



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

İNTERNET PROGRAMCILIĞI

Bu bölümde proje içinde kullanılacak olan Taghelper Kullanımı, Güncelleme Formu, Kayıt Güncelleme, Select Kutusu Oluşturma, Form Validations konularına yönelik uygulama yapılmıştır.

1. TAGHELPER KULLANIMI

ADIM 1: Create form için bir model oluşturalım (böylece var olan film bilgisini ilgili controller içine yükleyerek film güncellemesi yapabiliriz.)

```

1  @model Movie
2
3  <h1>Create Movie </h1>
4  <hr/>
5  <form method="post" action="/movies/create">
6      <div class="form-group">
7          <label for="Title"> Title </label>
8          <input type="text" name="Title" class="form-control" />
9      </div>
10     <div class="form-group">
11         <label for="Description"> Description </label>
12         <textarea name="Description" class="form-control"> </textarea>
13     </div>
14     <div class="form-group">
15         <label for="Director"> Director </label>
16         <input type="text" name="Title" class="form-control" />

```

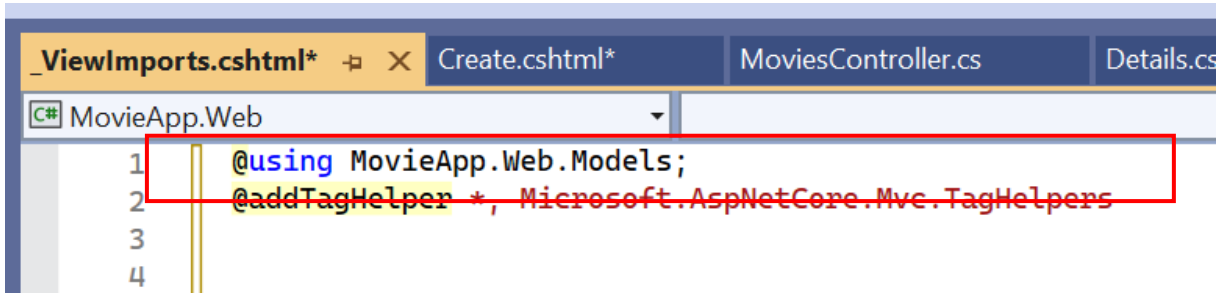
ADIM 2: TagHelper ı kullanabilmek için Create.cshtml de ilgili kod satırında değişiklik yapılır.

```

1  @model Movie
2
3  <h1>Create Movie </h1>
4  <hr/>
5  <form method="post" action="/movies/create">
6      <div class="form-group">
7          <label for="Title"> Title </label>
8          <input asp-for="Title" class="form-control" />
9      </div>
10     <div class="form-group">
11         <label for="Description"> Description </label>
12         <textarea name="Description" class="form-control"> </textarea>
13     </div>
14     <div class="form-group">
15         <label for="Director"> Director </label>

```

ADIM 3: _ViewImports.cshtml'e ilgili kod satırı eklenir.

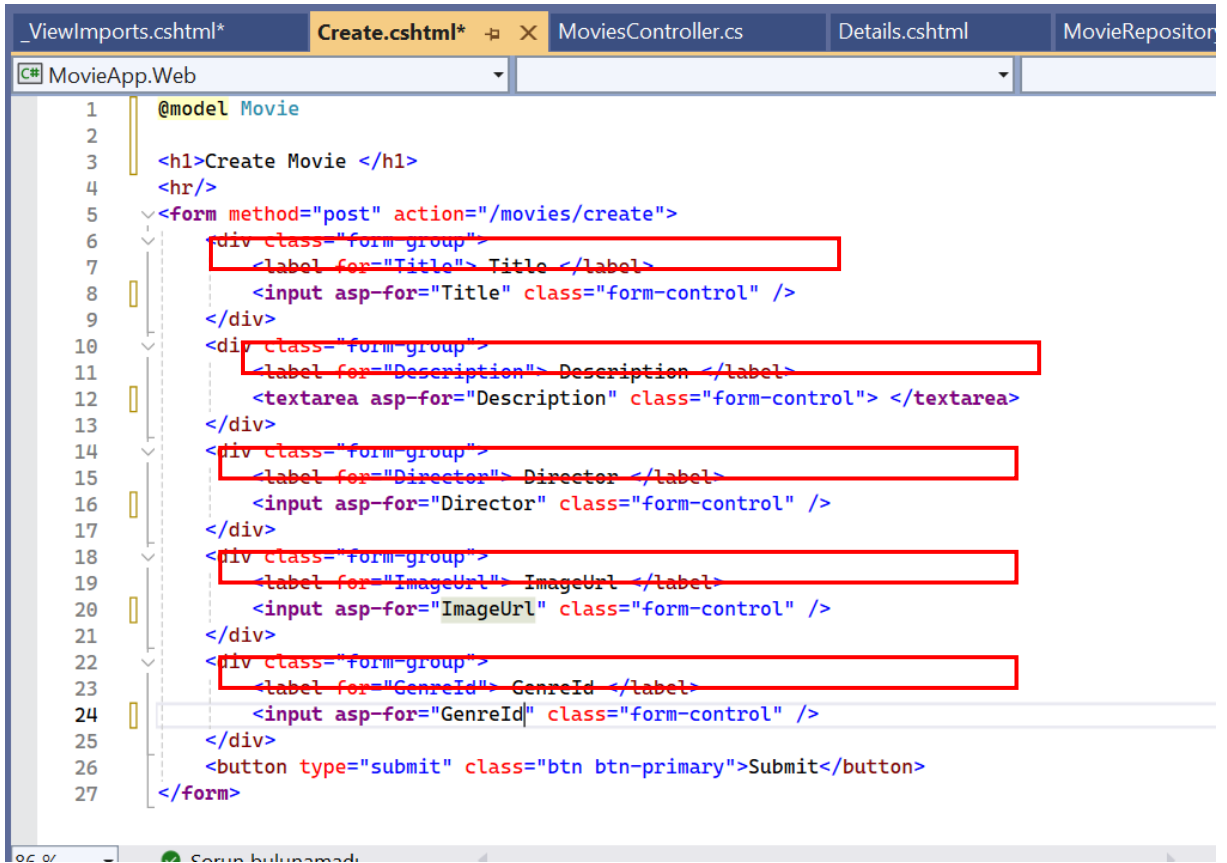


```

1  @using MovieApp.Web.Models;
2  @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
3
4

