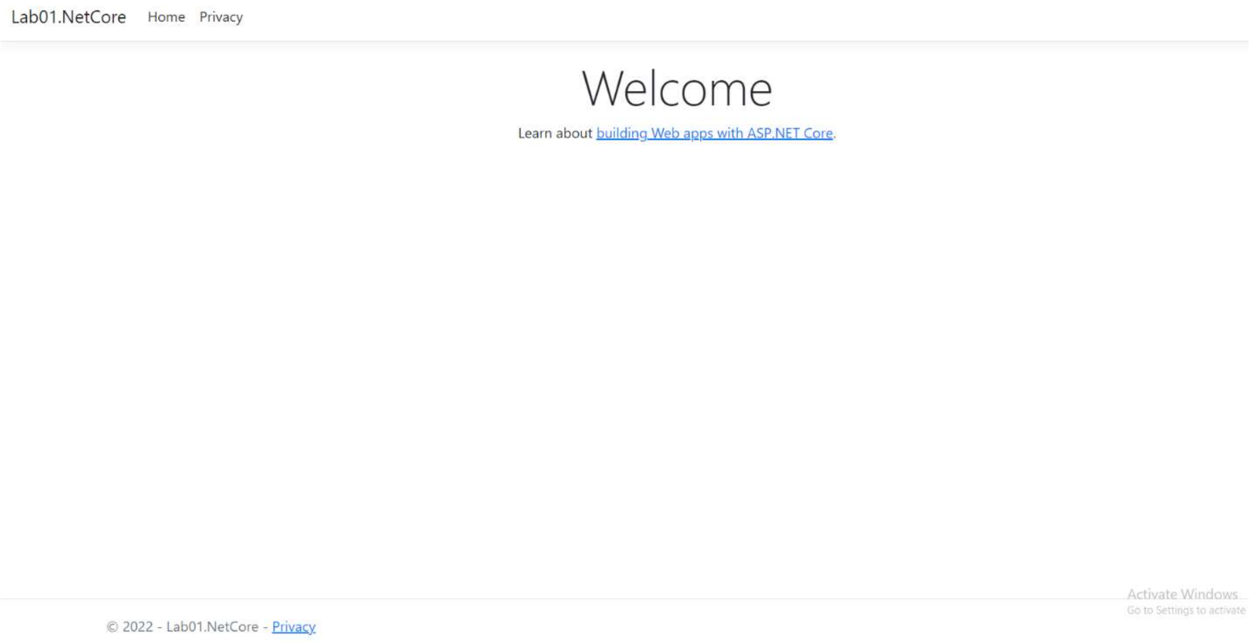
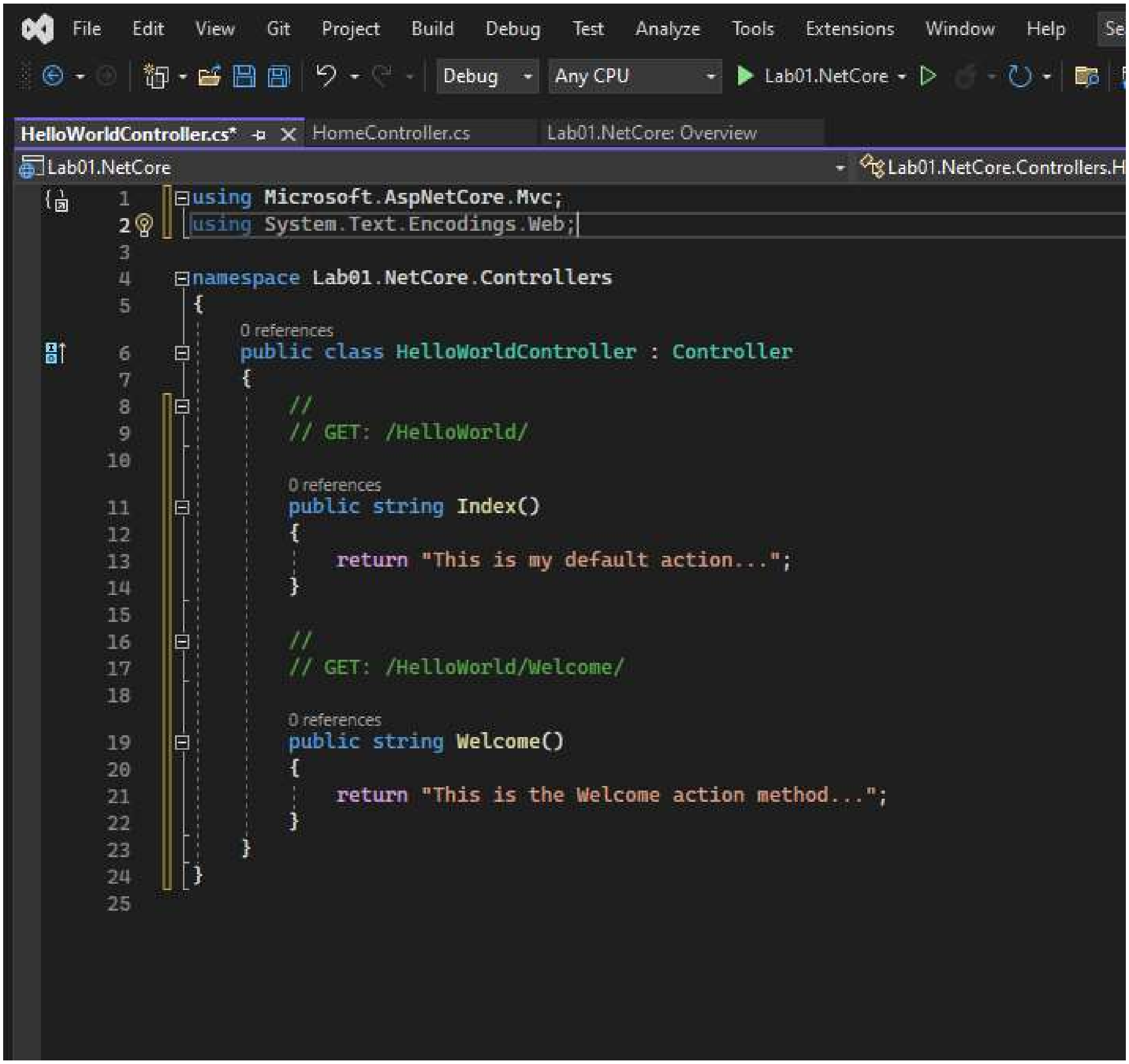
Câu 1



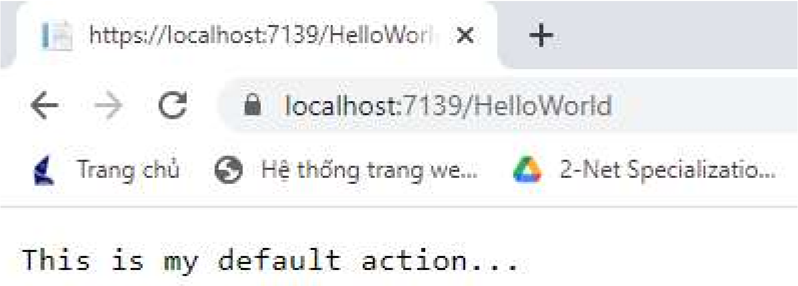
Welcome

Learn about building Web apps with ASP.NET Core.

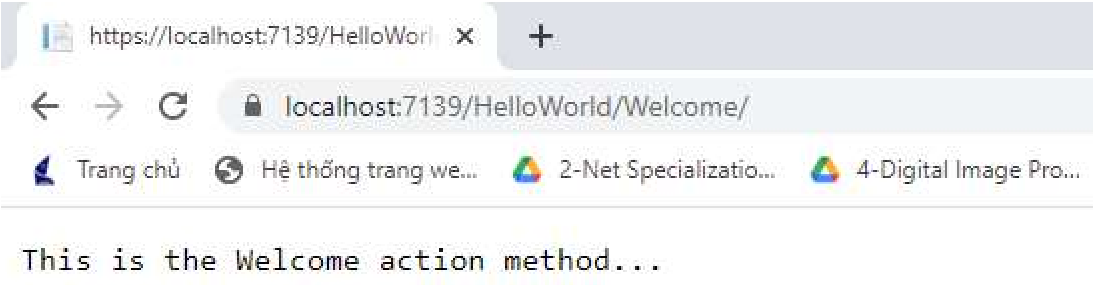
© 2022 - Lab01.NetCore - Privacy



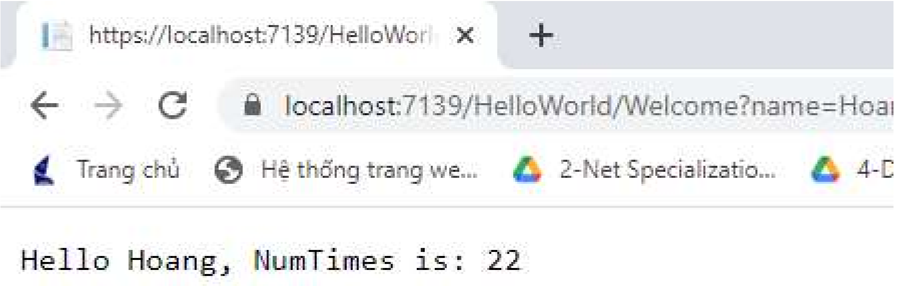
```
using Microsoft.AspNetCore.Mvc;
using System.Text.Encodings.Web;

namespace Lab01.NetCore.Controllers
{
    public class HelloWorldController : Controller
    {
        //
        // GET: /HelloWorld/

        public string Index()
        {
            return "This is my default action...";
        }

        //
        // GET: /HelloWorld/Welcome/

        public string Welcome()
        {
            return "This is the Welcome action method...";
        }
    }
}
```



