

Assignment Report

Name: Conor McKillen

Student ID: B00800202

Contents

- Salient points – Page 3
- MovieServiceDb.cs – Page 4 – 7
- TestMovieService.cs – Page 8 – 20
- MovieController.cs – Page 21 – 25
- ReviewController.cs – Page 26 – 27
- MovieViews
 - Index.cshtml – Page 28 – 29
 - Details.cshtml – Page 30
 - Create.cshtml – Page 31 – 32
 - Edit.cshtml – Page 33 – 34
 - CreateReview.cshtml – Page 35 -36
 - Delete.cshtml – Page 37
 - _Reviews.cshtml – Page 38
 - Directors.cshtml – Page 39 – 40
 - Titles.cshtml – Page 41 – 42
 - Years.cshtml – Page 43 – 44
- ReviewViews
 - Index.cshtml – Page 45
 - Create.cshtml – Page 46
 - Delete.csthml – Page 47
- Website screenshots – Page 48 - 51

Salient points

I begun the project gathering information for movies and their respective reviews for seeding data. I tried to implement a carousel for the movie posters on the home page. While I was able to provide the aesthetic of a carousel the functionality was unfortunately absent. With regards to styling I changed up the colour scheme and font colours and implemented a bootstrap client that implemented a dark background as I find this easier on the eye

MovieServiceDb.cs

```
13  namespace MMS.Data.Services
14  {
15      public class MovieServiceDb : IMovieService
16      {
17          private readonly MovieDbContext db;
18
19
20          public MovieServiceDb()
21          {
22              db = new MovieDbContext();
23          }
24
25          public Movie AddMovie(Movie m)
26          {
27              // Check if movie id actually exists
28              var existing = GetMovieById(m.Id);
29              if (existing != null)
30              {
31                  return null; // Movie id exists so it cannot be created
32              }
33
34              // Movie is unique so go ahead and create a movie
35              m = new Movie()
36              {
37                  Title = m.Title,
38                  Director = m.Director,
39                  Year = m.Year,
40                  Duration = m.Duration,
41                  Budget = m.Budget,
42                  PosterUrl = m.PosterUrl,
43                  Genre = m.Genre,
44                  Cast = m.Cast,
45                  Plot = m.Plot
46              };
47
48              db.Movies.Add(m);
49              db.SaveChanges(); // write to databases
50
51          }
52
53      }
```

```
31 references
54     public Review AddReview(Review r)
55     {
56         var existing = GetReviewById(r.Id);
57         if (existing != null)
58         {
59             return null; // Review id exists so it cannot be created
60         }
61
62         // Review is unique so go ahead and create review
63         r = new Review()
64         {
65             Id = r.Id,
66             Name = r.Name,
67             CreatedOn = r.CreatedOn,
68             Comment = r.Comment,
69             Rating = r.Rating,
70             MovieId = r.MovieId,
71             Movie = r.Movie
72         };
73
74         db.Reviews.Add(r);
75         db.SaveChanges(); // write to database
76
77         return r;
78     }
79
6 references
80     public bool DeleteMovie(int id)
81     {
82         // Check if the movie exists
83         // If null then there is nothing to delete
84         var m = GetMovieById(id);
85         if (m == null)
86         {
87             return false;
88         }
89
90         db.Movies.Remove(m);
91         db.SaveChanges(); // write to database
92
93         return true;
94     }
95
96
0 references
97     public Review GetReview(int id)
98     {
99         var review = db.Movies.SelectMany(m => m.Reviews).FirstOrDefault(r => r.Id == id);
100        return review;
101    }
```

```
 103     public bool DeleteReview(int id)
 104     {
 105         // find review
 106         // Check if the review exists
 107         // If null then there is nothing to delete
 108         var review = GetReviewById(id);
 109         if (review == null) return false;
 110
 111         // remove review
 112         db.Reviews.Remove(review);
 113
 114         db.SaveChanges();
 115
 116         return true;
 117     }
 118
 119     // Retrieve all reviews and movies associated with the ticket
 120     public IList<Review> GetAllReviews()
 121     {
 122         return db.Reviews
 123             .Include(r => r.Movie)
 124             .ToList();
 125     }
 126
 127
 128     public IList<Movie> GetAllMovies(string order = null)
 129     {
 130         return db.Movies.ToList();
 131     }
 132
 133     public Movie GetMovieById(int id)
 134     {
 135         return db.Movies.Include(m => m.Reviews).FirstOrDefault(m => m.Id == id);
 136     }
 137
 138     public Review GetReviewById(int id)
 139     {
 140         return db.Reviews.FirstOrDefault(r => r.Id == id);
 141     }
 142
 143     public void Initialise()
 144     {
 145         db.Initialise();
 146     }
 147
```

```
148     public bool UpdateMovie(Movie m)
149     {
150         //verify the movie exists
151         var movie = GetMovieById(m.Id);
152         if (movie == null)
153         {
154             return false;
155         }
156
157         // updated the details of the student retrieved and save
158         movie.Title = m.Title;
159         movie.Director = m.Director;
160         movie.Year = m.Year;
161         movie.Duration = m.Duration;
162         movie.Budget = m.Budget;
163         movie.PosterUrl = m.PosterUrl;
164         movie.Genre = m.Genre;
165         movie.Cast = m.Cast;
166         movie.Plot = m.Plot;
167
168         db.SaveChanges(); // write to database
169
170         return true;
171     }
172 }
173
174
175 }
```

TestMovieService.cs

```
11  public class TestMovieService
12  {
13      private readonly IMovieService _svc;
14
15      public TestMovieService()
16      {
17          // create and initialise MovieServiceDb service
18
19          // general arrangement
20          _svc = new MovieServiceDb();
21
22          // ensure data source is empty before each test
23          _svc.Initialise();
24      }
25
26      // add tests here - test should ensure your service implementation works
27      [Fact]
28      public void Movie_GetMoviesWhenNone_ShouldReturnNone()
29      {
30          // arrange
31
32          // act
33          var movies = _svc.GetAllMovies();
34          var count = movies.Count;
35
36          // assert
37          Assert.Equal(0, count);
38      }
39
40      [Fact]
41      public void Movie_AddMovieWhenUnique_ShouldSetAllProperties()
42      {
43          // arrange
44          var m = new Movie()
45          {
46              Title = "Rocky",
47              Budget = 1000000,
48              Year = 1976,
49              Cast = "Stallone",
50              Director = "John G. Avildsen",
51              Duration = 136,
52              Genre = Genre.Action,
53              Plot = "Boxing",
54              PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAJixeEDSg.jpg"
55          };
56      }

```

```

57     // act
58     var o = _svc.AddMovie(m);
59
60     var s = _svc.GetMovieById(o.Id); // retrieve movie saved
61
62     // assert - that movie is not null
63     Assert.NotNull(s);
64     Assert.Equal(s.Id, o.Id);
65     Assert.Equal("Rocky", s.Title);
66     Assert.Equal(1000000, s.Budget);
67     Assert.Equal(1976, s.Year);
68     Assert.Equal("Stallone", s.Cast);
69     Assert.Equal("John G. Avildsen", s.Director);
70     Assert.Equal(136, s.Duration);
71     Assert.Equal(Genre.Action, s.Genre);
72     Assert.Equal("Boxing", s.Plot);
73     Assert.Equal("https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAJixeEDSg.jpg", s.PosterUrl);
74 }
75
76
77 [Fact]
78 public void Movie_UpdateWhenExists_ShouldSetAllProperties()
79 {
80     // arrange - create test movie
81
82     var m = new Movie()
83     {
84         Title = "Rocky",
85         Budget = 1000000,
86         Year = 1976,
87         Cast = "Stallone",
88         Director = "John G. Avildsen",
89         Duration = 136,
90         Genre = Genre.Action,
91         Plot = "Boxing",
92         PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAJixeEDSg.jpg"
93     };
94
95
96     var o = _svc.AddMovie(m);
97
98     // act - update test movie
99     o.Title = "The Death of Stalin";
100    o.Budget = 13000000;
101    o.Year = 2017;
102    o.Cast = "Steve Buscemi, Simon Russel Beale, Paddy Considine";
103    o.Director = "Armando Iannucci";
104    o.Duration = 117;