```

ADIM 4: Create.cshtml sayfasındaki kod blokları verilerimize göre düzenlenir.



```

1  @model Movie
2
3  <h1>Create Movie </h1>
4  <hr/>
5  <form method="post" action="/movies/create">
6      <div class="form-group">
7          <label for="Title"> Title </label>
8          <input asp-for="Title" class="form-control" />
9      </div>
10     <div class="form-group">
11         <label for="Description"> Description </label>
12         <textarea asp-for="Description" class="form-control"> </textarea>
13     </div>
14     <div class="form-group">
15         <label for="Director"> Director </label>
16         <input asp-for="Director" class="form-control" />
17     </div>
18     <div class="form-group">
19         <label for="ImageUrl"> ImageUrl </label>
20         <input asp-for="ImageUrl" class="form-control" />
21     </div>
22     <div class="form-group">
23         <label for="GenreId"> GenreId </label>
24         <input asp-for="GenreId" class="form-control" />
25     </div>
26     <button type="submit" class="btn btn-primary">Submit</button>
27 </form>

```

ADIM 5: Create.cshtml sayfasında yine değişiklik yapılır.

```

1  @model Movie
2
3  <h1>Create Movie </h1>
4  <hr/>
5  <form method="post" action="/movies/create">
6      <div class="form-group">
7          <label asp-for="Title"></label>
8          <input asp-for="Title" class="form-control" />
9      </div>
10     <div class="form-group">
11         <label asp-for="Description"></label>
12         <textarea asp-for="Description" class="form-control"> </textarea>
13     </div>
14     <div class="form-group">
15         <label asp-for="Director"></label>
16         <input asp-for="Director" class="form-control" />
17     </div>
18     <div class="form-group">
19         <label asp-for="ImageUrl"></label>
20         <input asp-for="ImageUrl" class="form-control" />
21     </div>
22     <div class="form-group">
23         <label asp-for="GenreId"></label>
24         <input asp-for="GenreId" class="form-control" />
25     </div>
26     <button type="submit" class="btn btn-primary">Submit</button>
27 </form>
28
29
30

```

86 % Sorun bulunamadı

Hata Listesi

ADIM 6: Yapılan tüm değişiklikler sonrası sayfa görünümü şu şekilde olmalıdır.

```

_ViewImports.cshtml* Create.cshtml* MoviesController.cs Details.cshtml MovieRepository.cs
C# MovieApp.Web
1 @model Movie
2
3 <h1>Create Movie </h1>
4 <hr/>
5 <form method="post" action="/movies/create">
6   <div class="form-group">
7     <label asp-for="Title"></label>
8     <input asp-for="Title" class="form-control" />
9   </div>
10  <div class="form-group">
11    <label asp-for="Description"></label>
12    <textarea asp-for="Description" class="form-control"> </textarea>
13  </div>
14  <div class="form-group">
15    <label asp-for="Director"></label>
16    <input asp-for="Director" class="form-control" />
17  </div>
18  <div class="form-group">
19    <label asp-for="ImageUrl"></label>
20    <input asp-for="ImageUrl" class="form-control" />
21  </div>
22  <div class="form-group">
23    <label asp-for="GenreId"></label>
24    <input asp-for="GenreId" class="form-control" />
25  </div>
26  <button type="submit" class="btn btn-primary">Submit</button>
27 </form>
28
29
86 % Sorun bulunamadı Sa

```

ADIM 7: Çalıştırarak uygulamayı kontrol edelim. Biz yazmasak da TagHelper, model aracılığı ile ilgili başlıkları alıp ekrana yansıttı.

