

DataGym

Spor Salonu Yönetim Sistemi

MVC Core • EF Core • SQL Server

Üyelik, ders ve katılım yönetimi tek platformda

Hemen Kayıt Ol

Paketleri Gör



Spor Salonu Yönetim Sistemi(MVC Core & SQL Server)

**Web Programlama Dersi
Final Projesi**

BİRÜNİ ÜNİVERSİTESİ/BİLGİSAYAR MÜHENDİSLİĞİ

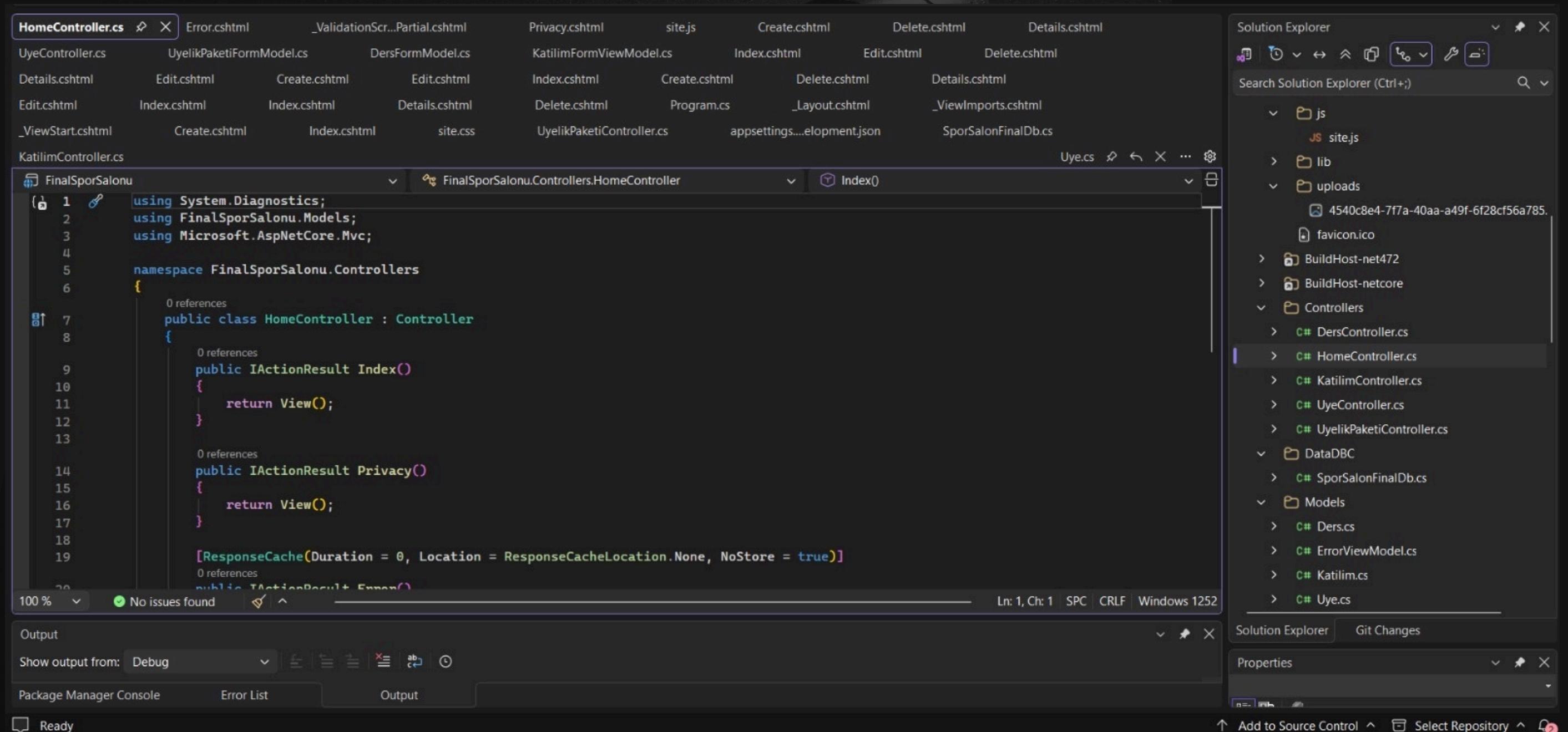
Grup Üyeleri:

ZAHRA ESPARGHAMİ 230404904

NARDIN FARIDASL 230404956

MOHAMMAD PARSA MALEK 230404929

Projenin Amacı



The screenshot shows the Microsoft Visual Studio interface. The code editor displays the `HomeController.cs` file with the following code:

```
1  using System.Diagnostics;
2  using FinalSporSalonu.Models;
3  using Microsoft.AspNetCore.Mvc;
4
5  namespace FinalSporSalonu.Controllers
6  {
7      public class HomeController : Controller
8      {
9          public IActionResult Index()
10         {
11             return View();
12         }
13
14         public IActionResult Privacy()
15         {
16             return View();
17         }
18
19         [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
20         public IActionResult Error()
21     }
22 }
```

The Solution Explorer window on the right lists the project structure:

- js (contains site.js)
- lib
- uploads (contains 4540c8e4-7f7a-40aa-a49f-6f28cf56a785.woff2, favicon.ico)
- Controllers (contains DersController.cs, HomeController.cs, KatilimController.cs, UyeController.cs, UyelikPaketiController.cs)
- DataDBC (contains SporSalonFinalDb.cs)
- Models (contains Ders.cs, ErrorViewModel.cs, Katilim.cs, Uye.cs)

Bu projenin amacı:

Spor salonu için üyelik, ders ve katılım yönetimini tek bir web platformu üzerinden gerçekleştirmektir.

Sistem:

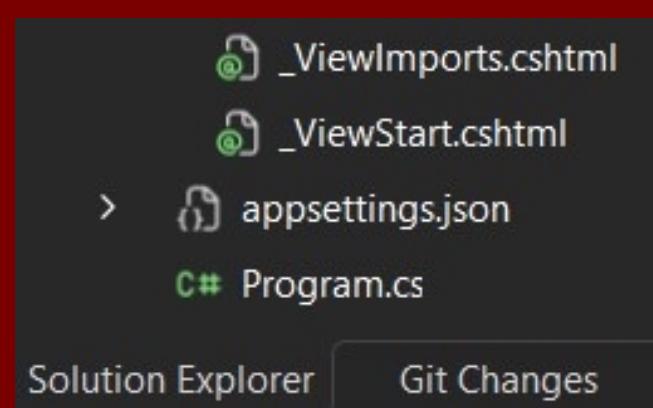
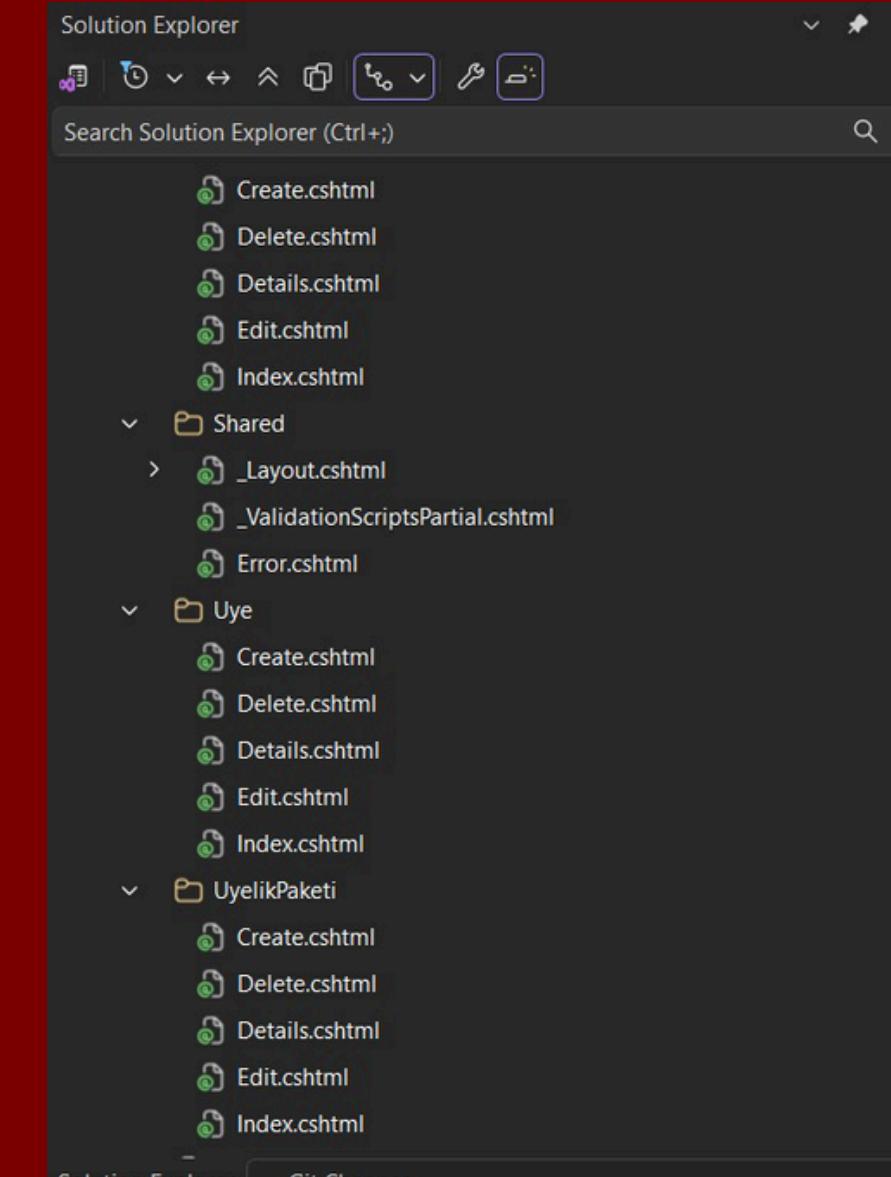