```

No issues found.

```

106     o.Plot = "When the tyrannical ruler Stalin dies, his hapless inner circle scrambles to come up with the next evolution of the revolution ? but it's clear everyone";
107     o.PosterUrl = "https://www.themoviedb.org/t/p/original/66nL9kPerEZMoecQuWD322Nm8g.jpg";
108     _svc.UpdateMovie(o);
109
110     // assert
111     Assert.NotNull(o);
112     Assert.Equal("The Death of Stalin", o.Title);
113     Assert.Equal(13000000, o.Budget);
114     Assert.Equal(2017, o.Year);
115     Assert.Equal("Steve Buscemi, Simon Russel Beale, Paddy Considine", o.Cast);
116     Assert.Equal("Armando Iannucci", o.Director);
117     Assert.Equal(117, o.Duration);
118     Assert.Equal(Genre.Comedy, o.Genre);
119
120     Assert.Equal("When the tyrannical ruler Stalin dies, his hapless inner circle scrambles to come up with the next evolution of the revolution ? but it's clear everyone";
121     Assert.Equal(https://www.themoviedb.org/t/p/original/66nL9kPerEZMoecQuWD322Nm8g.jpg, o.PosterUrl);
122 }
123
124
125 [Fact]
126 [References]
127 public void Movie_GetMoviesWhenTwoAdded_ShouldReturnTwo()
128 {
129     // arrange
130     var m = new Movie()
131     {
132         Title = "Rocky",
133         Budget = 1000000,
134         Year = 1976,
135         Cast = "Stallone",
136         Director = "John G. Avildsen",
137         Duration = 136,
138         Genre = Genre.Action,
139         Plot = "Boxing",
140         PosterUrl = "https://www.themoviedb.org/t/p/original/15xiwd5srecBvO7mifAJixeED5g.jpg"
141     };
142
143     var o = new Movie()
144     {
145         Title = "The Death of Stalin",
146         Budget = 13000000,
147         Year = 2017,
148         Cast = "Steve Buscemi, Simon Russel Beale, Paddy Considine",
149         Director = "Armando Iannucci",
150         Duration = 117,
151         Genre = Genre.Comedy,
152         Plot = "When the tyrannical ruler Stalin dies, his hapless inner circle scrambles to come up with the next evolution of the revolution ? but it's clear everyone";
153         PosterUrl = "https://www.themoviedb.org/t/p/original/66nL9kPerEZMoecQuWD322Nm8g.jpg"
```

```

153     PosterUrl = "https://www.themoviedb.org/t/p/original/66nL9hKPerEZMOeCQuWD322Nm8g.jpg"
154     };
155
156     var s1 = _svc.AddMovie(m);
157     var s2 = _svc.AddMovie(o);
158
159     // act
160     var movies = _svc.GetAllMovies();
161     var count = movies.Count;
162
163     // assert
164     Assert.Equal(2, count);
165
166 }
167
168 [Fact]
169 0 references
170 public void Movie_GetMovieWhenNone_ShouldReturnNull()
171 {
172     // arrange
173
174     // act
175     var movie = _svc.GetMovieById(1);
176
177     // assert
178     Assert.Null(movie);
179 }
180
181 [Fact]
182 0 references
183 public void Movie_GetMovieThatExists_ShouldReturnMovie()
184 {
185     // arrange
186     var m = new Movie()
187     {
188         Title = "Rocky",
189         Budget = 1000000,
190         Year = 1976,
191         Cast = "Stallone",
192         Director = "John G. Avildsen",
193         Duration = 136,
194         Genre = Genre.Action,
195         Plot = "Boxing",
196         PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAJixeEDSg.jpg"
197     };
198
199     // act
200     var ns = _svc.GetMovieById(s.Id);

```

```

201     // assert
202     Assert.NotNull(ns);
203     Assert.Equal(s.Id, ns.Id);
204 }
205
206 [Fact]
207 0 references
208 public void Movie_DeleteMovieThatExists_ShouldReturnTrue()
209 {
210     // arrange
211     var m = new Movie()
212     {
213         Title = "Rocky",
214         Budget = 1000000,
215         Year = 1976,
216         Cast = "Stallone",
217         Director = "John G. Avildsen",
218         Duration = 136,
219         Genre = Genre.Action,
220         Plot = "Boxing",
221         PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAjixeEDSg.jpg"
222     };
223
224     var s = _svc.AddMovie(m);
225
226     // act
227
228     var deleted = _svc.DeleteMovie(s.Id);
229     var s1 = _svc.GetMovieById(s.Id); // try to retrieve deleted movie
230
231     // assert
232
233     Assert.True(deleted); // delete movie should return true
234     Assert.Null(s1); // m1 should be null
235 }
236
237 [Fact]
238 0 references
239 public void Movie_DeleteMovieThatExists_ShouldReduceMovieCountByOne()
240 {
241     // arrange
242     var m = new Movie()
243     {
244         Title = "Rocky",
245         Budget = 1000000,
246         Year = 1976,
247         Cast = "Stallone",
248         Director = "John G. Avildsen",
249         Duration = 136,
250         Genre = Genre.Action

```

```

248     Genre = Genre.Action,
249     Plot = "Boxing",
250     PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAjlxeEDSg.jpg"
251   },
252 
253   var s = _svc.AddMovie(m);
254 
255   // act
256 
257   _svc.DeleteMovie(s.Id);
258   var movies = _svc.GetAllMovies();
259 
260   // assert
261 
262   Assert.Equal(0, movies.Count); // should be 0 movies
263 }
264 
265 [Fact]
266 0 references
267 public void Movie_DeleteMovieThatDoesntExist_ShouldNotChangeMovieCount()
268 {
269   // arrange
270   var m = new Movie()
271   {
272     Title = "Rocky",
273     Budget = 1000000,
274     Year = 1976,
275     Cast = "Stallone",
276     Director = "John G. Avildsen",
277     Duration = 136,
278     Genre = Genre.Action,
279     Plot = "Boxing",
280     PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAjlxeEDSg.jpg"
281   };
282 
283   _svc.AddMovie(m);
284 
285   // act
286   _svc.DeleteMovie(0); // delete non existant movie
287   var movies = _svc.GetAllMovies(); // retrieve list of movies
288 
289   // assert
290   Assert.Equal(1, movies.Count); // should be 1 student
291 }
292

```

```
293     // ----- REVIEW TESTS -----
294
295     [Fact]
296     public void Review_GetReviewsWhenNone_ShouldReturnNone()
297     {
298         // arrange
299
300         var reviews = _svc.GetAllReviews();
301         var count = reviews.Count();
302
303         // assert
304
305         Assert.Equal(0, count);
306     }
307
308     [Fact]
309     public void Review_AddReviewWhenUnique_ShouldSetAllProperties()
310     {
311         // arrange
312
313         var m = new Movie()
314         {
315             Title = "Rocky",
316             Budget = 1000000,
317             Year = 1976,
318             Cast = "Stallone",
319             Director = "John G. Avildsen",
320             Duration = 136,
321             Genre = Genre.Action,
322             Plot = "Boxing",
323             PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAJixeEDSg.jpg"
324         };
325
326         _svc.AddMovie(m);
327
328         var r = new Review()
329         {
330
331             MovieId = m.Id,
332             Name = "Joe Rogan",
333             CreatedOn = new DateTime(2019, 11, 06),
334             Comment = "Amazing",
335             Rating = 9
336         };
337
338
339         var o = _svc.AddReview(r);
340     }
```

```

340     var s = _svc.GetReviewById(o.Id); // retrieve review saved
341
342     // assert - that the review is not null
343     Assert.NotNull(s);
344     Assert.Equal(s.Id, s.Id);
345     Assert.Equal("Joe Rogan", s.Name);
346     Assert.Equal(s.CreatedOn, s.CreatedOn);
347     Assert.Equal("Amazing", s.Comment);
348     Assert.Equal(9, s.Rating);
349
350
351 }
352
353 [Fact]
354 0 references
355 public void Review_GetReviewsWhenTwoAdded_ShouldReturnTwo()
356 {
357     // arrange
358
359     var m = new Movie()
360     {
361         Title = "Rocky",
362         Budget = 1000000,
363         Year = 1976,
364         Cast = "Stallone",
365         Director = "John G. Avildsen",
366         Duration = 136,
367         Genre = Genre.Action,
368         Plot = "Boxing",
369         PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwd5srecBv07mIfAJixeED5g.jpg"
370     };
371
372     _svc.AddMovie(m);
373
374     var o = new Movie()
375     {
376         Title = "The Death of Stalin",
377         Budget = 13000000,
378         Year = 2017,
379         Cast = "Steve Buscemi, Simon Russel Beale, Paddy Considine",
380         Director = "Armando Iannucci",
381         Duration = 117,
382         Genre = Genre.Comedy,
383         Plot = "When the tyrannical ruler Stalin dies, his hapless inner circle scrambles to come up with the next evolution of the revolution ? but it's clear every",
384         PosterUrl = "https://www.themoviedb.org/t/p/original/66nL9hPerFZMOeCQuWD322Nm8g.jpg"
385     };
386
387     _svc.AddMovie(o);
388
389     var r1 = new Review()

