



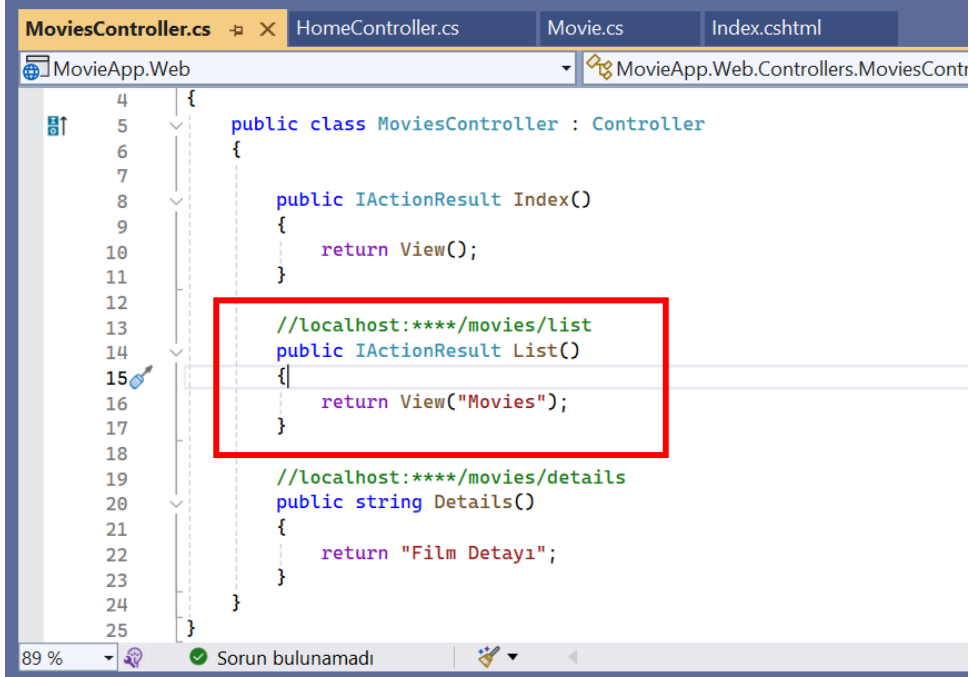
ANKARA
HACI BAYRAM VELİ ÜNİVERSİTESİ
**UZAKTAN EĞİTİM UYGULAMA VE
ARAŞTIRMA MERKEZİ**

İNTERNET PROGRAMCILIĞI

Bu bölümde bir liste yapısını Controller Tarafından View Tarafına Göndermek ve Razor Kullanımı Static Dosyalarla ile Çalışmak, Bootstrap Kütüphanesi kullanımı ve Layout Oluşturma konuları üzerine uygulama yapılmıştır.

1. BİR LİSTE YAPISINI CONTROLLER TARAFINDAN VIEW TARAFINA GÖNDERMEK VE RAZOR KULLANIMI

ADIM 1: Daha önce Movies Controller da bir list tanımlamıştık. (Movies/list in karşılığı)



```

4      {
5      }
6      public class MoviesController : Controller
7      {
8          public IActionResult Index()
9          {
10             return View();
11         }
12
13         //localhost:****/movies/list
14         public IActionResult List()
15         {
16             return View("Movies");
17         }
18
19         //localhost:****/movies/details
20         public string Details()
21         {
22             return "Film Detayı";
23         }
24     }
25

```

ADIM 2: Gerekli kütüphaneleri kontrol ediniz. Bu iki kütüphane ekli olmalı.

```

using MovieApp.Web.Models;
using System.Collections.Generic;

```

```

1  using Microsoft.AspNetCore.Mvc;
2  using MovieApp.Web.Models;
3  using System.Collections.Generic;
4
5  namespace MovieApp.Web.Controllers
6  {
7      public class MoviesController : Controller
8      {
9
10         public IActionResult Index()
11         {
12             return View();
13         }
14
15         //localhost:****/movies/list
16         public IActionResult List()
17         {
18             var filmlistesi = new List<Movie>()
19             {
20
21             };
22         }
23     }
24 }

```

ADIM 3: Film listesi için gerekli kodların MoviesController da yazımı.

```

14
15 //localhost:****/movies/list
16 public IActionResult List()
17 {
18     var filmlistesi = new List<Movie>()
19     {
20         new Movie
21         {
22             Title="film 1",
23             Description="açıklama 1",
24             Director="Yönetmen 1",
25             Players=new string[] {"oyuncu 1", "oyuncu 2"}
26         },
27         new Movie
28         {
29             Title="film 2",
30             Description="açıklama 1",
31             Director="Yönetmen 1",
32             Players=new string[] {"oyuncu 1", "oyuncu 2"}
33         },
34         new Movie
35         {
36             Title="film 3",
37             Description="açıklama 1",
38             Director="Yönetmen 1",
39             Players=new string[] {"oyuncu 1", "oyuncu 2"}
40         }
41     };
42     return View("Movies");
43 }
44

```

ADIM 4: Return kısmında ilk parametre çağırılacak olan view (Movies.cshtml) olmalı, ikinci parametre Model bilgisi (filmListesi) olmalıdır.

```

23         Description="açıklama 1",
24         Director="Yönetmen 1",
25         Players=new string[] { "oyuncu 1", "oyuncu 2" }
26     },
27     new Movie
28     {
29         Title="film 2",
30         Description="açıklama 1",
31         Director="Yönetmen 1",
32         Players=new string[] { "oyuncu 1", "oyuncu 2" }
33     },
34     new Movie
35     {
36         Title="film 3",
37         Description="açıklama 1",
38         Director="Yönetmen 1",
39         Players=new string[] { "oyuncu 1", "oyuncu 2" }
40     }
41 };
42 return View("Movies", filmListesi);
43

```

ADIM 5: Controller tarafından gönderdiğimiz listeyi Movies.cshtml de karşılığını oluşturan kod satırını yazıyoruz. **İlgili cshtml dosyalarında sürekli namespace i tekrarlamamak için adımlara devam ediniz.**

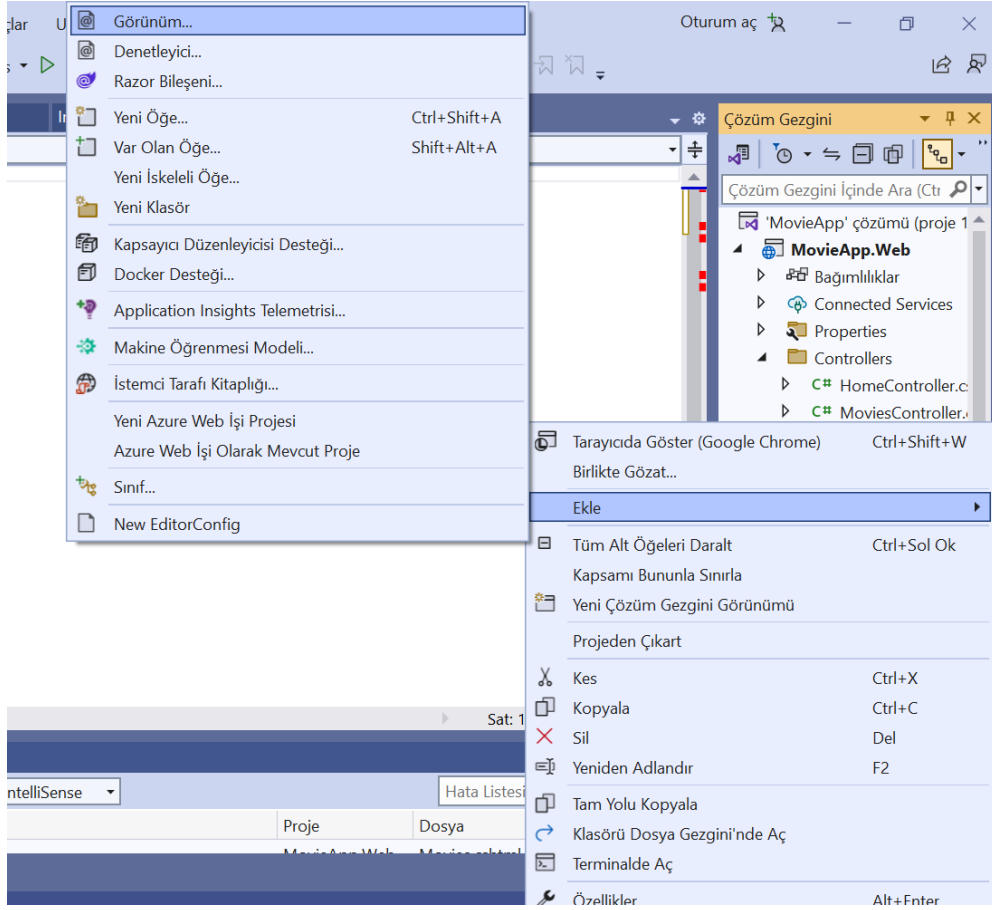
```

1 @model List<Movie>
2
3
4 <html xmlns="http://www.w3.org/1999/xhtml">
5     <head><title></title></head>
6     <body>
7         movies/movies.cshtml
8     </body>
9 </html>

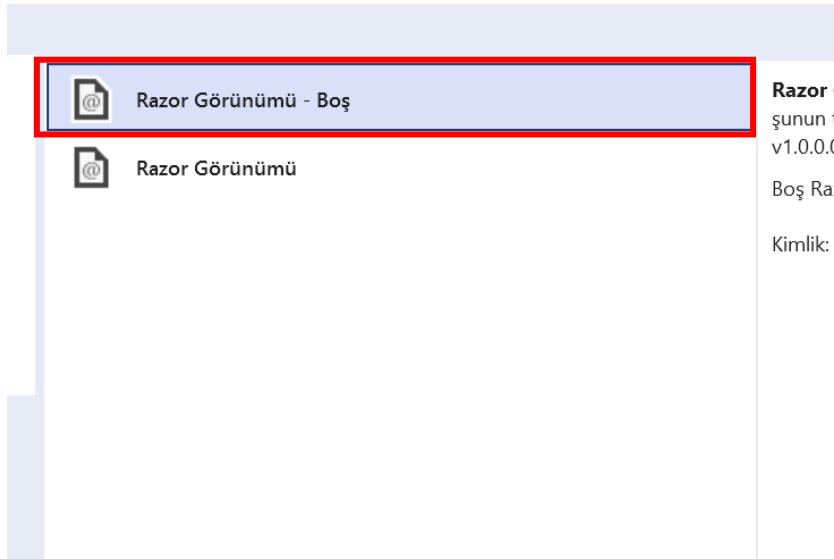
```

