



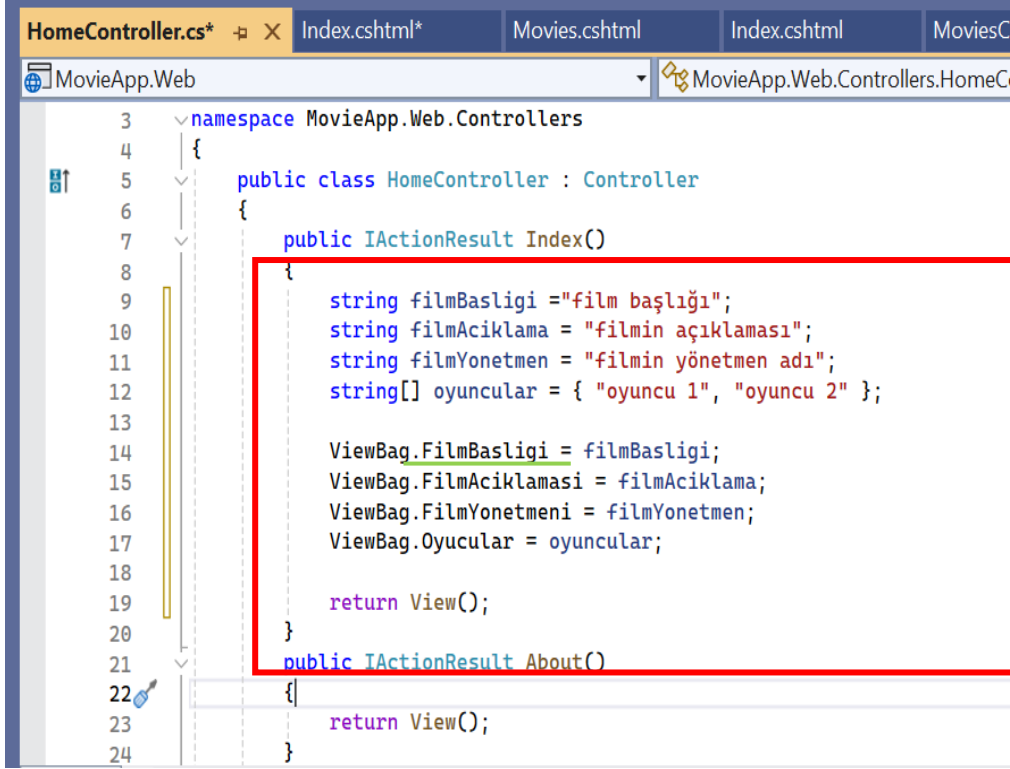
ANKARA
HACI BAYRAM VELİ ÜNİVERSİTESİ
**UZAKTAN EĞİTİM UYGULAMA VE
ARAŞTIRMA MERKEZİ**

İNTERNET PROGRAMCILIĞI

Bu bölümde dinamik bir Html sayfası oluşturma ve verileri model kullanarak view ortamına taşıma konuları üzerine uygulama yapılmıştır.

1. DİNAMİK BİR HTML SAYFASI OLUŞTURMAK

ADIM 1: Model kullanmadan HomeController içindeki bilgileri Index.cshtml sayfasına taşımak için ViewBag Komutunu kullanıyoruz. (Model Kullanarak aktarmak sonraki adımlarda)



```

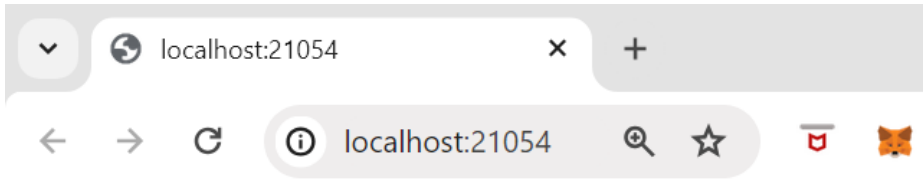
3  namespace MovieApp.Web.Controllers
4  {
5      public class HomeController : Controller
6      {
7          public IActionResult Index()
8          {
9              string filmBasligi = "film başlığı";
10             string filmAciklama = "filmin açıklaması";
11             string filmYonetmen = "filmin yönetmen adı";
12             string[] oyuncular = { "oyuncu 1", "oyuncu 2" };
13
14             ViewBag.FilmBasligi = filmBasligi;
15             ViewBag.FilmAciklamasi = filmAciklama;
16             ViewBag.FilmYonetmeni = filmYonetmen;
17             ViewBag.Oyucular = oyuncular;
18
19             return View();
20         }
21         public IActionResult About()
22         {
23             return View();
24         }
25     }