localhost:21054/movies/create

MovieApp Movies Create Movie

Create Movie

Title

Description

Director

ImageUrl

GenreId

Submit

ADIM 8: Açılacak sayfa bilgisi de TagHelper aracılığı ile dinamik olarak elde edilebilir. Bunun için ilgili kod satırında yapılacak değişiklik şu şekildedir.

```

_ViewImports.cshtml  Create.cshtml*  MoviesController.cs  Details.cshtml  MovieRepository
C# MovieApp.Web
1  @model Movie
2
3  <h1>Create Movie </h1>
4  <hr/>
5  <form method="post" asp-controller="Movies" asp-action="Create">
6      <div class="form-group">
7          <label asp-for="Title"></label>
8          <input asp-for="Title" class="form-control" />
9      </div>
10     <div class="form-group">
11         <label asp-for="Description"></label>
12         <textarea asp-for="Description" class="form-control"> </textarea>
13     </div>
14     <div class="form-group">
15         <label asp-for="Director"></label>
16         <input asp-for="Director" class="form-control" />
17     </div>
18     <div class="form-group">
19         <label asp-for="ImageUrl"></label>
20         <input asp-for="ImageUrl" class="form-control" />
21     </div>
22     <div class="form-group">
23         <label asp-for="GenreId"></label>

```

ADIM 9: Form da herhangi bir label adını değiştirmek isteseydik. Movie.cs dosyasına şu kod satırlarını ekleyebilirdik.

```

Movie.cs*  Create.cshtml*  _ViewImports.cshtml
MovieApp.Web  MovieApp.Web.Models  Title
1  using System.ComponentModel;
2
3  namespace MovieApp.Web.Models
4  {
5      {
6          public class Movie
7          {
8              public int MovieId { get; set; }
9              [DisplayName("Başlık")]
10             public string Title { get; set; }
11             public string Description { get; set; }
12             public string Director { get; set; }
13             public string [] Players { get; set; }
14             public string ImageUrl { get; set; }
15             public int GenreId { get; set; }
16         }
17     }
18

```

ADIM 10: Çalıştırılarak kontrol edilir.

localhost:21054/movies/create

MovieApp Movies Create Movie

Create Movie

Başlık

Description

Director

2. FİLM GÜNCELLEME FORMU

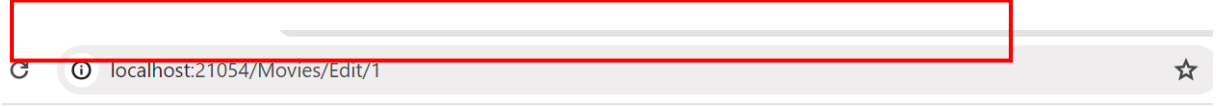
ADIM 1: **Components** klasörü içinde **_movie.cshtml** içerisinde, Form güncelleme için ilgili sayfayı oluşturalım ve yönlendirmeleri yapalım. Data önce yazdığımız kod bloğu ilgili şekilde düzenlenir.

```

_movie.cshtml*  Movie.cs  Create.cshtml  _ViewImports.cshtml  MoviesController.cs  Details.cshtml
MovieApp.Web
1  @model Movie
2
3  <div class="card mb-3" style="max-width: 540px;">
4  <div class="row g-0">
5  <div class="col-md-3">
6      
7  </div>
8  <div class="col-md-9">
9      <div class="card-body">
10         <h3 class="card-title">@Model.Title</h3>
11         <div class="card-text">@Model.Description</div>
12         <a class="btn btn-primary" asp-controller="Movies" asp-action="Details" asp-route-id="@Model.MovieId">details</a>
13         <a class="btn btn-primary" asp-controller="Movies" asp-action="Edit" asp-route-id="@Model.MovieId">edit</a>
14     </div>
15 </div>
16 </div>
17 </div>
18 </div>
19 </div>
20 </div>
21

```

ADIM 2: Çalıştırılarak edit butonunun ilgili URL e gidiyor olması kontrol edilir.



Bu localhost sayfası bulunamıyor

Şu web adresi için web sayfası bulunamadı: <http://localhost:21054/Movies/Edit/1>

HTTP ERROR 404

Yeniden Yükle

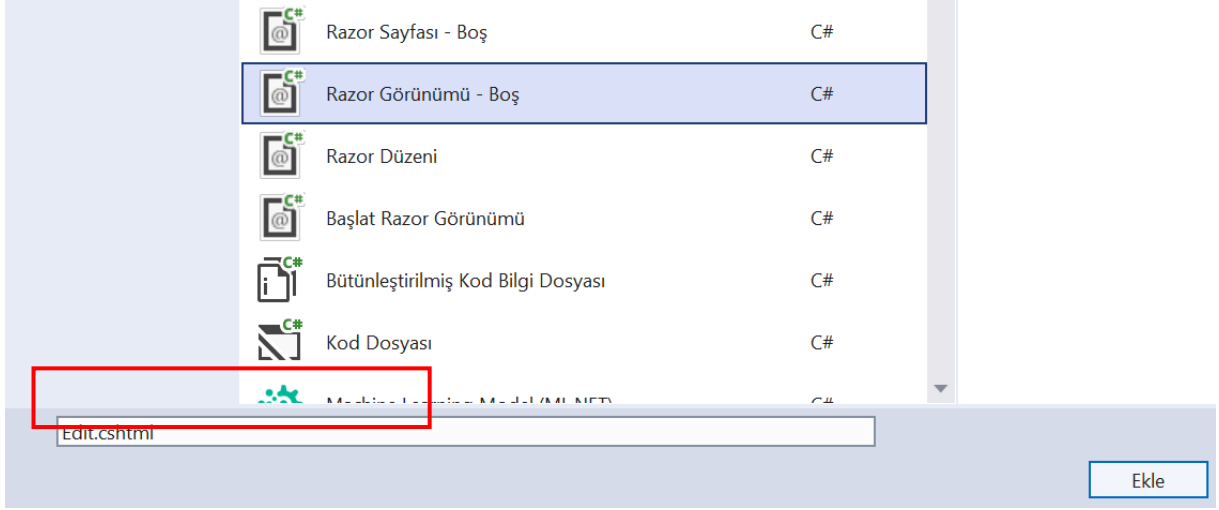
ADIM 3: **MoviesController.cs** dosyasında Edit i çağırılım.İlgili kod bloğu şu şekildedir.

```

MoviesController.cs*  _movie.cshtml  Movie.cs  Create.cshtml
MovieApp.Web
62      // GenreId = GenreId
63      // };
64      MovieRepository.Add(m);
65      return RedirectToAction("List");
66  }
67  [HttpGet]
68  public IActionResult Edit()
69  {
70      return View();
71  }
72  }
73  }
74

