile de sayfa üzerine yeni bilgi eklenebilecek ve mevcut bilgiler silinebilir ya da düzenlenlenebilecektir. Buradaki işlemlerin hepsinde işlem kaydedildikten sonra veri tabanına bilgiler görüntülenebilmektedir.

E-45: DENEYİM DÜZENLEME TASARIM VE VERİTABANI BAĞINTI KODU

```
1  @using MvcCv.Models.Entity
2  @model List<TblDeneyimler>
3  @{
4      ViewBag.Title = "Index";
5      Layout = "~/Views/Shared/_AdminLayout.cshtml";
6  }
7  <br />
8  <h2>Deneyim Listesi</h2>
9  <table class="table table-bordered">
10 <tr>
11     <th>ID</th>
12     <th>Başlık</th>
13     <th>Alt Başlık</th>
14     <th>Tarih</th>
15     <th>Sil</th>
16     <th>Düzenle</th>
17 </tr>
18 <foreach (var item in Model)>
19     {
20         <tr>
21             <td>@item.ID</td>
22             <td>@item.Baslik</td>
23             <td>@item.AltBaslik</td>
24             <td>@item.Tarih</td>
25             <td><a href="/Deneyim/DeneyimSil/@item.ID" class="btn btn-danger">Sil</a></td>
26             <td><a href="/Deneyim/DeneyimGetir/@item.ID" class="btn btn-warning">Düzenle</a></td>
27         </tr>
28     }
29 </table>
30 <a href="/Deneyim/DeneyimEkle/" class="btn btn-primary"> Yeni Deneyim Ekle </a>
31
32
```

E-46: DENEYİM LİSTESİ DÜZENLEME SAYFASI CONTROLLER KODU(1)

The screenshot shows the Visual Studio IDE interface with the following details:

- Top Bar:** Dosya, Düzen, Görünüm, Git, Proje, Derle, Hata Ayıklama, Test, Analiz, Araçlar, Uzantılar, Pencere, Yardım, Ara, MvcCv.
- Solution Explorer:** Sürücü Gezegini, Aşağı Kütüphanesi.
- Current File:** DeneyimController.cs
- Code Editor:** The code for the DeneyimController is displayed, which includes imports for Entity, Repositories, System, Collections.Generic, Linq, Web, and Mvc. It defines a DeneyimController class that inherits from Controller. It contains three actions: Index (GET), DeneyimEkle (GET), and DeneyimEkle (POST). The POST action adds a new entry to the repository and returns a redirect to the index page.
- Toolbars and Status Bar:** Standard Visual Studio toolbars and status bar are visible.

```
1  using MvcCv.Models.Entity;
2  using MvcCv.Repositories;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Web;
7  using System.Web.Mvc;
8
9  namespace MvcCv.Controllers
10 {
11     public class DeneyimController : Controller
12     {
13         // GET: Deneyim
14         DeneyimRepository repo = new DeneyimRepository();
15         public ActionResult Index()
16         {
17             var değerler = repo.List();
18             return View(değerler);
19         }
20         [HttpGet]
21         public ActionResult DeneyimEkle()
22         {
23             return View();
24         }
25         [HttpPost]
26         public ActionResult DeneyimEkle(TblDeneyimler p)
27         {
28             repo.Tadd(p);
29             return RedirectToAction("Index");
30         }
31     }
32 }
```

E-40: DENEYİM LİSTESİ DÜZENLEME SAYFASI CONTROLLER KODU(2)

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "MvcCv". The main window displays the code for "DeneyimController.cs". The code implements three actions: DeneyimSil (Delete), DeneyimGetir (Get), and DeneyimGetir (Update). It uses Entity Framework's Repository pattern via a "repo" variable. The "DeneyimSil" method finds the entity by ID and deletes it. The "DeneyimGetir" methods find the entity by ID and return it either as a View or updated with new values from a "p" parameter. The "DeneyimGetir" update method updates the entity's properties and returns to the index.

```
0 başvuru
31     public ActionResult DeneyimSil(int id)
32     {
33         TblDeneyimler t = repo.Find(x => x.ID == id);
34         repo.TDelete(t);
35         return RedirectToAction("Index");
36     }
37     [HttpGet]
38     0 başvuru
39     public ActionResult DeneyimGetir(int id)
40     {
41         TblDeneyimler t = repo.Find(x => x.ID == id);
42         return View(t);
43     }
44     [HttpPost]
45     0 başvuru
46     public ActionResult DeneyimGetir(TblDeneyimler p )
47     {
48         TblDeneyimler t = repo.Find(x => x.ID == p.ID);
49         t.Baslik = p.Baslik;
50         t.AltBaslik = p.AltBaslik;
51         t.Tarih = p.Tarih;
52         t.Aciklama = p.Aciklama;
53         repo.TUpdate(t);
54         return RedirectToAction("Index");
55     }
56 }
57 }
```

E-41: DENEYİM DÜZENLEME VERİTABANI BAĞINTISI SAYFASI KODU

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "MvcCv". The main window displays the code for "DeneyimGetir.cshtml", which is a Razor view. It starts with a model declaration for "TblDeneyimler". The view contains HTML for a form to update an experiment record. It includes labels and text boxes for "ID", "Başlık", "Alt Başlık", "Tarih", and "Acıklama". A button at the bottom is labeled "Güncelle".

```
@model MvcCv.Models.Entity.TblDeneyimler
@{
    ViewBag.Title = "DeneyimGetir";
    Layout = "~/Views/Shared/_AdminLayout.cshtml";
}

<br />
<h4>Deneyim Güncelleme Sayfası</h4>
<br />
<form method="post">
    @Html.Label("Deneyim ID")
    @Html.TextBoxFor(x => x.ID, new { @class = "form-control" })
    <br />
    @Html.Label("Başlık")
    @Html.TextBoxFor(x => x.Baslik, new { @class = "form-control" })
    <br />
    @Html.Label("Alt Başlık")
    @Html.TextBoxFor(x => x.AltBaslik, new { @class = "form-control" })
    <br />
    @Html.Label("Tarih")
    @Html.TextBoxFor(x => x.Tarih, new { @class = "form-control" })
    <br />
    @Html.Label("Acıklama")
    @Html.TextAreaFor(x => x.Aciklama, new { @class = "form-control" })
    <br />
    <button class="btn btn-info">Güncelle</button>
</form>
```

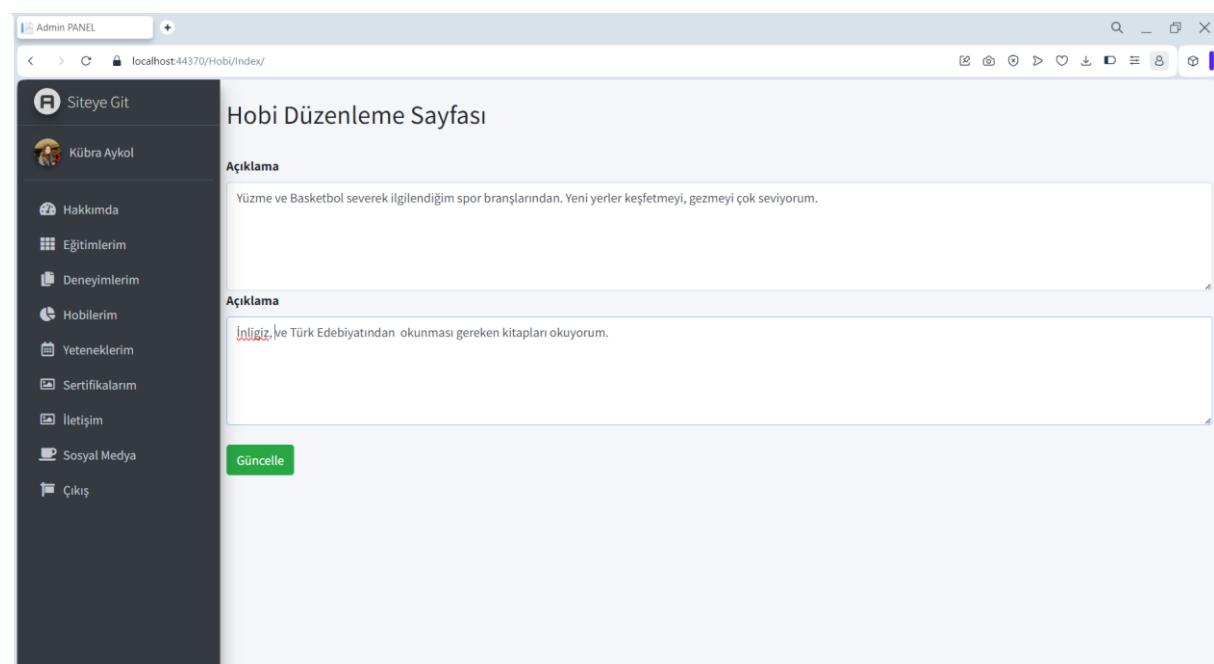
E-42: YENİ DENEYİM EKLEME SAYFASI TASARIM KOD EKRAN GÖRÜNTÜSÜ

The screenshot shows the Visual Studio code editor with the file 'DeneyimEkle.cshtml' open. The code is written in C# Razor syntax. It includes a title assignment, a layout reference, and a form for adding a new experience. The form contains fields for Title, Subtitle, Date, Description, and a 'Save' button.

```
1  @{
2      ViewBag.Title = "DeneyimEkle";
3      Layout = "~/Views/Shared/_AdminLayout.cshtml";
4  }
5  <br/>
6  <h4>Yeni Deneyim Ekleme Sayfası</h4>
7  <br/>
8  <form method="post">
9      <input type="text" name="Baslik" class="form-control" placeholder="Başlığı yazınız" />
10     <br />
11     <input type="text" name="AltBaslik" class="form-control" placeholder="Alt Başlık" />
12     <br />
13     <input type="text" name="Tarih" class="form-control" placeholder="Çalışma tarihi yazınız" />
14     <br />
15     <textarea placeholder="Açıklama giriniz" name="Aciklama" class="form-control" rows="5" ></textarea>
16     <br />
17     <button class="btn btn-info">Kaydet</button>
18
19 </form>
20
21
```

MVC çalışma prensibi gereğince sayfalar ve veri tabanı arasındaki bağıntılar daha az kodla daha işlevsel ve daha yüksek performanslı web projeleri üretmeye yaramaktadır.