```

```

387     var r1 = new Review()
388     {
389
390         Name = "Joe Rogan",
391         CreatedOn = new DateTime(2019, 11, 06),
392         Comment = "Amazing",
393         Rating = 9
394     };
395
396     var s1 = _svc.AddReview(r1);
397
398     var r2 = new Review()
399     {
400
401         Name = "Shayne",
402         CreatedOn = new DateTime(2018, 01, 12),
403         Comment = "Awful",
404         Rating = 2
405     };
406
407     var s2 = _svc.AddReview(r2);
408
409     // act
410
411     var reviews = _svc.GetAllReviews();
412     var count = reviews.Count;
413
414     // assert
415     Assert.Equal(2, count);
416 }
417
418 [Fact]
419 0 references
420 public void Review_GetReviewWhenNone_ShouldReturnNull()
421 {
422     // arrange
423
424     // act
425     var review = _svc.GetReviewById(1); // non existent review
426
427     // assert
428     Assert.Null(review);
429 }
430
431 [Fact]
432 0 references
433 public void Review_GetReviewThatExists_ShouldReturnReview()
434 {
435     // arrange
436     var m = new Movie()
437     {

```

```
430  [Fact]
431  0 references
432  public void Review_GetReviewThatExists_ShouldReturnReview()
433  {
434      // arrange
435      var m = new Movie()
436      {
437          Title = "Rocky",
438          Budget = 1000000,
439          Year = 1976,
440          Cast = "Stallone",
441          Director = "John G. Avildsen",
442          Duration = 136,
443          Genre = Genre.Action,
444          Plot = "Boxing",
445          PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAJixeEDSg.jpg"
446      };
447
448      _svc.AddMovie(m);
449
450      var r = new Review()
451      {
452          Name = "Joe Rogan",
453          CreatedOn = new DateTime(2019, 11, 06),
454          Comment = "Amazing",
455          Rating = 9
456      };
457
458      var s = _svc.AddReview(r);
459
460      // act
461      var ns = _svc.GetReviewById(r.Id);
462
463      // assert
464      Assert.NotNull(ns);
465      Assert.Equal(s.Id, ns.Id);
466  }
467
468  [Fact]
469  0 references
470  public void Review_DeleteReviewThatExists_ShouldReturnTrue()
471  {
472      // arrange
473      var m = new Movie()
474      {
475          Title = "Rocky",
476          Budget = 1000000,
```

```

468     [Fact]
469     0 references
470     public void Review_DeleteReviewThatExists_ShouldReturnTrue()
471     {
472         // arrange
473
474         var m = new Movie()
475         {
476             Title = "Rocky",
477             Budget = 1000000,
478             Year = 1976,
479             Cast = "Stallone",
480             Director = "John G. Avildsen",
481             Duration = 136,
482             Genre = Genre.Action,
483             Plot = "Boxing",
484             PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAJixeEDSG.jpg"
485         };
486
487         _svc.AddMovie(m);
488
489         var r = new Review()
490         {
491             Name = "Joe Rogan",
492             CreatedOn = new DateTime(2019, 11, 06),
493             Comment = "Amazing",
494             Rating = 9
495         };
496
497         var s = _svc.AddReview(r);
498
499         // act
500         var deleted = _svc.DeleteReview(s.Id);
501         var s1 = _svc.GetReviewById(s.Id);
502
503         // assert
504         Assert.True(deleted); // delete student should return true
505         Assert.Null(s1); // r1 should be null
506     }
507
508     [Fact]
509     0 references
510     public void Review_DeleteReviewThatExists_ShouldReduceReviewCountByOne()
511     {
512         // arrange
513         var m = new Movie()
514         {
515             Title = "Rocky",
516             Budget = 1000000

```

```

508     [Fact]
509     public void Review_DeleteReviewThatExists_ShouldReduceReviewCountByOne()
510     {
511         // arrange
512         var m = new Movie()
513         {
514             Title = "Rocky",
515             Budget = 1000000,
516             Year = 1976,
517             Cast = "Stallone",
518             Director = "John G. Avildsen",
519             Duration = 136,
520             Genre = Genre.Action,
521             Plot = "Boxing",
522             PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBvO7mIfAJixeEDSg.jpg"
523         };
524
525         _svc.AddMovie(m);
526
527         var r = new Review()
528         {
529             MovieId = m.Id,
530             Name = "Joe Rogan",
531             CreatedOn = new DateTime(2019, 11, 06),
532             Comment = "Amazing",
533             Rating = 9
534         };
535
536         var s = _svc.AddReview(r);
537
538         // act
539         var deleted = _svc.DeleteReview(s.Id);
540         var reviews = _svc.GetAllReviews();
541
542         // assert
543         Assert.Equal(0, reviews.Count); // should be 0 reviews
544     }
545
546     [Fact]
547     public void Review_DeleteReviewThatDoesntExist_ShouldNotChangeReviewCount()
548     {
549         // arrange
550         var m = new Movie()
551         {
552             Title = "Rocky",
553             Budget = 1000000,
554             Year = 1976,
555             Cast = "Stallone"

```

```

543     // assert
544     Assert.Equal(0, reviews.Count); // should be 0 reviews
545 }
546 }
547 [Fact]
548 0 references
549 public void Review_DeleteReviewThatDoesntExist_ShouldNotChangeReviewCount()
550 {
551     // arrange
552     var m = new Movie()
553     {
554         Title = "Rocky",
555         Budget = 1000000,
556         Year = 1976,
557         Cast = "Stallone",
558         Director = "John G. Avildsen",
559         Duration = 136,
560         Genre = Genre.Action,
561         Plot = "Boxing",
562         PosterUrl = "https://www.themoviedb.org/t/p/original/i5xiwdSsrecBv07mIfAJixeEDSg.jpg"
563     };
564
565     _svc.AddMovie(m);
566
567     var r = new Review()
568     {
569         MovieId = m.Id,
570         Name = "Joe Rogan",
571         CreatedOn = new DateTime(2019, 11, 06),
572         Comment = "Amazing",
573         Rating = 9
574     };
575
576     var s = _svc.AddReview(r);
577
578     // act
579     _svc.DeleteReview(0); // delete non existent review
580     var reviews = _svc.GetAllReviews(); // retrieve list of reviews
581
582     // assert
583     Assert.Equal(1, reviews.Count); // should be 1 review
584 }
585 }
586 }
587 }
588 }
589 }