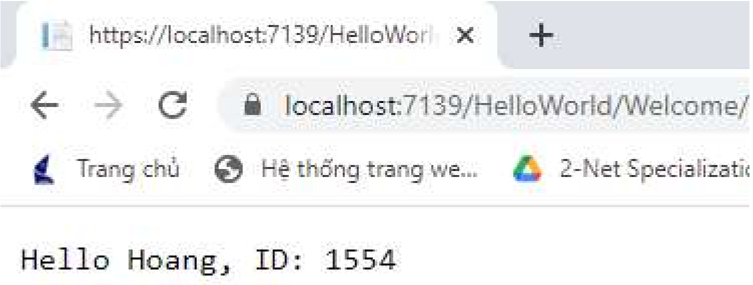
```
This is my default action...
```



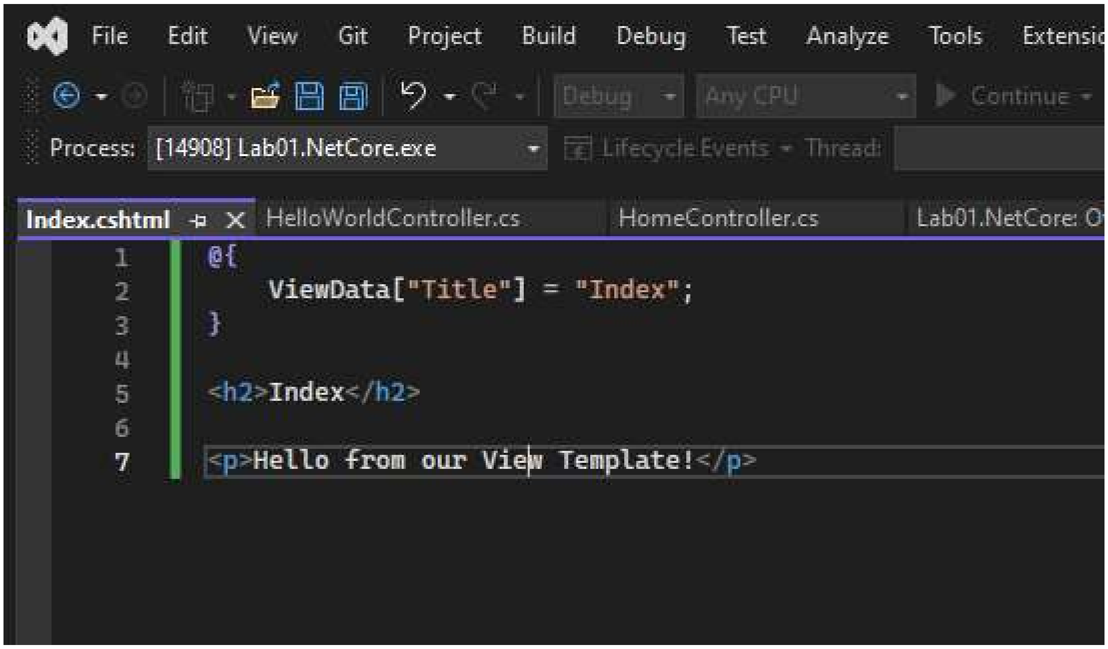
```
This is the Welcome action method...
```


https://localhost:7139/HelloWorl
localhost:7139/HelloWorld/Welcome?name=Hoa
Trang chủ
Hệ thống trang we...
2-Net Specializatio...
4-D

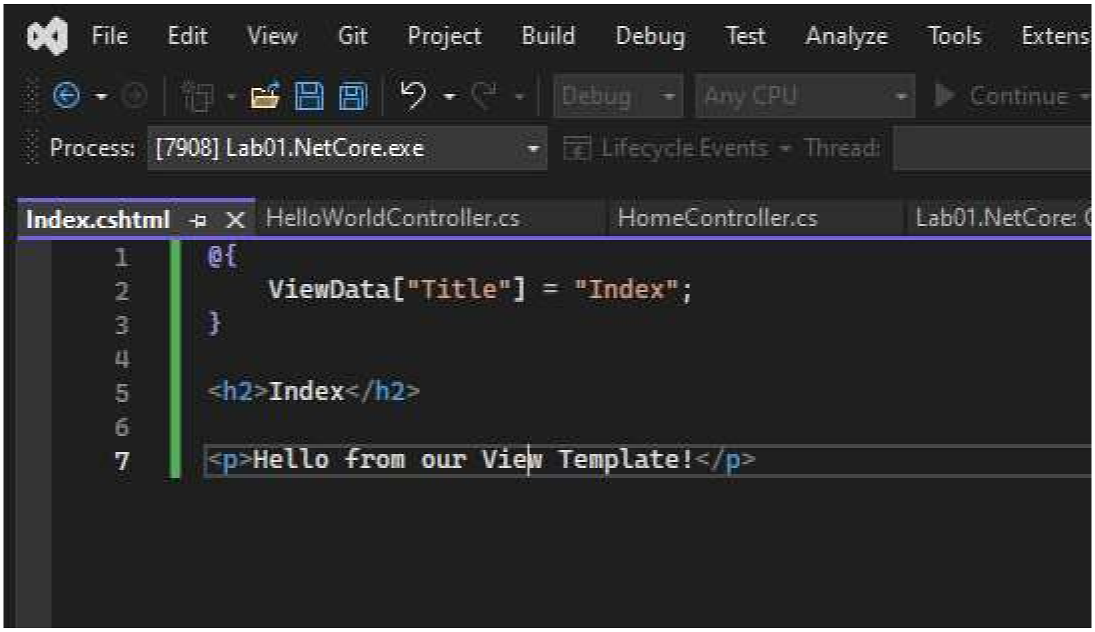
```
Hello Hoang, NumTimes is: 22
```


https://localhost:7139/HelloWorl
localhost:7139/HelloWorld/Welcome/
Trang chủ
Hệ thống trang we...
2-Net Specializati

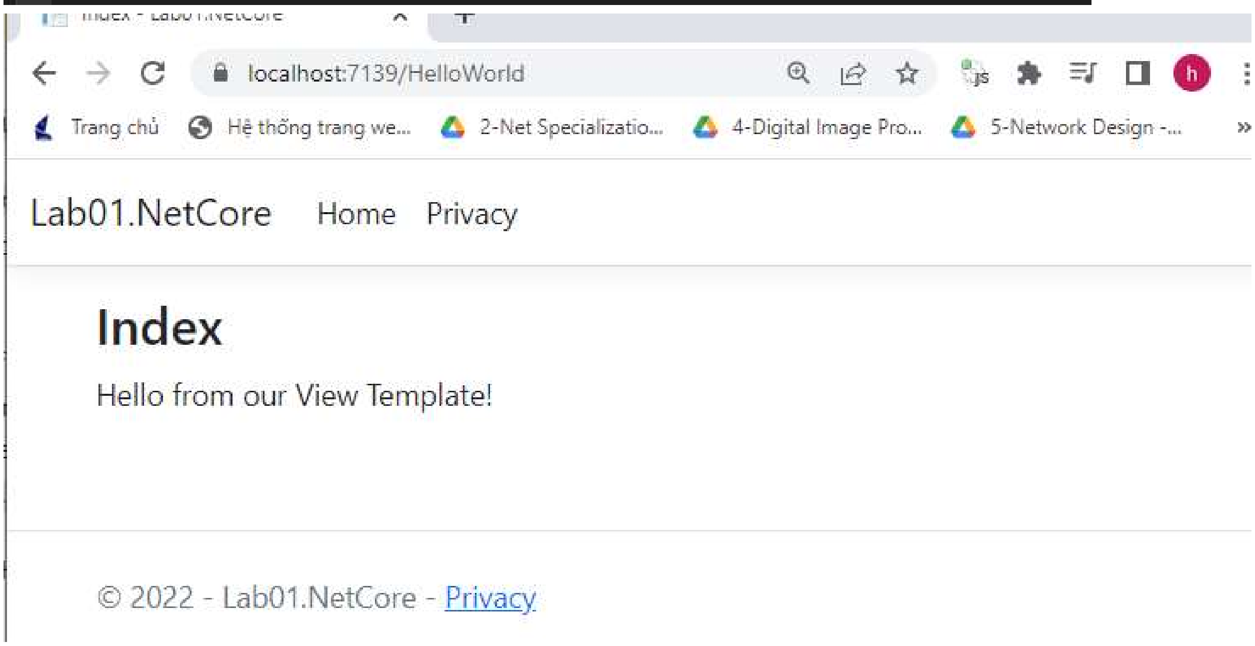
```
Hello Hoang, ID: 1554
```



```
@{
    ViewData["Title"] = "Index";
}

<h2>Index</h2>

<p>Hello from our View Template!</p>
```



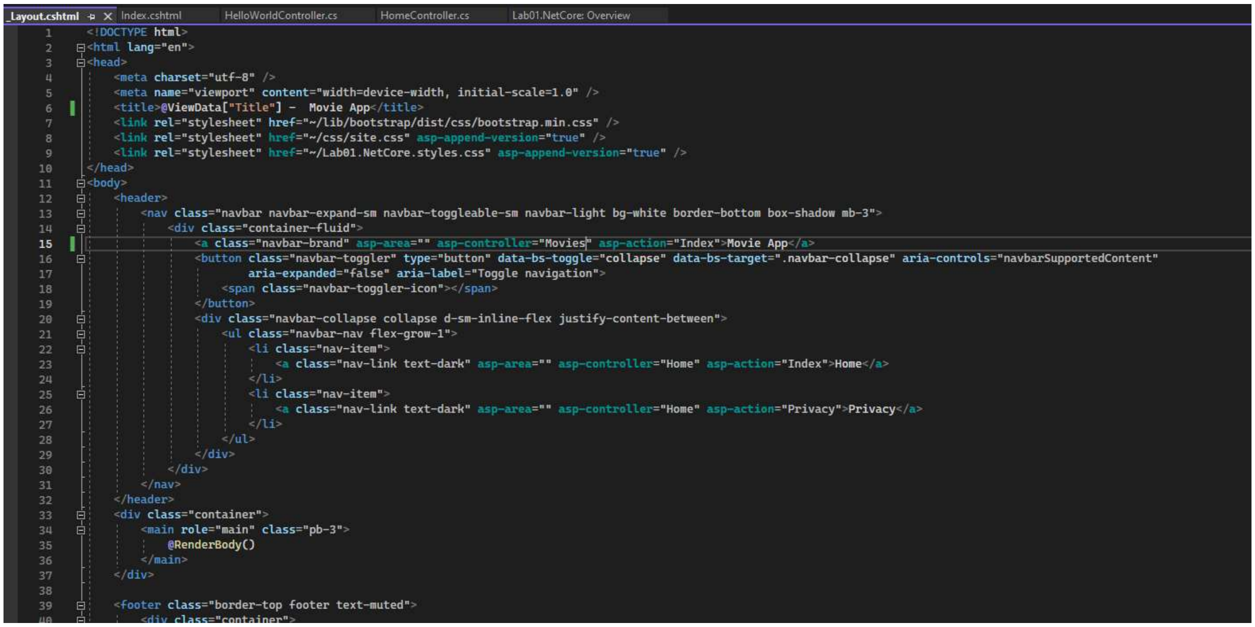
```
@{
    ViewData["Title"] = "Index";
}

<h2>Index</h2>

<p>Hello from our View Template!</p>
```