E-43: HOBİLERİM DÜZENLEME SAYFASI EKRAN GÖRÜNTÜSÜ



E-44: HOBİLERİ DÜZENLEME SAYFASI KOD EKRAN GÖRÜNTÜSÜ

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "HobiController.cs" and "MvcCv". The main editor window displays the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using MvcCv.Models.Entity;
7  using MvcCv.Repositories;
8
9  namespace MvcCv.Controllers
10 {
11     public class HobiController : Controller
12     {
13         // GET: Hobi
14         GenericRepository<TblHobiler> repo = new GenericRepository<TblHobiler>();
15         [HttpGet]
16         public ActionResult Index()
17         {
18             var hobiler = repo.List();
19             return View(hobiler);
20         }
21         [HttpPost]
22         public ActionResult Index(TblHobiler p)
23         {
24             var t = repo.Find(x => x.ID == 1);
25             t.Aciklama1 = p.Aciklama1;
26             t.Aciklama2 = p.Aciklama2;
27             repo.TUpdate(t);
28             return RedirectToAction("Index");
29         }
30     }
31 }
```

The status bar at the bottom shows "Sat: 1 Krkt: 1 BŞL CRLF".

“Hobilerim” sayfası admin paneli üzerinde yalnızca “Güncelleme” ile işlevsellik kazandırılmıştır. Burada yalnızca metin kutuları üzerinde istenilen bilgiler eklenerek güncelleme yapılır ve veriler veri tabanına kaydedildikten sonra “default” anasayfada görüntülenir.

E-45: HOBİLERİM DÜZENLEME SAYFASI EKRAN GÖRÜNTÜSÜ

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "Index.cshtml" and "HobiController.cs". The main editor window displays the following ASP.NET MVC view code:

```
1  @using MvcCv.Models.Entity
2  @model List<TblHobiler>
3  @{
4      ViewBag.Title = "Index";
5      Layout = "~/Views/Shared/_AdminLayout.cshtml";
6  }
7  <br />
8  <h2>Hobi Düzenleme Sayfası</h2>
9  <br />
10 <form method="post">
11     @foreach (var item in Model)
12     {
13         <label>Açıklama</label>
14         <textarea rows="5" class="form-control" name="Aciklama1">@item.Aciklama1</textarea>
15         <label>Açıklama</label>
16         <textarea rows="5" class="form-control" name="Aciklama2">@item.Aciklama2</textarea>
17     }
18     <br />
19     <button class="btn btn-success">Güncelle</button>
20 </form>
21
22
23
```

The status bar at the bottom shows "Sat: 1 Krkt: 1 BŞL CRLF".

E-46: YETENEKLERİM DÜZENLEME SAYFASI EKRAN GÖRÜNTÜSÜ

#	Yetenek	Derece	Oran	Düzenle	Sil
1	Java Programlama Dili	<div style="width: 62%;"></div>	62%	<button>Düzenle</button>	<button>Sil</button>
2	Spring Initializr	<div style="width: 50%;"></div>	50%	<button>Düzenle</button>	<button>Sil</button>
3	Maven	<div style="width: 50%;"></div>	50%	<button>Düzenle</button>	<button>Sil</button>
4	UnitTest	<div style="width: 50%;"></div>	50%	<button>Düzenle</button>	<button>Sil</button>
5	C# Programlama Dili	<div style="width: 60%;"></div>	60%	<button>Düzenle</button>	<button>Sil</button>
6	ASP.NET Mvc5	<div style="width: 65%;"></div>	65%	<button>Düzenle</button>	<button>Sil</button>
7	MS SQL Veri Tabanı	<div style="width: 70%; background-color: #007bff;"></div>	70%	<button>Düzenle</button>	<button>Sil</button>
8	OOP	<div style="width: 72%; background-color: #007bff;"></div>	72%	<button>Düzenle</button>	<button>Sil</button>
9	Postman	<div style="width: 50%; background-color: #007bff;"></div>	50%	<button>Düzenle</button>	<button>Sil</button>
12	Python Programlama	<div style="width: 40%; background-color: #dc3545;"></div>	40%	<button>Düzenle</button>	<button>Sil</button>

“Yeni Yetenek Ekle” butonuna basıldıktan sonra sayfa aşağıdaki düzene geçmektedir. Buradan gerekli bilgilern girişi sağlandıktan sonra yine veriler veri tabanına kaydedilir ve buradan da anasayfada görüntülenmiş olunur.

E-47: YETENEKLERİM DÜZENLEME SAYFASI EKRAN GÖRÜNTÜSÜ

Yeni Yetenek Ekleme Sayfası

Yetenek Açıklaması

Derece

Kaydet

Aşağıda Controller ile gerçekleştirilen işlemlerin kod ekran görüntüsü iki sayfa şeklinde devam niteliğinde eklenmiştir.

E-48: YETENEKLERİM SAYFASI CONTROLLER EKRAN GÖRÜNTÜSÜ (1)

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "MvcCv". The main window displays the "YetenekController.cs" file under the "MvcCv" project. The code implements a controller for managing talents (Yetenekler). It includes methods for listing talents, adding a new talent, deleting a talent, and updating a talent's status.

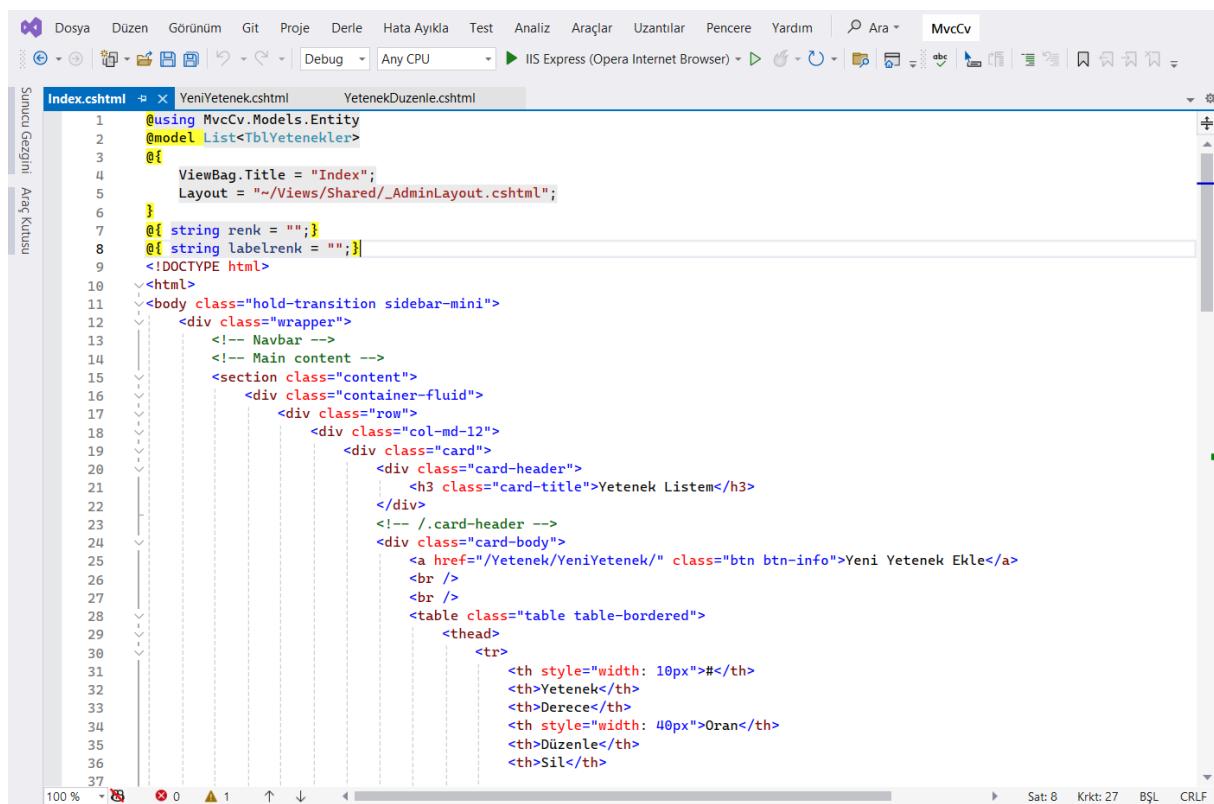
```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using MvcCv.Models.Entity;
7  using MvcCv.Repositories;
8  namespace MvcCv.Controllers
9  {
10     public class YetenekController : Controller
11     {
12         GenericRepository<TblYetenekler> repo = new GenericRepository<TblYetenekler>();
13
14         public ActionResult Index()
15         {
16             var yetenekler = repo.List();
17             return View(yetenekler);
18         }
19
20         [HttpPost]
21         public ActionResult YeniYetenek()
22         {
23             return View();
24         }
25
26         [HttpPost]
27         public ActionResult YeniYetenek(TblYetenekler p)
28         {
29             repo.Tadd(p);
30             return RedirectToAction("Index");
31         }
32
33         public ActionResult YetenekSil(int id)
34         {
35             var yetenek = repo.Find(x => x.ID == id);
36             repo.TDelete(yetenek);
37             return RedirectToAction("Index");
38         }
39
40         [HttpGet]
41         public ActionResult YetenekDuzenle(int id)
42         {
43             var yetenek = repo.Find(x => x.ID == id);
44             return View(yetenek);
45         }
46
47         [HttpPost]
48         public ActionResult YetenekDuzenle(TblYetenekler t)
49         {
50             var y = repo.Find(x => x.ID == t.ID);
51             y.Yetenek = t.Yetenek;
52             y.Oran = t.Oran;
53             repo.TUpdate(y);
54             return RedirectToAction("Index");
55         }
56     }
57 }
```

E-48: YETENEKLERİM SAYFASI CONTROLLER EKRAN GÖRÜNTÜSÜ (2)

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "MvcCv". The main window displays the "YetenekController.cs" file under the "MvcCv" project. The code is identical to the first screenshot, implementing a controller for managing talents (Yetenekler) with methods for listing, adding, deleting, and updating talents.

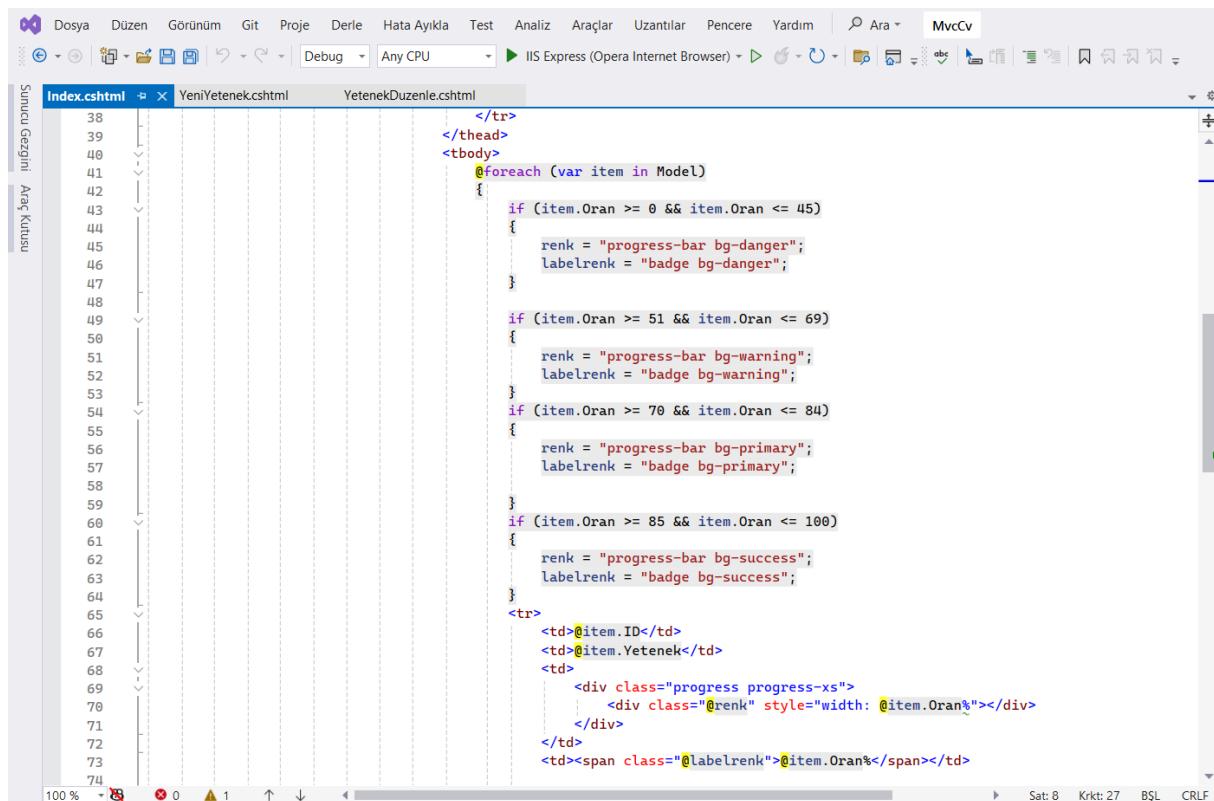
```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using MvcCv.Models.Entity;
7  using MvcCv.Repositories;
8  namespace MvcCv.Controllers
9  {
10     public class YetenekController : Controller
11     {
12         GenericRepository<TblYetenekler> repo = new GenericRepository<TblYetenekler>();
13
14         public ActionResult Index()
15         {
16             return View();
17         }
18
19         [HttpPost]
20         public ActionResult YeniYetenek(TblYetenekler p)
21         {
22             repo.Tadd(p);
23             return RedirectToAction("Index");
24         }
25
26         public ActionResult YetenekSil(int id)
27         {
28             var yetenek = repo.Find(x => x.ID == id);
29             repo.TDelete(yetenek);
30             return RedirectToAction("Index");
31         }
32
33         [HttpGet]
34         public ActionResult YetenekDuzenle(int id)
35         {
36             var yetenek = repo.Find(x => x.ID == id);
37             return View(yetenek);
38         }
39
40         [HttpPost]
41         public ActionResult YetenekDuzenle(TblYetenekler t)
42         {
43             var y = repo.Find(x => x.ID == t.ID);
44             y.Yetenek = t.Yetenek;
45             y.Oran = t.Oran;
46             repo.TUpdate(y);
47             return RedirectToAction("Index");
48         }
49
50     }
51 }
```

E-49: YETENEKLERİM SAYFASI TASARIM EKRAN GÖRÜNTÜSÜ (1)