```

MovieController.cs

```
12     namespace MMS.Web.Controllers
13     {
14
15         public class MovieController : BaseController
16         {
17             private IMovieService svc;
18
19             public MovieController()
20             {
21                 svc = new MovieServiceDb();
22             }
23
24             //GET /movie
25             public IActionResult Index()
26             {
27                 // retrieve all movies
28                 var movies = svc.GetAllMovies();
29
30                 return View(movies);
31             }
32
33             public IActionResult Directors()
34             {
35                 // retrieve all movies
36                 var movies = svc.GetAllMovies();
37
38                 return View(movies);
39             }
40
41             public IActionResult Titles()
42             {
43                 // retrieve all movies
44                 var movies = svc.GetAllMovies();
45
46                 return View(movies);
47             }
48
49             public IActionResult Years()
50             {
51                 // retrieve all movies
52                 var movies = svc.GetAllMovies();
53
54                 return View(movies);
55             }
56
57             0 references
```

```
0 references
57     public IActionResult Details(int id)
58     {
59         // retrieve the movie with specified id from the service
60         var m = svc.GetMovieById(id);
61
62         // Check if m is null and return NotFound()
63         if (m == null)
64         {
65             Alert("Movie not found", AlertType.Warning);
66             return RedirectToAction(nameof(Index));
67         }
68
69         // pass movie as parameter to the view
70
71         return View(m);
72     }
73
74
75     // GET /movie/create
76     public IActionResult Create()
77     {
78         // display blank form to create a movie
79         var m = new Movie();
80
81         if (m == null)
82         {
83             Alert("Movie not found", AlertType.Info);
84             return RedirectToAction(nameof(Index));
85         }
86
87         return View(m);
88     }
89
90     // POST /movie/create
91     [HttpPost]
92     public IActionResult Create(Movie m)
93     {
94         // verify the model is valid
95         if (ModelState.IsValid)
96         {
97             // pass data to service to store
98             svc.AddMovie(m);
99
100            Alert("Movie was created", AlertType.Success);
101
102            return RedirectToAction(nameof(Index));
103        }
104
105        // redisplay the form for editing and there are validation errors
```

```
104     // redisplay the form for editing and there are validation errors
105     return View(m);
106 }
107 }
108 }
109 // GET /movie/edit/{id}
110 public IActionResult Edit(int id)
111 {
112     // load the movie using the service
113     var m = svc.GetMovieById(id);
114
115     // check if m is null and return NotFound()
116     if (m == null)
117     {
118         Alert("Movie not found", AlertType.Warning);
119         return RedirectToAction(nameof(Index));
120     }
121
122     // pass movie to view for editing
123     return View(m);
124 }
125
126 // POST /movie/edit/{id}
127 [HttpPost]
128 public IActionResult Edit(int id, Movie m)
129 {
130     // verify movie model is valid
131     if (ModelState.IsValid)
132     {
133         // pass data to service to update
134         var updated = svc.UpdateMovie(m);
135
136         Alert("Movie was updated", AlertType.Success);
137
138         return RedirectToAction(nameof(Index));
139     }
140
141     // redisplay the form for editing as validation errors
142     return View(m);
143 }
144
145 // GET /movie/delete/{id}
146 public IActionResult Delete(int id)
147 {
148     // load the movie using the service
149     var m = svc.GetMovieById(id);
150
151     // check the returned movie is not null and if so return NotFound()
```

```
151     // check the returned movie is not null and if so return NotFound()
152     if (m == null)
153     {
154         Alert("Movie not found", AlertType.Warning);
155
156         return RedirectToAction(nameof(Index));
157     }
158
159     // pass movie to view for deletion confirmation
160     return View(m);
161 }
162
163 // POST /movie/delete/{id}
164 [HttpPost]
165 public IActionResult DeleteConfirm(int id)
166 {
167     // delete movie via service
168     svc.DeleteMovie(id);
169
170     // redirect to the index view
171     Alert("Movie was deleted", AlertType.Success);
172     return RedirectToAction(nameof(Index));
173 }
174
175 // GET /movies/createresview
176 [HttpGet]
177 public IActionResult CreateReview(int id)
178 {
179     var m = svc.GetMovieById(id);
180
181     // check the returned movie is not null and if so alert
182     if (m == null)
183     {
184         Alert($"No such movie {id}", AlertType.Warning);
185         return RedirectToAction(nameof(Index));
186     }
187
188     // create the review and populate the MovieId property
189     var r = new Review()
190     {
191         MovieId = id
192     };
193
194     return View(r);
195 }
196
197 // POST /movie/createresview
198 [HttpPost]
199
```

```
182
183     Alert($"No such movie {id}", AlertType.Warning);
184     return RedirectToAction(nameof(Index));
185 }
186
187 // create the review and populate the MovieId property
188 var r = new Review()
189 {
190     MovieId = id
191 };
192
193 return View(r);
194
195
196
197 // POST /movie/createresview
198 [HttpPost]
199 public IActionResult CreateReview(Review r)
200 {
201
202     // check the returned movie is not null
203     if (r == null)
204     {
205         Alert($"No such movie {r.MovieId}", AlertType.Warning);
206         return RedirectToAction(nameof(Index));
207     }
208
209     // create the review and populate the MovieId property
210     svc.AddReview(r);
211     Alert($"Review created successfully", AlertType.Success);
212
213     return RedirectToAction("Details", new {Id = r.MovieId});
214
215
216
217 }  
218 }  
219  
220  
221 }
```

ReviewController.cs

```
5  namespace MMS.Web.Controllers
6  {
7      public class ReviewController : BaseController
8      {
9          private IMovieService svc;
10
11         public ReviewController()
12         {
13             svc = new MovieServiceDb();
14         }
15
16         // GET /review/index
17         public IActionResult Index()
18         {
19             // retrieve all movies
20             var r = svc.GetAllReviews();
21
22             return View(r);
23         }
24
25         public IActionResult Details(int id)
26         {
27             // retrieve the review with specified id from the service
28             var r = svc.GetReviewById(id);
29
30             // Check if r is null
31             if (r == null)
32             {
33                 Alert("Review not found", AlertType.Warning);
34                 return RedirectToAction(nameof(Index));
35             }
36
37             // pass review as parameter to the view
38
39             return View(r);
40         }
41
42         // GET: /review/create
43         public IActionResult Create()
44         {
45             // display blank form to create a review
46             var r = new Review();
47             return View(r);
48         }
49     }
```

```
50     // POST /review/create
51     [HttpPost]
52     public IActionResult Create(Review r)
53     {
54         if (ModelState.IsValid)
55         {
56             // pass data to service to store
57             svc.AddReview(r);
58
59             Alert("Review created successfully", AlertType.Info);
60             return RedirectToAction(nameof(Index));
61         }
62
63         //redisplay the form for editing as there are validation errors
64         return View(r);
65     }
66
67     // GET /review/delete/{id}
68     public IActionResult Delete(int id)
69     {
70         // load the review using the service
71         var r = svc.GetReviewById(id);
72
73         // check the returned review is not null
74         if (r == null)
75         {
76             Alert("Review not found", AlertType.Warning);
77             return RedirectToAction(nameof(Index));
78         }
79         // pass review to view for deletion confirmation
80         return View(r);
81     }
82
83     // POST /review/delete/{id}
84     [HttpPost]
85     public IActionResult DeleteConfirm(int id)
86     {
87         // delete review via service
88         svc.DeleteReview(id);
89
90         Alert($"Review {id} deleted successfully", AlertType.Success);
91
92         // redirect to index view
93
94         return RedirectToAction(nameof(Index));
95     }
96 }
97 }
```

Movie Views

Index.cshtml

```
1  @model IList<Movie>
2
3  <style type="text/css">
4      .moviestitle {
5          font-weight: bold;
6      }
7  </style>
8
9  <partial name="_BreadCrumbs" model=@( new [] { ('/', "Home"), ("~/movie", "Movies"), ("", "Index") } )>
10 <a img src="https://fontmeme.com/permalink/210422/6a5d2e749bff56c52846a2401f8c3ale.png" alt="polential-neon-font" border="0"></a>
11
12 <div class="moviestitle">
13     <h2>
14         <a class="mb-3"></a>
15     </h2>
16 </div>
17
18 <p>
19     <a class="btn btn-lg btn-secondary my-5" asp-controller="Movie" asp-action="Create">Add Movie</a>
20 </p>
21
22 <table class="table">
23     <thead>
24         <tr>
25             <th>Id</th>
26             <th>Title</th>
27             <th>Director</th>
28             <th>Year</th>
29             <th>Duration</th>
30             <th>Budget</th>
31             <th>Poster</th>
32             <th>Genre</th>
33             <th>Cast</th>
34             <th>Options</th>
35         </tr>
36     </thead>
37
38     <tbody>
39         <foreach (var m in Model)>
40             <tr>
41                 <td>m.Id</td>
42                 <td>m.Title</td>
43                 <td>m.Director</td>
44                 <td>m.Year</td>
45                 <td>m.Duration minutes</td>
46                 <td><a href="#">Edit</a> | <a href="#">Delete</a></td>
47             </tr>
48         </foreach>
49     </tbody>
50 </table>
```

```

30     <th>Year</th>
31     <th>Duration</th>
32     <th>Budget</th>
33     <th>Poster</th>
34     <th>Genre</th>
35     <th>Cast</th>
36     <th>Options</th>
37   </tr>
38 </thead>
39
40 <tbody>
41   @foreach (var m in Model)
42   {
43
44     <tr>
45       <td>@m.Id</td>
46       <td>@m.Title</td>
47       <td>@m.Director</td>
48       <td>@m.Year</td>
49       <td>@m.Duration minutes</td>
50       <td>$@m.Budget</td>
51       <td></td>
52       <td>@m.Genre</td>
53       <td>@m.Cast</td>
54
55
56       <td>
57         <!--Link to details view-->
58         <a asp-controller="Movie" asp-action="Details" asp-route-id=@m.Id>Details</a>
59         <p></p>
60         <!-- Link to edit view -->
61         <a asp-controller="Movie" asp-action="Edit" asp-route-id=@m.Id>Edit</a>
62         <p></p>
63         <!-- Link to create a review -->
64         <a asp-controller="Movie" asp-action="CreateReview" asp-route-id=@m.Id>Add Review</a>
65         <p></p>
66         <!-- Link to Delete view -->
67         <a asp-controller="Movie" asp-action="Delete" asp-route-id=@m.Id>Delete</a>
68       </td>
69     </tr>
70   }
71 </tbody>
72 </table>