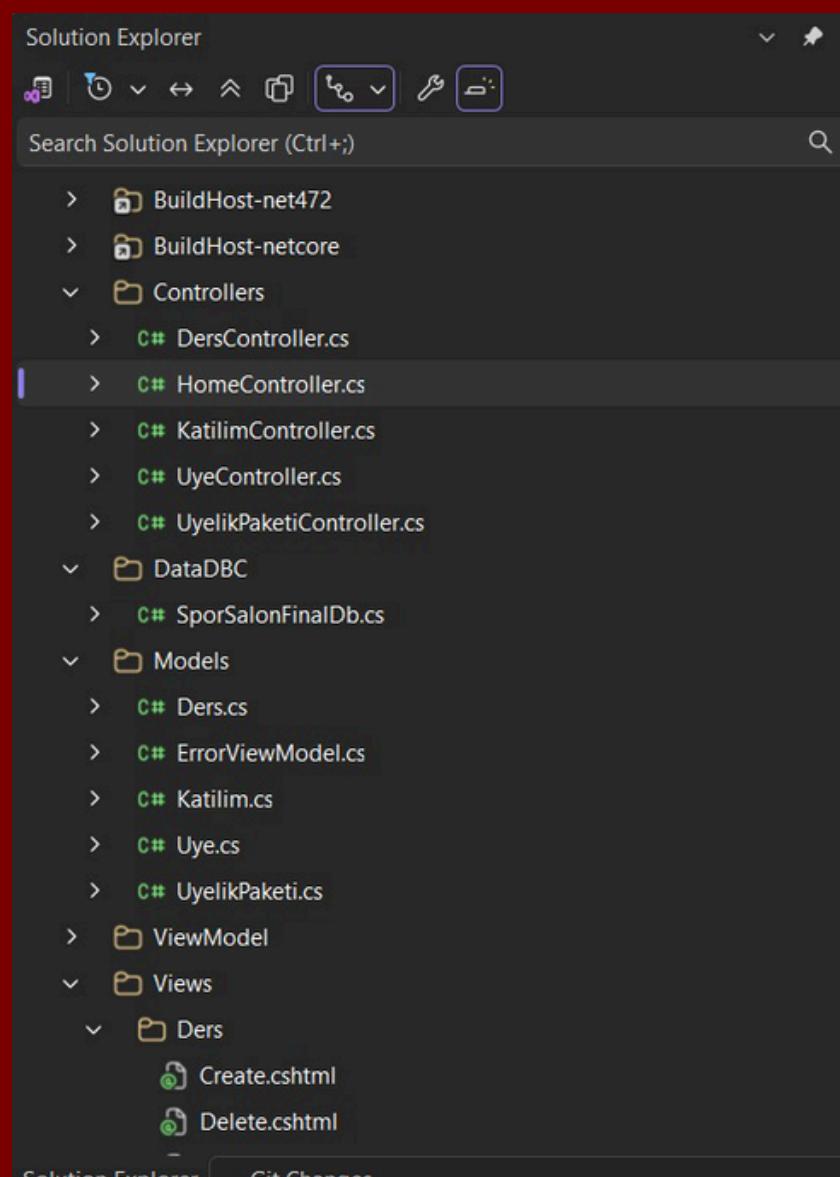
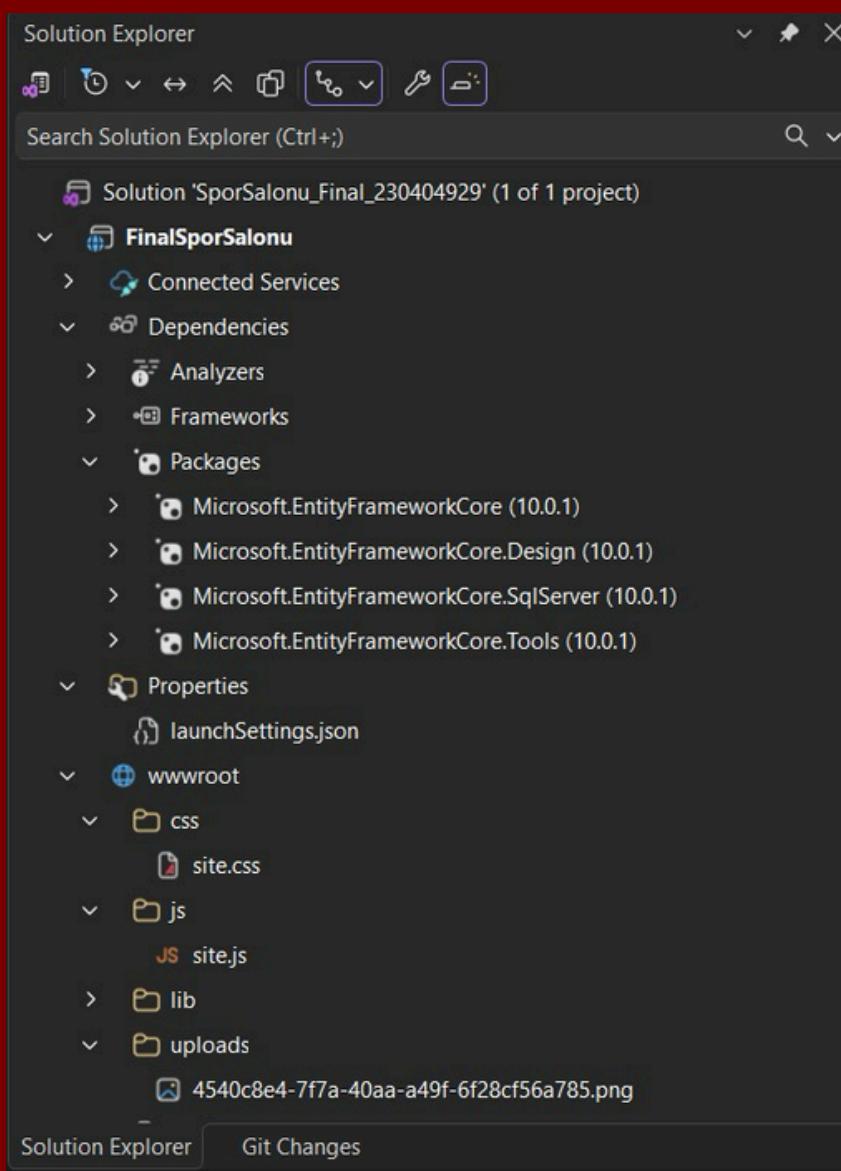
Manuel işlemleri azaltır