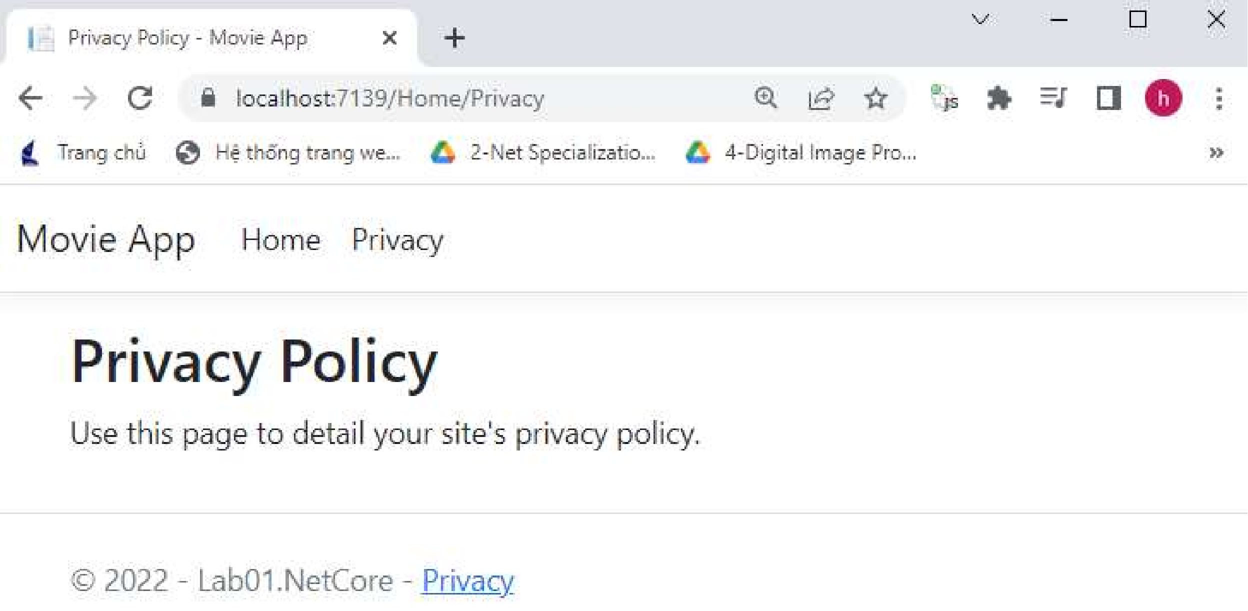
Index

Hello from our View Template!

© 2022 - Lab01.NetCore - Privacy



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] -  Movie App</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
    <link rel="stylesheet" href="~/Lab01.NetCore.styles.css" asp-append-version="true" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
            <div class="container-fluid">
                <a class="navbar-brand" asp-area="" asp-controller="Movies" asp-action="Index">Movie App</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"
                        aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>
    </header>
    <div class="container">
        <main role="main" class="pb-3">
            @RenderBody()
        </main>
    </div>

    <footer class="border-top footer text-muted">
        <div class="container">
```

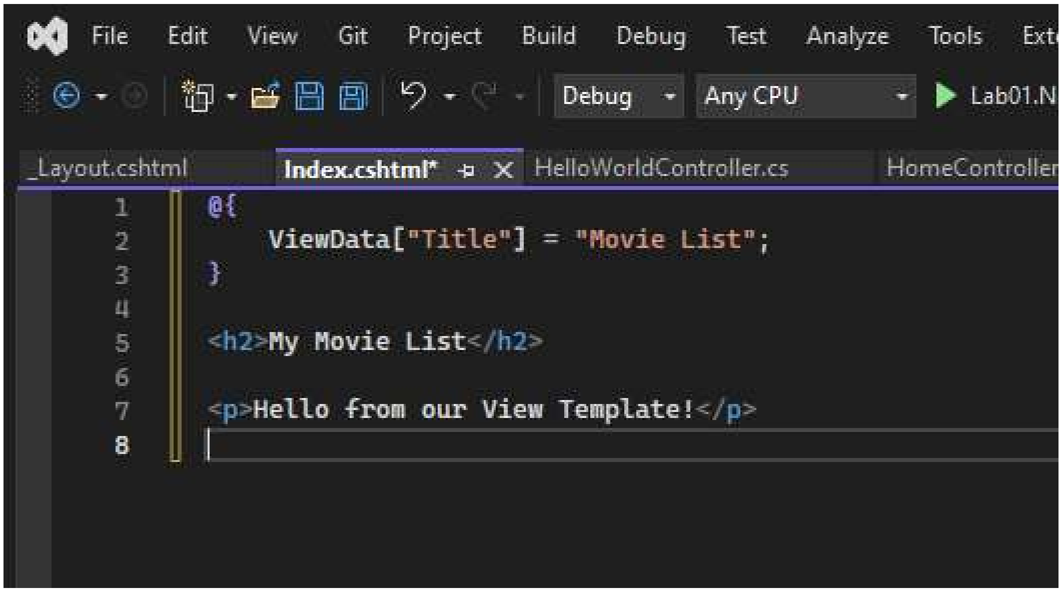


Movie App Home Privacy

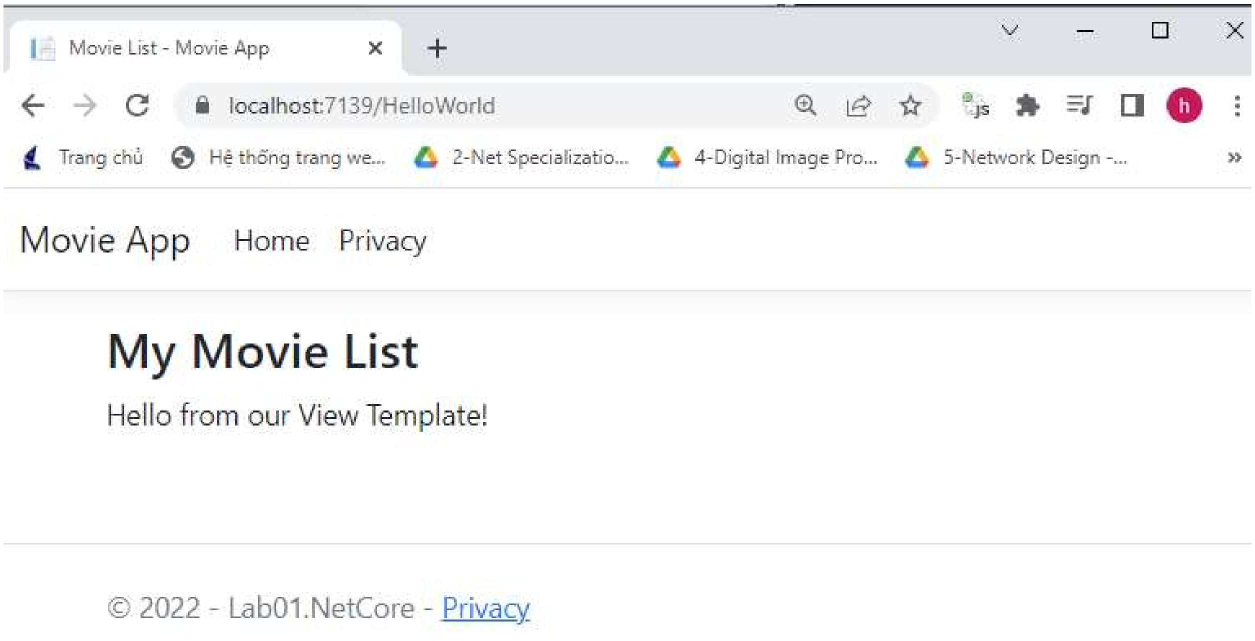
Privacy Policy

Use this page to detail your site's privacy policy.

© 2022 - Lab01.NetCore - Privacy



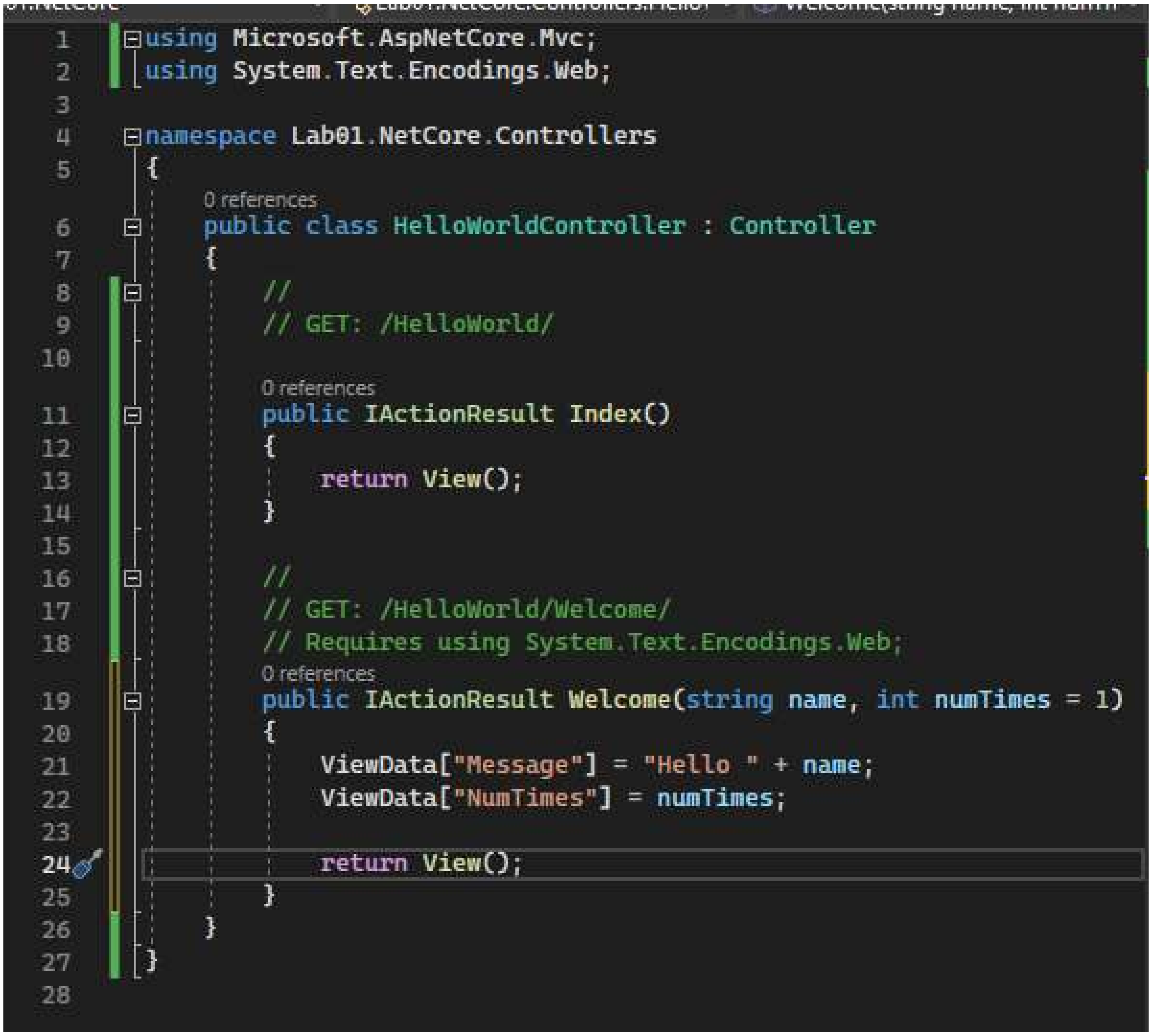
```
@{
    ViewData["Title"] = "Movie List";
}