```

ADIM 2: HomeController a karşılık gelecek Index.cshtml sayfasının hazırlanması için gerekli kodlar. ViewBag aracılığı ile ilgili View e değişkenleri taşıyabiliriz. Oyuncular değişkeninin dizi olarak tanımlandığına ve döngü ile çağırıldığına dikkat ediniz.

```
Index.cshtml  HomeController.cs
MovieApp.Web
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head><title></title></head>
3 <body>
4 <h1> Film Listesi</h1>
5 <div id="filmler">
6 <div class="film">
7 <p>@ViewBag.FilmBasligi</p>
8 <p>@ViewBag.FilmAciklamasi</p>
9 <p>@ViewBag.FilmYonetmeni</p>
10 <p>
11 <ul>
12 <li>@foreach (var item in ViewBag.Oyuncular)
13 {
14 <li>@item</li>
15 }
16 </ul>
17 </p>
18 </div>
19
20 </div>
21
22
```

ADIM 3: Çalıştırılır.



Film Listesi

film başlığı

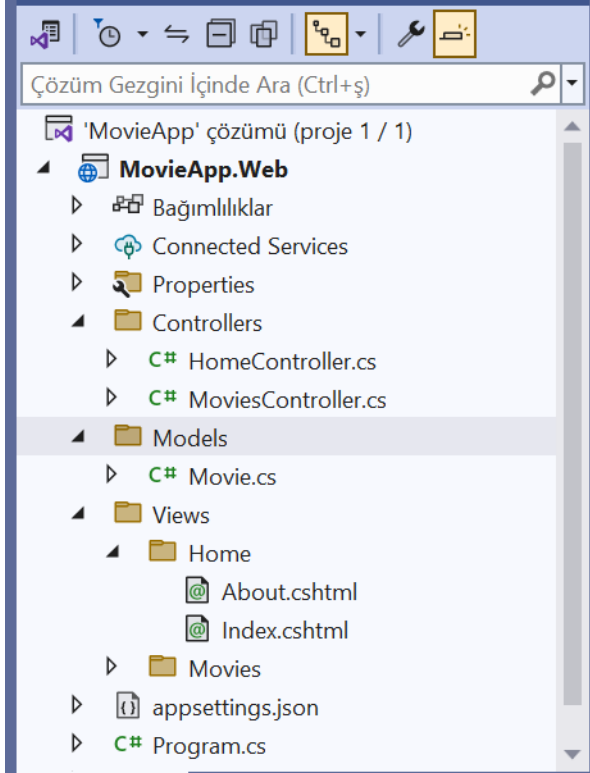
filmin açıklaması

filmin yönetmen adı

- oyuncu 1
- oyuncu 2

2. AYNI VERİLERİ MODEL KULLANARAK VIEW E TAŞIMAK

ADIM 1: Model Sağ tıklanarak yeni bir class/sınıf eklenir. Movie olarak isimlendirilir.