```
1 @using MvcCv.Models.Entity
2 @model List<TblYetenekler>
3 @{
4     ViewBag.Title = "Index";
5     Layout = "~/Views/Shared/_AdminLayout.cshtml";
6 }
7 @{
8     string renk = "";
9     string labelrenk = "";
10}
11<!DOCTYPE html>
12<html>
13     <body class="hold-transition sidebar-mini">
14         <div class="wrapper">
15             <!-- Navbar -->
16             <!-- Main content -->
17             <section class="content">
18                 <div class="container-fluid">
19                     <div class="row">
20                         <div class="col-md-12">
21                             <div class="card">
22                                 <div class="card-header">
23                                     <h3 class="card-title">Yetenek Listem</h3>
24                                 </div>
25                                 <div class="card-body">
26                                     <a href="/Yetenek/YeniYetenek/" class="btn btn-info">Yeni Yetenek Ekle</a>
27                                     <br />
28                                     <br />
29                                     <table class="table table-bordered">
30                                         <thead>
31                                             <tr>
32                                                 <th style="width: 10px">#</th>
33                                                 <th>Yetenek</th>
34                                                 <th>Derece</th>
35                                                 <th style="width: 40px">Oran</th>
36                                                 <th>Düzenle</th>
37                                                 <th>Sil</th>
38                                         </tr>
39                                         </thead>
40                                         <tbody>
41                                             @foreach (var item in Model)
42                                             {
43                                                 if (item.Oran >= 0 && item.Oran <= 45)
44                                                 {
45                                                     renk = "progress-bar bg-danger";
46                                                     labelrenk = "badge bg-danger";
47                                                 }
48                                                 if (item.Oran >= 51 && item.Oran <= 69)
49                                                 {
50                                                     renk = "progress-bar bg-warning";
51                                                     labelrenk = "badge bg-warning";
52                                                 }
53                                                 if (item.Oran >= 70 && item.Oran <= 84)
54                                                 {
55                                                     renk = "progress-bar bg-primary";
56                                                     labelrenk = "badge bg-primary";
57                                                 }
58                                                 if (item.Oran >= 85 && item.Oran <= 100)
59                                                 {
60                                                     renk = "progress-bar bg-success";
61                                                     labelrenk = "badge bg-success";
62                                                 }
63                                             <tr>
64                                                 <td>@item.ID</td>
65                                                 <td>@item.Yetenek</td>
66                                                 <td>
67                                                     <div class="progress progress-xs">
68                                                         <div class="@renk" style="width: @item.Oran%"></div>
69                                                     </div>
70                                                 </td>
71                                                 <td><span class="@labelrenk">@item.Oran%</span></td>
72                                             </tr>
73                                         }
74                                         </tbody>
75                                     </table>
76                                 </div>
77                             </div>
78                         </div>
79                     </div>
80                 </div>
81             </section>
82         </div>
83     </body>
84 </html>
```

E-50: YETENEKLERİM SAYFASI TASARIM EKRAN GÖRÜNTÜSÜ (2)



```
38 </tr>
39 </thead>
40 <tbody>
41     @foreach (var item in Model)
42     {
43         if (item.Oran >= 0 && item.Oran <= 45)
44         {
45             renk = "progress-bar bg-danger";
46             labelrenk = "badge bg-danger";
47         }
48
49         if (item.Oran >= 51 && item.Oran <= 69)
50         {
51             renk = "progress-bar bg-warning";
52             labelrenk = "badge bg-warning";
53         }
54         if (item.Oran >= 70 && item.Oran <= 84)
55         {
56             renk = "progress-bar bg-primary";
57             labelrenk = "badge bg-primary";
58         }
59         if (item.Oran >= 85 && item.Oran <= 100)
60         {
61             renk = "progress-bar bg-success";
62             labelrenk = "badge bg-success";
63         }
64     <tr>
65         <td>@item.ID</td>
66         <td>@item.Yetenek</td>
67         <td>
68             <div class="progress progress-xs">
69                 <div class="@renk" style="width: @item.Oran%"></div>
70             </div>
71         </td>
72         <td><span class="@labelrenk">@item.Oran%</span></td>
73     </tr>
74     </tbody>
75 </table>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </body>
82 </html>
```

E-53: YETENEKLERİM SAYFASI TASARIM EKRAN GÖRÜNTÜSÜ (3)

The screenshot shows the 'YetenekDuzenle.cshtml' file in Visual Studio. The code is a table definition with several columns. The first column contains a progress bar with a width corresponding to the item's percentage. The second column contains a link to edit the item. The third column contains a link to delete the item.