<h2>My Movie List</h2>

<p>Hello from our View Template!</p>

```



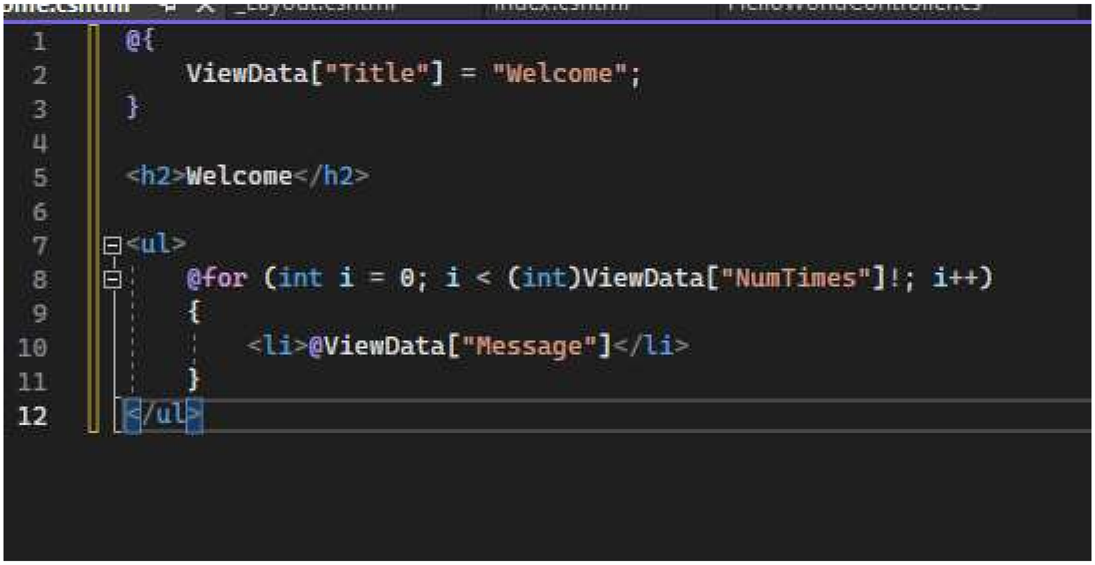
My Movie List

Hello from our View Template!

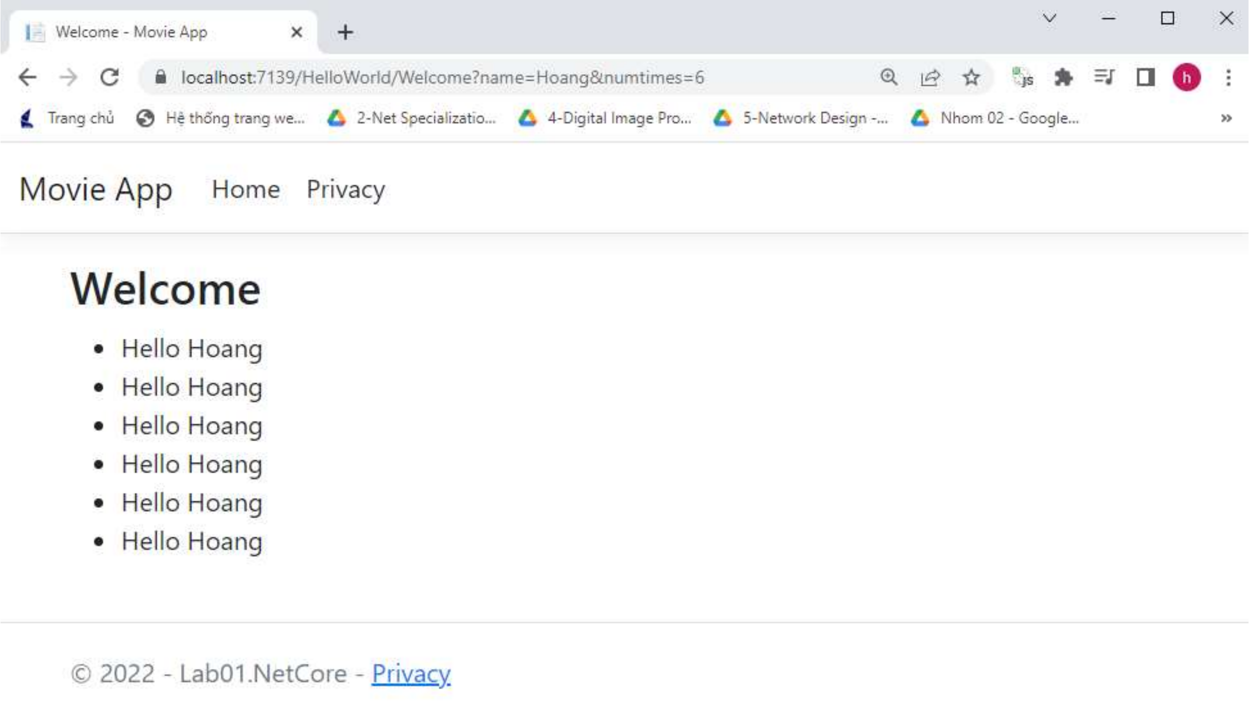
© 2022 - Lab01.NetCore - Privacy



```
using Microsoft.AspNetCore.Mvc;
using System.Text.Encodings.Web;

namespace Lab01.NetCore.Controllers
{
    public class HelloWorldController : Controller
    {
        //
        // GET: /HelloWorld/

        public IActionResult Index()
        {
            return View();
        }

        //
        // GET: /HelloWorld/Welcome/
        // Requires using System.Text.Encodings.Web;
        public IActionResult Welcome(string name, int numTimes = 1)
        {
            ViewData["Message"] = "Hello " + name;
            ViewData["NumTimes"] = numTimes;

            return View();
        }
    }
}
```



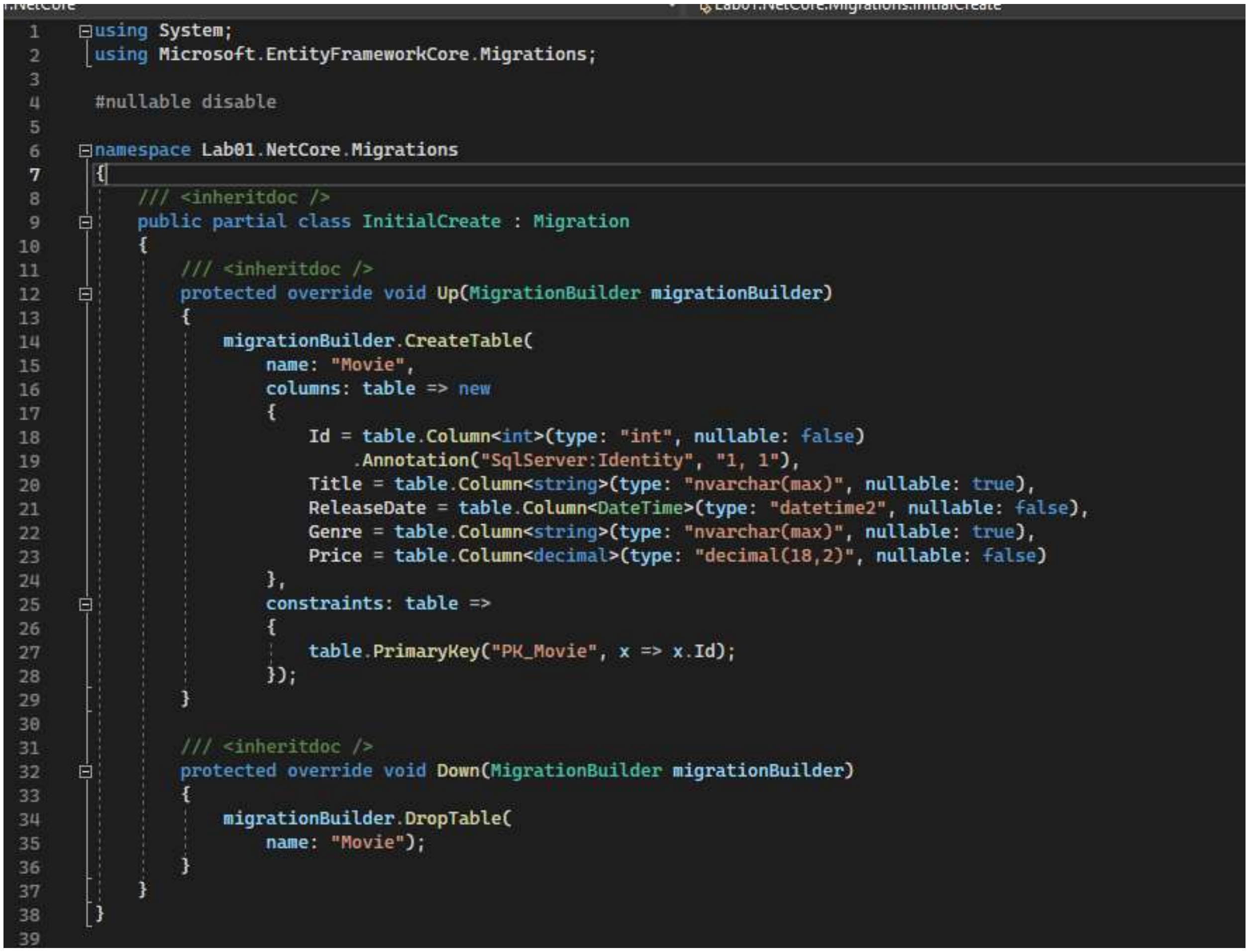
```
@{
    ViewData["Title"] = "Welcome";
}