ADIM 6: Views klasörüne sağ tıklama görünüm ekle

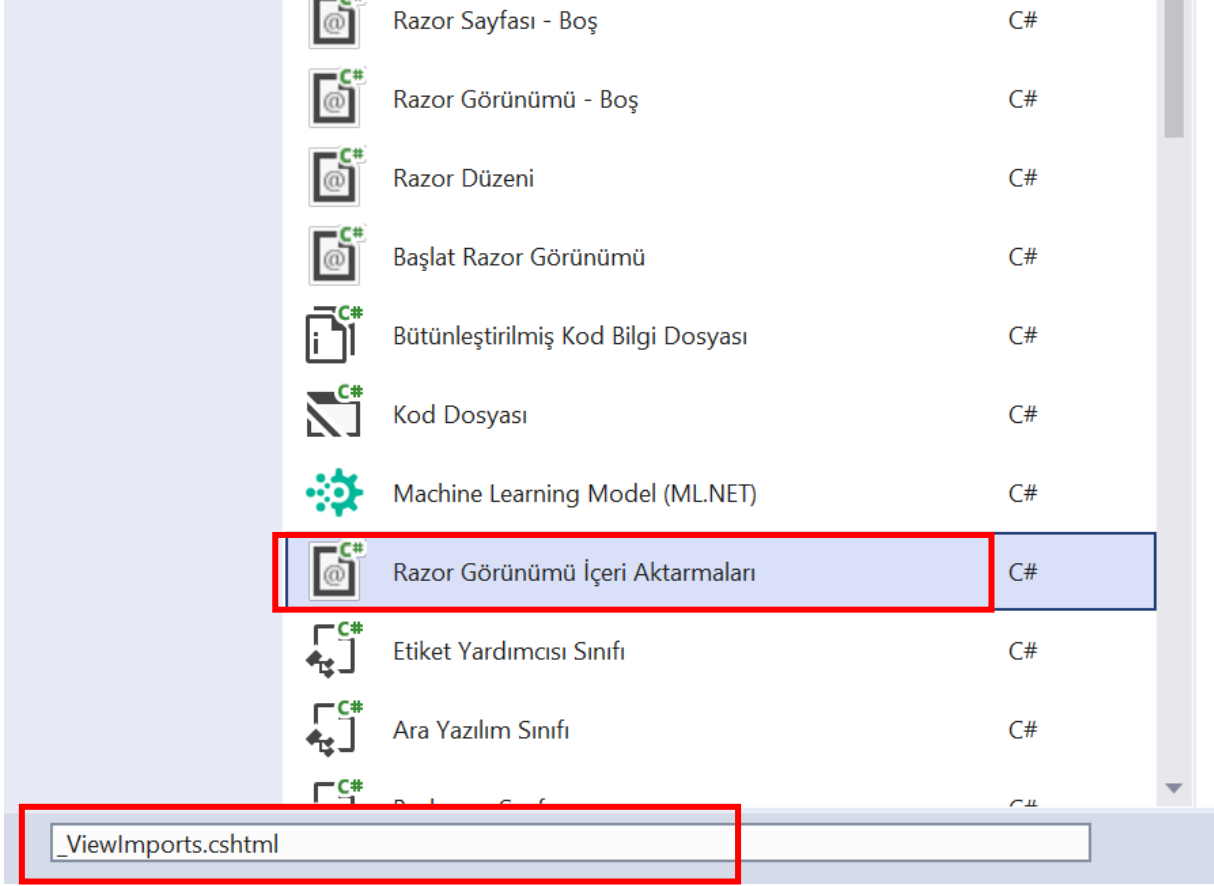




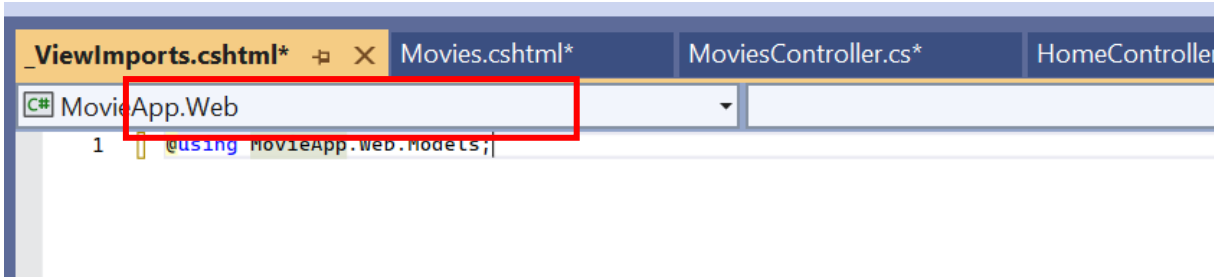
ADIM 7: Boş Razor Dosyası eklenir.



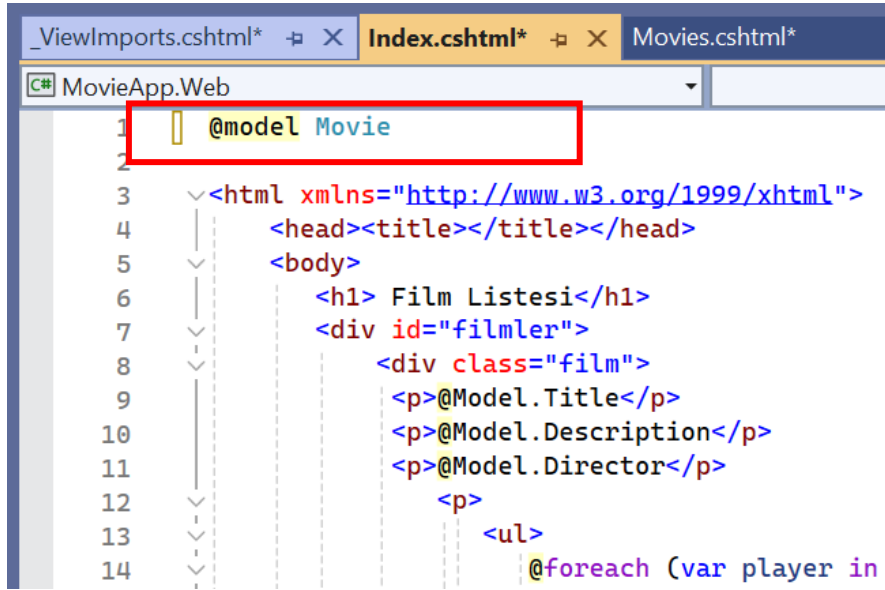
ADIM 8: Razor Görünümünün tipi seçilir.



ADIM 9: İlgili cshtml dosyalarında sürekli namespace i tekrarlamamak için; gerekli kod satırı '_ViewImports.cshtml' 'e eklenir.



ADIM 10: Aynı işlem Index.cshtml için de tekrarlanır.



```
1  @model 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
```

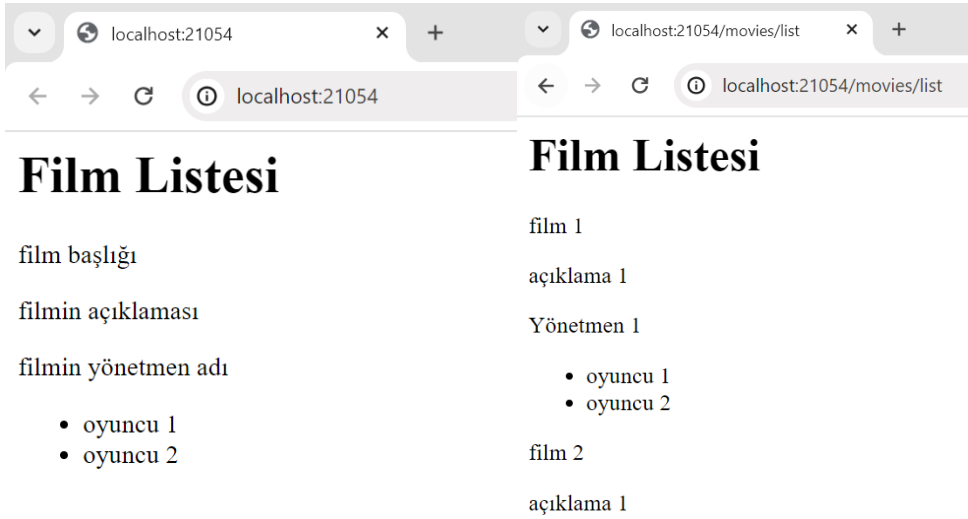
ADIM 11: MoviesController da oluşturduğumuz liste yapısının karşılığını Movies.cshtml de oluşturuyoruz.

```

5 <head><title></title></head>
6 <body>
7 <h1> Film Listesi</h1>
8 <div id="filmler">
9     @foreach (var movie in Model)
10     {
11         <div class="film">
12             <p>@movie.Title</p>
13             <p>@movie.Description</p>
14             <p>@movie.Director</p>
15             <p>
16                 <ul>
17                     @foreach (var player in movie.Players)
18                     {
19                         <li>@player</li>
20                     }
21                 </ul>
22             </p>
23         </div>
24     }
25 </div>
26 </body>
27 </html>

```

ADIM 12: Çalıştırarak kontrol ediyoruz. Movies/List de üç film bilgisi (film listesi), Anasayfada tek film bilgisi olmalı.