Veri tutarlığını sağlar

Yöneticiye kolay kontrol imkânı sunar

Kullanılan Teknolojiler

ASP.NET Core MVC
Entity Framework Core
SQL Server
HTML – CSS – JavaScript
Bootstrap & Bootstrap Icons
ViewModel mimarisi



Veritabanı Tasarımı

Sistem toplam 4 tablo içermektedir:

UyelikPaketi

Uye

Ders

Katilim

Tablolar arasında:

1 – N ilişkileri

Foreign Key yapıları

Normalize edilmiş tasarım bulunmaktadır

100 % No issues found

Results Messages

	Uyeld	Ad	Soyad	Email	Telefon	FotoYolu	UyelikPaketild
1	1	Zahra	Esparghami	zahra@mail.com	0500	NULL	1
2	2	Mohammadparsa	Malek	parsa@gmail.com	05527374227	NULL	6
3	3	Nardin	Faridasl	nardin@gmail.com	0535357423	NULL	5
4	4	Ali	Yilmaz	ali@mail.com	0501	NULL	2

```
1 -- 1) DATABASE
2 CREATE DATABASE SporSalonFinalDb;
3 GO
4 USE SporSalonFinalDb;
5 GO
6
7 -- 2) UYELIK PAKETI
8 CREATE TABLE UyelikPaketi1 (
9     UyelikPaketiId INT IDENTITY PRIMARY KEY,
10    PaketAdi NVARCHAR(50) NOT NULL,
11    SureGun INT NOT NULL,
12    Fiyat DECIMAL(10,2) NOT NULL,
13    PaketResimYolu NVARCHAR(255)
14 );
15 GO
16
17 -- 3) UYE (1 Paket -> Cok Uye)
18 CREATE TABLE Uye1 (
19     UyeId INT IDENTITY PRIMARY KEY,
20     Ad NVARCHAR(50) NOT NULL,
21     Soyad NVARCHAR(50) NOT NULL,
22     Email NVARCHAR(100) NOT NULL,
23     Telefon NVARCHAR(20),
24     FotoYolu NVARCHAR(255),
25     UyelikPaketiId INT NOT NULL,
26
27     CONSTRAINT FK_Uye_Paket
28         FOREIGN KEY (UyelikPaketiId)
29             REFERENCES UyelikPaketi(UyelikPaketiId)
30 );
31 GO
32
33 -- 4) DERS
34 CREATE TABLE Ders (
35     DersId INT IDENTITY PRIMARY KEY,
36     DersAdi NVARCHAR(50) NOT NULL,
37     Gun NVARCHAR(10) NOT NULL,
38     Saat NVARCHAR(5) NOT NULL,
39     Kontenjan INT NOT NULL
40 );
41 GO
42
43 -- 5) KATILIM
44 CREATE TABLE Katilim (
45     UyeId INT NOT NULL,
46     DersId INT NOT NULL,
47     PRIMARY KEY (UyeId, DersId),
48     CONSTRAINT FK_Katilim_Uye
49         FOREIGN KEY (UyeId)
50             REFERENCES Uye1(UyeId),
51     CONSTRAINT FK_Katilim_Ders
52         FOREIGN KEY (DersId)
53             REFERENCES Ders(DersId)
54 );
55 GO
56
57 -- 6) ORNEK VERI
58 INSERT INTO UyelikPaketi (PaketAdi, SureGun, Fiyat)
59 VALUES
60     ('Aylik', 30, 1200),
61     ('3 Aylik', 90, 3200),
62     ('Yillik', 365, 11000);
63
64 INSERT INTO Ders (DersAdi, Gun, Saat, Kontenjan)
65 VALUES
66     ('Yoga', 'Pazartesi', '18:00', 15),
67     ('Pilates', 'Carsamba', '19:00', 20),
68     ('Spinning', 'Cuma', '20:00', 25);
69
70 INSERT INTO Uye (Ad, Soyad, Email, Telefon, UyelikPaketiId)
71 VALUES
72     ('Zahra', 'Esparghami', 'zahra@mail.com', '0500', 1),
73     ('Ali', 'Yilmaz', 'ali@mail.com', '0501', 2);
74
75 INSERT INTO Katilim (UyeId, DersId)
76 VALUES (1,1), (1,2), (2,3);
77 GO
78
79 SELECT name FROM sys.tables;
80 SELECT * FROM sys.foreign_keys;
81
82 SELECT * FROM UyelikPaketi;
83 SELECT * FROM Uye;
84 SELECT * FROM Ders;
85 SELECT * FROM Katilim;
```

100 % ▼  No issues found

	KatilimId	Uyelid	DersId	KayitTarihi
1	1	1	1	2026-01-12 13:44:26.150
2	2	1	2	2026-01-12 13:44:26.150
3	3	2	3	2026-01-12 13:44:26.150
4	4	1	1	2026-01-12 16:04:28.163
5	5	1	2	2026-01-12 16:04:28.163
6	6	2	3	2026-01-12 16:04:28.163

100 % ▼ ✔ No issues found

	UyelikPaketId	PaketAdi	SureGun	Fiyat	PaketResimYolu
1	1	Aylık	30	10000.00	NULL
2	2	3 Aylık	90	3200.00	NULL
3	3	Yıllık	365	11000.00	NULL
4	4	Aylık	30	1200.00	NULL
5	5	3 Aylık	90	4000.00	NULL
6	6	Yıllık	365	15000.00	NULL