<h2>Welcome</h2>

<ul>
    @for (int i = 0; i < (int)ViewData["NumTimes"]!; i++)
    {
        <li>@ViewData["Message"]</li>
    }
</ul>
```



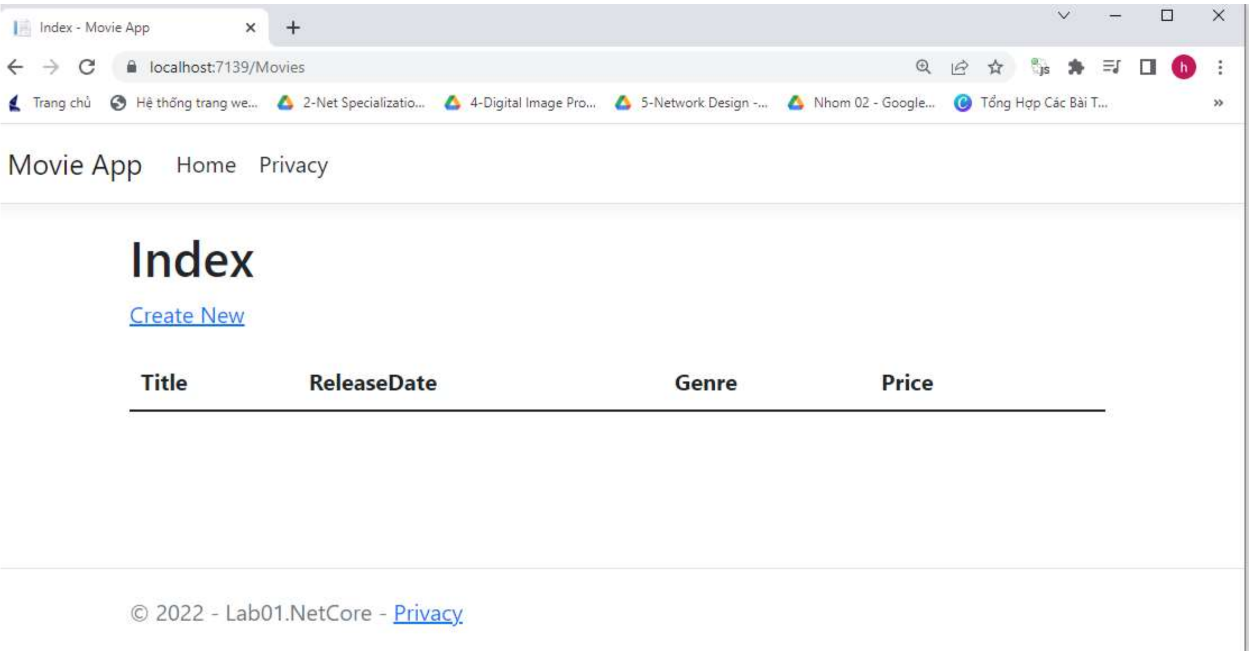
Welcome

- Hello Hoang
- Hello Hoang
- Hello Hoang
- Hello Hoang
- Hello Hoang
- Hello Hoang

© 2022 - Lab01.NetCore - Privacy



```
using System;
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace Lab01.NetCore.Migrations
{
    /// <inheritdoc />
    public partial class InitialCreate : Migration
    {
        /// <inheritdoc />
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "Movie",
                columns: table => new
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    Title = table.Column<string>(type: "nvarchar(max)", nullable: true),
                    ReleaseDate = table.Column<DateTime>(type: "datetime2", nullable: false),
                    Genre = table.Column<string>(type: "nvarchar(max)", nullable: true),
                    Price = table.Column<decimal>(type: "decimal(18,2)", nullable: false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Movie", x => x.Id);
                });
        }

        /// <inheritdoc />
        protected override void Down(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.DropTable(
                name: "Movie");
        }
    }
}
```

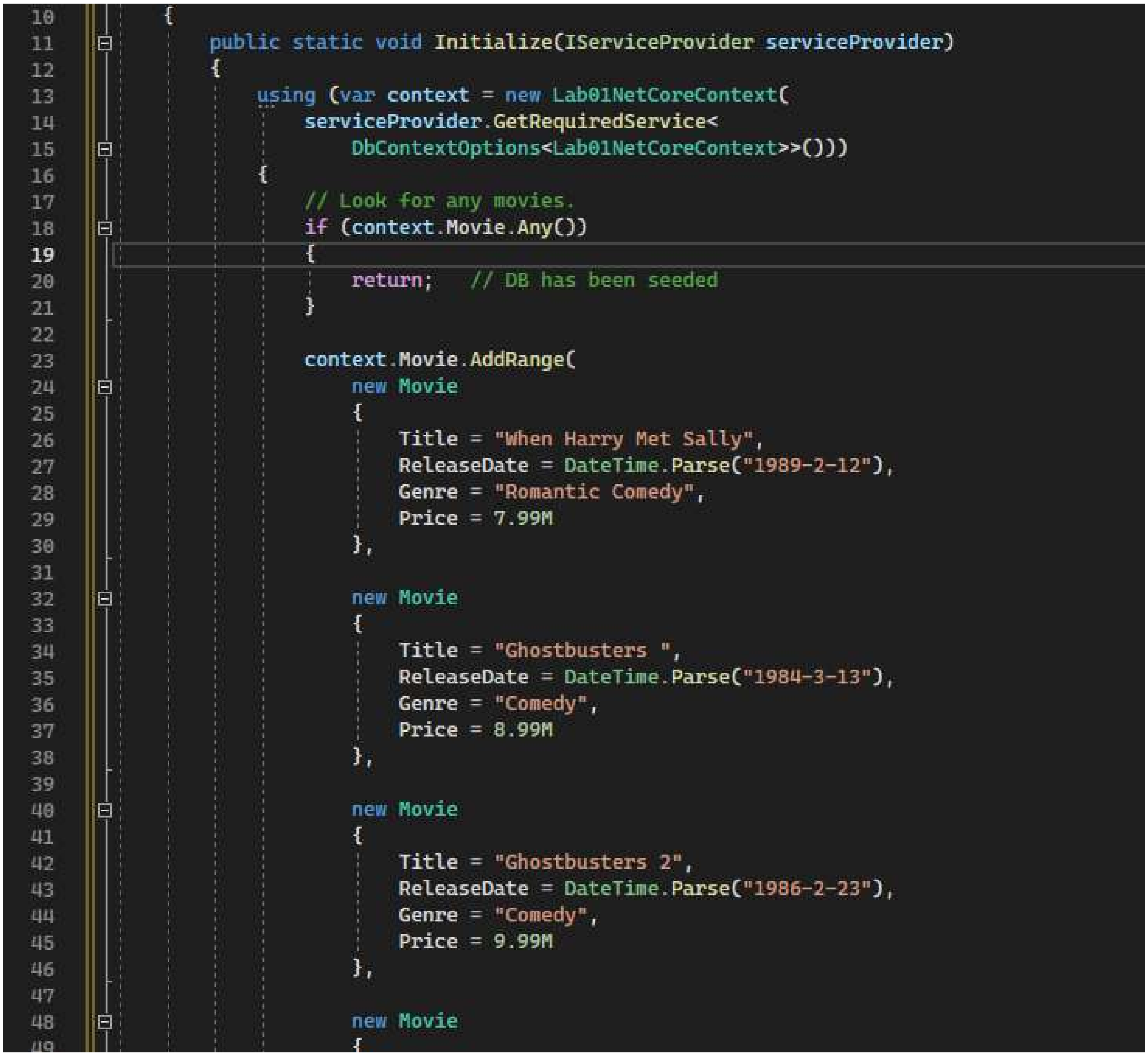


Index

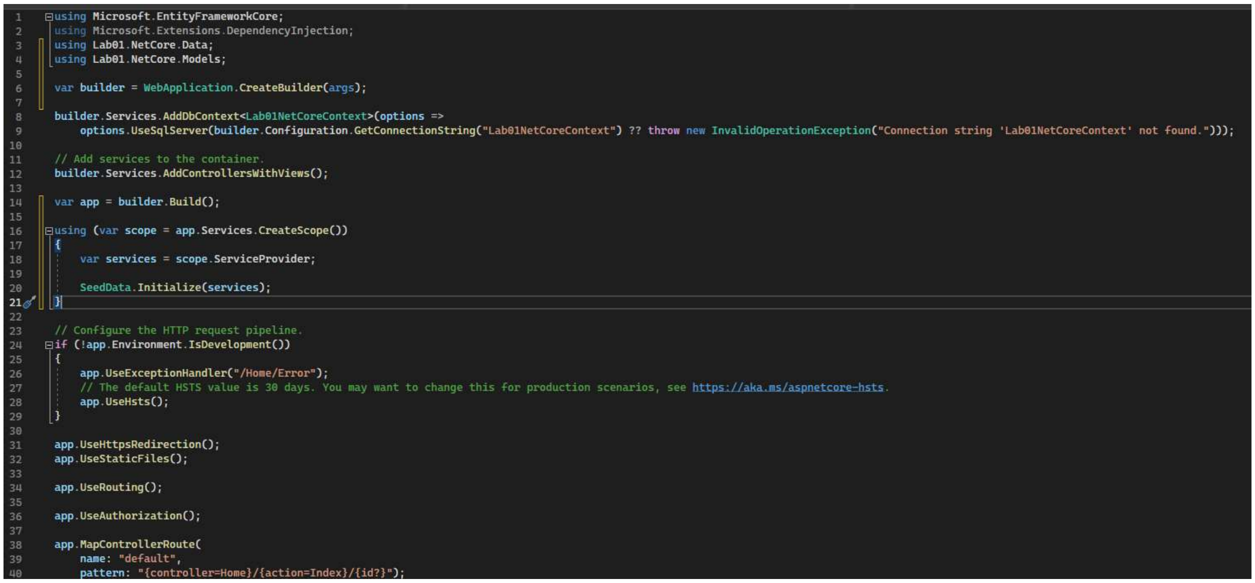
Create New

Title	ReleaseDate	Genre	Price

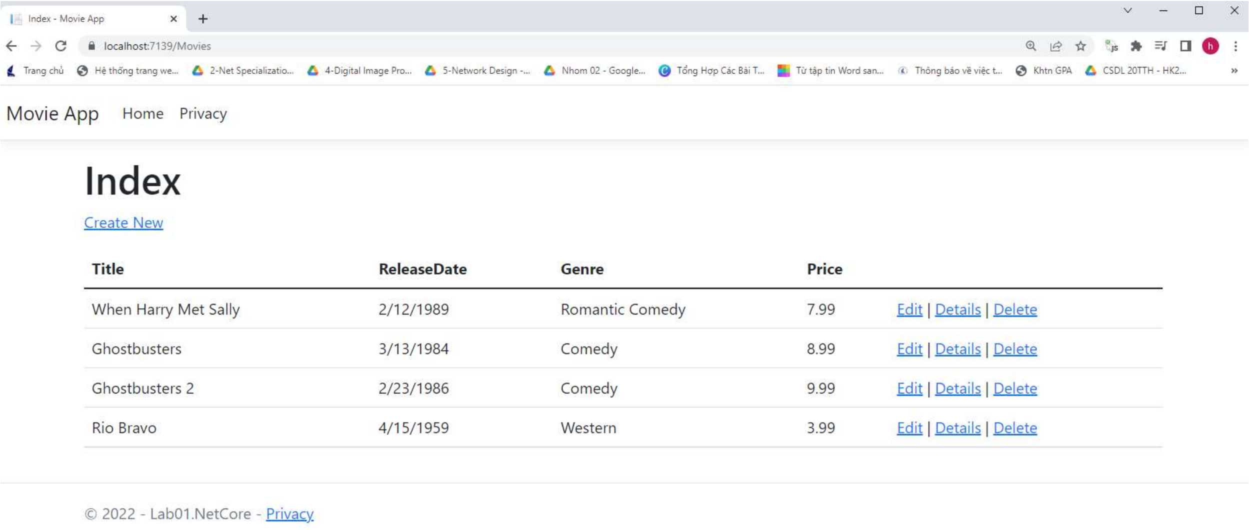
© 2022 - Lab01.NetCore - Privacy