ADIM 13: Daha güzel görünüm için Movies.cshtml dosyasına css kod bloğu ekleyelim.

```
Index.cshhtml  Movies.cshhtml*  _ViewImports.cshhtml
C# MovieApp.Web
1  @model List<Movie>
2
3
4  <html xmlns="http://www.w3.org/1999/xhtml">
5    <head>
6      <title></title>
7      <style>
8        .film{
9          border:1px solid #000000;
10         margin-bottom:10px;
11         padding:10px;
12       }
13     </style>
14   </head>
15   <body>
16
17     <h1> Film Listesi</h1>
18     <div id="filmler">
19       @foreach (var movie in Model)
20       {
21         <div class="film">
```

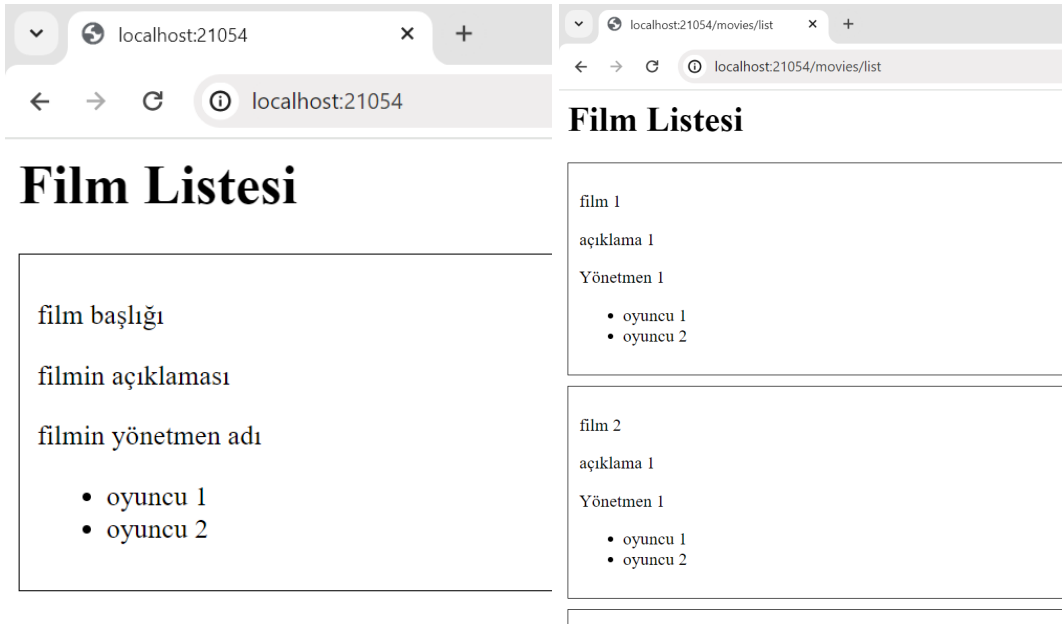
ADIM 14: Aynı kodlar Index.cshtml ye de kopyalanır.

```

1  @model Movie
2
3  <html xmlns="http://www.w3.org/1999/xhtml">
4  <head><title></title>
5  <style>
6      .film {
7          border: 1px solid #000000;
8          margin-bottom: 10px;
9          padding: 10px;
10     }
11
12 </style>
13 </head>
14 <body>
15     <h1> Film Listesi</h1>
16     <div id="filmler">
17         <div class="film">

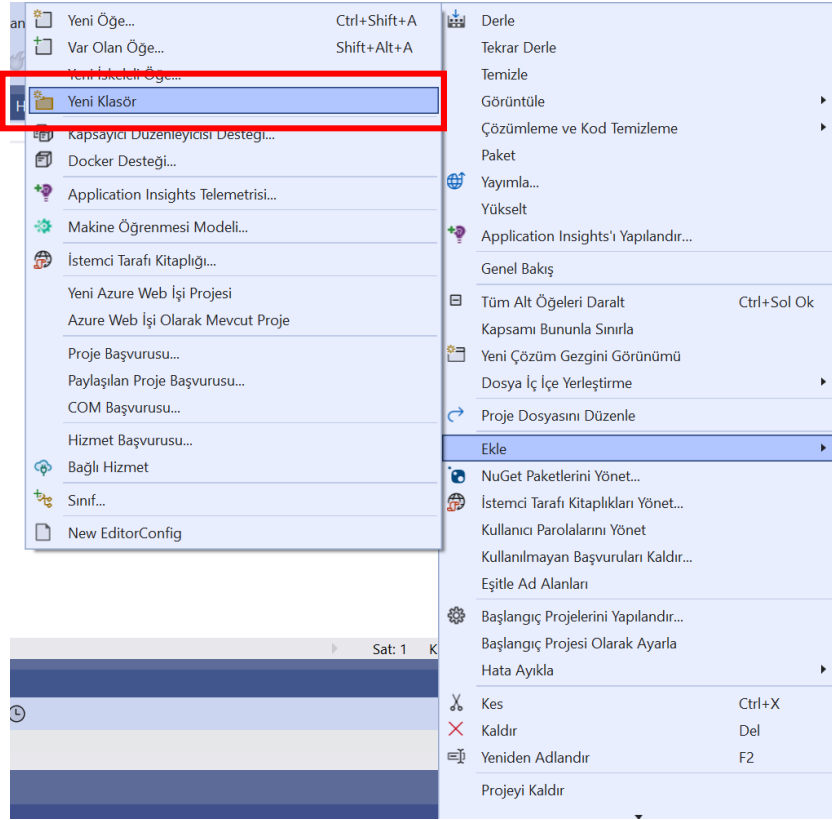
```

ADIM 15: Çalıştırarak deniyoruz.

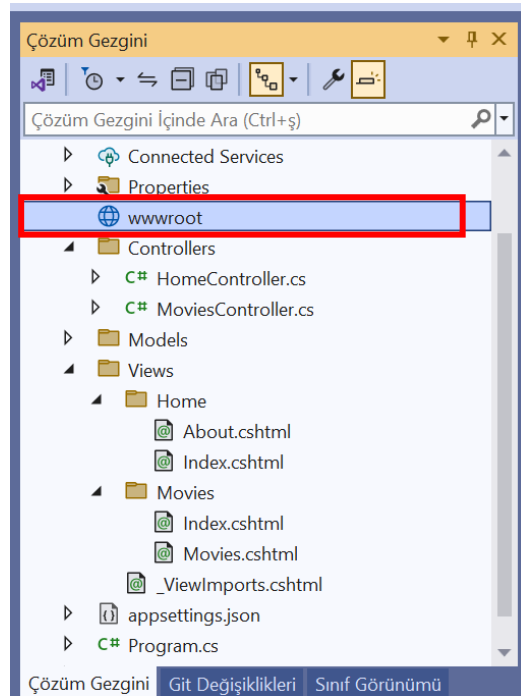


2. STATIC DOSYALARLA (RESİM DOSYALARI) İLE ÇALIŞMAK

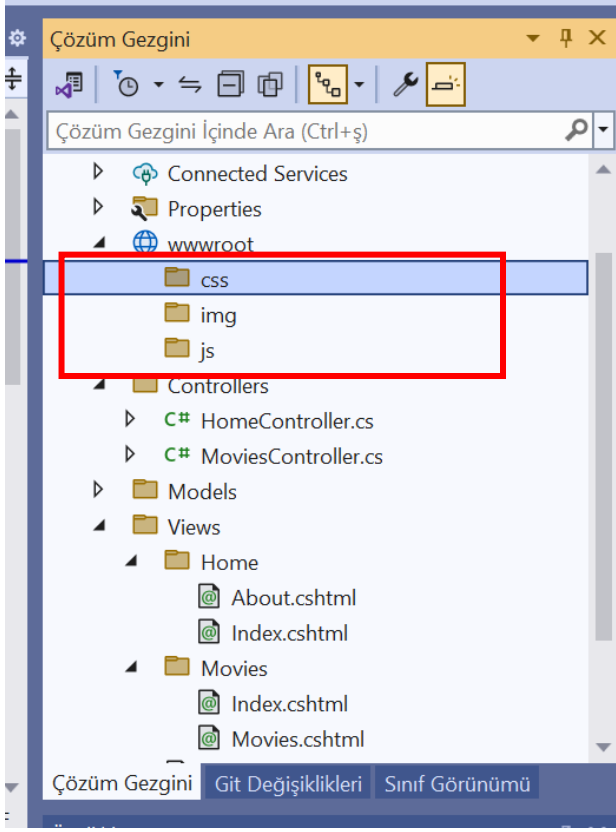
ADIM 1: Uygulamada bir resim dosyası kullanmak için öncelikle **web projesine** sağ tıklanır ve yeni klasör eklenir.



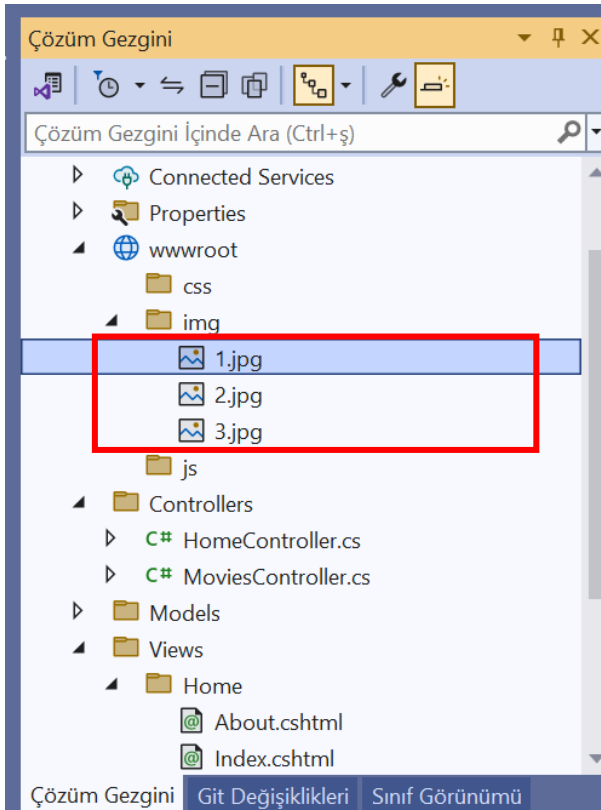
ADIM 2: Klasörü **wwwroot** olarak isimlendir. İsimlendirme sonrası dosyanın ikonunun değiştiği görülmektedir.