```
Index.cshtml  X YeniYetenekcshtml  YetenekDuzenle.cshtml
68 >|>
69 >|>
70 >|>    <td>
71 >|>        <div class="progress progress-xs">
72 >|>            <div class="bar" style="width: @item.Oran%></div>
73 >|>        </div>
74 >|>    <td>
75 >|>        <span class="@labelrenk">@item.Oran%</span></td>
76 >|>
77 >|>    <td>
78 >|>        <a class="btn btn-info btn-sm" href="/Yetenek/YetenekDuzenle/@item.ID">
79 >|>            <i class="fas fa-pencil-alt">
80 >|>            Duzenle
81 >|>        </a>
82 >|>    <td>
83 >|>        <a class="btn btn-danger btn-sm" href="/Yetenek/YetenekSil/@item.ID">
84 >|>            <i class="fas fa-trash">
85 >|>            Sil
86 >|>        </a>
87 >|>    </td>
88 >|>    </tr>
89 >|>
90 >|>    </tbody>
91 >|>  </table>
92 >|>  </div>
93 >|>  </div>
94 >|>  </div>
95 >|>  </div>
96 >|>  </div>
97 >|>  </div>
98 >|>  </div>
99 >|>  </div>
100 >|>  </body>
101 >|>  </html>
102 >|>
103 >|>
104 >|>
```

E-54: YENİ YETENEK SAYFASI VERİTABANI BAĞINTISI EKRAN GÖRÜNTÜSÜ

The screenshot shows the 'YeniYetenek.cshtml' file in Visual Studio. It defines a form for adding a new talent. The form includes fields for 'Yetenek Açıklaması' (Description) and 'Derece' (Degree), both using the 'form-control' class.

```
YeniYetenek.cshtml  X YetenekDuzenle.cshtml
1 @model MvcCv.Models.Entity.TblYetenekler
2 @{
3     ViewBag.Title = "YeniYetenek";
4     Layout = "~/Views/Shared/_AdminLayout.cshtml";
5 }
6
7 <br />
8 <h4>Yeni Yetenek Ekleme Sayfası</h4>
9 <br />
10 <form method="post">
11     @Html.Label("Yetenek Açıklaması")
12     @Html.TextBoxFor(x => x.Yetenek, new { @class = "form-control" })
13     <br />
14     @Html.Label("Derece")
15     @Html.TextBoxFor(x => x.Oran, new { @class = "form-control" })
16     <br />
17     <button class="btn btn-info">Kaydet</button>
18 </form>
19
20
```

E-55: YETENEK DÜZENLE SAYFASI VERİTABANI BAĞINTISI EKRAN GÖRÜNTÜSÜ

The screenshot shows the Visual Studio code editor with the file 'YetenekDuzenle.cshtml' open. The code is written in C# Razor syntax. It includes a model declaration, a ViewBag assignment, and a form for updating a record from the database. The code is as follows:

```
1 @model MvcCv.Models.Entity.TblYetenekler
2 @{
3     ViewBag.Title = "YetenekDuzenle";
4     Layout = "~/Views/Shared/_AdminLayout.cshtml";
5 }
6
7 <br />
8 <h4>Yetenek Düzenleme Sayfası</h4>
9 <br />
10 <><form method="post">
11     @Html.Label("Yetenek Açıklaması")
12     @Html.TextBoxFor(x => x.Yetenek, new { @class = "form-control" })
13     <br />
14     @Html.Label("Derece")
15     @Html.TextBoxFor(x => x.Oran, new { @class = "form-control" })
16     <br />
17     <button class="btn btn-success">Güncelle</button>
18 </form>
19
20
```

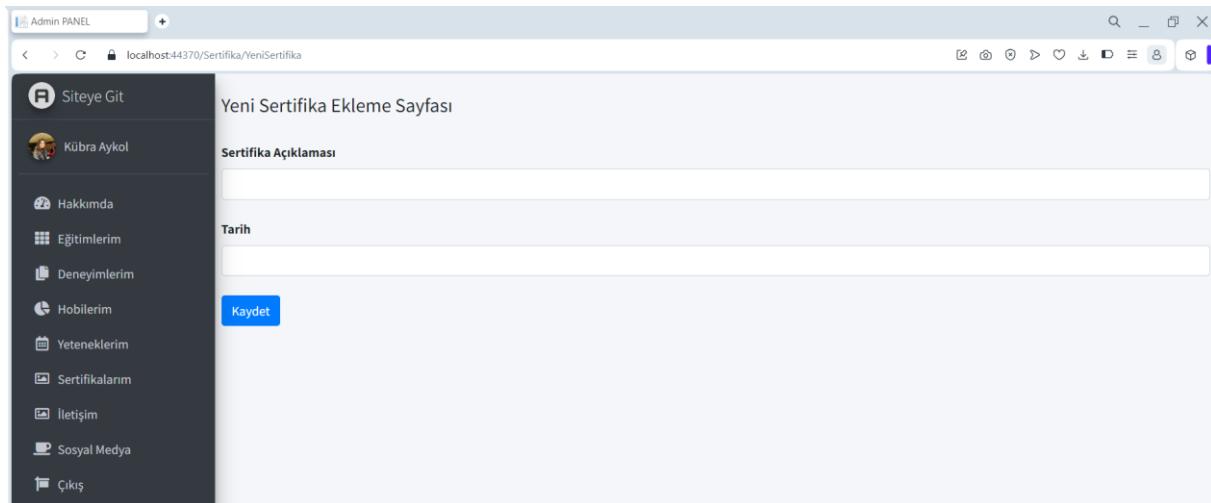
“Yeteneklerim” sayfası admin paneli üzerinde mevcut bilgiler üzerinde düzenleme ve silme işlemleri yine veri tabanıyla bağlantılı şekilde gerçekleştirilebilir.

E-56: SERTİFİKALARIM DÜZENLEME SAYFASI EKRAN GÖRÜNTÜSÜ

The screenshot shows a web browser window titled 'Admin PANEL'. The left sidebar contains navigation links such as 'Siteye Git', 'Kübra Aykol', 'Hakkında', 'Eğitimlerim', 'Deneyimlerim', 'Hobilerim', 'Yeteneklerim', 'Sertifikalarım', 'İletişim', 'Sosyal Medya', and 'Çıkış'. The main content area displays a list of certificates organized by month: May 2024, May 2024, December 2022, and November 2022. Each entry shows the name of the certificate and its provider. At the bottom of the list is a blue button labeled 'Yeni Sertifika Giriş'i.

“Yeni Sertifika Giriş'i” butonuna tıkladıktan sonra aşağıdaki ekran görüntüsüne geçiş sağlanmaktadır web sayfasında.

E-57: YENİ SERTİFİKA EKLEME SAYFASI EKRAN GÖRÜNTÜSÜ



Burada işlemler iki ayrı metin kutusuna girilen bilgilerin “Kaydet” butonuna tıklanmasıından sonra veri tabanına kaydı düşen bilgilerin anasayfada görüntülenmesini sağlar. Bunlara yönelik kodlar ve tasarım kodları da aşağıdaki görsellerle eklenmiştir. Bu kodların bazıları birden fazla ekran görüntüsü ile 1,2,... şeklinde isimlendirilmiştir.

E-58: SERTİFİKALARIM SAYFASI CONTROLLER EKRAN GÖRÜNTÜSÜ(1)

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using MvcCv.Models.Entity;
7  using MvcCv.Repositories;
8
9  namespace MvcCv.Controllers
10 {
11     public class SertifikaController : Controller
12     {
13         // GET: Sertifika
14         GenericRepository<TblSertifikalar> repo= new
15             GenericRepository<TblSertifikalar>();
16
17         public ActionResult Index()
18         {
19             var sertifika = repo.List();
20             return View(sertifika);
21         }
22
23         [HttpGet]
24         public ActionResult SertifikaGetir(int id)
25         {
26             var sertifika = repo.Find(x => x.ID == id);
27             ViewBag.d = id;
28             return View(sertifika);
29         }
30
31         [HttpPost]
32         public ActionResult SertifikaGetir(TblSertifikalar t)
33         {
34             var sertifika = repo.Find(x => x.ID == t.ID);
35             sertifika.Aciklama = t.Aciklama;
36         }
37     }
38 }
```

E-59: SERTİFİKALARIM SAYFASI CONTROLLER EKRAN GÖRÜNTÜSÜ(2)

The screenshot shows the code editor in Visual Studio with the file 'SertifikaController.cs' open. The code implements a controller for managing certificates. It includes methods for getting a certificate by ID, adding a new certificate, and deleting a certificate. The code uses Entity Framework to interact with a database repository.