```csharp
    {
        public static void Initialize(IServiceProvider serviceProvider)
        {
            using (var context = new Lab01NetCoreContext(
                serviceProvider.GetRequiredService<
                    DbContextOptions<Lab01NetCoreContext>>()))
            {
                // Look for any movies.
                if (context.Movie.Any())
                {
                    return;   // DB has been seeded
                }

                context.Movie.AddRange(
                    new Movie
                    {
                        Title = "When Harry Met Sally",
                        ReleaseDate = DateTime.Parse("1989-2-12"),
                        Genre = "Romantic Comedy",
                        Price = 7.99M
                    },

                    new Movie
                    {
                        Title = "Ghostbusters ",
                        ReleaseDate = DateTime.Parse("1984-3-13"),
                        Genre = "Comedy",
                        Price = 8.99M
                    },

                    new Movie
                    {
                        Title = "Ghostbusters 2",
                        ReleaseDate = DateTime.Parse("1986-2-23"),
                        Genre = "Comedy",
                        Price = 9.99M
                    },

                    new Movie
                    {
```



```csharp
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.DependencyInjection;
using Lab01.NetCore.Data;
using Lab01.NetCore.Models;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddDbContext<Lab01NetCoreContext>(options =>
    options.UseSqlServer(builder.Configuration.GetConnectionString("Lab01NetCoreContext") ?? throw new InvalidOperationException("Connection string 'Lab01NetCoreContext' not found.")));

// Add services to the container.
builder.Services.AddControllersWithViews();

var app = builder.Build();

using (var scope = app.Services.CreateScope())
{
    var services = scope.ServiceProvider;