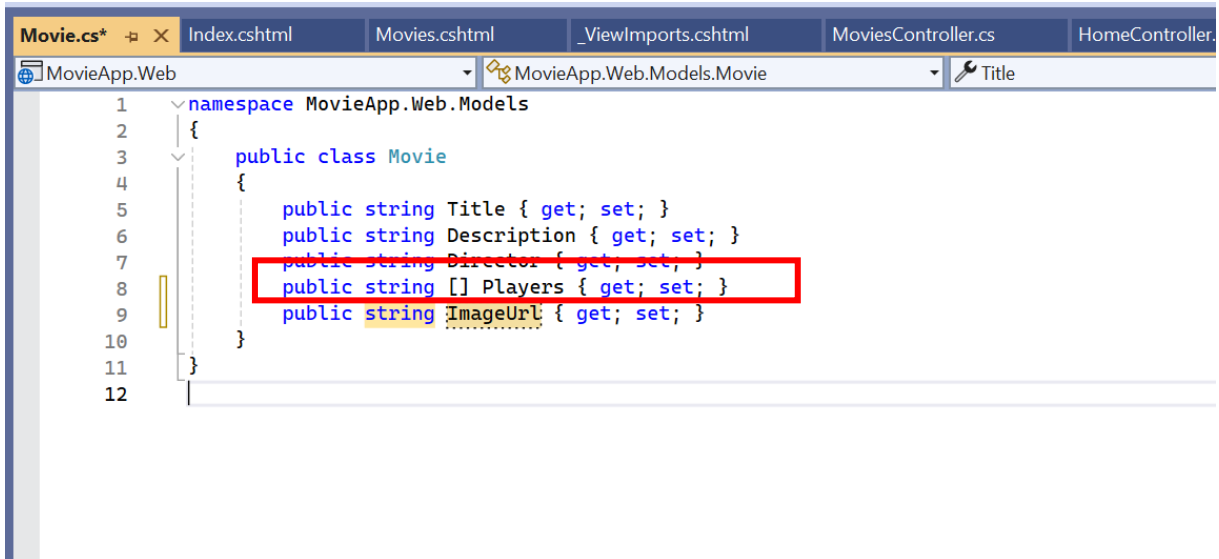
ADIM 3: wwwroot klasörü altına üç yeni klasör eklenir. Klasörlerden birincisi js, ikincisi img, üçüncüsü css olarak isimlendirilir.



ADIM 4: Daha önce masaüstümüze indirmiş olduğumuz film resimlerini img klasörü içerisine kopyalıyoruz. Yüklemeden önce Resim boyutlarının aynı olmasına dikkat ediniz.



ADIM 5: Daha önce filmlere ilişkin taşıma işlemini model kullanarak yapmıştık. Model klasörü altındaki Movie.cs dosyasındaki kodları resimleri ekleyecek şekilde güncelliyoruz.



ADIM 6: MoviesController da da kodlar karşılayacak şekilde düzenlenir. (3 kod bloğunda da)

```

MoviesController.cs*  Index.cshtml  Movies.cshtml  _ViewImports.cshtml
b  MovieApp.Web.Controllers.MoviesController  List()

new Movie
{
    Title="film 1",
    Description="açıklama 1",
    Director="Yönetmen 1",
    Players=new string[] {"oyuncu 1", "oyuncu 2"},
    ImageUrl="1.jpg"
},
new Movie
{
    Title="film 2",
    Description="açıklama 1",
    Director="Yönetmen 1",
    Players=new string[] {"oyuncu 1", "oyuncu 2"},
    ImageUrl="2.jpg"
},
new Movie
{
    Title="film 3",
    Description="açıklama 1",
    Director="Yönetmen 1",
    Players=new string[] {"oyuncu 1", "oyuncu 2"},
    ImageUrl="3.jpg"
}
};
return View("Movies", filmlistesi);

```

Sorun bulunamadı

ADIM 6: Movies.cshtml içinde resimlerin yolu gösterilir.

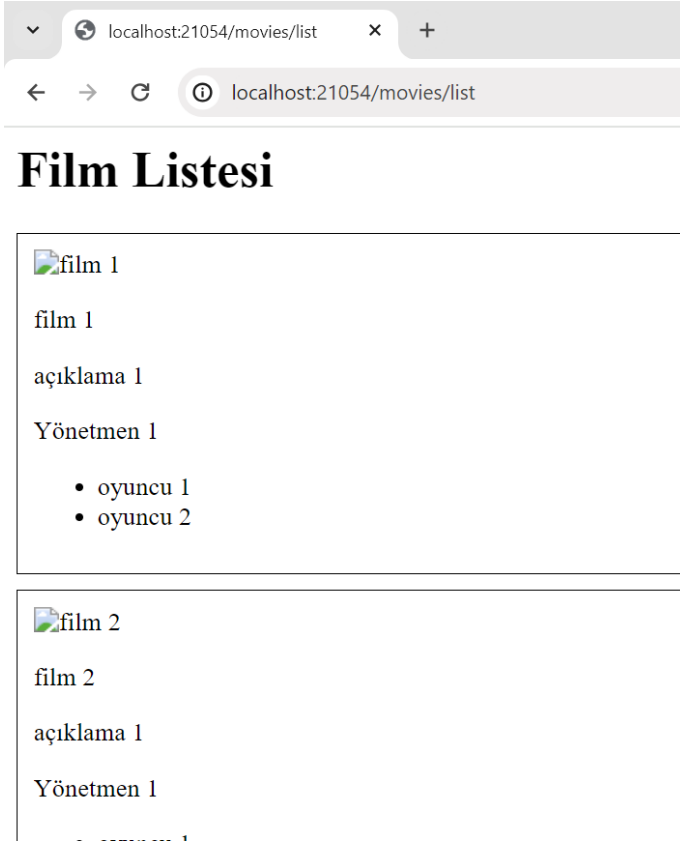
```

Movies.cshtml*  MoviesController.cs*  Movie.cs*  _ViewImports.cs
</head>
<body>

<h1> Film Listesi</h1>
<div id="filmler">
    @foreach (var movie in Model)
    {
        <div class="film">
            
            <p>@movie.Title</p>
            <p>@movie.Description</p>
            <p>@movie.Director</p>
            <p>
                <ul>
                    @foreach (var player in movie.Players)
                    {
                        <li>@player</li>
                    }
                </ul>
            </p>
        </div>
    }
</div>

```

ADIM 7: Çalıştırılır. Resimlerin alanlarının olduğu fakat resimlerin aktif olmadığını görüyoruz. Resimlerin aktif olması için bir düzenlemeye ihtiyacımız var.



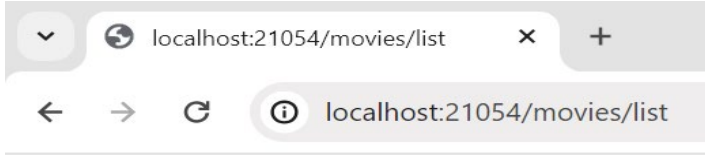
ADIM 8: **Startup.cs** dosyası içine **wwwroot** klasörünü kullanıma açacak kod satırı eklenir. Configure kod bloğu içine.

```
// This method gets called by the runtime. Use this method to configure the HTTP
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseStaticFiles(), //wwwroot

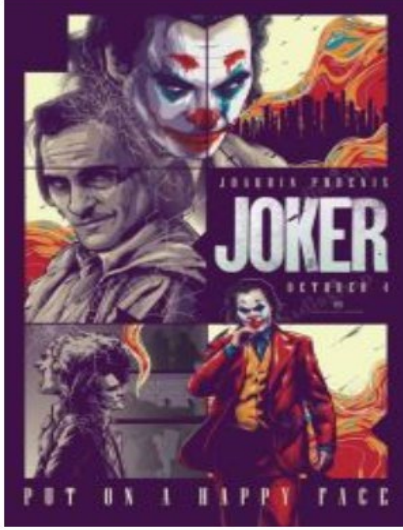
    app.UseRouting();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}")
    })
}
```

ADIM 9: Çalıştırılır.



Film Listesi

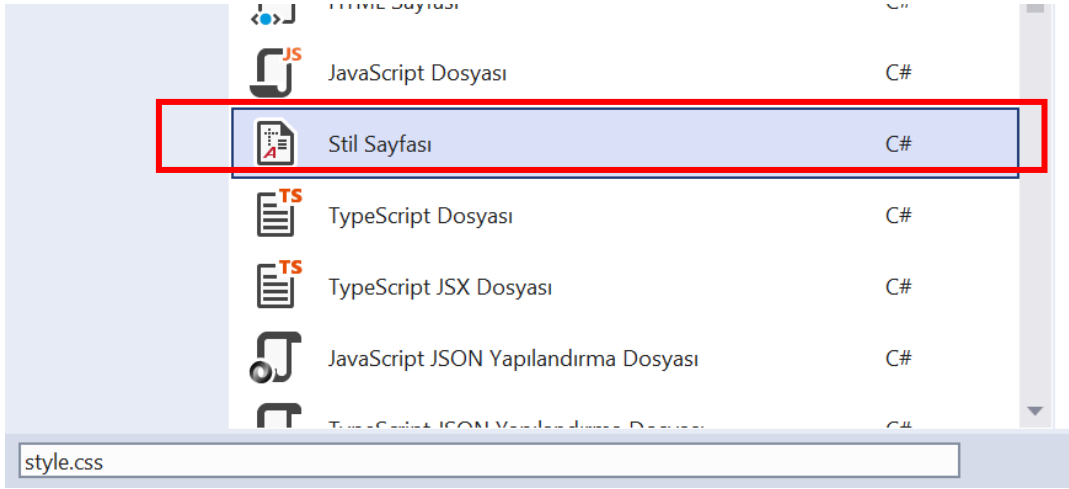


film 1

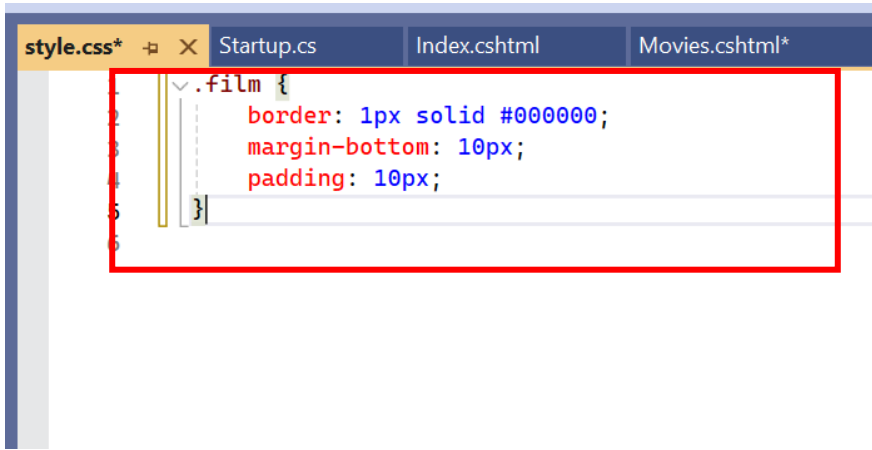
açıklama 1

Daha önce eklemiş olduğumuz border'ları bir style.css dosyasına taşıyalım.