```

ADIM 4: Movies klasörü altına edit formu eklemek için. Yeni Razor View- Boş eklenir. Edit.cshtml olarak isimlendirilir.



ADIM 5: Create için oluşturmuş olduğumuz Create.cshtml içindeki html kodlarının tamamını kopyalayıp Edit.cshtml e yapıştıralım. İlgili değişikliği yapalım.

```

1  @model Movie
2
3  <h1>Edit Movie </h1>
4  <hr />
5  <form method="post" asp-controller="Movies" asp-action="Create">
6      <div class="form-group">
7          <label asp-for="Title"></label>
8          <input asp-for="Title" class="form-control" />
9      </div>
10     <div class="form-group">
11         <label asp-for="Description"></label>
12         <textarea asp-for="Description" class="form-control"> </textarea>
13     </div>
14     <div class="form-group">
15         <label asp-for="Director"></label>
16         <input asp-for="Director" class="form-control" />
17     </div>
18     <div class="form-group">
19         <label asp-for="ImageUrl"></label>
20         <input asp-for="ImageUrl" class="form-control" />
21     </div>

```

86 % Sorun bulunamadı

Çıktı

ADIM 6: MoviesController.cs de ilgili kod bloğu

```

56     // var m = new Movie()
57     // {
58     //     Title = Title,
59     //     Description = Description,
60     //     Director = Director,
61     //     ImageUrl = ImageUrl,
62     //     GenreId = GenreId
63     // };
64     MovieRepository.Add(m);
65     return RedirectToAction("List");
66 }
67 [HttpGet]
68 public IActionResult Edit()
69 {
70     return View();
71 }
72 [HttpPost]
73 public IActionResult Edit (Movie m)
74 {
75     return View();
76 }
77 }
78 }
79

```

ADIM 7: Controllerdan sayfaya gönderdiğimiz bilgiyi alabilmek için **Edit.cshtml** e ilgili kodlar eklenir.

```

Edit.cshtml*  MoviesController.cs*  _movie.cshtml  Movie.cs  Create.cshtml  _ViewImports.cshtml
MovieApp.Web
1  @model Movie
2
3  <h1>Edit Movie </h1>
4  <hr />
5  <form method="post" asp-controller="Movies" asp-action="Create">
6      <div class="form-group">
7          <label asp-for="Title"></label>
8          <input asp-for="Title" class="form-control" value="@Model.Title" />
9      </div>
10     <div class="form-group">
11         <label asp-for="Description"></label>
12         <textarea asp-for="Description" class="form-control" value="@Model.Description"></textarea>
13     </div>
14     <div class="form-group">
15         <label asp-for="Director"></label>
16         <input asp-for="Director" class="form-control" value="@Model.Director" />
17     </div>
18     <div class="form-group">
19         <label asp-for="ImageUrl"></label>
20         <input asp-for="ImageUrl" class="form-control" value="@Model.ImageUrl" />
21     </div>
22     <div class="form-group">
23         <label asp-for="GenreId"></label>
24         <input asp-for="GenreId" class="form-control" value="@Model.GenreId" />
25     </div>
26     <button type="submit" class="btn btn-primary">Submit</button>

```

86 % Sorun bulunamadı Sat: 28 Krkt: 1

ADIM 8: ilgili form çağırıldığında id parametresini **MoviesController.cs** de bekliyor olmamız gerek bunun için daha önce yazdığımız kod bloğuna gerekli eklemeler yapılır.

```

Edit.cshtml*  MoviesController.cs*  _movie.cshtml  Movie.cs  Create.cshtml
MovieApp.Web  MovieApp.Web.Controllers.MoviesController
56  // var m = new Movie()
57  // {
58  //     Title = Title,
59  //     Description = Description,
60  //     Director = Director,
61  //     ImageUrl = ImageUrl,
62  //     GenreId = GenreId
63  // };
64  MovieRepository.Add(m);
65  return RedirectToAction("List");
66  }
67  [HttpGet]
68  public IActionResult Edit(int id)
69  {
70      return View(MovieRepository.GetById(id));
71  }
72  [HttpPost]
73  public IActionResult Edit (Movie m)
74  {
75      return View();
76  }
77  }
78  }

```

ADIM 9: Uygulamayı Çalıştırarak edit butonuna basıp kontrol edelim.

localhost:21054/Movies/Edit/3

MovieApp Movies Create Movie

Edit Movie

Başlık

Black Swan

Description

açıklama 1

Director

Darren Aronofsky

ImageUrl

3.jpg

GenreId

1

Submit

Herhangi bir film için edit butonuna tıkladığımızda bilgilerin formda ilgili yerlere geldiğine dikkat ediniz. Şimdi değiştirdiğimiz bilgileri güncellemek için adımlara devam edelim.