ADIM 2: Daha önce HomeController da ViewBag metodu kullanarak gönderdiğimiz değişkenlere karşılık gelecek şekilde yeni değişken isimleri, oluşturduğumuz yeni movie ismindeki class a yazılır . Değişken isimleri İngilizce olarak düzenlensin.

```

1 namespace MovieApp.Web.Models
2 {
3     public class Movie
4     {
5         public string Title { get; set; }
6         public string Description { get; set; }
7         public string Director { get; set; }
8         public string [] Players { get; set; }
9     }
10 }
11

```

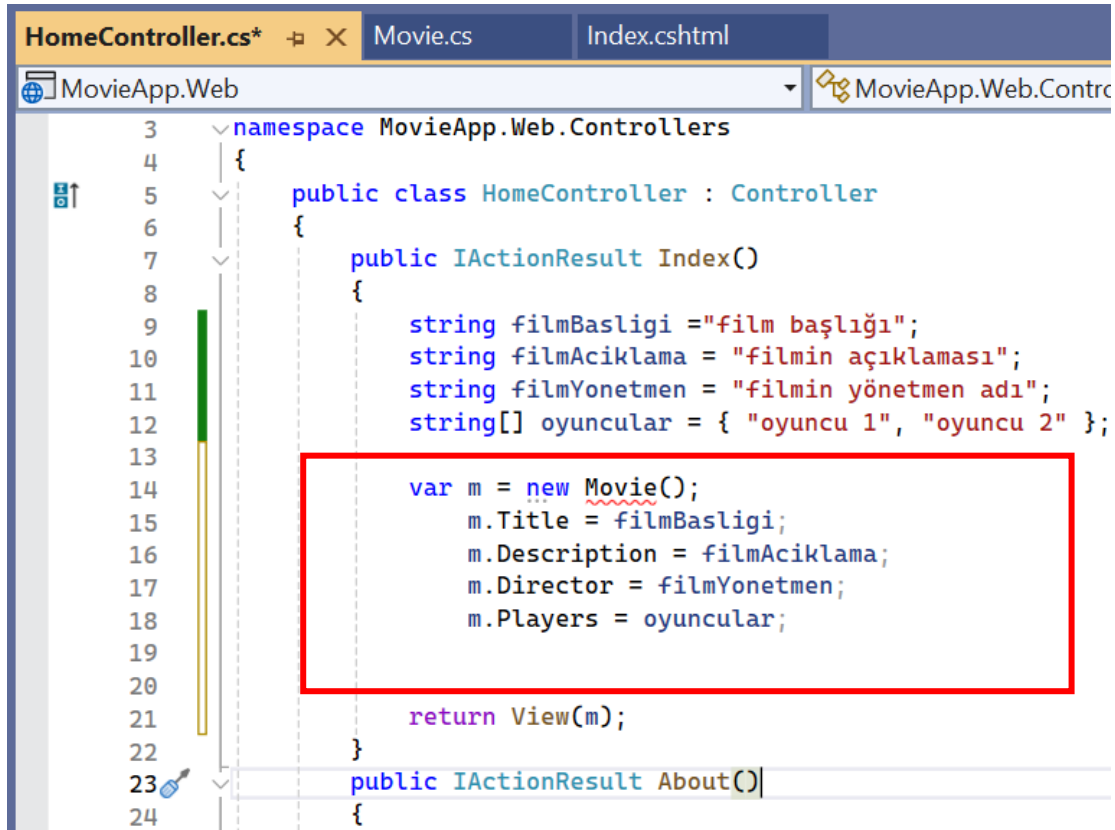
ADIM 3: ViewBag komutuna artık ihtiyacımız kalmadı. **Kod Bloğu silinir**

```

4 {
5     public class HomeController : Controller
6     {
7         public IActionResult Index()
8         {
9             string filmBasligi = "film başlığı";
10            string filmAciklama = "filmin açıklaması";
11            string filmYonetmen = "filmin yönetmen adı";
12            string[] oyuncular = { "oyuncu 1", "oyuncu 2" };
13
14            ViewBag.FilmBasligi = filmBasligi;
15            ViewBag.FilmAciklama = filmAciklama;
16            ViewBag.FilmYonetmeni = filmYonetmen;
17            ViewBag.Oyuncular = oyuncular;
18
19            return View();
20        }
21        public IActionResult About()
22        {
23            return View();
24        }
25    }

```

ADIM 3 : ViewBag kod bloğu yerine geçecek olan ve model kullanarak verileri aktarmamızı sağlayacak yeni kod bloğu HomeController a eklenir.