100 % ▼  No issues found

	DersId	DersAdı	Gun	Saat	Kontenjan
1	1	Yoga	Pazartesi	18:00	15
2	2	Pilates	Carsamba	19:00	20
3	3	Spinning	Cuma	20:00	25
4	4	Yoga	Pazartesi	18:00	15
5	5	Pilates	Carsamba	19:00	20
6	6	Spinning	Cuma	20:00	25

Entity ve DbContext Yapısı

- Her tablo için ayrı Entity sınıfı
- Tüm DbSet'ler SporSalonFinalDb içinde tanımlıdır
- OnModelCreating ile tablo isimleri eşleştirilmiştir
- modelBuilder.Entity<Uye>().ToTable("Uye");

The screenshot shows a Visual Studio interface. On the left is a code editor window displaying the `SporSalonFinalDb.cs` file. The code defines a `SporSalonFinalDb` class that inherits from `DbContext`. It contains four DbSet properties: `UyelikPaketleri`, `Uyeler`, `Dersler`, and `Katiliklar`. A protected override method `OnModelCreating` is defined to map entities to tables. On the right is a Solution Explorer window showing the project structure. The `Models` folder contains `Ders.cs`, `ErrorViewModel.cs`, `Katilik.cs`, `Uye.cs`, and `UyelikPaketi.cs`. The `ViewModel` folder is collapsed. The `Views` folder is expanded, showing subfolders `Ders` and `Home`. The `Ders` folder contains `Create.cshtml`, `Delete.cshtml`, `Details.cshtml`, `Edit.cshtml`, and `Index.cshtml`. The `Home` folder contains `Index.cshtml`.

```
4 namespace FinalSporSalonu.DataDBC
5 {
6     12 references
7     public class SporSalonFinalDb : DbContext
8     {
9         0 references
10        public SporSalonFinalDb(DbContextOptions<SporSalonFinalDb> options) : base(options) { }
11
12        11 references
13        public DbSet<UyelikPaketi> UyelikPaketleri { get; set; }
14        11 references
15        public DbSet<Uye> Uyeler { get; set; }
16        11 references
17        public DbSet<Ders> Dersler { get; set; }
18        8 references
19        public DbSet<Katilik> Katiliklar { get; set; }
20
21        0 references
22        protected override void OnModelCreating(ModelBuilder modelBuilder)
23        {
24            modelBuilder.Entity<UyelikPaketi>().ToTable("UyelikPaketi");
25            modelBuilder.Entity<Uye>().ToTable("Uye");
26            modelBuilder.Entity<Ders>().ToTable("Ders");
27            modelBuilder.Entity<Katilik>().ToTable("Katilik");
28        }
29    }
```

ViewModel Kullanımı

Projede Entity yerine ViewModel kullanılmıştır.

Örnek ViewModeller:

UyeFormViewModel

KatilimFormViewModel

Amaç:

Dropdown listeleri yönetmek

Form kontrolünü kolaylaştmak

MVC kurallarına uygunluk

```
using Microsoft.AspNetCore.Mvc.Rendering;
using System.ComponentModel.DataAnnotations;

namespace FinalSporSalonu.ViewModels
{
    8 references
    public class KatilimFormViewModel
    {
        3 references
        public int KatilimId { get; set; }

        [Required]
        7 references
        public int UyeId { get; set; }

        [Required]
        7 references
        public int DersId { get; set; }

        6 references
        public List<SelectListItem> Uyeler { get; set; } = new();
        6 references
        public List<SelectListItem> Dersler { get; set; } = new();
    }
}
```

CRUD İşlemleri

Tüm tablolar için:

Create

Read

Update

Delete

İşlemleri uygulanmıştır.