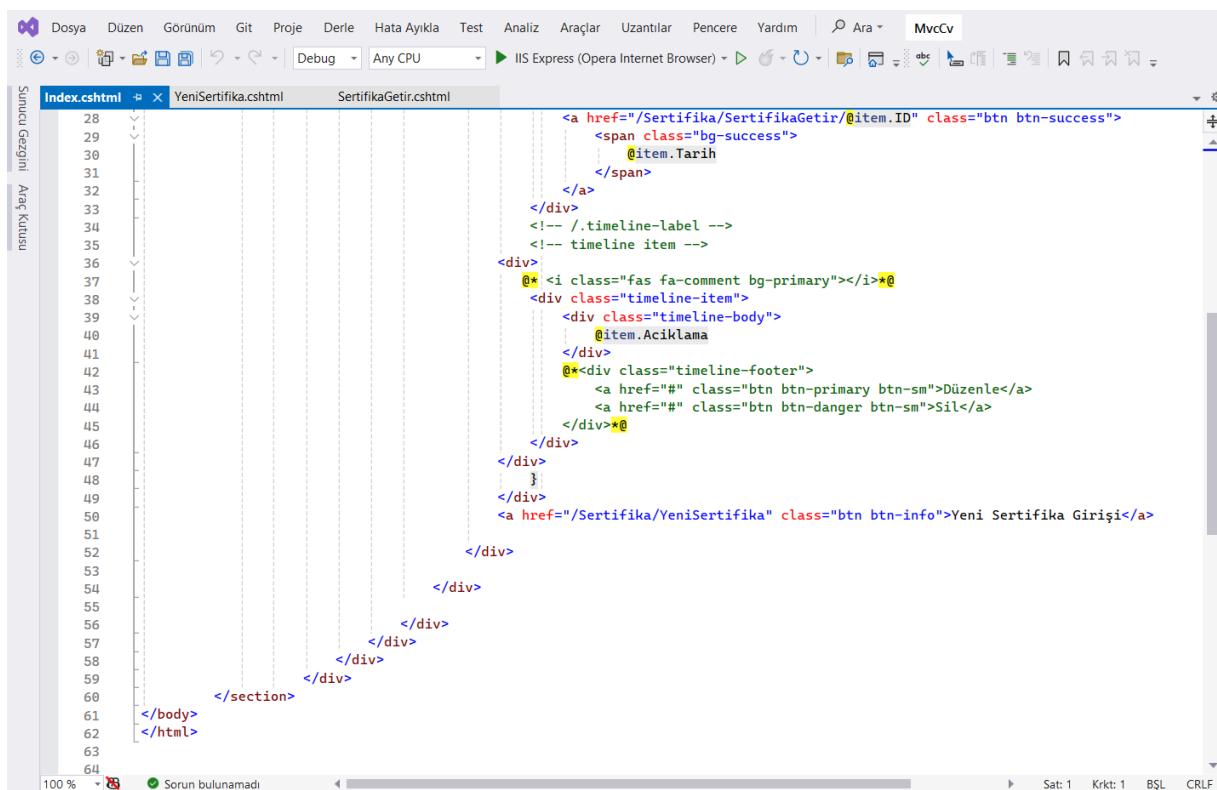
```
25     ViewBag.o = t;
26     return View(sertifika);
27 }
28 [HttpPost]
29 0 basvuru
30 public ActionResult SertifikaGetir(TblSertifikalar t)
31 {
32     var sertifika = repo.Find(x => x.ID == t.ID);
33     sertifika.Aciklama = t.Aciklama;
34     sertifika.Tarih = t.Tarih;
35     repo.Update(sertifika);
36     return RedirectToAction("Index");
37 }
38 [HttpGet]
39 0 basvuru
40 public ActionResult YeniSertifika()
41 {
42     return View();
43 }
44 [HttpPost]
45 0 basvuru
46 public ActionResult YeniSertifika(TblSertifikalar p)
47 {
48     repo.Tadd(p);
49     return RedirectToAction("Index");
50 }
51 0 basvuru
52 public ActionResult SertifikaSil(int id)
53 {
54     var sertifika=repo.Find(x=> x.ID == id);
55     repo.TDelete(sertifika);
56     return RedirectToAction("Index");
57 }
```

E-60: SERTİFİKALARIM SAYFASI TASARIM EKRAN GÖRÜNTÜSÜ(1)

The screenshot shows the code editor in Visual Studio with the file 'Index.cshtml' open. This is a Razor view for displaying a list of certificates. It uses a timeline-based layout where each certificate is represented as a card with a date label and a success button.

```
1 @using MvcCv.Models.Entity
2 @model List<TblSertifikalar>
3 @{
4     ViewBag.Title = "Index";
5     Layout = "~/views/Shared/_AdminLayout.cshtml";
6 }
7
8 <!DOCTYPE html>
9 <html>
10 <body class="hold-transition sidebar-mini">
11     <section class="content">
12         <div class="container-fluid">
13             <div class="row">
14                 <!-- /.col -->
15                 <div class="col-md-12">
16                     <div class="card">
17                         <div class="card-header p-2">
18                         </div><!-- /.card-header -->
19                         <div class="card-body">
20
21                             <div class="tab-pane id="timeline">
22                                 <!-- The timeline -->
23                                 <div class="timeline timeline-inverse">
24                                     <!-- timeline time label -->
25                                     @foreach (var item in Model)
26                                     {
27                                         <div class="time-label">
28                                             <a href="/Sertifika/SertifikaGetir/@item.ID" class="btn btn-success">
29                                                 <span class="bg-success">
30                                                 | @item.Tarih
31                                                 </span>
32                                             </a>
33                                         </div>
34                                         <!-- /.timeline-label -->
35                                         <!-- timeline item -->
36                                         <div>
37                                             | @* <i class="fas fa-comment bg-primary"></i>*@
38                                         </div>
39                                     }
40
41                                 </div>
42                             <!-- /.tab-pane -->
43                         </div>
44                     </div>
45                 </div>
46             </div>
47         </div>
48     </section>
49 </body>
50 </html>
```

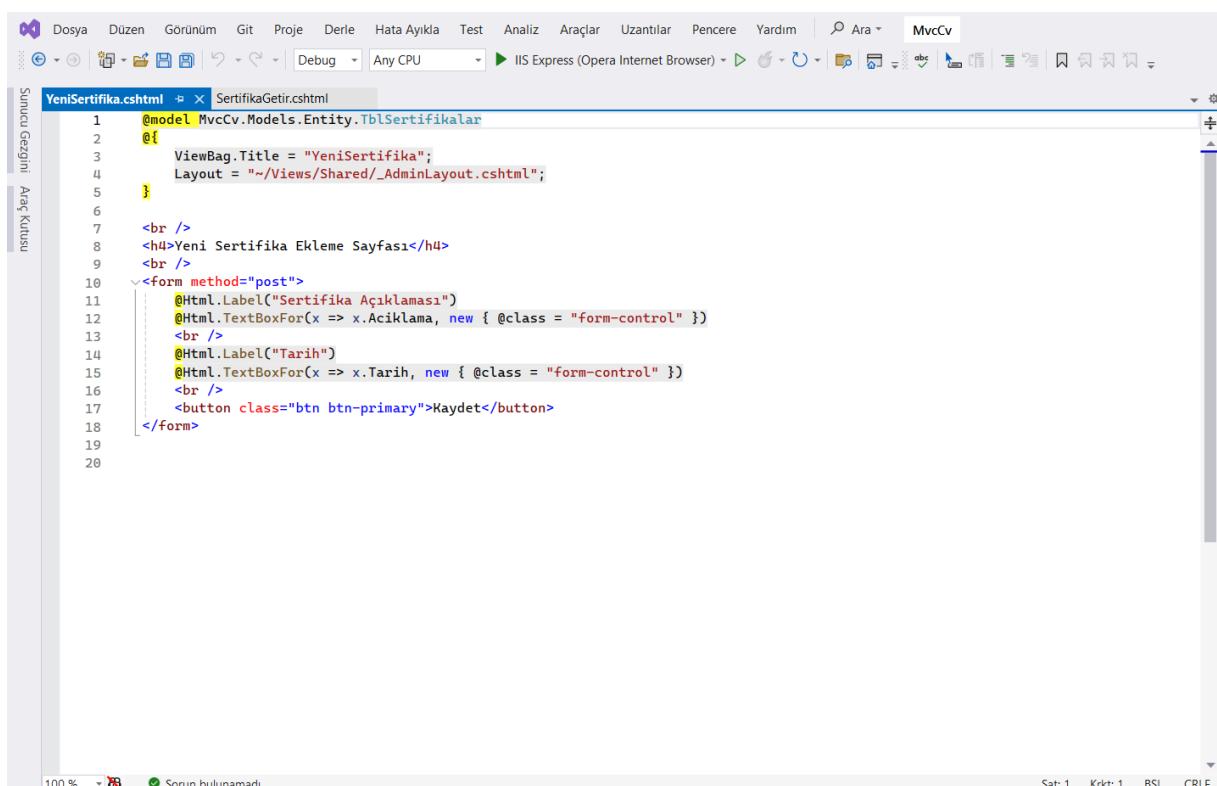
E-61: SERTİFİKALARIM SAYFASI TASARIM EKRAN GÖRÜNTÜSÜ(2)



The screenshot shows the Visual Studio IDE with the Index.cshtml file open. The code is a template for displaying certificate items in a timeline. It includes sections for the header, a main loop for items, and a footer. The footer contains links for 'Düzenle' (Edit) and 'Sil' (Delete). The file uses Bootstrap classes like 'btn btn-success' and 'btn btn-primary'.

```
28 <div>
29   <div>
30     <a href="/Sertifika/SertifikaGetir/@item.ID" class="btn btn-success">
31       <span class="bg-success">
32         @item.Tarih
33       </span>
34     </a>
35   </div>
36   <!-- /.timeline-label -->
37   <!-- timeline item -->
38   <div>
39     <i class="fas fa-comment bg-primary"></i>*
40     <div class="timeline-item">
41       <div class="timeline-body">
42         @item.Aciklama
43       </div>
44       <div class="timeline-footer">
45         <a href="#" class="btn btn-primary btn-sm">Düzenle</a>
46         <a href="#" class="btn btn-danger btn-sm">Sil</a>
47       </div>
48     </div>
49   </div>
50   <a href="/Sertifika/YeniSertifika" class="btn btn-info">Yeni Sertifika Giriş'i</a>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </section>
61 </body>
62 </html>
```

E-62: YENİ SERTİFİKA SAYFASI VERİTABANI BAĞINTISI EKRAN GÖRÜNTÜSÜ



The screenshot shows the Visual Studio IDE with the YeniSertifika.cshtml file open. This is a form page for adding a new certificate. It includes a title, a subtitle, and a form with fields for 'Sertifika Açıklaması' (Certificate Description) and 'Tarih' (Date). The form uses the 'post' method.