```

Details.cshtml

```
1  @model Movie
2
3
4  <partial name="_Breadcrumbs" model=@(new [] {("/", "Home"), ("/movie", "Movies"), ("", "Details")})>
5
6  <h3 class="mt-4 mb-4">Movie Details</h3>
7
8  <div class="card shadow rounded p-3">
9    <div class="row pl-3">
10      <div class="col-8 pt-3">
11        <dl class="row">
12          <dt class="col-5">Title</dt>
13          <dd class="col-7">@Model.Title</dd>
14
15          <dt class="col-5">Director</dt>
16          <dd class="col-7">@Model.Director</dd>
17
18          <dt class="col-5">Genre</dt>
19          <dd class="col-7">@Model.Genre</dd>
20
21          <dt class="col-5">Year</dt>
22          <dd class="col-7">@Model.Year</dd>
23
24          <dt class="col-5">Duration</dt>
25          <dd class="col-7">@Model.Duration minutes</dd>
26
27          <dt class="col-5">Budget</dt>
28          <dd class="col-7">$@Model.Budget</dd>
29
30          <dt class="col-5">Cast</dt>
31          <dd class="col-7">@Model.Cast</dd>
32
33          <dt class="col-5">Plot</dt>
34          <dd class="col-7">@Model.Plot</dd>
35
36          <dt class="col-5">Rating</dt>
37          <dd class="col-7">@Model.Rating</dd>
38        </dl>
39      </div>
40      <div class="col-4">
41        
42      </div>
43
44    </div>
45  </div>
46
47  <div>
48    <partial name="_Reviews">
49  </div>
```

Create.cshtml

```
1  @model Movie
2
3  @<partial name="_Breadcrumbs" model=@(new [ ] {
4      ("/", "Home"),
5      ("/movie", "Movies"),
6      ("", "Create")
7 })>
8
9  <div class="card rounded shadow p-3">
10     <h3 class="mb-4">Create Movie</h3>
11
12     <form asp-action="Create">
13         <div asp-validation-summary="ModelOnly" class="text-danger"></div>
14
15         <div class="form-group">
16             <label asp-for="Title" class="control-label"></label>
17             <input asp-for="Title" class="form-control" />
18             <span asp-validation-for="Title" class="text-danger"></span>
19         </div>
20
21         <div class="form-group">
22             <label asp-for="Director" class="control-label"></label>
23             <input asp-for="Director" class="form-control" />
24             <span asp-validation-for="Director" class="text-danger"></span>
25         </div>
26
27         <div class="form-group">
28             <label asp-for="Year" class="control-label"></label>
29             <input asp-for="Year" class="form-control" />
30             <span asp-validation-for="Year" class="text-danger"></span>
31         </div>
32
33         <div class="form-group">
34             <label asp-for="Duration" class="control-label"></label>
35             <input asp-for="Duration" class="form-control" />
36             <span asp-validation-for="Duration" class="text-danger"></span>
37         </div>
38
39
40         <div class="form-group">
41             <label asp-for="Genre" class="control-label"></label>
42             <input asp-for="Genre" class="form-control" />
43             <span asp-validation-for="Genre" class="text-danger"></span>
44         </div>
45
46
47         <!-- add PhotoUrl input group - ensure input field type is url -->
48         <div class="form-group">
49             <label asp-for="PosterUrl" class="control-label"></label>
50             <input type="url" asp-for="PosterUrl" class="form-control" />
```

```
28     <div class="form-group">
29         <label asp-for="Year" class="control-label"></label>
30         <input asp-for="Year" class="form-control" />
31         <span asp-validation-for="Year" class="text-danger"></span>
32     </div>
33
34     <div class="form-group">
35         <label asp-for="Duration" class="control-label"><label>
36         <input asp-for="Duration" class="form-control" />
37         <span asp-validation-for="Duration" class="text-danger"></span>
38     </div>
39
40     <div class="form-group">
41         <label asp-for="Genre" class="control-label"></label>
42         <input asp-for="Genre" class="form-control" />
43         <span asp-validation-for="Genre" class="text-danger"></span>
44     </div>
45
46     <!-- add PhotoUrl input group - ensure input field type is url -->
47     <div class="form-group">
48         <label asp-for="PosterUrl" class="control-label"></label>
49         <input type="url" asp-for="PosterUrl" class="form-control" />
50         <span asp-validation-for="PosterUrl" class="text-danger"></span>
51     </div>
52
53     <div class="form-group">
54         <input type="submit" value="Create" class="btn btn-primary" />
55         <a class="btn btn-secondary" asp-action="Index">Cancel</a>
56     </div>
57
58 </form>
59 </div>
60
61
62 @section Scripts {
63     @await Html.RenderPartialAsync("_ValidationScriptsPartial");
64 }
65
66 }
```

Edit.cshtml

```
1  @model Movie
2
3  <partial name="_Breadcrumbs" model=@(new [] {
4      ("/", "Home"),
5      ("/movie", "Movies"),
6      ($"/movie/details/{Model.Id}", $"{Model.Id}"),
7      ("","Edit")
8 })>
9
10
11 <div class="card rounded shadow p-3">
12     <h3 class="mb-4">Edit Movie</h3>
13
14     <form asp-action="Edit">
15
16         <div asp-validation-summary="ModelOnly" class="text-danger"></div>
17
18         <!-- we hide the id as its not user editable but needs to be in the -->
19         <!-- form to ensure the student passed to the controller is complete -->
20         <input type="hidden" asp-for="Id" class="form-control" />
21
22         <!-- add form groups for each student attributed being edited -->
23
24         <div class="form-group">
25             <label asp-for="Title" class="control-label"></label>
26             <input asp-for="Title" class="form-control" />
27             <span asp-validation-for="Title" class="text-danger"></span>
28         </div>
29
30         <div class="form-group">
31             <label asp-for="Director" class="control-label"></label>
32             <input asp-for="Director" class="form-control" />
33             <span asp-validation-for="Director" class="text-danger"></span>
34         </div>
35
36         <div class="form-group">
37             <label asp-for="Year" class="control-label"></label>
38             <input asp-for="Year" class="form-control" />
39             <span asp-validation-for="Year" class="text-danger"></span>
40         </div>
41
42         <div class="form-group">
43             <label asp-for="Duration" class="control-label"></label>
44             <input asp-for="Duration" class="form-control" />
45             <span asp-validation-for="Duration" class="text-danger"></span>
46         </div>
47
48         <div class="form-group">
49             <label asp-for="Year" class="control-label"></label>
50             </div>
```

```
37
38
39     <div class="form-group">
40         <label asp-for="Year" class="control-label"></label>
41         <input asp-for="Year" class="form-control" />
42         <span asp-validation-for="Year" class="text-danger"></span>
43     </div>
44
45     <div class="form-group">
46         <label asp-for="Duration" class="control-label"></label>
47         <input asp-for="Duration" class="form-control" />
48         <span asp-validation-for="Duration" class="text-danger"></span>
49     </div>
50
51
52     <div class="form-group">
53         <label asp-for="Genre" class="control-label"></label>
54         <input asp-for="Genre" class="form-control" />
55         <span asp-validation-for="Genre" class="text-danger"></span>
56     </div>
57
58     <!-- add PhotoUrl input group - ensure input field type is url -->
59     <div class="form-group">
60         <label asp-for="PosterUrl" class="control-label"></label>
61         <input type="url" asp-for="PosterUrl" class="form-control" />
62         <span asp-validation-for="PosterUrl" class="text-danger"></span>
63     </div>
64
65     <!-- you can copy the necessary groups from Create.cshtml --->
66     <!-- submit button and cancel link which navigates to Index action -->
67     <div class="form-group">
68         <input type="submit" value="Save" class="btn btn-primary" />
69         <a class="btn btn-secondary" asp-action="Details" asp-route-id="@Model.Id">Cancel</a>
70     </div>
71
72 </div>
73
74 @section Scripts {
75     @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
76 }
77
```