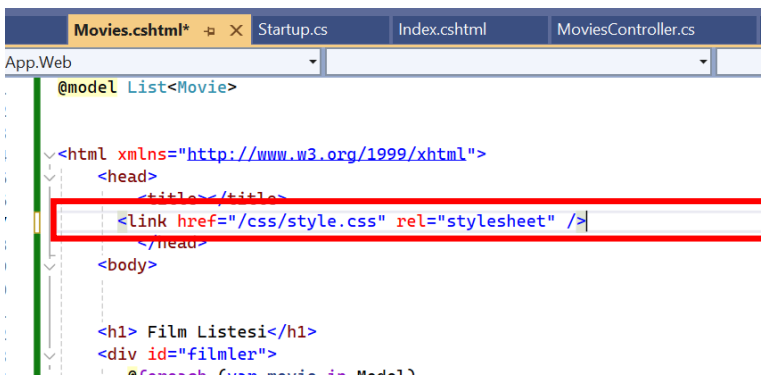
ADIM 10: css klasörü altına yeni öge (new item) ile bir css dosyası eklenir. style.css olarak isimlendirilir.



ADIM 11: Daha önce **Movies.cshtml** klasöründe oluşturduğumuz css kod bloğu style.css dosyasına taşınır.



ADIM 12: **Movies.cshtml** de ilgisiz alanlar temizlenir. Css altındaki style.css'i aktarmak için gerekli kodlar yazılır.



ADIM 13: Aynı düzenleme Index.cshtml dosyasında da yapılır.

```

style.css* Movies.cshtml* Startup.cs Index.cshtml* MoviesController.cs
MovieApp.Web
1 @model Movie
2
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head><title></title>
5 <link href="/css/style.css" rel="stylesheet" />
6 </head>
7 <body>
8 <h1> Film Listesi</h1>
9 <div id="filmler">
10 <div class="film">
11 <p>@Model.Title</p>
12 <p>@Model.Description</p>
13 <p>@Model.Director</p>
14 <p>
15 <ul>
16 @foreach (var player in Model.Players)
17 {
18 <li>@player</li>
19 }

```

ADIM 14: Çalıştırılır. Oluşturmuş olduğumuz border lar **style.css** dosyası aracılığı ile görünümüne taşınmış olur.

localhost:21054

localhost:21054

localhost:21054/movies/list

localhost:21054/movies/list

Film Listesi

film başlığı

filmin açıklaması

filmin yönetmen adı

- oyuncu 1
- oyuncu 2



film 1

açıklama 1

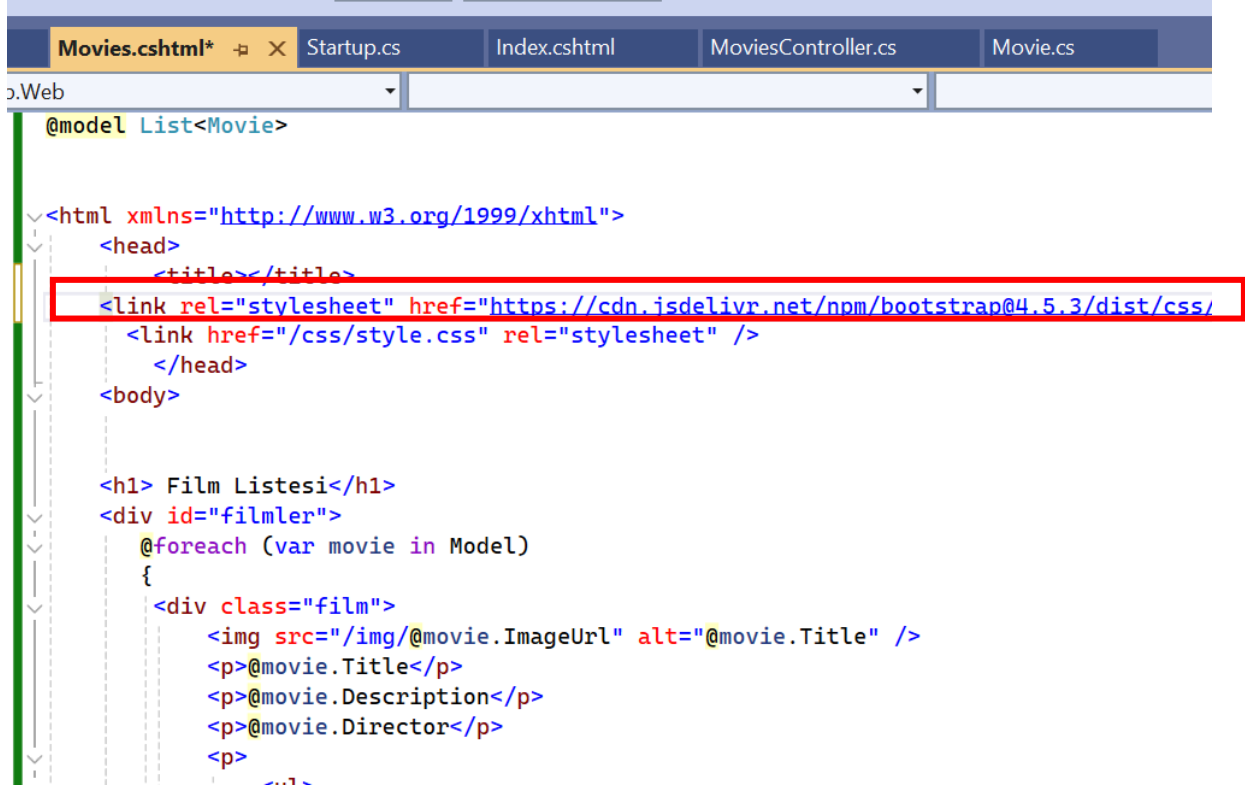
Yönetmen 1

- oyuncu 1
- oyuncu 2

3. TASARIMIN YAPILMASI (BOOTSTRAP KÜTÜPHANESİ KULLANMAK)

ADIM 1: Bootstrap kütüphanesinden elde edilen ilgili kod bloğu **Movies.cshtml** dosyası içinde ilgili alana eklenir.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uKz4rEkgIXeMed4M0JlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
```



```

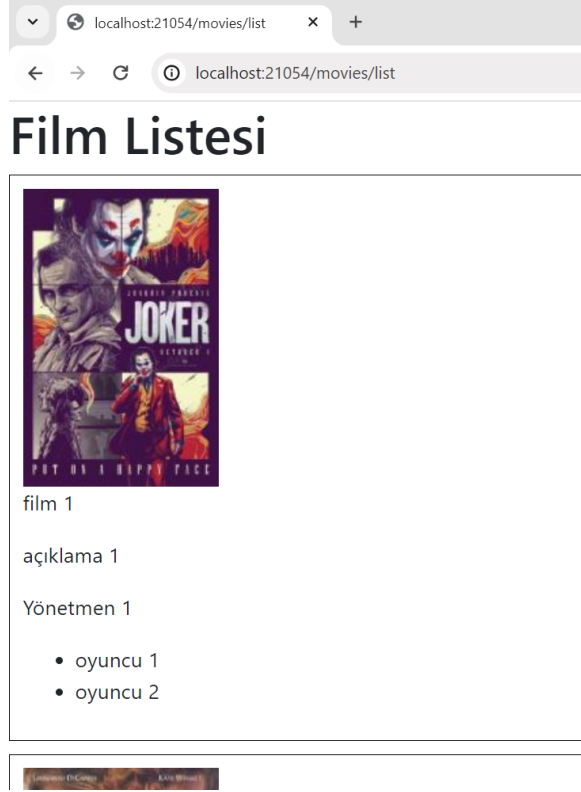
Movies.cshtml*  Startup.cs  Index.cshtml  MoviesController.cs  Movie.cs
p.Web
@model List<Movie>

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/
<link href="/css/style.css" rel="stylesheet" />
</head>
<body>

<h1> Film Listesi</h1>
<div id="filmler">
    @foreach (var movie in Model)
    {
        <div class="film">
            
            <p>@movie.Title</p>
            <p>@movie.Description</p>
            <p>@movie.Director</p>
            <p>
                ...
            </p>
        </div>
    }
</div>

```

ADIM 2: Çalıştırılır kontrol edilir. Kopyalanan kod satırı bazı html etiketlerine css özelliklerini varsayılan olarak ekler. Örneğin; Yazı stilinin değişimine dikkat ediniz.



ADIM 3: Navbar eklemek için gerekli ilgili kod bloğu **Movies.cshtml** dosyası içinde ilgili alana eklenir.

```
<nav class="navbar navbar-expand-lg navbar-dark bg-danger">
  <div class="container">
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarToggler">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarToggler">
      <a href="#" class="navbar-brand">MovieApp</a>
      <ul class="navbar-nav mr-auto">
        <li class="navbar-item active">
          <a href="/movies/list" class="nav-link">content</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

```

Movies.cshtml* x Startup.cs Index.cshtml MoviesController.cs Movie.cs _ViewImports.cs
Veb
@model List<Movie>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title></title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css
    <link href="/css/style.css" rel="stylesheet" />
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-danger">
      <div class="container">
        <button class="navbar-toggler" type="button" data-toggle="collapse" d
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarToggler">
          <a href="#" class="navbar-brand">MovieApp</a>
          <ul class="navbar-nav mr-auto">
            <li class="navbar-item active">
              <a href="/movies/list" class="nav-link">content</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </body>
</html>