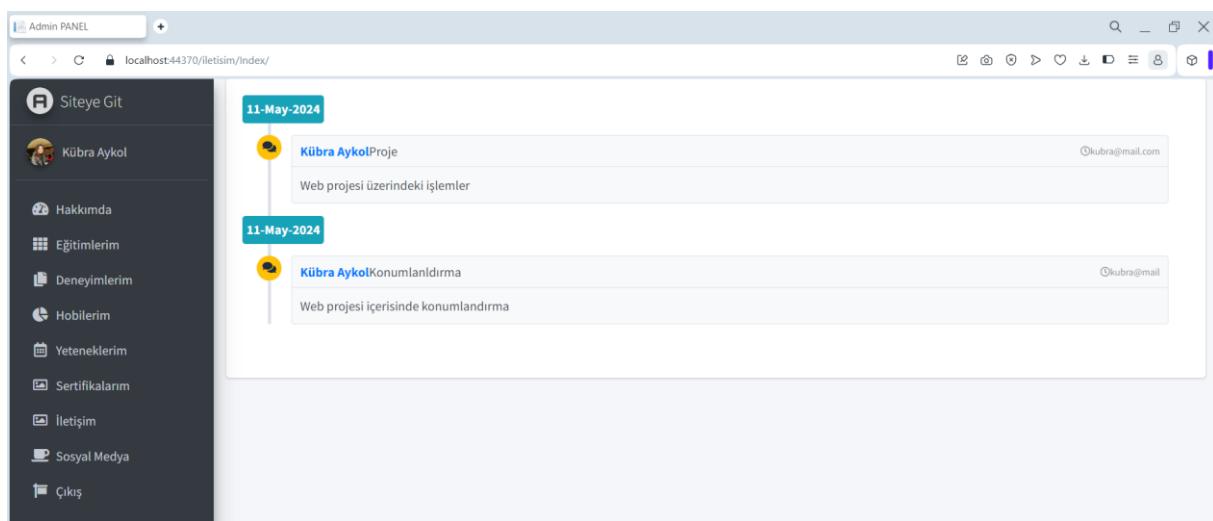
```
1 @model MvcCv.Models.Entity.TblSertifikalar
2 @{
3   ViewBag.Title = "YeniSertifika";
4   Layout = "~/Views/Shared/_AdminLayout.cshtml";
5 }
6
7 <br />
8 <h4>Yeni Sertifika Ekleme Sayfası</h4>
9 <br />
10 <form method="post">
11   @Html.Label("Sertifika Açıklaması")
12   @Html.TextBoxFor(x => x.Aciklama, new { @class = "form-control" })
13   <br />
14   @Html.Label("Tarih")
15   @Html.TextBoxFor(x => x.Tarih, new { @class = "form-control" })
16   <br />
17   <button class="btn btn-primary">Kaydet</button>
18 </form>
```

E-63: SERTİFİKA GÜNCELLEME SAYFASI VERİTABANI BAĞINTISI EKRAN GÖRÜNTÜSÜ

The screenshot shows the Visual Studio code editor with the file 'SertifikaGetir.cshtml' open. The code is written in C# Razor syntax. It includes a model declaration, view bag title, layout, and a form for updating a certificate record. The code uses Bootstrap classes for styling.

```
1 @model MvcCv.Models.Entity.TblSertifikalar
2 @{
3     ViewBag.Title = "SertifikaGetir";
4     Layout = "~/Views/Shared/_AdminLayout.cshtml";
5 }
6
7 <br />
8 <h4>Sertifika Düzenleme Sayfası</h4>
9 <br />
10 <form method="post">
11     @Html.Label("Sertifika Açıklaması")
12     @Html.TextBoxFor(x => x.Aciklama, new { @class = "form-control" })
13     <br />
14     @Html.Label("Tarih")
15     @Html.TextBoxFor(x => x.Tarih, new { @class = "form-control" })
16     <br />
17     <button class="btn btn-success">Güncelle</button>
18     <a href="/Sertifika/SertifikaSil/@ViewBag.d" class="btn btn-danger">Sil</a>
19 </form>
20
21
22
```

E-64: İLETİŞİM SAYFASI EKRAN GÖRÜNTÜSÜ



Anasayfa üzerinde bulunan “Bana Ulaşın” sayfası üzerinden gönderilen maillerin Admin paneline gelen maillerin bilgileri ve içeriklerini görüntülemektedir. Aynı zamanda veri tabanında da kaydolan mesaj içerikleri gönderinin mail bilgilerini içermektedir.

E-65: İLETİŞİM SAYFASI CONTROLLER EKRAN GÖRÜNTÜSÜ

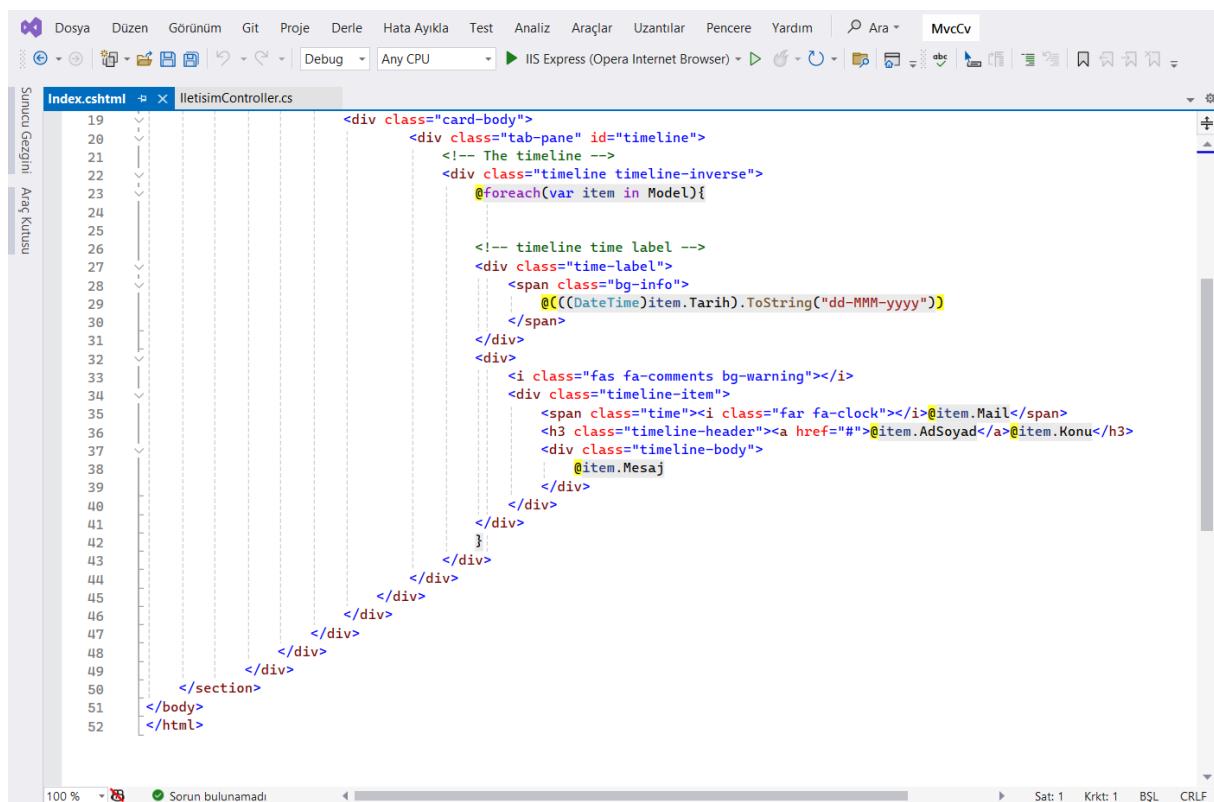
The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes 'Dosya', 'Düzen', 'Görünüm', 'Git', 'Proje', 'Derle', 'Hata Ayıklama', 'Test', 'Analiz', 'Araçlar', 'Uzantılar', 'Pencere', 'Yardım', 'Ara', and 'MvcCv'. Below the menu is a toolbar with icons for file operations like 'New', 'Open', 'Save', etc. The title bar displays 'IletisimController.cs' and 'Any CPU'. A status bar at the bottom shows 'Sat: 1 Krkt: 1 BSL CRLF'.