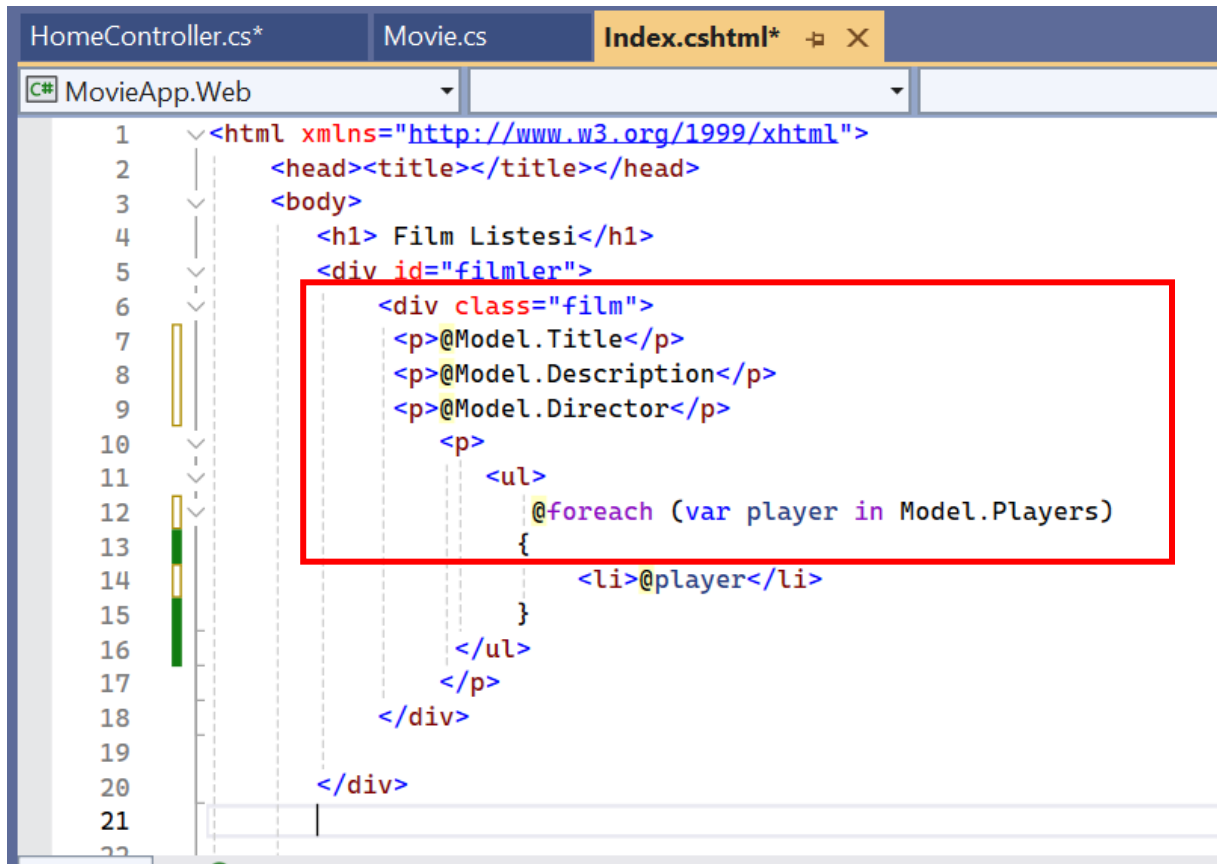


```

3 namespace MovieApp.Web.Controllers
4 {
5     public class HomeController : Controller
6     {
7         public IActionResult Index()
8         {
9             string filmBasligi = "film başlığı";
10            string filmAciklama = "filmin açıklaması";
11            string filmYonetmen = "filmin yönetmen adı";
12            string[] oyuncular = { "oyuncu 1", "oyuncu 2" };
13
14            var m = new Movie();
15            m.Title = filmBasligi;
16            m.Description = filmAciklama;
17            m.Director = filmYonetmen;
18            m.Players = oyuncular;
19
20
21            return View(m);
22        }
23        public IActionResult About()
24    {