Ders Programı				
Haftalık aktif spor dersleri, saatler ve doluluk oranları.				
Yoga Grup Dersi	Pazartesi	18:00	15 Kiplık	
Pilates Grup Dersi	Carsamba	19:00	20 Kiplık	
Spinning Grup Dersi	Cuma	20:00	25 Kiplık	
Yoga Grup Dersi	Pazartesi	18:00	15 Kiplık	
Pilates Grup Dersi	Carsamba	19:00	20 Kiplık	

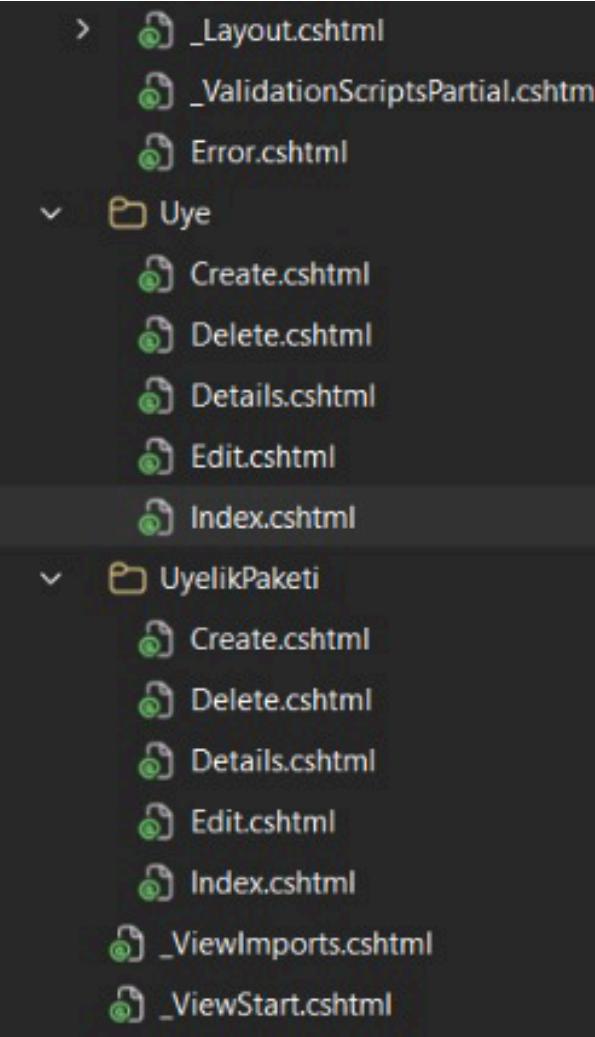
```
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
```

```
<td>
|   <span class="lz-badge-success">Aktif</span>
</td>

<td class="text-end pe-4">
    <div class="d-flex justify-content-end gap-2">
        <a asp-action="Details" asp-route-id="@item.UyeId"
            class="btn btn-sm btn-outline-info d-flex align-items-center gap-1">
            <i class="bi bi-eye-fill"></i> Detay
        </a>

        <a asp-action="Edit" asp-route-id="@item.UyeId"
            class="btn btn-sm btn-outline-primary d-flex align-items-center gap-1">
            <i class="bi bi-pencil-square"></i> Düzenle
        </a>

        <a asp-action="Delete" asp-route-id="@item.UyeId"
            class="btn btn-sm btn-outline-danger d-flex align-items-center gap-1">
            <i class="bi bi-trash3-fill"></i> Sil
        </a>
    </div>
</td>
```



Üyelik Paketleri

FitLogic • Premium Deneyim

Hedeflerinize en uygun paketi seçin ve Parsa-Zahra-Nardin ailesine katılın.



Aylık

Süre: 30 Gün

10,000 ₺

[Detay](#)

[Kayıt Ol](#)



3 Aylık

Süre: 90 Gün

3,200 ₺

[Detay](#)

[Kayıt Ol](#)



Yıllık

Süre: 365 Gün

11,000 ₺

[Detay](#)

[Kayıt Ol](#)



Aylık

Süre: 30 Gün

1,200 ₺

[Detay](#)

[Kayıt Ol](#)



Kullanıcı Arayüzü Tasarımı

Dark Theme

Glassmorphism kartlar

Icon destekli butonlar

Responsive (mobil uyumlu)

Kullanıcı deneyimi ön planda tutulmuştur.