CreateReview.cshtml

```
1  @model Review
2
3  <partial name="_BreadCrumbs" model=@(new [] {
4      ("/", "Home"),
5      ("/movie", "Movies"),
6      ($"/movie/details/{Model.MovieId}", $"{Model.MovieId}"),
7      ("", "CreateReview") })>
8
9  <h3 class="mt-4 mb-4">Create Review</h3>
10
11 <form asp-action="CreateReview">
12     <div asp-validation-summary="ModelOnly" class="text-danger"></div>
13
14     <!-- we hide the Model MovieId as its not user editable but need to be in the form
15        to ensure the Review passed to the controller is complete -->
16
17     <input type="hidden" asp-for="MovieId" class="form-control" />
18
19     <div class="form-group">
20         <label asp-for="Name" class="control-label"></label>
21         <textarea rows="1" asp-for="Name" class="form-control"></textarea>
22         <span asp-validation-for="Name" class="text-danger"></span>
23     </div>
24
25     <!--
26     <div class="form-group">
27         <label asp-for="CreatedOn" class="control-label"></label>
28         <input type="date" value="CreatedOn" />
29         <span asp-validation-for="CreatedOn" class="text-danger"></span>
30     </div> -->
31
32     <div class="form-group">
33         <label asp-for="Comment" class="control-label"></label>
34         <textarea rows="5" asp-for="Comment" class="form-control"></textarea>
35         <span asp-validation-for="Comment" class="text-danger"></span>
36     </div>
37
38     <div class="form-group">
39         <label asp-for="Rating" class="control-label"></label>
40         <textarea rows="1" asp-for="Rating" class="form-control"></textarea>
41         <span asp-validation-for="Rating" class="text-danger"></span>
42     </div>
43
44     <div class="form-group">
45         <input type="submit" value="Create" class="btn btn-secondary" />
46         <a class="btn btn-secondary" asp-action="Details" asp-route-id=@Model.MovieId>Cancel</a>
47     </div>
```

```
19     <div class="form-group">
20         <label asp-for="Name" class="control-label"></label>
21         <textarea rows="1" asp-for="Name" class="form-control"></textarea>
22         <span asp-validation-for="Name" class="text-danger"></span>
23     </div>
24
25     <!--
26     <div class="form-group">
27         <label asp-for="CreatedOn" class="control-label"></label>
28         <input type="date" value="CreatedOn" />
29         <span asp-validation-for="CreatedOn" class="text-danger"></span>
30     </div> -->
31
32     <div class="form-group">
33         <label asp-for="Comment" class="control-label"></label>
34         <textarea rows="5" asp-for="Comment" class="form-control"></textarea>
35         <span asp-validation-for="Comment" class="text-danger"></span>
36     </div>
37
38     <div class="form-group">
39         <label asp-for="Rating" class="control-label"></label>
40         <textarea rows="1" asp-for="Rating" class="form-control"></textarea>
41         <span asp-validation-for="Rating" class="text-danger"></span>
42     </div>
43
44     <div class="form-group">
45         <input type="submit" value="Create" class="btn btn-secondary" />
46         <a class="btn btn-secondary" asp-action="Details" asp-route-id=@Model.MovieId>Cancel</a>
47     </div>
48
49
50
51
52
53
54 </form>
55
56     @section Scripts {
57         @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
58     }
```

Delete.cshtml

```
1  @model Movie
2
3  <partial name="_Breadcrumbs" model=@new [] {
4      ("/", "Home"),
5      ("/movie", "Movies"),
6      ($"/movie/details/{Model.Id}", $"{Model.Id}"),
7      ("","Delete")
8}>
9
10 <div>
11     <!-- display the movie using a &lt;dl&gt; as per Details.cshtml --&gt;
12     &lt;div class="col card rounded shadow pt-3"&gt;
13
14         &lt;h3 class="mb-4"&gt;Delete Movie&lt;/h3&gt;
15
16         &lt;h5 class="mb-4"&gt;Are you sure you want to delete this movie?&lt;/h5&gt;
17
18         &lt;dl class="row"&gt;
19             &lt;dt class="col-5"&gt;Title&lt;/dt&gt;
20             &lt;dd class="col-7"&gt;@Model.Title&lt;/dd&gt;
21
22             &lt;dt class="col-5"&gt;Director&lt;/dt&gt;
23             &lt;dd class="col-7"&gt;@Model.Director&lt;/dd&gt;
24
25             &lt;dt class="col-5"&gt;Year&lt;/dt&gt;
26             &lt;dd class="col-7"&gt;@Model.Year&lt;/dd&gt;
27
28             &lt;dt class="col-5"&gt;Genre&lt;/dt&gt;
29             &lt;dd class="col-7"&gt;@Model.Genre&lt;/dd&gt;
30
31             &lt;dt class="col-5"&gt;Poster&lt;/dt&gt;
32             &lt;dd class="col-7"&gt;&lt;img src="@Model.PosterUrl" class="img-fluid" width="150" height="400"/&gt;&lt;/dd&gt;
33         &lt;/dl&gt;
34     &lt;/div&gt;
35
36     &lt;div class="mt-4"&gt;
37         <!-- create a form to submit post request to DeleteConfirm action
38             student Id should be a hidden input field named 'Id' in the form
39             add type submit input button
40             add cancel link which calls Index action --&gt;
41         &lt;form asp-controller="Movie" asp-action="DeleteConfirm" type="post"&gt;
42             &lt;input type="hidden" asp-for="Id"/&gt;
43             &lt;input type="submit" value="Confirm" class="btn btn-danger" /&gt;
44             &lt;a class="btn btn-link" asp-controller="Movie" asp-action="Index"&gt;Cancel&lt;/a&gt;
45         &lt;/form&gt;
46     &lt;/div&gt;
47 &lt;/div&gt;</pre>
```

_Reviews.cshtml

```
1  @model Movie
2  <partial name="_BreadCrumbs" model=@(new [] {
3      ("/", "Home"),
4      ("/movie", "Movies"),
5      ("$/movie/details/{Model.Title}", $"{Model.Title}"),
6      ("", "CreateReview") })>
7
8  <table class="table">
9    <thead>
10   <tr>
11     <th>Name</th>
12     <th>Created On</th>
13     <th>Comment</th>
14     <th>Rating</th>
15   </tr>
16 </thead>
17 <tbody>
18   @foreach (var r in Model.Reviews)
19   {
20     <tr>
21       <td>@r.Name</td>
22       <td>@r.CreatedOn.ToString("MM/dd/yy H:mm")</td>
23       <td>@r.Comment</td>
24       <td>@r.Rating</td>
25     </tr>
26
27   }
28 </tbody>
29
30 </table>
31
32
```

Directors.cshtml

```
1  @model IList<Movie>
2
3  <style type="text/css">
4      .movieTitle {
5          font-weight: bold;
6      }
7  </style>
8
9  <partial name="_Breadcrumbs" model=@( new [] { ('/', "Home"), ("movie", "Movies"), ("", "Directors") } )>
10
11     <a img src="https://fontmeme.com/permalink/210422/6a5d2e749bfff56c52846a2401f8c3ale.png" alt="polientical-neon-font" border="0"></a>
12
13     <div class="movieTitle">
14         <h2>
15             <a class="mb-3"></a>
16         </h2>
17     </div>
18
19     <p>
20         <a class="btn btn-lg btn-secondary my-5" asp-controller="Movie" asp-action="Create">Add Movie</a>
21     </p>
22
23     <table class="table">
24         <thead>
25             <tr>
26                 <th>Id</th>
27                 <th>Title</th>
28                 <th>Director</th>
29                 <th>Year</th>
30                 <th>Duration</th>
31                 <th>Budget</th>
32                 <th>Poster</th>
33                 <th>Genre</th>
34                 <th>Cast</th>
35                 <th>Options</th>
36             </tr>
37         </thead>
38         <tbody>
39             &gt;foreach (var m in Model.OrderBy(d => d.Director))
40             &gt;
41             <tr>
42                 <td>@m.Id</td>
43                 <td>@m.Title</td>
44                 <td>@m.Director</td>
45                 <td>@m.Year</td>
46                 <td>@m.Duration_minutes</td>
47             </tr>
48         </tbody>
49     </table>
50 
```

```

33     <th>Budget</th>
34     <th>Poster</th>
35     <th>Genre</th>
36     <th>Cast</th>
37     <th>Options</th>
38   </tr>
39 </thead>
40
41   <tbody>
42     @foreach (var m in Model.OrderBy(d => d.Director))
43     {
44
45       <tr>
46         <td>@m.Id</td>
47         <td>@m.Title</td>
48         <td>@m.Director</td>
49         <td>@m.Year</td>
50         <td>@m.Duration minutes</td>
51         <td>@m.Budget</td>
52         <td></td>
53         <td>@m.Genre</td>
54         <td>@m.Cast</td>
55
56
57         <td>
58           <!--Link to details view-->
59           <a asp-controller="Movie" asp-action="Details" asp-route-id=@m.Id>Details</a>
60           <p></p>
61           <!-- Link to edit view -->
62           <a asp-controller="Movie" asp-action="Edit" asp-route-id=@m.Id>Edit</a>
63           <p></p>
64           <!-- Link to create a review -->
65           <a asp-controller="Movie" asp-action="CreateReview" asp-route-id=@m.Id>Add Review</a>
66           <p></p>
67           <!-- Link to Delete view -->
68           <a asp-controller="Movie" asp-action="Delete" asp-route-id=@m.Id>Delete</a>
69         </td>
70       </tr>
71     </tbody>
72   </table>
73
74