```

ADIM 3: Index.cshtml sayfası Movie.cs ve HomeController a göre düzenlenir. Artık ViewBag değil Model kullanıyoruz.



```

1 <html xmlns="http://www.w3.org/1999/xhtml">
2     <head><title></title></head>
3     <body>
4         <h1> Film Listesi</h1>
5         <div id="filmler">
6             <div class="film">
7                 <p>@Model.Title</p>
8                 <p>@Model.Description</p>
9                 <p>@Model.Director</p>
10                <p>
11                    <ul>
12                        @foreach (var player in Model.Players)
13                        {
14                            <li>@player</li>
15                        }
16                    </ul>
17                </p>
18            </div>
19        </div>
20    </div>
21
22

```

ADIM 4: @model MovieApp.Web.Models.Movie kod satırı ile Index.cshtml ve Model olan Movie.cs dosyası arasındaki bağlantı kurulur.

```

1  @model MovieApp.Web.Models.Movie
2
3  <html xmlns="http://www.w3.org/1999/xhtml">
4  <head><title></title></head>
5  <body>
6  <h1> Film Listesi</h1>
7  <div id="filmler">
8  <div class="film">
9  <p>@Model.Title</p>
10 <p>@Model.Description</p>
11 <p>@Model.Director</p>
12 <p>
13 <ul>
14 <foreach (var player in Model.Players)
15 {
16 <li>@player</li>
17 }
18 </ul>
19 </p>
20 </div>
21 </div>
22 </body>
23 </html>

```

89 % Sorun bulunamadı Sat: 10 Krkt: 38

Çıktı

ADIM 5: Home Controller a using MovieApp.Web.Models; kod satırı eklenir.

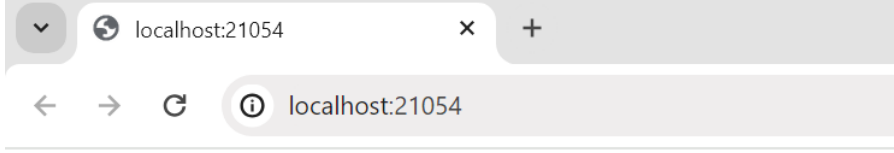
```

1  using Microsoft.AspNetCore.Mvc;
2  using MovieApp.Web.Models;
3
4  namespace MovieApp.Web.Controllers
5  {
6  public class HomeController : Controller
7  {
8  public IActionResult Index()
9  {
10 string filmBasligi = "film başlığı";
11 string filmAciklama = "filmin açıklaması";
12 string filmYonetmen = "filmin yönetmen adı";
13 string[] oyuncular = { "oyuncu 1", "oyuncu 2" };
14
15 var m = new Movie();
16 m.Title = filmBasligi;
17 m.Description = filmAciklama;
18 m.Director = filmYonetmen;
19 m.Players = oyuncular;
20
21 return View(m);
22 }
23 }

```

89 % Sorun bulunamadı

ADIM 6: Çalıştırılır.



Film Listesi

film başlığı

filmin açıklaması

filmin yönetmen adı

- oyuncu 1
- oyuncu 2

ÇALIŞMA SORUSU

Bölüm 2’de daha önceden ViewBag komutu kullanarak yapılan veri taşıma işlemi bu bölümde Model kullanarak gerçekleştirildi.

Sizde dinamik bir html sayfası oluşturarak Model katmanı aracılığı ile veri taşıma işlemini gerçekleştiriniz.