IT2030 Assignment 4 Matt Brown

**HomeController.cs**

namespace MovieList2.Controllers

{

public class HomeController : Controller

{

private MovieContext context { get; set; }

public HomeController(MovieContext ctx)

{

context = ctx;

}

public IActionResult Index()

{

var movies = context.Movies.Include(m => m.Genre).OrderBy(m => m.Name).ToList();

return View(movies);

}

}

}

**MovieController.cs**

namespace MovieList2.Controllers

{

public class MovieController : Controller

{

private MovieContext context { get; set; }

public MovieController(MovieContext ctx)

{

context = ctx;

}

[HttpGet]

public IActionResult Add()

{

ViewBag.Action = "Add";

ViewBag.Genres = context.Genres.OrderBy(g => g.Name).ToList();

return View("Edit", new Movie());

}

[HttpGet]

public IActionResult Edit(int id)

{

ViewBag.Action = "Edit";

ViewBag.Genres = context.Genres.OrderBy(g => g.Name).ToList();

var movie = context.Movies.Find(id);

return View(movie);

}

[HttpPost]

public IActionResult Edit(Movie movie)

{

if (ModelState.IsValid)

{

if (movie.MovieId == 0)

context.Movies.Add(movie);

else

context.Movies.Update(movie);

context.SaveChanges();

return RedirectToAction("Index", "Home");

} else

{

ViewBag.Action = (movie.MovieId == 0) ? "Add" : "Edit";

ViewBag.Genres = context.Genres.OrderBy(g => g.Name).ToList();

return View(movie);

}

}

[HttpGet]

public IActionResult Delete(int id)

{

var movie = context.Movies.Find(id);

return View(movie);

}

[HttpPost]

public IActionResult Delete(Movie movie)

{

context.Movies.Remove(movie);

context.SaveChanges();

return RedirectToAction("Index", "Home");

}

}

}

**Genre.cs**

namespace MovieList2.Models

{

public class Genre

{

public string GenreId { get; set; }

public string Name { get; set; }

}

}

**Movie.cs**

namespace MovieList2.Models

{

public class Movie

{

// EF Core will configure the database to generate this value

public int MovieId { get; set; }

[Required(ErrorMessage = "Please enter a name.")]

public string Name { get; set; }

[Required(ErrorMessage = "Please enter a year.")]

[Range(1889, 2999, ErrorMessage = "Year must be after 1889.")]

public int? Year { get; set; }

[Required(ErrorMessage = "Please enter a rating.")]

[Range(1, 5, ErrorMessage = "Rating must be between 1 and 5.")]

public int? Rating { get; set; }

[Required(ErrorMessage = "Please Enter a genre.")]

public string GenreId { get; set; }

public Genre Genre { get; set; }

public string Slug => Name?.Replace(' ', '-').ToLower() + '-' + Year?.ToString();

}

}

**MovieContext.cs**

namespace MovieList2.Models

{

public class MovieContext : DbContext

{

public MovieContext(DbContextOptions<MovieContext> options) : base(options)

{ }

public DbSet<Movie> Movies { get; set; }

public DbSet<Genre> Genres { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

base.OnModelCreating(modelBuilder);

modelBuilder.Entity<Genre>().HasData(

new Genre { GenreId = "A", Name = "Action"},

new Genre { GenreId = "C", Name = "Comedy" },

new Genre { GenreId = "D", Name = "Drama" },

new Genre { GenreId = "H", Name = "Horror" },

new Genre { GenreId = "M", Name = "Musical" },

new Genre { GenreId = "R", Name = "RomCom" },

new Genre { GenreId = "S", Name = "SciFi" }

);

modelBuilder.Entity<Movie>().HasData(

new Movie

{

MovieId = 1,

Name = "Casablanca",

Year = 1942,

Rating = 5,

GenreId = "D"

},

new Movie

{

MovieId = 2,

Name = "Wonder Woman",

Year = 2017,

Rating = 3,

GenreId = "A"

},

new Movie

{

MovieId = 3,

Name = "Moonstruck",

Year = 1988,

Rating = 4,

GenreId = "R"

}

);

}

}

}

**Index.cshtml**

@model List<Movie>

@{

ViewBag.Title = "My Movies";

}

<h2>Movie List</h2>

<a asp-controller="Movie" asp-action="Add">Add New Movie</a>

<table class="table table-bordered table-striped">

<thead>

<tr>

<th>Name</th>

<th>Year</th>

<th>Genre</th>

<th>Rating</th>

<th></th>

</tr>

</thead>

<tbody>

@foreach (var movie in Model)

{

<tr>

<td>@movie.Name</td>

<td>@movie.Year</td>

<td>@movie.Genre.Name</td>

<td>@movie.Rating</td>

<td>

<a asp-controller="Movie" asp-action="Edit"

asp-route-id="@movie.MovieId"

asp-route-slug="@movie.Slug">Edit</a>

<a asp-controller="Movie" asp-action="Delete"

asp-route-id="@movie.MovieId"

asp-route-slug="@movie.Slug">Delete</a>

</td>

</tr>

}

</tbody>

</table>

**Delete.cshtml**

@model Movie

@{

ViewBag.Title = "Delete Movie";

}

<h2>Confirm Deletion</h2>

<h3>@Model.Name (@Model.Year) </h3>

<form asp-action="Delete" method="post">

<input type="hidden" asp-for="MovieId" />

<button type="submit" class="btn btn-primary">Delete</button>

<a asp-controller="Home" asp-action="Index"

class="btn btn-primary">Cancel</a>

</form>

**Edit.cshtml**

@model Movie

@{

string title = ViewBag.Action + " Movie";

ViewBag.Title = title;

}

<h2>@ViewBag.Title</h2>

<form asp-action="Edit" method="post">

<div asp-validation-summary="All" class="text-danger"></div>

<div class="form-group">

<label asp-for="Name">Name</label>

<input asp-for="Name" class="form-control">

</div>

<div class="form-group">

<label asp-for="Year">Year</label>

<input asp-for="Year" class="form-control">

</div>

<div class="form-group">

<label asp-for="GenreId">Genre</label>

<select asp-for="GenreId" class="form-control">

<option value="">Select a genre</option>

@foreach (Genre g in ViewBag.Genres)

{

<option value="@g.GenreId">@g.Name</option>

}

</select>

</div>

<div class="form-group">

<label asp-for="Rating">Rating</label>

<input asp-for="Rating" class="form-control">

</div>

<input type="hidden" asp-for="MovieId" />

<button type="submit" class="btn btn-primary">@ViewBag.Action</button>

<a asp-controller="Home" asp-action="Index"

class="btn btn-primary">Cancel</a>

</form>

**\_Layout.cshtml**

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>@ViewBag.Title</title>

<link rel="stylesheet" type="text/css"

href="~/lib/bootstrap/css/bootstrap.min.css">

</head>

<body>

<div class="container">

<header class="jumbotron">

<h1>My Movies</h1>

</header>

@RenderBody()

</div>

</body>

</html>

**Appsettings.json**

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

"MovieContext": "Server=(localdb)\\mssqllocaldb;Database=MoviesAssignment;Trusted\_Connection=True;MultipleActiveResultSets=true"

}

}

**Startup.cs**

namespace MovieList2

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

services.AddControllersWithViews();

services.AddRouting(options =>

{

options.LowercaseUrls = true;

options.AppendTrailingSlash = true;

});

services.AddDbContext<MovieContext>(options =>

options.UseSqlServer(

Configuration.GetConnectionString("MovieContext")));

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app)

{

app.UseDeveloperExceptionPage();

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}/{slug?}");

});

}

}

}