3. KAYIT GÜNCELLEME

ADIM 1: MoviesController.cs de ilgili kod bloğunda gerekli güncellemeler yapılır.

```

59      //      Description = Description,
60      //      Director = Director,
61      //      ImageUrl = ImageUrl,
62      //      GenreId = GenreId
63      // };
64      MovieRepository.Add(m);
65      return RedirectToAction("List");
66  }
67  [HttpGet]
68  public IActionResult Edit(int id)
69  {
70      return View(MovieRepository.GetById(id));
71  }
72  [HttpPost]
73  public IActionResult Edit (Movie m)
74  {
75      MovieRepository.Edit(m);
76      return RedirectToAction("Details", "Movies", new { @id = m.MovieId });
77  }
78  }
79  }
80

```

ADIM 2: MovieRepository.cs e Controller a karşılık gelecek şekilde ilgili kod bloğu eklenir.

```

88  }
89
90  public static Movie GetById (int id)
91  {
92      // ilgili liste üzerinde bütün elemanlara tek tek döngü şeklinde bakar
93      // m yukarıdaki listenin her bir elemanını ifade eder
94      return _movies.FirstOrDefault(m => m.MovieId == id);
95  }
96  public static void Edit (Movie m)
97  {
98      foreach (var movie in _movies)
99      {
100          if(movie.MovieId ==m.MovieId)
101          {
102              movie.Title = m.Title;
103              movie.Description = m.Description;
104              movie.Director = m.Director;
105              movie.ImageUrl = m.ImageUrl;
106              movie.GenreId = m.GenreId;
107              break;
108          }
109      }
110  }
111  }
112  }
113

```

ADIM 3: Edit.cshtml kodlarını model ile eşleştirmek için gereken kod satırı yazılır.

```

1  @model Movie
2
3  <h1>Edit Movie </h1>
4  <hr />
5  <form method="post" asp-controller="Movies" asp-action="Create">
6    <input type="hidden" name="MovieId" value="@Model.MovieId" />
7    <div class="form-group">
8      <label asp-for="Title"></label>
9      <input asp-for="Title" class="form-control" value="@Model.Title" />
10   </div>
11   <div class="form-group">
12     <label asp-for="Description"></label>
13     <textarea asp-for="Description" class="form-control">@Model.Description </textarea>
14   </div>
15   <div class="form-group">
16     <label asp-for="Director"></label>
17     <input asp-for="Director" class="form-control" value="@Model.Director" />
18   </div>
19   <div class="form-group">
20     <label asp-for="ImageUrl"></label>
21     <input asp-for="ImageUrl" class="form-control" value="@Model.ImageUrl" />
22   </div>
23   <div class="form-group">
24     <label asp-for="GenreId"></label>
25     <input asp-for="GenreId" class="form-control" value="@Model.GenreId" />
26   </div>

```

ADIM 4: Edit.cshtml de son bir güncelleme.

```

1  @model Movie
2
3  <h1>Edit Movie </h1>
4  <hr />
5  <form method="post" asp-controller="Movies" asp-action="Edit">
6    <input type="hidden" name="MovieId" value="@Model.MovieId" />
7    <div class="form-group">
8      <label asp-for="Title"></label>
9      <input asp-for="Title" class="form-control" value="@Model.Title" />
10   </div>
11   <div class="form-group">
12     <label asp-for="Description"></label>
13     <textarea asp-for="Description" class="form-control">@Model.Description </textarea>
14   </div>
15   <div class="form-group">
16     <label asp-for="Director"></label>
17     <input asp-for="Director" class="form-control" value="@Model.Director" />
18   </div>
19   <div class="form-group">
20     <label asp-for="ImageUrl"></label>
21     <input asp-for="ImageUrl" class="form-control" value="@Model.ImageUrl" />
22   </div>
23   <div class="form-group">
24     <label asp-for="GenreId"></label>
25     <input asp-for="GenreId" class="form-control" value="@Model.GenreId" />
26   </div>

```

ADIM 4: Uygulamayı çalıştırarak güncelleme işlemi kontrol edilir. Örneğin 3 numaralı film bilgilerinin her birini güncelleyelim.

localhost:21054/Movies/Edit/1

MovieApp

Movies Create Movie

Edit Movie

Başlık

Joker

Description

Joker, 2019 yapımı Amerikan psikolojik gerilim filmidir ve Todd Phillips tarafından yönetilmiş, senaryosu Scott Silvey karakterlerine dayanan film, başarısız bir palyaço ve hedefli stand-up komedyeni olmak olan Arthur Fleck'i, 1981'c