    SeedData.Initialize(services);
}

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

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

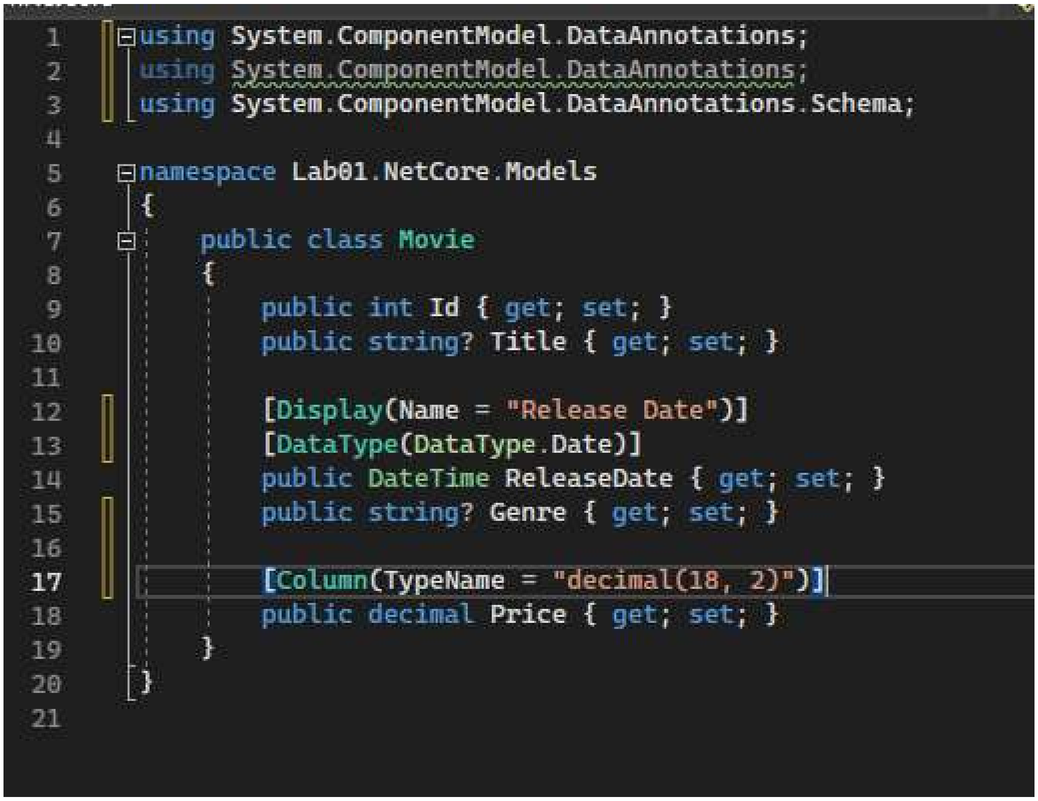


Index

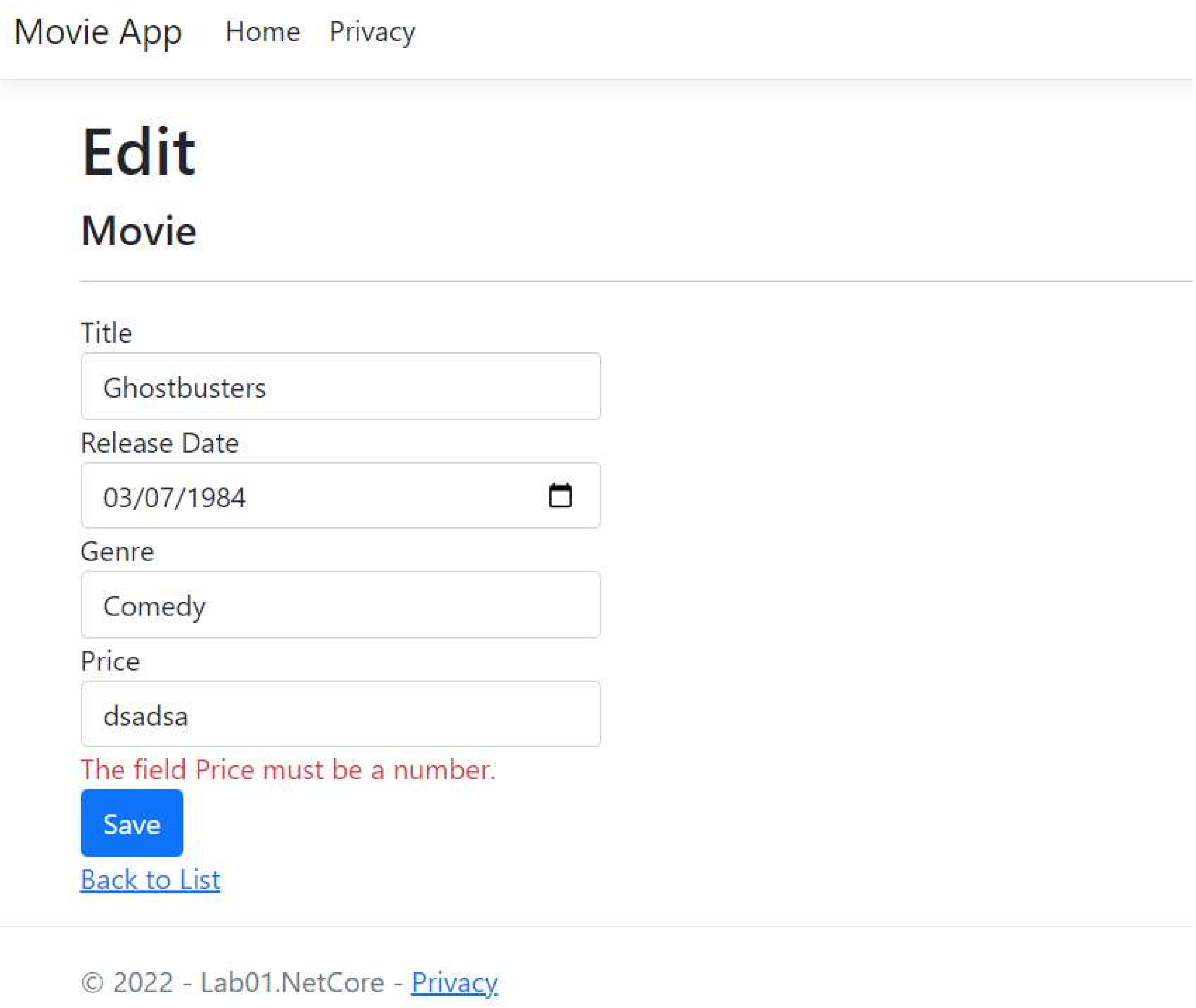
Create New

Title	ReleaseDate	Genre	Price	
When Harry Met Sally	2/12/1989	Romantic Comedy	7.99	Edit | Details | Delete
Ghostbusters	3/13/1984	Comedy	8.99	Edit | Details | Delete
Ghostbusters 2	2/23/1986	Comedy	9.99	Edit | Details | Delete
Rio Bravo	4/15/1959	Western	3.99	Edit | Details | Delete

© 2022 - Lab01.NetCore - Privacy



```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace Lab01.NetCore.Models
{
    public class Movie
    {
        public int Id { get; set; }
        public string? Title { get; set; }

        [Display(Name = "Release Date")]
        [DataType(DataType.Date)]
        public DateTime ReleaseDate { get; set; }
        public string? Genre { get; set; }

        [Column(TypeName = "decimal(18, 2)")]
        public decimal Price { get; set; }
    }
}
```



Edit

Movie

Title

Ghostbusters

Release Date

03/07/1984

Genre

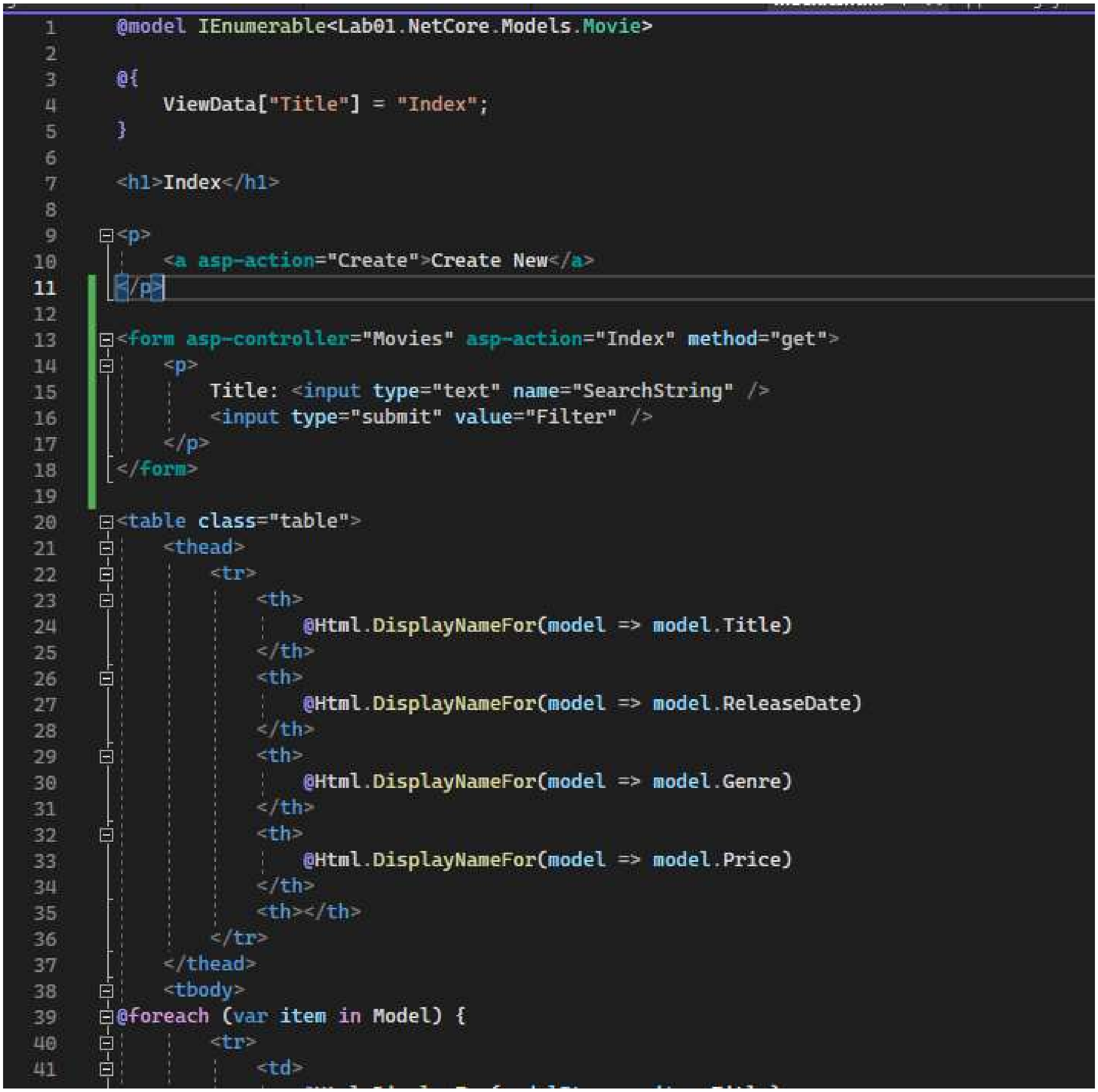
Comedy

Price

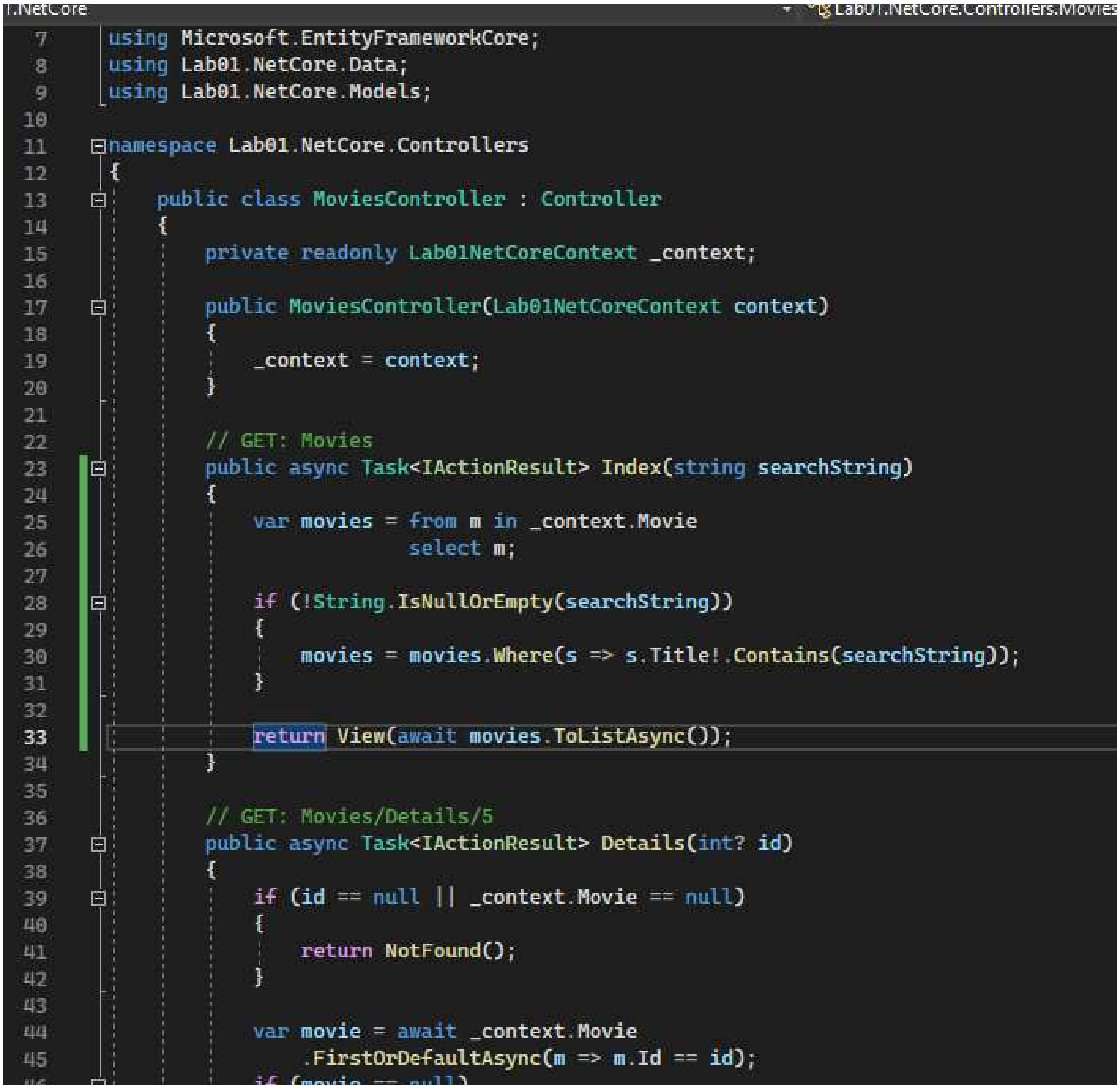
dsadsa

The field Price must be a number.