```

Sorun bulunamadı

ADIM 4: Bootstrap kütüphanesinden elde edilen ikon için gerekli js kod bloğu Movies.cshtml dosyasında ilgili alana eklenir.

```

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous"></script>

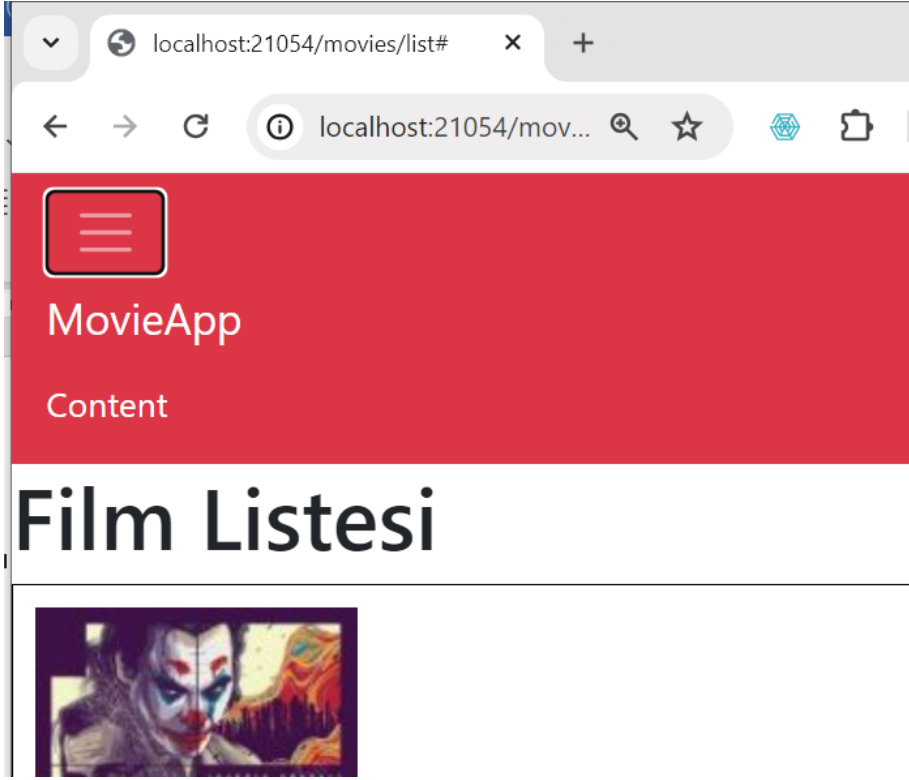
```

```

Movies.cshtml x Startup.cs Index.cshtml MoviesController.cs Movie.cs _ViewImports.cs
Veb
@foreach (var movie in Model)
{
  <div class="film">
    
    <p>@movie.Title</p>
    <p>@movie.Description</p>
    <p>@movie.Director</p>
    <p>
      <ul>
        @foreach (var player in movie.Players)
        {
          <li>@player</li>
        }
      </ul>
    </p>
  </div>
}
</div>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstr
</body>
</html>

```

ADIM 5: Çalıştırılır. Ekranı küçülttüğümüzde açılır ikon görüntüsünü elde etmiş oluruz.



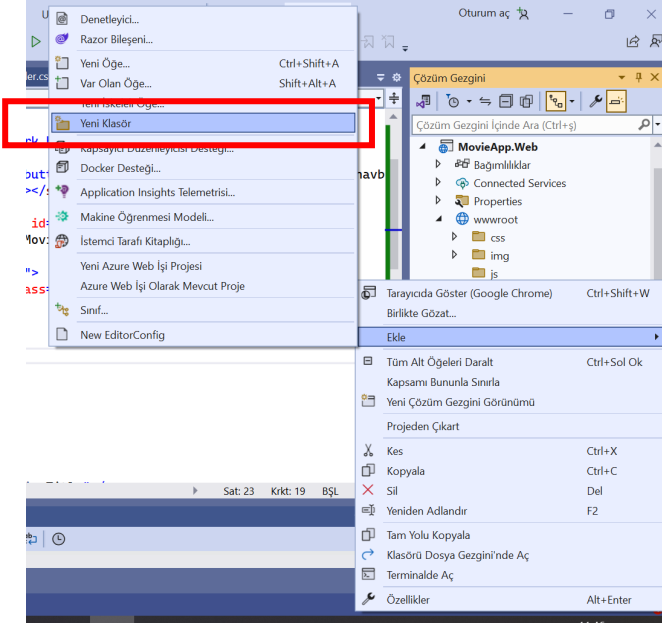
ÖNEMLİ NOT

Her sayfada sabit olması gereken içerikler olduğu gibi değişmesini istediğimiz içerikler de var.

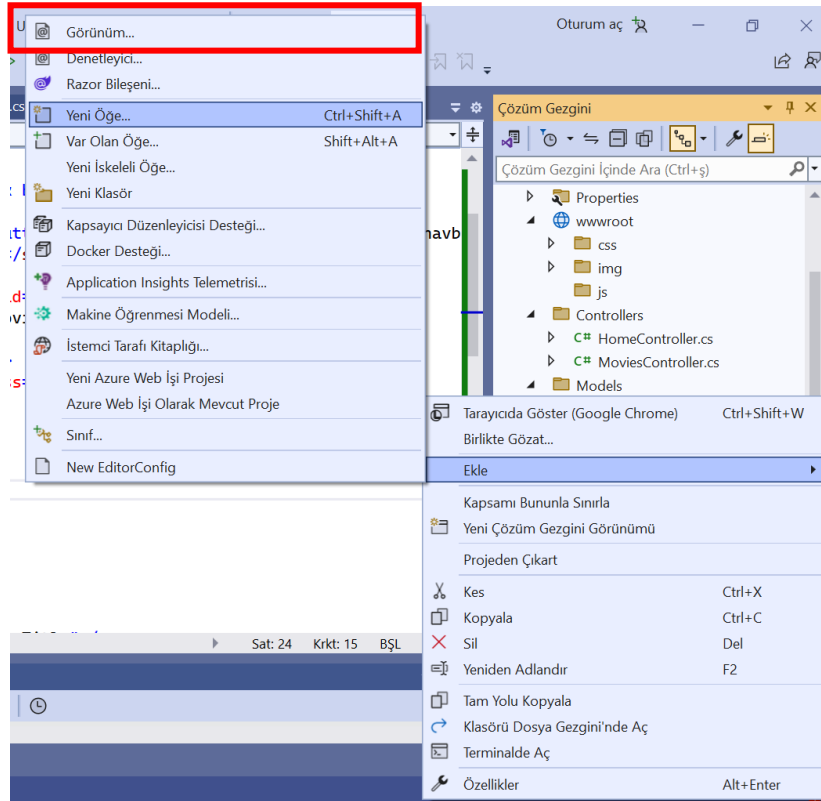
Bir şablon ekleyerek sabit içerikler ile ilgili işlemleri şablon üzerinden gerçekleştirelim.

4. LAYOUT OLUŞTURMAK

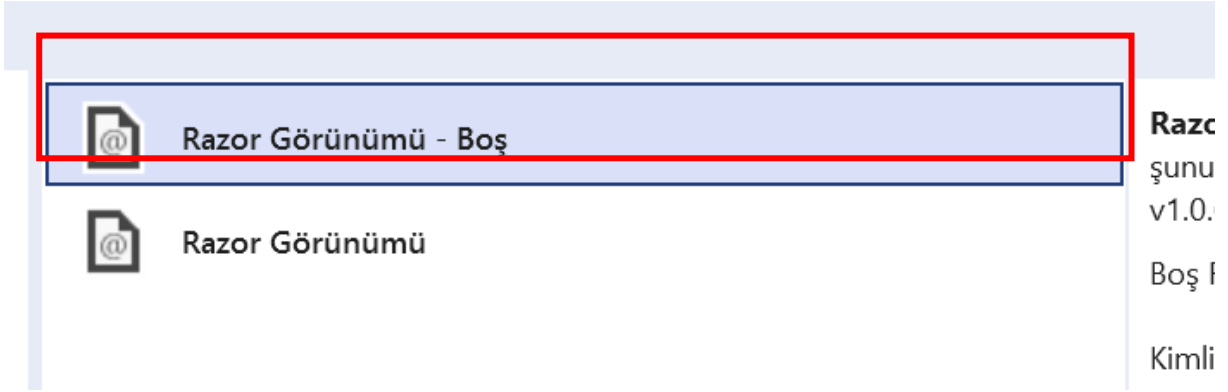
ADIM 1: Views klasörü üzerine sağ tıklayarak yeni bir Klasör oluşturup **Shared** olarak isimlendiriyoruz.



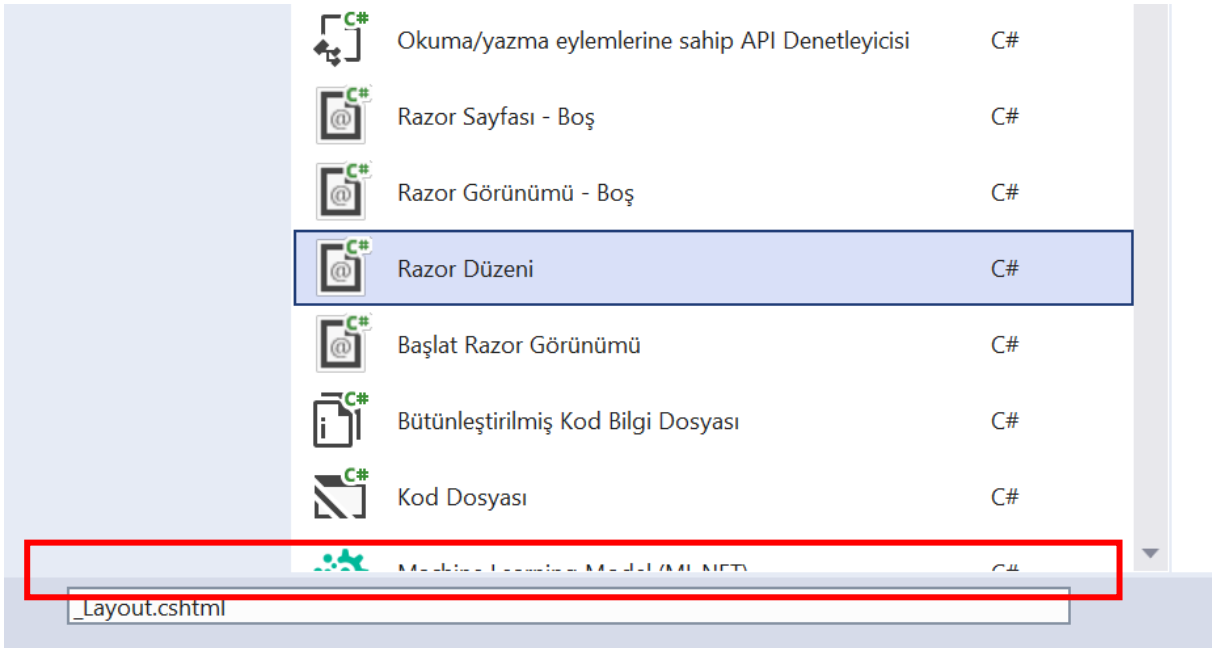
ADIM 2: Shared Klasörü içine bir layout sayfası ekleyebilmek için. Shared klasörüne sağ tıklayarak **Görünüm (View)**, boş Razor Layout sayfası seçiyoruz.