Director

Todd Phillips

ImageUrl

1.jpg

GenreId

4

Submit

localhost:21054/Movies/Edit/1

MovieApp

Movies Create Movie

Edit Movie

Başlık

Film Adı

Description

Açıklaması

Director

Yönetmeni

ImageUrl

2.jpg

GenreId

2

Submit

localhost:21054/Movies/Details/1

MovieApp

Movies Create Movie



Film Adı

Açıklaması

Yönetmeni

- Ed Helms
- Justin Bartha

4. FİLM TÜRLERİ İÇİN SELECT KUTUSU OLUŞTURMA

ADIM 1: Edit.cshtml içinde select kutusu için gerekli güncellemeler dikkatli şekilde yapılır.

```

Edit.cshtml*  MovieRepository.cs  MoviesController.cs  Create.cshtml  _ViewImports.cshtml
MovieApp.Web
10  </div>
11  <div class="form-group">
12  <label asp-for="Description"></label>
13  <textarea asp-for="Description" class="form-control">@Model.Description </textarea>
14  </div>
15  <div class="form-group">
16  <label asp-for="Director"></label>
17  <input asp-for="Director" class="form-control" value="@Model.Director" />
18  </div>
19  <div class="form-group">
20  <label asp-for="ImageUrl"></label>
21  <input asp-for="ImageUrl" class="form-control" value="@Model.ImageUrl" />
22  </div>
23  <div class="form-group">
24  <label asp-for="GenreId"></label>
25  <select asp-for="GenreId" asp-items="@ViewBag.Genres" class="form-control">
26  <option selected disabled>Seçiniz</option>
27  </select>
28  </div>
29  <button type="submit" class="btn btn-primary">Submit</button>
30  </form>
31
32

```

ADIM 2: MoviesController.cs de güncellenmesi gereken kod bloğu.

```

Edit.cshtml*  MovieRepository.cs  MoviesController.cs*  Create.cshtml  _ViewImports.cshtml
MovieApp.Web  MovieApp.Web.Controllers.MoviesController  List(int? id, string q)
57  // var m = new Movie()
58  // {
59  //     Title = Title,
60  //     Description = Description,
61  //     Director = Director,
62  //     ImageUrl = ImageUrl,
63  //     GenreId = GenreId
64  // };
65  MovieRepository.Add(m);
66  return RedirectToAction("List");
67  }
68  [HttpGet]
69  public IActionResult Edit(int id)
70  {
71  ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
72  return View(MovieRepository.GetById(id));
73  }
74  [HttpPost]
75  public IActionResult Edit (Movie m)
76  {
77  MovieRepository.Edit(m);
78  return RedirectToAction("Details", "Movies", new {id = m.MovieId });
79  }
80  }
81  }
82

```

ADIM 3: Uygulamayı Çalıştırarak edit formu üzerinde tür bilgilerini kontrol edelim.

localhost:21054/Movies/Edit/1

MovieApp

Movies

Create Movie

Film Ara

B

Edit Movie

Başlık

Joker

Description

Joker, 2019 yapımı Amerikan psikolojik gerilim filmidir ve Todd Phillips tarafından yönetilmiş, senaryosu Scott Silver ile birlikte yazılmıştır. DC Comics karakterlerine dayanan film, başarısız bir palyaço ve hedefi stand-up komedyeni olmak olan Arthur Fleck'i, 1981'de geçen bir hikâyede ele alır.

Director

Todd Phillips

ImageUrl

1.jpg

Genrel

Savaş

Submit

ADIM 4: Aynı işlemleri Create Formu üzerinde de uygulayalım. Edit.cshtml içinden ilgili kod bloğu kopyalanarak Create.cshtml içine yapıştırılır.

```

3 <div>Create movie </div>
4 <hr/>
5 <form method="post" asp-controller="Movies" asp-action="Create">
6   <div class="form-group">
7     <label asp-for="Title"></label>
8     <input asp-for="Title" class="form-control" />
9   </div>
10  <div class="form-group">
11    <label asp-for="Description"></label>
12    <textarea asp-for="Description" class="form-control"> </textarea>
13  </div>
14  <div class="form-group">
15    <label asp-for="Director"></label>
16    <input asp-for="Director" class="form-control" />
17  </div>
18  <div class="form-group">
19    <label asp-for="ImageUrl"></label>
20    <input asp-for="ImageUrl" class="form-control" />
21  </div>
22  <div class="form-group">
23    <label asp-for="GenreId"></label>
24    <select asp-for="GenreId" asp-items="@ViewBag.Genres" class="form-control">
25      <option selected disabled>Seçiniz</option>
26    </select>
27  </div>
28  <button type="submit" class="btn btn-primary">Submit</button>
29 </form>
30
31

```

ADIM 5: MoviesController.cs de Create formu alanına ilgili kod satırı eklenir.

```

49 [HttpGet]
50 public IActionResult Create()
51 {
52   ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
53   return View();
54 }
55 [HttpPost]
56 public IActionResult Create(Movie m)
57 {
58   // var m = new Movie()
59   // {
60   //   Title = Title,
61   //   Description = Description,
62   //   Director = Director,
63   //   ImageUrl = ImageUrl,
64   //   GenreId = GenreId
65   // };
66   MovieRepository.Add(m);
67   return RedirectToAction("List");
68 }
69 [HttpGet]
70 public IActionResult Edit(int id)
71 {
72   ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
73   return View(MovieRepository.GetById(id));
74 }
75 [HttpPost]
76 public IActionResult Edit (Movie m)

```

ADIM 6: Çalıştırarak Create Movie form alanı kontrol edilir.