```

Titles.cshtml

```
1  @model IList<Movie>
2
3  <style type="text/css">
4      .movieTitle {
5          font-weight: bold;
6      }
7  </style>
8
9  <partial name="_Breadcrumbs" model=> new [] {("/", "Home"), ("/movie", "Movies"), ("", "Titles") } >
10
11     <a img src="https://fontmeme.com/permalink/210422/6a5d2e749bfff56c52846a2401f8c3ale.png" alt="polentical-neon-font" border="0"></a>
12
13     <div class="movieTitle">
14         <h2>
15             <a class="mb-3"></a>
16         </h2>
17     </div>
18
19
20     <p>
21         <a class="btn btn-lg btn-secondary my-5" asp-controller="Movie" asp-action="Create">Add Movie</a>
22     </p>
23
24     <table class="table">
25         <thead>
26             <tr>
27                 <th>Id</th>
28                 <th>Title</th>
29                 <th>Director</th>
30                 <th>Year</th>
31                 <th>Duration</th>
32                 <th>Budget</th>
33                 <th>Poster</th>
34                 <th>Genre</th>
35                 <th>Cast</th>
36                 <th>Options</th>
37             </tr>
38         </thead>
39
40         <tbody>
41             <foreach var m in Model.OrderBy(t => t.Title)>
42                 <tr>
43                     <td>m.Id</td>
44                     <td>m.Title</td>
45                     <td>m.Director</td>
46                     <td>m.Year</td>
47                     <td>m.Duration minutes</td>
48                 </tr>
49             </foreach>
50         </tbody>
51     </table>
52
53     <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha256-3P8yE3dSkuKagrr7+XJ40ZlExGd29wqGnTdlRwJ40=" crossorigin="anonymous"></script>
54
55     <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js" integrity="sha384-kA1af5gC4ddCbQ3kXW39Va2HrOj1i5k12QFZ4ZNaGCaT1WQf2dVbiZxQ8LuJO" crossorigin="anonymous"></script>
56
57     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-gQXoLd03uKZjxNtEEoIz8AcxUv2ew6M5ar1dfzq4+uqBp4l3" crossorigin="anonymous"></script>
```

```
33     <th>Poster</th>
34     <th>Genre</th>
35     <th>Cast</th>
36     <th>Options</th>
37   </tr>
38 </thead>
39
40 <tbody>
41   @foreach (var m in Model.OrderBy(t => t.Title))
42   {
43
44     <tr>
45       <td>@m.Id</td>
46       <td>@m.Title</td>
47       <td>@m.Director</td>
48       <td>@m.Year</td>
49       <td>@m.Duration minutes</td>
50       <td>@m.Budget</td>
51       <td></td>
52       <td>@m.Genre</td>
53       <td>@m.Cast</td>
54
55
56       <td>
57         <!--Link to details view-->
58         <a asp-controller="Movie" asp-action="Details" asp-route-id=@m.Id>Details</a>
59         <p></p>
60         <!-- Link to edit view -->
61         <a asp-controller="Movie" asp-action="Edit" asp-route-id=@m.Id>Edit</a>
62         <p></p>
63         <!-- Link to create a review -->
64         <a asp-controller="Movie" asp-action="CreateReview" asp-route-id=@m.Id>Add Review</a>
65         <p></p>
66         <!-- Link to Delete view -->
67         <a asp-controller="Movie" asp-action="Delete" asp-route-id=@m.Id>Delete</a>
68       </td>
69     </tr>
70   </tbody>
71 </table>
```

Years.cshtml

```
1  @model IList<Movie>
2
3  <style type="text/css">
4      .moviestitle {
5          font-weight: bold;
6      }
7  </style>
8
9  <partial name="Breadcrumbs" model=>8( new [] { ("", "Home"), ("/movie", "Movies"), ("", "Years") } )>
10
11     <a href="#">https://fontmeme.com/permalink/210422/6a5d2e749bff56c52846a2401f8c3a1e.png</a>
12
13     <div class="moviestitle">
14         <h2>
15             <a href="#">https://fontmeme.com/permalink/210422/6a5d2e749bff56c52846a2401f8c3a1e.png</a> polential-neon-font border="0" width="150" height="29"></h2>
16         </div>
17
18
19     <p>
20         <a href="#" class="btn btn-lg btn-secondary my-5" asp-controller="Movie" asp-action="Create">Add Movie</a>
21     </p>
22
23
24     <table class="table">
25         <thead>
26             <tr>
27                 <th>Id</th>
28                 <th>Title</th>
29                 <th>Director</th>
30                 <th>Year</th>
31                 <th>Duration</th>
32                 <th>Budget</th>
33                 <th>Poster</th>
34                 <th>Genre</th>
35                 <th>Cast</th>
36                 <th>Options</th>
37             </tr>
38         </thead>
39
40         <tbody>
41             <foreach (var m in Model.OrderBy(y => y.Year))>
42                 <tr>
43                     <td>m.Id</td>
44                     <td>m.Title</td>
45                     <td>m.Director</td>
46                     <td>m.Year</td>
47                     <td>m.Duration minutes</td>
48                     <td>...</td>
49                 </tr>
50             </foreach>
51         </tbody>
52     </table>
53
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
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
727
728
729
729
730
731
732
733
734
735
735
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
154
```

```
33     <th>Poster</th>
34     <th>Genre</th>
35     <th>Cast</th>
36     <th>Options</th>
37   </tr>
38 </thead>
39
40 <tbody>
41   @foreach (var m in Model.OrderBy(y => y.Year))
42   {
43
44     <tr>
45       <td>@m.Id</td>
46       <td>@m.Title</td>
47       <td>@m.Director</td>
48       <td>@m.Year</td>
49       <td>@m.Duration minutes</td>
50       <td>@m.Budget</td>
51       <td></td>
52       <td>@m.Genre</td>
53       <td>@m.Cast</td>
54
55
56       <td>
57         <!--Link to details view-->
58         <a asp-controller="Movie" asp-action="Details" asp-route-id=@m.Id>Details</a>
59         <p></p>
60         <!-- Link to edit view -->
61         <a asp-controller="Movie" asp-action="Edit" asp-route-id=@m.Id>Edit</a>
62         <p></p>
63         <!-- Link to create a review -->
64         <a asp-controller="Movie" asp-action="CreateReview" asp-route-id=@m.Id>Add Review</a>
65         <p></p>
66         <!-- Linke to Delete view -->
67         <a asp-controller="Movie" asp-action="Delete" asp-route-id=@m.Id>Delete</a>
68       </td>
69     </tr>
70   }
71 </tbody>
72 </table>
```

Review views

Index.cshtml

```
1  @model IList<Review>
2
3
4  <partial name="_BreadCrumbs" model=@( new [] {("/", "Home"), ("/review", "Reviews"), ("", "Index") } )>
5
6  <h3>Review List</h3>
7
8  <table class="table">
9    <thead>
10      <tr>
11        <th>Id</th>
12        <th>Movie</th>
13        <th>Name</th>
14        <th>Created On</th>
15        <th>Comment</th>
16        <th>Rating</th>
17        <th>Delete</th>
18      </tr>
19    </thead>
20
21
22    <tbody>
23
24
25    @foreach (var r in Model)
26    {
27      <tr>
28        <td>@r.Id</td>
29        <td>@r.Movie.Title</td>
30        <td>@r.Name</td>
31        <td>@r.CreatedOn.ToShortDateString()</td>
32        <td>@r.Comment</td>
33        <td>@r.Rating</td>
34
35        <td class="btn btn-danger btn-sm rounded">
36          <a asp-controller="Review" asp-action="Delete" asp-route-id=@r.Id>Delete</a>
37        </td>
38
39      </tr>
40
41    }
42
43
44  </tbody>
45  </table>
```

Create.cshtml

```
1  @model Review
2
3  <partial name="_Breadcrumbs" model=@(new [] {("/", "Home"), ("review", "Reviews"), ("", "Create")))>
4
5  <h3 class="mt-4 mb-4">Create Review</h3>
6
7  <form asp-action="Create">
8    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
9
10   <input type="hidden" asp-for="MovieId" class="form-control" />
11
12   <div class="form-group">
13     <label asp-for="Name" class="control-label"></label>
14     <textarea rows="1" asp-for="Name" class="form-control"></textarea>
15     <span asp-validation-for="Name" class="text-danger"></span>
16   </div>
17
18   <!--
19   <div class="form-group">
20     <label asp-for="CreatedOn" class="control-label"></label>
21     <input type="date" value="CreatedOn" />
22     <span asp-validation-for="CreatedOn" class="text-danger"></span>
23   </div>-->
24
25   <div class="form-group">
26     <label asp-for="Comment" class="control-label"></label>
27     <textarea rows="5" asp-for="Comment" class="form-control"></textarea>
28     <span asp-validation-for="Comment" class="text-danger"></span>
29   </div>
30
31   <div class="form-group">
32     <label asp-for="Rating" class="control-label"></label>
33     <textarea rows="1" asp-for="Rating" class="form-control"></textarea>
34     <span asp-validation-for="Rating" class="text-danger"></span>
35   </div>
36
37
38   <input type="submit" value="Create" class="btn btn-secondary" />
39   <a class="btn btn-secondary" asp-action="Index">Cancel</a>
40
41
42
43 </form>
44
45 @section Scripts
46 {
47   @{ await Html.RenderPartialAsync("_ValidationScriptsPartial");}
48 }
```