ADIM 3: boş Razor Layout sayfası seçilir.



ADIM 4: Razor Düzeni (Razor Layout) seçilir. Adı _Layout.cshtml olarak bırakılır.



ADIM 5: Her sayfada sabit kalmasını istediğimiz css kod bloğu oluşturmuş olduğumuz Layout a taşınır.

```

_Layout.cshtml* Movies.cshtml style.css Startup.cs Index.cshtml MoviesController.cs
MovieApp.Web
1 <!DOCTYPE html>
2
3 <html>
4 <head>
5   <meta name="viewport" content="width=device-width" />
6   <title>@ViewBag.Title</title>
7
8   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5
9   <link href="/css/style.css" rel="stylesheet" />
10
11 </head>
12 <body>
13   <div>
14     @RenderBody()
15   </div>
16 </body>
17 </html>
18

```

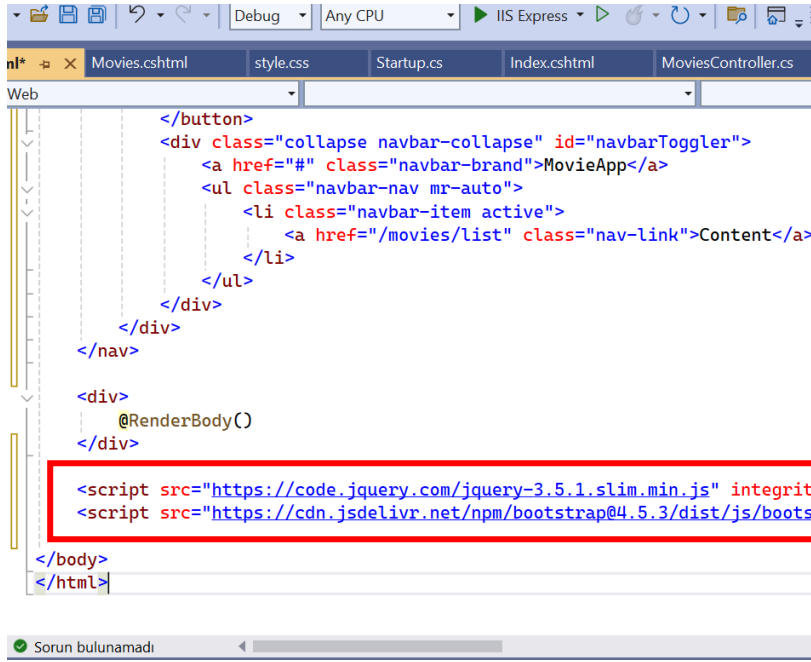
ADIM 6: Navbar için oluşturduğumuz kod bloğu **_Layout** sayfasına taşınır.

```

_Layout.cshtml* Movies.cshtml style.css Startup.cs Index.cshtml MoviesController.cs
MovieApp.Web
9 <link href="/css/style.css" rel="stylesheet" />
10
11 </head>
12 <body>
13
14   <nav class="navbar navbar-expand-lg navbar-dark bg-danger">
15     <div class="container">
16       <button class="navbar-toggler" type="button" data-toggle="col
17       <span class="navbar-toggler-icon"></span>
18     </button>
19     <div class="collapse navbar-collapse" id="navbarToggler">
20       <a href="#" class="navbar-brand">MovieApp</a>
21       <ul class="navbar-nav mr-auto">
22         <li class="navbar-item active">
23           <a href="/movies/list" class="nav-link">Content</
24         </li>
25       </ul>
26     </div>
27   </div>
28 </nav>
29
30   <div>
31     @RenderBody()
32   </div>
33

```

ADIM 7: js kod bloğu **_Layout** sayfasına taşınır.



```
</button>
<div class="collapse navbar-collapse" id="navbarToggler">
  <a href="#" class="navbar-brand">MovieApp</a>
  <ul class="navbar-nav mr-auto">
    <li class="navbar-item active">
      <a href="/movies/list" class="nav-link">Content</a>
    </li>
  </ul>
</div>
</nav>

<div>
  @RenderBody()
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/boots

</body>
</html>
```

Sorun bulunamadı

ADIM 8: Taşıma işlemleri sonrası **Movies.cshtml** dosyasının son görünümü aşağıdaki gibi olmalıdır.

```

Layout.cshtml*  Movies.cshtml*  style.css  Startup.cs  Index.cshtml  Movies
MovieApp.Web
1  @model List<Movie>
2
3      <h1> Film Listesi </h1>
4      <div id="filmler">
5          @foreach (var movie in Model)
6          {
7              <div class="film">
8                  
9                  <p>@movie.Title</p>
10                 <p>@movie.Description</p>
11                 <p>@movie.Director</p>
12                 <p>
13                     <ul>
14                         @foreach (var player in movie.Players)
15                         {
16                             <li>@player</li>
17                         }
18                     </ul>
19                 </p>
20             </div>
21         }
22     </div>

```

07 % Sorun bulunamadı

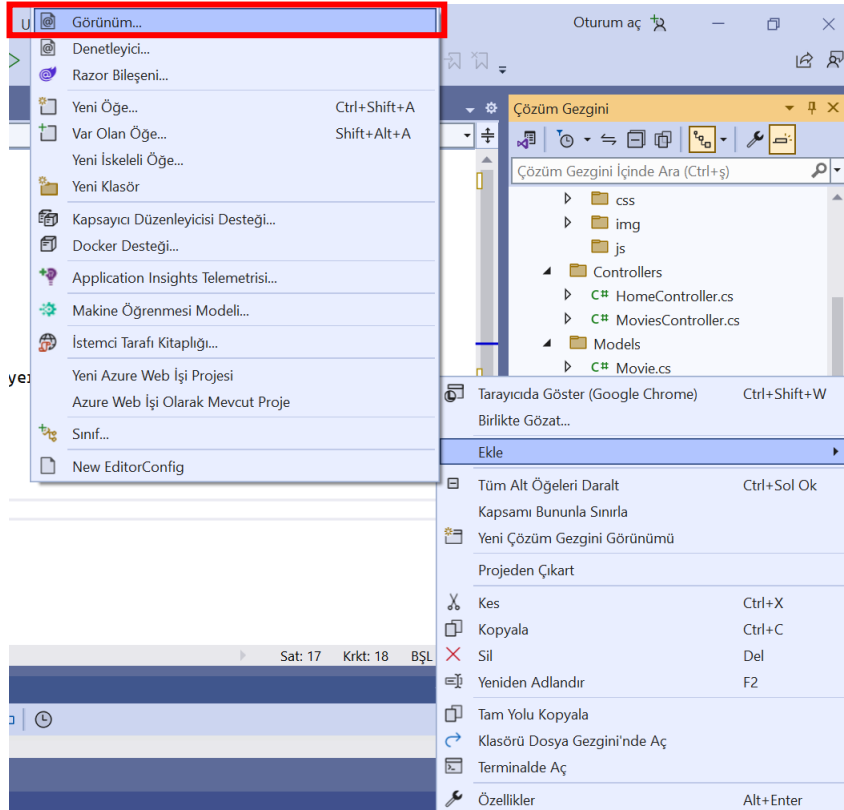
ADIM 9: Home **Index.cshtml** sayfası kodları da aynı şekilde düzenlenir.

```

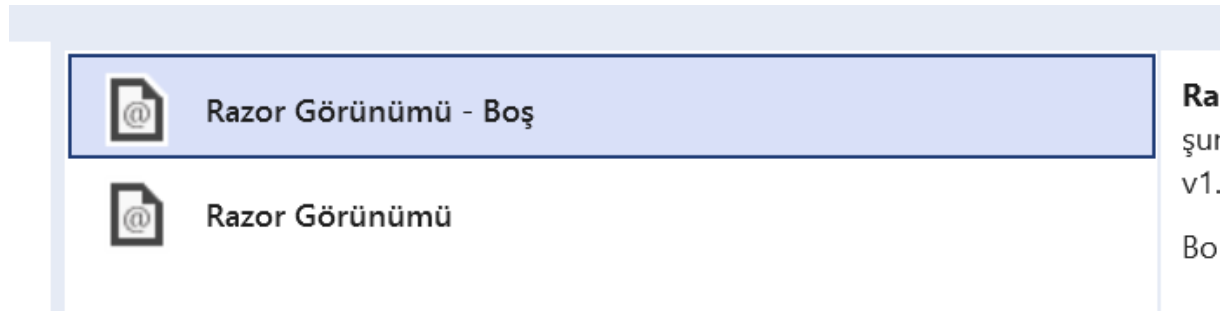
Index.cshtml*  _Layout.cshtml*  Movies.cshtml*
MovieApp.Web
1  @model Movie
2
3      <h1> Film Listesi</h1>
4      <div id="filmler">
5          <div class="film">
6              <p>@Model.Title</p>
7              <p>@Model.Description</p>
8              <p>@Model.Director</p>
9              <p>
10                 <ul>
11                     @foreach (var player in Model.Players)
12                     {
13                         <li>@player</li>
14                     }
15                 </ul>
16             </p>
17         </div>
18     </div>
19
20

```

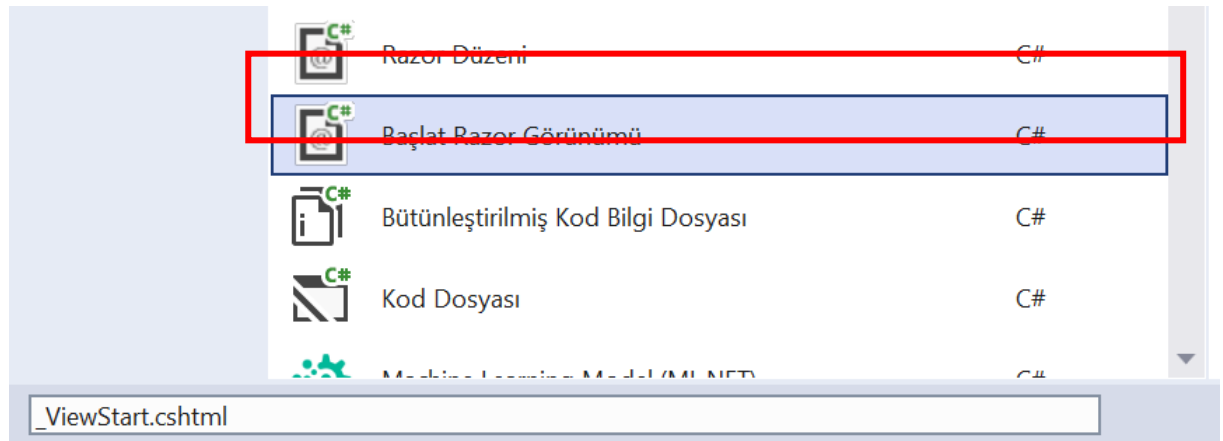
ADIM 10: oluşturmuş olduğumuz **_Layout** u uygulamamıza tanıtmak için. **View** klasörüne sağ tıklayarak görünümünden bir **Razor** daha eklenir.