MovieApp Movies Create Movie

Create Movie

Başlık

Description

Director

ImageUrl

GenreId

Komedi

Submit

ADIM 7: Yeni bir film türü eklemek isteseydik. **GenreRepository.cs** dosyası üzerinde ilgili kod satırı yazılırdı.

```

1  using MovieApp.Web.Models;
2  using System.Collections.Generic;
3  using System.Linq;
4  namespace MovieApp.Web.Data
5  {
6      public class GenreRepository
7      {
8          private static readonly List<Genre> _genres = null;
9          static GenreRepository()
10         {
11             _genres = new List<Genre>()
12             {
13                 new Genre {GenreId=1, Name="Macera"},
14                 new Genre {GenreId=2, Name="Komedi"},
15                 new Genre {GenreId=3, Name="Romantik"},
16                 new Genre {GenreId=4, Name="Savaş"},
17                 new Genre {GenreId=5, Name="Bilim Kurgu"}
18             };
19         }
20         public static List<Genre> Genres
21         {
22             get
23             {
24                 return _genres;
25             }
26         }
27         public static void Add(Genre genre)
28         {

```

86 % Sorun bulunamadı

Çıktı

ADIM 8: Uygulamayı çalıştırarak yeni tür bilgisine uygun bir film eklenir.

localhost:21054/movies/create

MovieApp Movies Create Movie

Create Movie

Başlık

Yeni Film

Description

Açıklama

Director

Yönetmeni

ImageUrl

1.jpg

Genrelid

Bilim Kurgu

Submit

ADIM 9: Submit ettikten sonra bilim kurguya tıklanınca yeni eklenen filmin geldiği görülür.

localhost:21054/movies/list/5

MovieApp Movies Create Movie

Film Listesi

Tüm Filmler

Macera

Komedi

Romantik

Savaş

Bilim Kurgu

Yeni Film

details edit



5. FORM VALIDATIONS

ADIM 1: Create Form da belirli alanların boş bırakılmasını istemiyorsak **Movie.cs** dosyasına ilgili kod satırları eklenir.

```

1  using System.ComponentModel;
2  using System.ComponentModel.DataAnnotations;
3
4  namespace MovieApp.Web.Models
5  {
6
7      public class Movie
8      {
9          public int MovieId { get; set; }
10         [DisplayName("Başlık")]
11         [Required(ErrorMessage="film başlığı eklemelisiniz.")]
12         [StringLength(50,MinimumLength =5,ErrorMessage ="film başlığı 5-10 karakter arasında olmalıdır.")]
13         public string Title { get; set; }
14         [Required(ErrorMessage = "film açıklaması eklemelisiniz.")]
15         public string Description { get; set; }
16         public string Director { get; set; }
17         public string [] Players { get; set; }
18         [Required]
19         public string ImageUrl { get; set; }
20         [Required]
21         public int GenreId { get; set; }
22     }
23 }
24

```

ADIM 2: Gelen film içerisindeki parametrelerin istediğimiz gibi olup olmadığını anlamak için **MoviesController.cs** içine if bloğu oluşturulur ve ilgili kod bloğu aşağıdaki gibi düzenlenir.

```

52  ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
53  return View();
54  }
55  [HttpPost]
56  public IActionResult Create(Movie m)
57  {
58      if(ModelState.IsValid)
59      {
60          MovieRepository.Add(m);
61          return RedirectToAction("List");
62      }
63      return View();
64  }
65
66  [HttpGet]
67  public IActionResult Edit(int id)
68  {
69      ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
70      return View(MovieRepository.GetById(id));
71  }
72  [HttpPost]
73  public IActionResult Edit (Movie m)
74  {
75

```

ADIM 3: Uygulamayı Çalıştırarak Create Form un boş şekilde girilip girilemediğini kontrol edelim.

localhost:21054/Movies/Create

MovieApp Movies Create Movie

Create Movie

Bağlık

Description

Director

ImageUrl

GenreId

Seçiniz

Submit

ADIM 4: Bu durumda uygulamanın hata mesajı göstermesi için Create.cshtml de her bloğa ilgili kod satırı eklenir.

```

Create.cshtml* MoviesController.cs GenreRepository.cs Edit.cshtml MovieRepository.cs
MovieApp.Web
5 <form method="post" asp-controller="Movies" asp-action="Create">
6   <div class="form-group">
7     <label asp-for="Title"></label>
8     <input asp-for="Title" class="form-control" />
9     <span asp-validation-for="Title" class="text-danger"></span>
10  </div>
11  <div class="form-group">
12    <label asp-for="Description"></label>
13    <textarea asp-for="Description" class="form-control"></textarea>
14    <span asp-validation-for="Description" class="text-danger"></span>
15  </div>
16  <div class="form-group">
17    <label asp-for="Director"></label>
18    <input asp-for="Director" class="form-control" />
19    <span asp-validation-for="Director" class="text-danger"></span>
20  </div>
21  <div class="form-group">
22    <label asp-for="ImageUrl"></label>
23    <input asp-for="ImageUrl" class="form-control" />
24    <span asp-validation-for="ImageUrl" class="text-danger"></span>
25  </div>
26  <div class="form-group">
27    <label asp-for="GenreId"></label>
28    <select asp-for="GenreId" asp-items="@ViewBag.Genres" class="form-control">
29      <option selected disabled>Seçiniz</option>
30    </select>
31    <span asp-validation-for="GenreId" class="text-danger"></span>
32  </div>
33 </form>
86 % Sorun bulunamadı