The main code editor window contains the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using MvcCv.Models.Entity;
7  using MvcCv.Repositories;
8
9  namespace MvcCv.Controllers
10 {
11     public class IletisimController : Controller
12     {
13         // GET: Iletisim
14         GenericRepository<Tbliletisim>
15             repo= new GenericRepository<Tbliletisim>();
16         public ActionResult Index()
17         {
18             var mesajlar = repo.List();
19             return View(mesajlar);
20         }
21     }
}
```

E-66: İLETİŞİM SAYFASI VERİTABANI BAĞINTISI EKRAN GÖRÜNTÜSÜ(1)

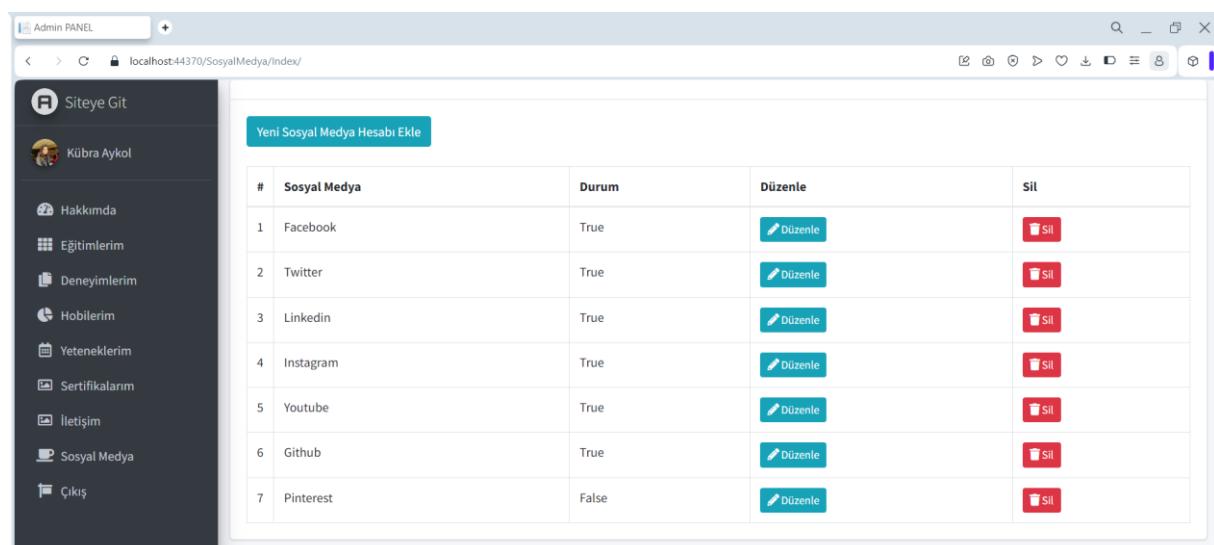
E-67: İLETİŞİM SAYFASI VERİTABANI BAĞINTISI EKRAN GÖRÜNTÜSÜ(2)



```
19 <div class="card-body">
20   <div class="tab-pane" id="timeline">
21     <!-- The timeline -->
22     <div class="timeline timeline-inverse">
23       @foreach(var item in Model){
24
25
26         <!-- timeline time label -->
27         <div class="time-label">
28           <span class="bg-info">
29             @((DateTime)item.Tarih.ToString("dd-MMM-yyyy"))
30           </span>
31         </div>
32         <div>
33           <i class="fas fa-comments bg-warning"></i>
34           <div class="timeline-item">
35             <span class="time"><i class="far fa-clock"></i>@item.Mail</span>
36             <h3 class="timeline-header"><a href="#">@item.AdSoyad</a>@item.Konu</h3>
37             <div class="timeline-body">
38               @item.Mesaj
39             </div>
40           </div>
41         </div>
42       }
43     </div>
44   </div>
45   </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </body>
51 </html>
```

“İletişim” sekmesinde veri tabanı bağlantıları ve tasarım aynı index sayfası üzerinde kodlamaları yapılmıştır. Buradaki işlemlerde maillere herhangi bir güncelleme veya silme müdahalesi yapılmayıp yalnızca iletişim amaçlı gelen iletileri görebilmek amaçlanmıştır.

E-68: SOSYAL MEDYA SAYFASI EKRAN GÖRÜNTÜSÜ

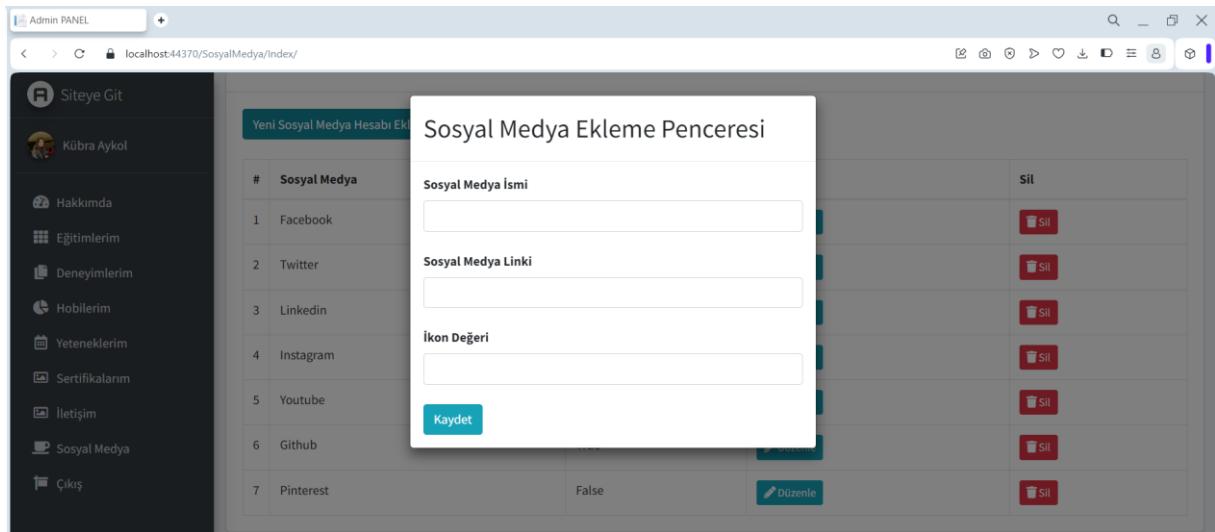


The screenshot shows a web application interface for managing social media accounts. On the left, there is a sidebar with user information and navigation links. The main content area displays a table of social media accounts with columns for ID, Name, Status, Edit, and Delete.

#	Sosyal Medya	Durum	Düzenle	Sil
1	Facebook	True	<button>Düzenle</button>	<button>Sil</button>
2	Twitter	True	<button>Düzenle</button>	<button>Sil</button>
3	Linkedin	True	<button>Düzenle</button>	<button>Sil</button>
4	Instagram	True	<button>Düzenle</button>	<button>Sil</button>
5	Youtube	True	<button>Düzenle</button>	<button>Sil</button>
6	Github	True	<button>Düzenle</button>	<button>Sil</button>
7	Pinterest	False	<button>Düzenle</button>	<button>Sil</button>

Bu sayfa aracılığı ile web sitesi üzerine diğer sosyal medya hesapları eklenebilecek. Kişi dilerse sosyal medya hesabını aktif veya pasif duruma çekebilecek bu panel üzerinden. Böylece “Güncelleme” işlemi veya “Silme” işlemi gerçekleşmiş olacak. “Yeni Sosyal Medya Hesabı Ekle” butonu ile de aşağıdaki ekran görüntüsü ortaya çıkmış olacak ve yeni bir sosyal medya hesabı ekleme işlemi gerçekleşmiş olacak.

E-69: YENİ SOSYAL MEDYA HESABI EKLEME SAYFASI EKRAN GÖRÜNTÜSÜ



E-70: YENİ SOSYAL MEDYA HESABI CONTROLLER EKRAN GÖRÜNTÜSÜ