ADIM 11: Boş olan tercih edilir.



ADIM 12: `_ViewStart.cshtml` uzantılı olan **Başlat Razor Görünümü (Razor View Start)** seçilir.



ÖNEMLİ NOT

Yapdığımız işlemler sonucunda, uygulamamız tüm layout ları (sayfadaki sabit değişkenleri) Shared klasörü altındaki **_Layout.cshtml** dosyasından alır.

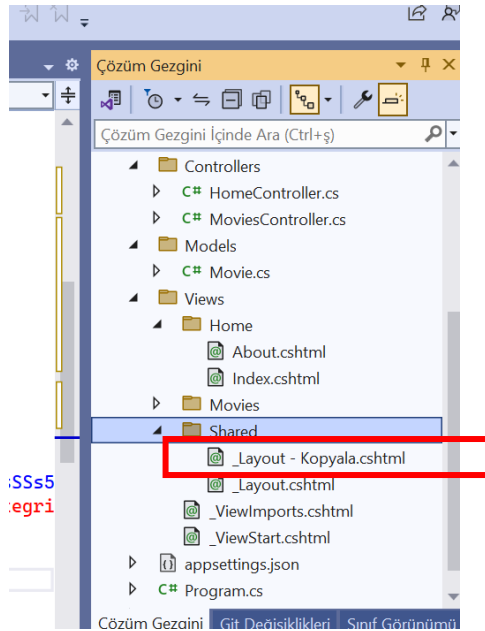
_Layout.cshtml'nin bu ilişkisi ise **_ViewStart.cshtml** oluşturulması ile mümkün olur.

Tüm cshtml sayfalarında sabit olmayan özellikler ise **_Layout.cshtml** klasöründeki

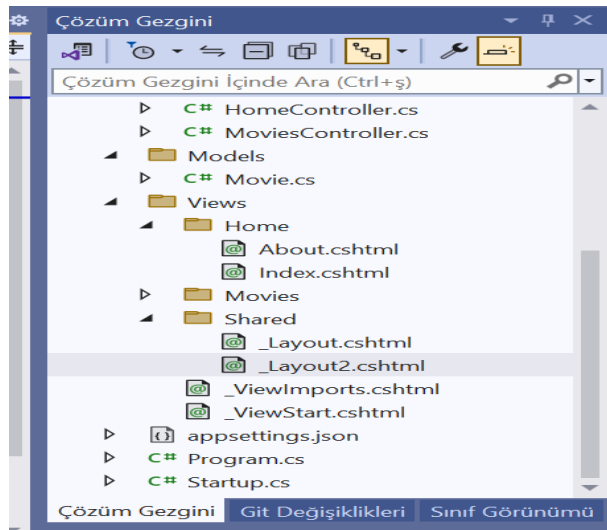
```
<div>
    @RenderBody()
</div>
```

kod bloğu aracılığı sağlanır.

ADIM 13: İki farklı Layout tanımlamak isteseydik. Shared içindeki **_Layout.cshtml** Shared klasörü içinde kopyala yapıştır diyerek çoğaltılabildi.



ADIM 14: **_Layout2.cshtml** olarak yeniden isimlendirilir.



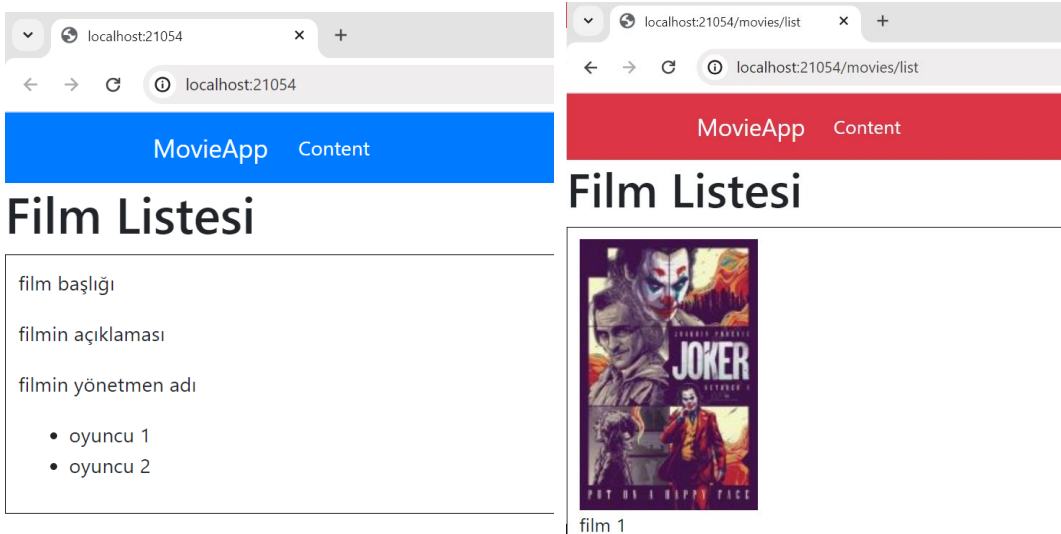
ADIM15: Farklılığı görebilmek için **_Layout2.cshtml** de navbar renginde değişiklik yapılır. Navbar arkaplan rengi mavi olarak değiştirildi.

```
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">
  <div class="container">
    <button class="navbar-toggler" type="button" data-toggle="collap
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarToggler">
      <a href="#" class="navbar-brand">MovieApp</a>
      <ul class="navbar-nav mr-auto">
        <li class="navbar-item active">
          <a href="/movies/list" class="nav-link">Content</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

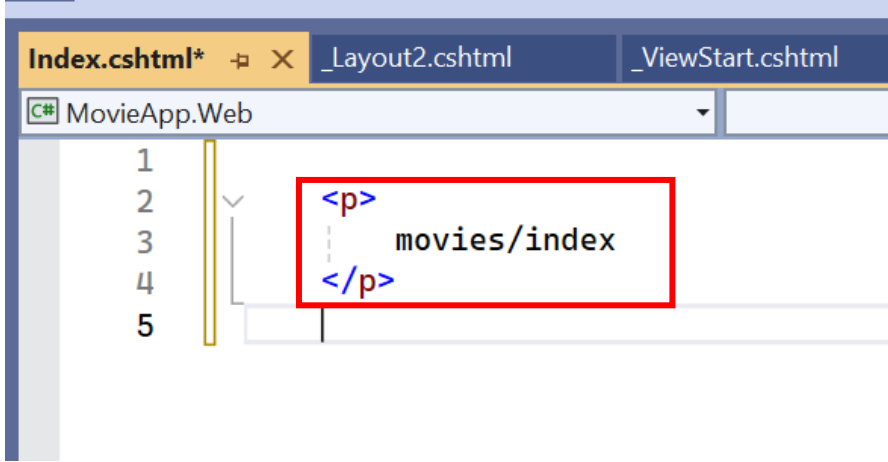
ADIM 15: Sadece **Home Index** de **Layout2** yi diğerlerinde **Layout'** u kullansın isteniyorsa. **Home Index.cshtml** dosyasına kod bloğu eklenir.

```
1 @model Movie
2
3 @{
4     Layout = "_Layout2";
5 }
6
7
8 <h1> Film Listesi</h1>
9 <div id="filmler">
10   <div class="film">
11     <p>@Model.Title</p>
12     <p>@Model.Description</p>
13     <p>@Model.Director</p>
14     <p>
15       <ul>
```

ADIM 16: Çalıştırılır.



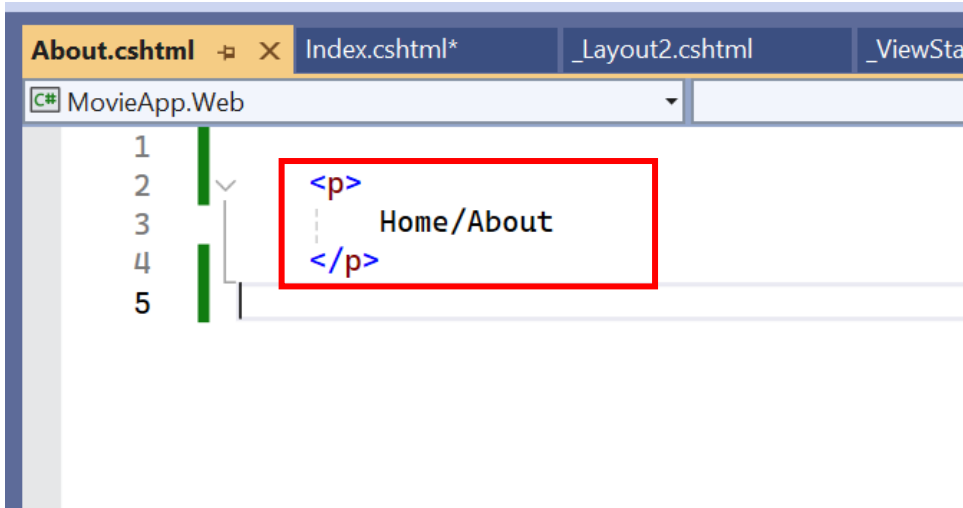
ADIM 17: Diğer sayfalar da düzenlenerek kullanıma hazır hale getirilir. **Movies** klasörü altındaki **Index** düzenlenir. Html kodları silinir (artık gereksiz)



```
1
2
3
4
5
```

`<p>movies/index</p>`

ADIM 18: **Home** klasörü altındaki **About** düzenlenir. Html kodları silinir (artık gereksiz)



```
1
2
3
4
5
```

`<p>Home/About</p>`

ÇALIŞMA SORUSU

Razor View nedir? Kullanım amaçları nelerdir açıklayınız.