```

ADIM 5: Uygulamayı çalıştırarak Form boşken Submit butonuna basıp kontrol edelim. Submit butonuna bastıktan sonra Genrelid alanında tür bilgilerinin gelmediği görülecektir. Bu problemin çözülmesi için bir sonraki adımı uygulayınız.

localhost:21054/Movies/Create

MovieApp Movies Create Movie

Create Movie

Başlık

film başlığı eklemelisiniz.

Description

film açıklaması eklemelisiniz.

Director

ImageUrl

The ImageUrl field is required.

Genrelid

Seçiniz

Submit

ADIM 6: Movie.cs dosyasında Genrelid değerinin int olarak tanımlandığını görüyoruz. İnt? Olarak kod satırını güncelliyoruz.

```

1  using System.ComponentModel;
2  using System.ComponentModel.DataAnnotations;
3
4  namespace MovieApp.Web.Models
5  {
6
7      public class Movie
8      {
9          public int MovieId { get; set; }
10         [DisplayName("Başlık")]
11         [Required(ErrorMessage="film başlığı eklemelisiniz.")]
12         [StringLength(50,MinimumLength=5,ErrorMessage="film başlığı 5-10 karakter arasında olmalıdır.")]
13         public string Title { get; set; }
14         [Required(ErrorMessage="film açıklaması eklemelisiniz.")]
15         public string Description { get; set; }
16         public string Director { get; set; }
17         public string [] Players { get; set; }
18         [Required]
19         public string ImageUrl { get; set; }
20         [Required]
21         public int? GenreId { get; set; }
22     }
23 }
24

```

ADIM 7: MoviesController.cs de Genrelid tür bilgilerine ilişkin satırı kopyalayıp ilgili alana yapıştırarak submit edildikten sonra da tür bilgilerinin görünmesini sağlıyoruz.

```

47     return View(MovieRepository.GetById(id));
48 }
49 [HttpGet]
50 public IActionResult Create()
51 {
52     ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
53     return View();
54 }
55 [HttpPost]
56 public IActionResult Create(Movie m)
57 {
58     if(ModelState.IsValid)
59     {
60         MovieRepository.Add(m);
61         return RedirectToAction("List");
62     }
63     ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
64     return View();
65 }
66
67 [HttpGet]
68 public IActionResult Edit(int id)
69 {
70     ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
71     return View(MovieRepository.GetById(id));
72 }
73 [HttpPost]
74 public IActionResult Edit (Movie m)

```

ADIM 8: Uygulamayı çalıştırdıktan sonra submit butonuna basıp tür bilgilerinin gelip gelmediğini kontrol ediyoruz.

localhost:21054/Movies/Create

MovieApp Movies Create Movie

Create Movie

Başlık

 film başlığı eklemelisiniz.

Description

 film açıklaması eklemelisiniz.

Director

ImageUrl

 The ImageUrl field is required.

Genrelid

 The Genrelid field is required.

Submit

ADIM 9: Create form için yaptığımız if kontrolünü Edit form için de yapalım.

```

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

    {
        MovieRepository.Add(m);
        return RedirectToAction("List");
    }
    ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
    return View();
}

[HttpGet]
public IActionResult Edit(int id)
{
    ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
    return View(MovieRepository.GetById(id));
}

[HttpPost]
public IActionResult Edit (Movie m)
{
    if (ModelState.IsValid)
    {
        MovieRepository.Edit(m);
        return RedirectToAction("Details", "Movies", new { @id = m.MovieId });
    }
    ViewBag.Genres = new SelectList(GenreRepository.Genres, "GenreId", "Name");
    return View(m);
}

```

ADIM 10: Create.cshtml kodlarında daha önce yaptığımız değişiklikleri Edit.cshtml kodlarında da yapalım

```

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

<form method="post" asp-controller="Movies" asp-action="Edit">
    <input type="hidden" name="MovieId" value="@Model.MovieId" />
    <div class="form-group">
        <label asp-for="Title"></label>
        <input asp-for="Title" class="form-control" value="@Model.Title" />
        <span asp-validation-for="Title" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="Description"></label>
        <textarea asp-for="Description" class="form-control" value="@Model.Description" />
        <span asp-validation-for="Description" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="Director"></label>
        <input asp-for="Director" class="form-control" value="@Model.Director" />
        <span asp-validation-for="Director" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="ImageUrl"></label>
        <input asp-for="ImageUrl" class="form-control" value="@Model.ImageUrl" />
        <span asp-validation-for="ImageUrl" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="GenreId"></label>
        <select asp-for="GenreId" asp-items="@ViewBag.Genres" class="form-control">
            <option selected disabled>Seçiniz</option>
        </select>
        <span asp-validation-for="GenreId" class="text-danger"></span>
    </div>
</form>

```

ÇALIŞMA SORUSU

TagHelper metodu uygulamada ne amaçla kullanılmıştır. Açıklayınız