Delete.cshtml

```
1 @model Review
2
3 @<partial name="_BreadCrumbs" model=>(new [] {("/", "Home"), ("/movie", "Movies"), ($"/movie/details/{Model.MovieId}", $"{Model.MovieId}"), ("", "Delete Review")}).
4 .
5 <div>
6     <div class="col card rounded shadow pt-3">
7         <h3 class="mb-4">Delete Review</h3>
8         <h5 class="mb-4">Are you sure you want to delete this review?</h5>
9
10        <dl class="row">
11            <dt class="col-5">Movie</dt>
12            <dd class="col-7">{Model.Movie}</dd>
13
14            <dt class="col-5">Name</dt>
15            <dd class="col-7">{Model.Name}</dd>
16
17            <dt class="col-5">CreatedOn</dt>
18            <dd class="col-7">{Model.CreatedOn}</dd>
19
20            <dt class="col-5">Comment</dt>
21            <dd class="col-7">{Model.Comment}</dd>
22
23            <dt class="col-5">Rating</dt>
24            <dd class="col-7">{Model.Rating}</dd>
25        </dl>
26    </div>
27
28    <div class="mt-4">
29        <form asp-controller="Review" asp-action="DeleteConfirm" type="post">
30            <input type="hidden" asp-for="Model.Id" />
31            <input type="submit" value="Confirm" class="btn btn-danger" />
32            <a class="btn btn-link" asp-controller="Review" asp-action="Index">Cancel</a>
33        </form>
34    </div>
35
36 </div>
37
38
39
40
41
42
43
44
45
46
47
48
```

Website screenshots

Home

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

Welcome to Retrospective Reviews

The people's movie review site.



About

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

Retrospective Reviews

Retrospective Reviews was set up for everyone to be able to submit reviews for any movie of their choosing.

Formed on 24 March 2021 (47 days ago)

© 2021, Retrospective Reviews - Privacy

© 2021, Retrospective Reviews - Privacy

Reviews

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

Home / Reviews / Index

Review List

ID	Movie	Name	Created On	Comment	Rating	Delete
1	The Matrix	Patry	06/05/2021	Loved it. One of the best movies of all time. Do yourself a favour if you haven't seen it and watch it!	9	<button>Update</button>
2	Killer Klowns From Outer Space	Seamus	06/05/2021	Stupid movie from the 80s. Soundtrack is surprisingly brilliant and goofy.	7	<button>Update</button>
3	The Founder	Korstantin	06/05/2021	Who knew that the story of McDonald's would be so entertaining.	8	<button>Update</button>
4	They Live	Jeff	06/05/2021	Cult classic of the 80s. Some great one liners	8	<button>Update</button>
5	El Camino: A Breaking Bad Movie	Donmacha	06/05/2021	Great conclusion to Jessie Pinkman's story. Loved the ending of Breaking Bad, but couldn't help but feel like Jessie's story should of been elaborated a bit more at the end. This criticism has been well and truly rectified.	9	<button>Update</button>
6	Easy Rider	Francis	06/05/2021	You will never be as cool as this movie.	10	<button>Update</button>
7	Airheads	Max	06/05/2021	Overlooked classic from the 90s.	8	<button>Update</button>
8	The Punisher	Shane	06/05/2021	This movie deserved a sequel. Performed poorly with the critics, but I thought it was great.	8	<button>Update</button>
9	Rambo	Kev	06/05/2021	First Blood is directed by Ted Kotcheff and adapted by Michael Korda, William Suckheim and	9	<button>Update</button>

Privacy

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

Privacy Policy

Last updated: April 28, 2021

This Privacy Policy describes Our policies and procedures on the collection, use and disclosure of Your information when You use the Service and tells You about Your privacy rights and how the law protects You.

We use Your Personal data to provide and improve the Service. By using the Service, You agree to the collection and use of information in accordance with this Privacy Policy. This Privacy Policy has been created with the help of the Privacy Policy Generator.

Interpretation and Definitions

Interpretation

The words of which the initial letter is capitalized have meanings defined under the following conditions. The following definitions shall have the same meaning regardless of whether they appear in singular or in plural.

Definitions

For the purposes of this Privacy Policy:

- Account means a unique account created for You to access our Service or parts of our Service.
- Company referred to as either "the Company", "We", "Us" or "Our" in this Agreement refers to Retrospective, Grove Street, Los Santos, San Andreas.
- Cookies are small files that are placed on Your computer, mobile device or any other device by a website, containing the details of Your browsing history on that website among its many uses.
- Country refers to: Tonga
- Device means any device that can access the Service such as a computer, a cellphone or a digital tablet.

Movies by Id

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

Home / Movies / Index

MOVIES

Add Movie

ID	Title	Director	Year	Duration	Budget	Poster	Genre	Cast	Options
1	The Matrix	Lilly Wachowski, Lana Wachowski	2001	136 minutes	\$63000000		Action	Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss, Hugo Weaving, Joe Pantoliano	Details Edit Add Review Delete
2	Killer Klowns From Outer Space	Stephen Chiodo	1988	88 minutes	\$200000		Horror	Grant Cramer, Suzanne Snyder, John Vernon, Royal Dano	Details Edit Add Review

Movies by Year

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

Home / Movies / Years

MOVIES

Add Movie

ID	Title	Director	Year	Duration	Budget	Poster	Genre	Cast	Options
6	Easy Rider	Dennis Hopper	1969	95 minutes	\$360000		Drama	Peter Fonda, Dennis Hopper, Antonio Mendoza, Phil Spector	Details Edit Add Review Delete
11	Apocalypse Now	Francis Ford Coppola	1979	147 minutes	\$3150000		War	Martin Sheen, Marlon Brando, Robert Duvall, Frederic Forrest, Laurence Fishbourne	Details Edit Add Review

Movies by Title

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

Home / Movies / Titles

MOVIES

Add Movie

ID	Title	Director	Year	Duration	Budget	Poster	Genre	Cast	Options
7	Airheads	Michael Lehmann	1994	92 minutes	\$11200000		Comedy	Brendon Frazer, Steve Buscemi, Adam Sandler, Chris Farley, Judd Nelson, Michael McKean	Details Edit Add Review Delete
11	Apocalypse Now	Francis Ford Coppola	1979	147 minutes	\$3150000		War	Martin Sheen, Marlon Brando, Robert Duvall, Frederic Forrest, Laurence Fishbourne	Details Edit Add Review

Movies by Director

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

Home / Movies / Directors

MOVIES

Add Movie

ID	Title	Director	Year	Duration	Budget	Poster	Genre	Cast	Options
15	Se7en	David Fincher	1995	127 minutes	\$33000000		Crime	Brad Pitt, Morgan Freeman, Gwyneth Paltrow, R. Lee Ermey, John C. McGinley	Details Edit Add Review Delete
6	Easy Rider	Dennis Hopper	1969	95 minutes	\$360000		Drama	Peter Fonda, Dennis Hopper, Antonio Mendoza, Phil Spector	Details Edit Add Review

Create Movie

[Home](#) / [Movies](#) / Create

Create Movie

Title

Director

Year

Duration

Genre

Action

PosterUrl

Movie Details

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy



[Home](#) / [Movies](#) / [The Matrix](#) / [CreateReview](#)

Name	Created On	Comment	Rating
Patsy	05/06/21 18:05	Loved It. One of the best movies of all time. Do yourself a favour if you haven't seen It and watch this!	9

Edit Movie

[Home](#) / [Movies](#) / [1](#) / [Edit](#)

Edit Movie

Title:

Director:

Year:

Duration:

Genre:

Poster URL:

[Cancel](#)

Delete Movie

[RETROSPECTIVE REVIEWS](#) | [Home](#) [About](#) [Movies](#) [Reviews](#) [Privacy](#)



Delete Movie

Comments

www.ijerpi.org

Create Review

Delete Review

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

RETROSPECTIVE REVIEWS Home About Movies Reviews Privacy

Home / Movies / 1 / CreateReview

Create Review

Name

Comment

Rating

[Create](#) [Cancel](#)

© 2021 - Retrospective Reviews - Privacy

© 2021 - Retrospective Reviews - Privacy

Home / Movies / 1 / DeleteReview

Delete Review

Are you sure you want to delete this review?

Movie	Patsy
Name	06/05/2021 18:05:45
CreatedOn	Loved it. One of the best movies of all time. Do yourself a favour if you haven't seen it and watch this!
Comment	
Rating	9

[Confirm](#) [Cancel](#)