Güvenlik ve Veri Kontrolleri

Foreign Key kısıtlamaları

Null kontrolü

Try-Catch ile hata yönetimi

ModelState doğrulama

Bu sayede:

Hatalı veri girişi engellenmiştir

Sistem kararlılığı sağlanmıştır

```
// VM Create
0 references
public IActionResult Create()
{
    return View(new DersFormViewModel());
}

[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create(DersFormViewModel vm)
{
    if (!ModelState.IsValid) return View(vm);

    var ders = new Ders
    {
        DersAdi = vm.DersAdi,
        Gun = vm.Gun,
        Saat = vm.Saat,
        Kontenjan = vm.Kontenjan
    };

    _context.Add(ders);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}
```

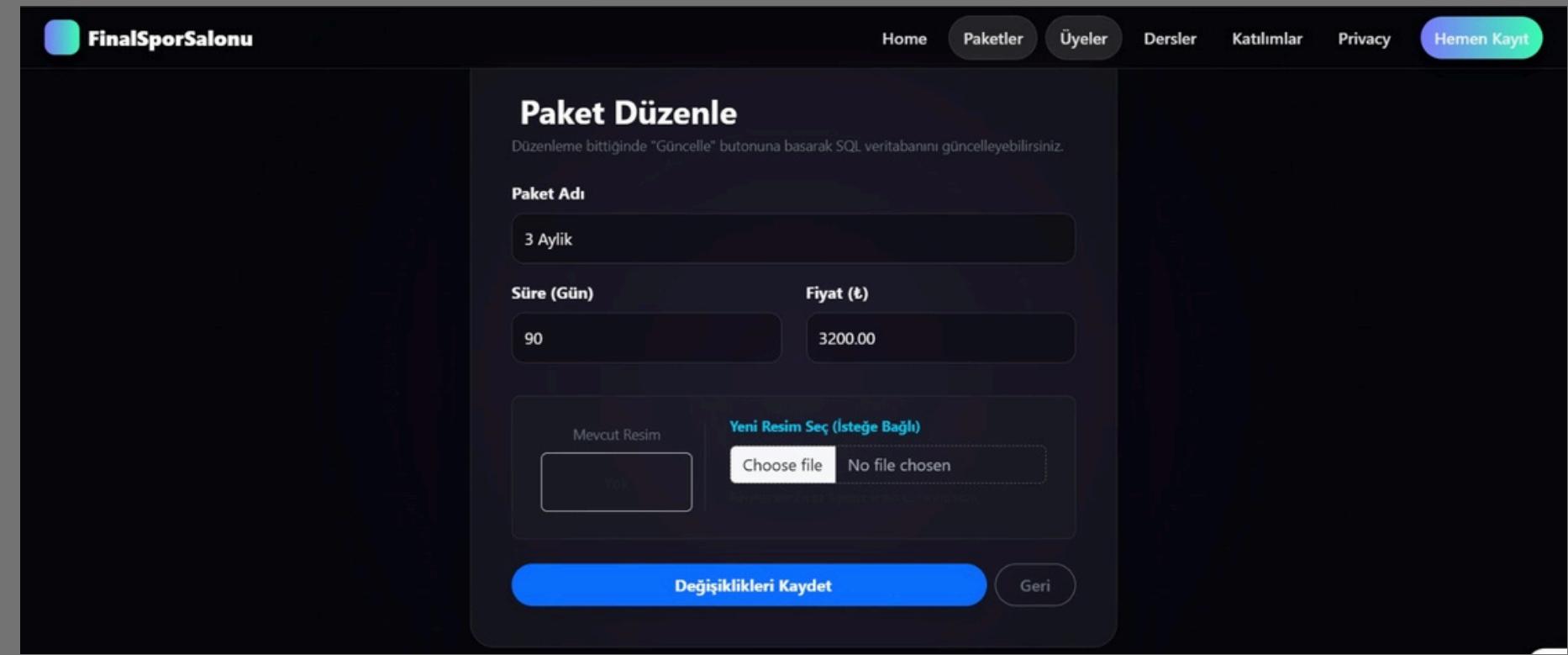
```
// VM Edit
0 references
public async Task<IActionResult> Edit(int? id)
{
    if (id == null) return NotFound();

    var ders = await _context.Dersler.FindAsync(id);
    if (ders == null) return NotFound();

    var vm = new DersFormViewModel
    {
        DersId = ders.DersId,
        DersAdi = ders.DersAdi,
        Gun = ders.Gun,
        Saat = ders.Saat,
        Kontenjan = ders.Kontenjan
    };

    return View(vm);
}

[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Edit(int id, DersFormViewModel vm)
```



Sonuç ve Değerlendirme

Bu proje sayesinde:

MVC mimarisi pekiştirilmiştir

EF Core – SQL Server entegrasyonu öğrenilmiştir

Gerçek hayata uygun bir sistem geliştirilmiştir