```
0 başvuru
public PartialViewResult SosyalMedya()
{
    var sosyalmedya = db.TblSosyalMedya.ToList();
    return PartialView(sosyalmedya);
}
0 başvuru
```

Burada bütün sayfa yeniden eklenmek yerine projeye sonradan dahil edilen fikrin eklenmesine yönelik yalnızca o kısma ait kodların ekran görüntüsü eklendi. Aynı zamanda veri tabanında verilerin kaydolması için de yeniden tablolama yapmak yerine DDL ile işlem yapılarak Sosyal Medya Hesabı içeriğine yönelik veri tabanında yer açıldı.

E71: DDL ile SOSYAL MEDYA İKON ve LINKLERİNE YÖNELİK TABLOLAMA İŞLEMİ EKRAN GÖRÜNTÜSÜ

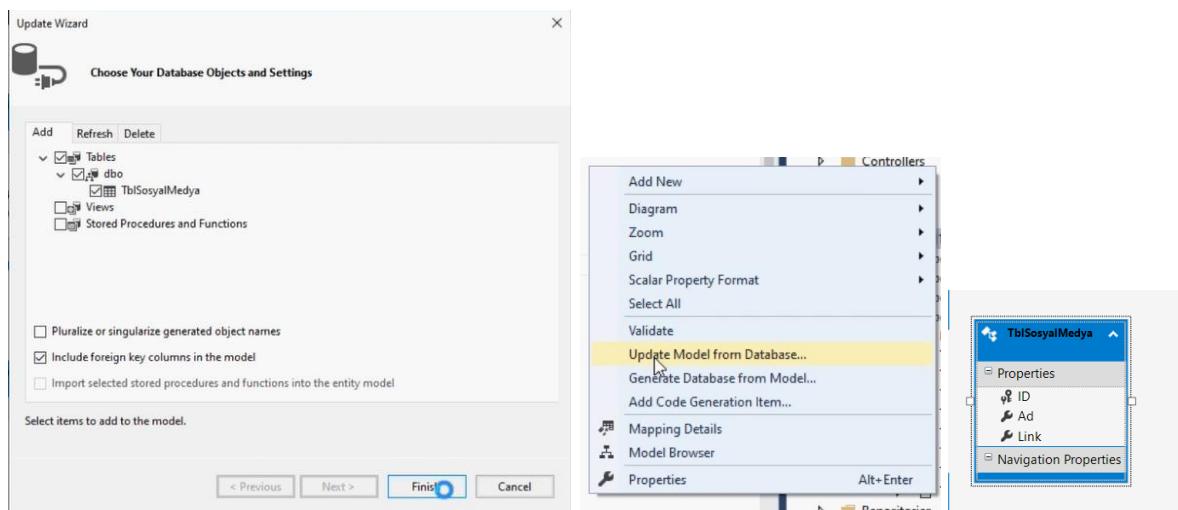
```
Use DbCv
Create Table TblSosyalMedya
(
    ID Int Primary Key Identity(1,1),
    Ad Varchar(30),
    Link varchar(50)
)
```

159 %

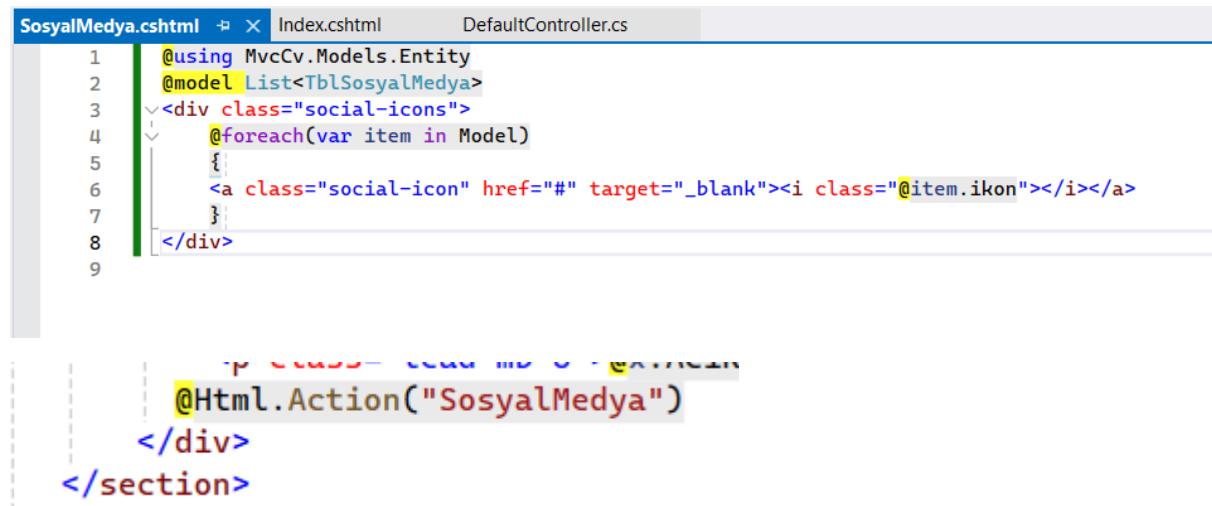
Messages

Commands completed successfully.

E72: DDL SORGU İŞLEMİ SONRASINDA VERİLERİN MODEL DENİLEN ALANA VİSUAL STUDIO'DA YANSITILMA AŞAMALARI EKRAN GÖRÜNTÜLERİ (3ADET)



E73:SOSYAL MEDYA HESAPLARI CONTROLLER EKRAN GÖRÜNTÜSÜ

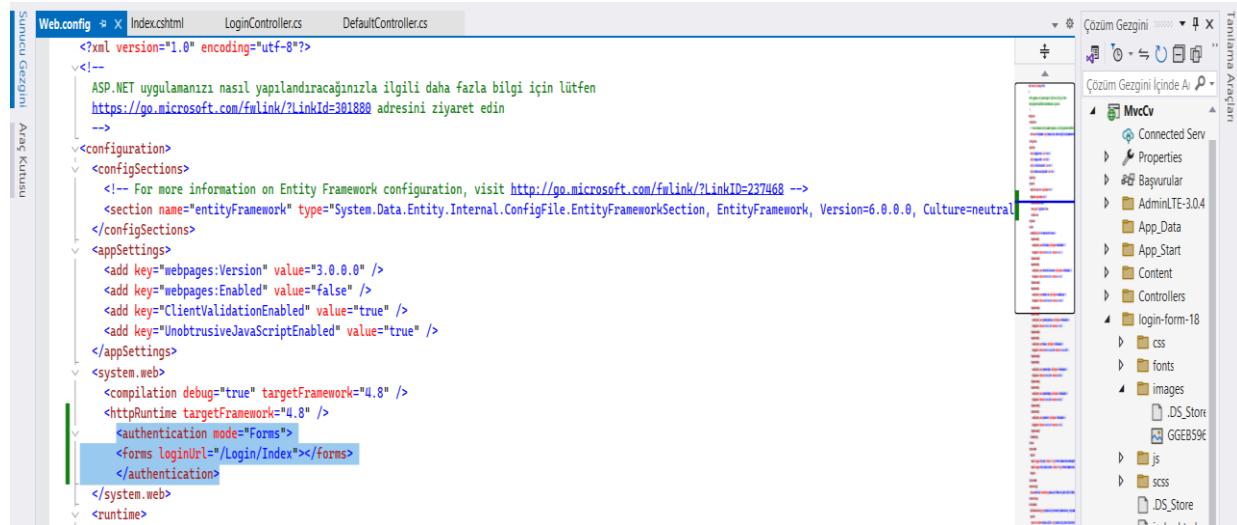


```
1 @using MvcCv.Models.Entity
2 @model List<TblSosyalMedya>
3 <div class="social-icons">
4     @foreach(var item in Model)
5     {
6         <a class="social-icon" href="#" target="_blank"><i class="@item.ikon"></i></a>
7     }
8 </div>
9 
```

```
    @Html.Action("SosyalMedya")
</div>
</section>
```

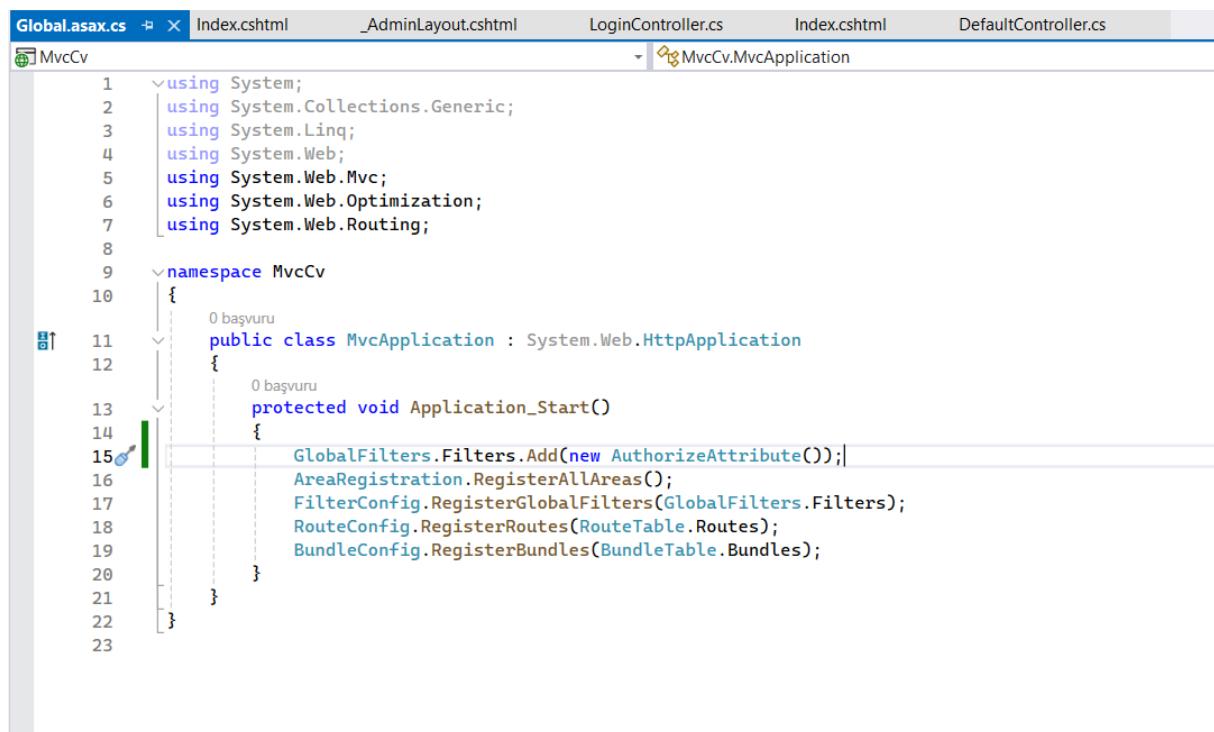
Bu kısımda yine bütün ekran görüntüsü eklenmek yerine projeye sonradan dahil olan kısımları kısa ekran görüntüleri şeklinde eklemiştir.

E-74: AUTHORIZED İŞLEMİ EKRAN GÖRÜNTÜSÜ



Buradaki işlem ile projede son olarak yetkisi olamayan kullanıcıların girişi engellenmek amacıyla gereken kodlama yapılarak güvenlik sağlanmıştır. Bütün sayfalara tek tek bu işlem yapılması uzun ve zahmetli olacağından projeye “Global” sekmesiyle gereken tanımlamalar yapılmıştır. Aynı zamanda Giriş ve Yeni şifre ve Anasayfa gibi sayfaların bundan etkilenmemesi için de gereken ayrılmıştır kodlar üzerinden.

E-75: AUTHORIZED İŞLEMİ SONRASI ERİŞİM GÜVENSİĞİNİN BÜTÜN PROJELYE YANSITILMASI EKRAN GörÜNTÜSÜ



```
Global.asax.cs  ✖  Index.cshtml  _AdminLayout.cshtml  LoginController.cs  Index.cshtml  DefaultController.cs
MvcCv
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using System.Web.Optimization;
7  using System.Web.Routing;
8
9  namespace MvcCv
10 {
11     public class MvcApplication : System.Web.HttpApplication
12     {
13         protected void Application_Start()
14         {
15             GlobalFilters.Filters.Add(new AuthorizeAttribute());
16             AreaRegistration.RegisterAllAreas();
17             FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
18             RouteConfig.RegisterRoutes(RouteTable.Routes);
19             BundleConfig.RegisterBundles(BundleTable.Bundles);
20         }
21     }
22 }
23
```

Artık proje Hosting ve Domain işlemleri ile canlıya alınmaya hazır hale gelmiştir.