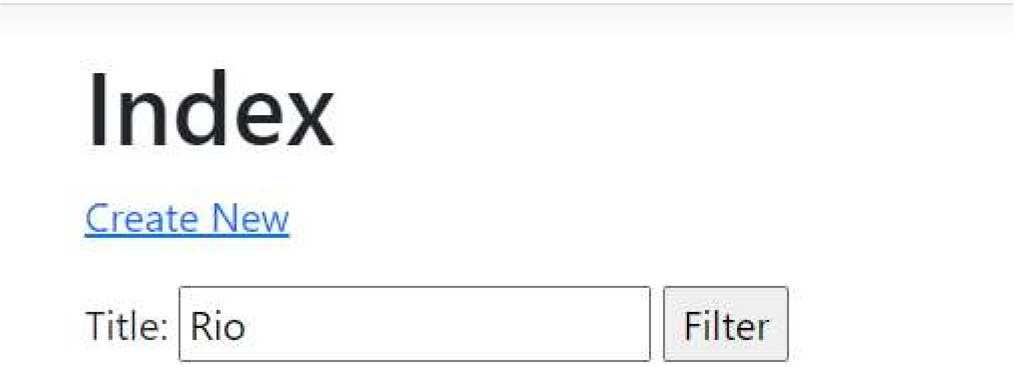
© 2022 - Lab01.NetCore - Privacy



```
@model IEnumerable<Lab01.NetCore.Models.Movie>

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>

<form asp-controller="Movies" asp-action="Index" method="get">
    <p>
        Title: <input type="text" name="SearchString" />
        <input type="submit" value="Filter" />
    </p>
</form>

<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Title)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.ReleaseDate)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Genre)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Price)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
@foreach (var item in Model) {
        <tr>
            <td>
```



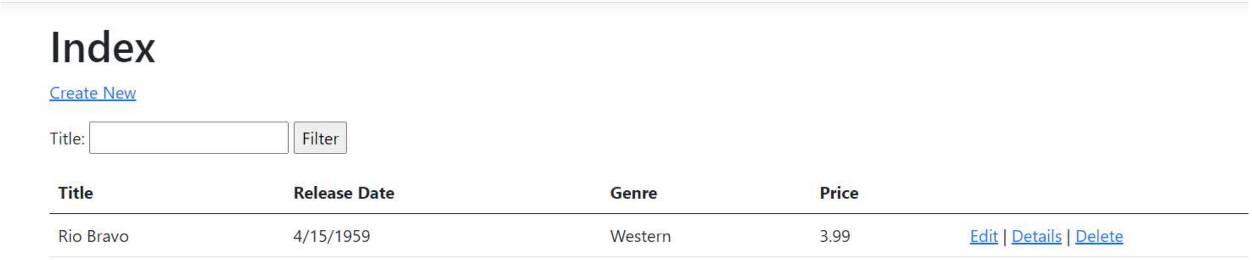
```
using Microsoft.EntityFrameworkCore;
using Lab01.NetCore.Data;
using Lab01.NetCore.Models;

namespace Lab01.NetCore.Controllers
{
    public class MoviesController : Controller
    {
        private readonly Lab01NetCoreContext _context;

        public MoviesController(Lab01NetCoreContext context)
        {
            _context = context;
        }

        // GET: Movies
        public async Task<IActionResult> Index(string searchString)
        {
            var movies = from m in _context.Movie
                         select m;

            if (!String.IsNullOrEmpty(searchString))
            {
                movies = movies.Where(s => s.Title!.Contains(searchString));
            }

            return View(await movies.ToListAsync());
        }

        // GET: Movies/Details/5
        public async Task<IActionResult> Details(int? id)
        {
            if (id == null || _context.Movie == null)
            {
                return NotFound();
            }

            var movie = await _context.Movie
                .FirstOrDefaultAsync(m => m.Id == id);
            if (movie == null)
```



Index

Create New

Title: Rio Filter

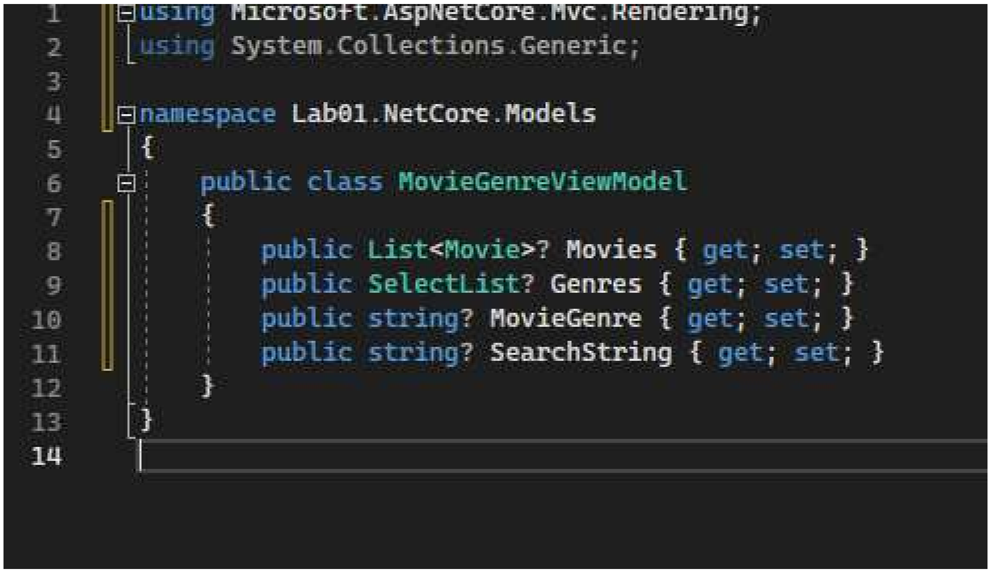


Index

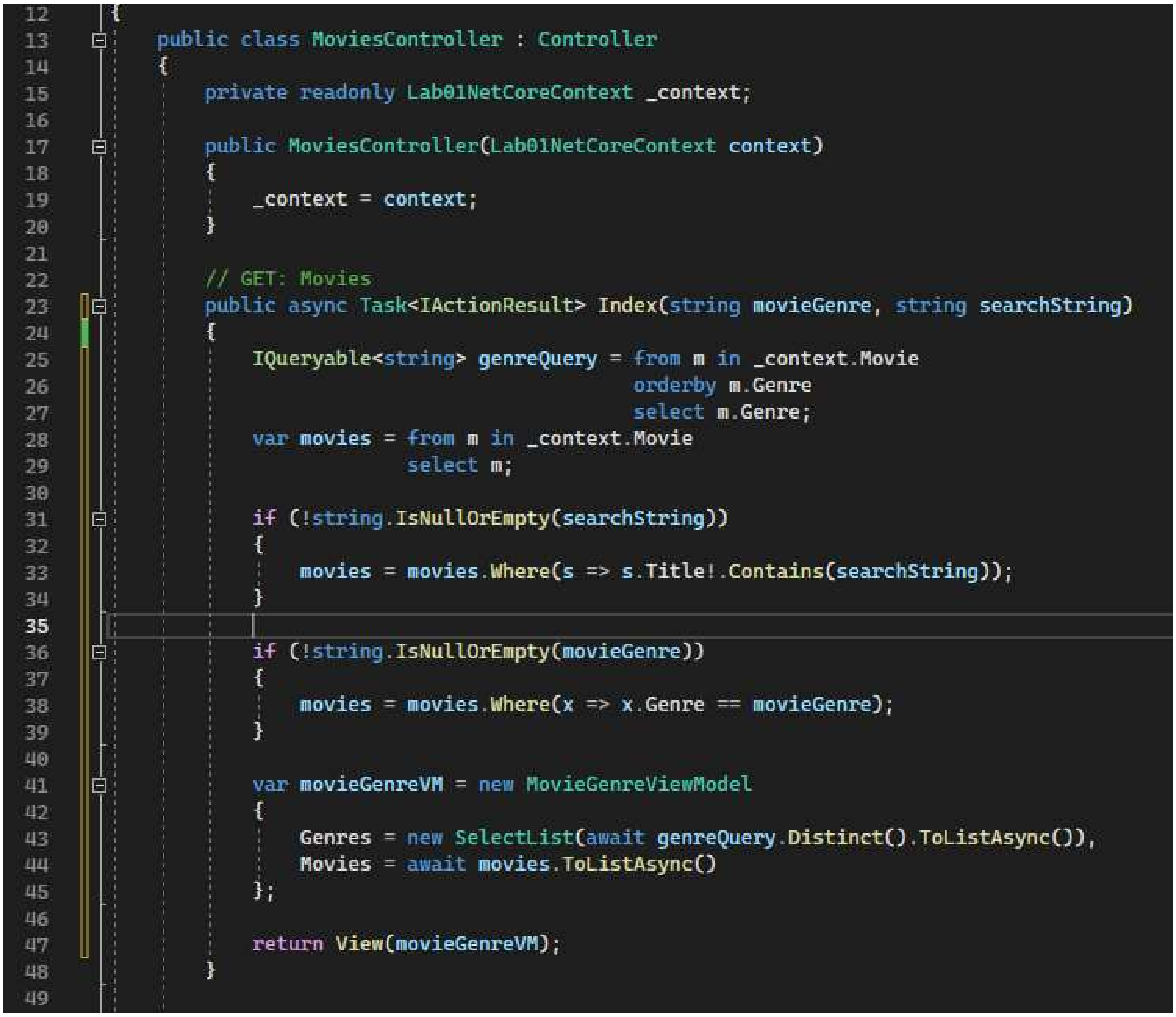
Create New

Title: [ ] Filter

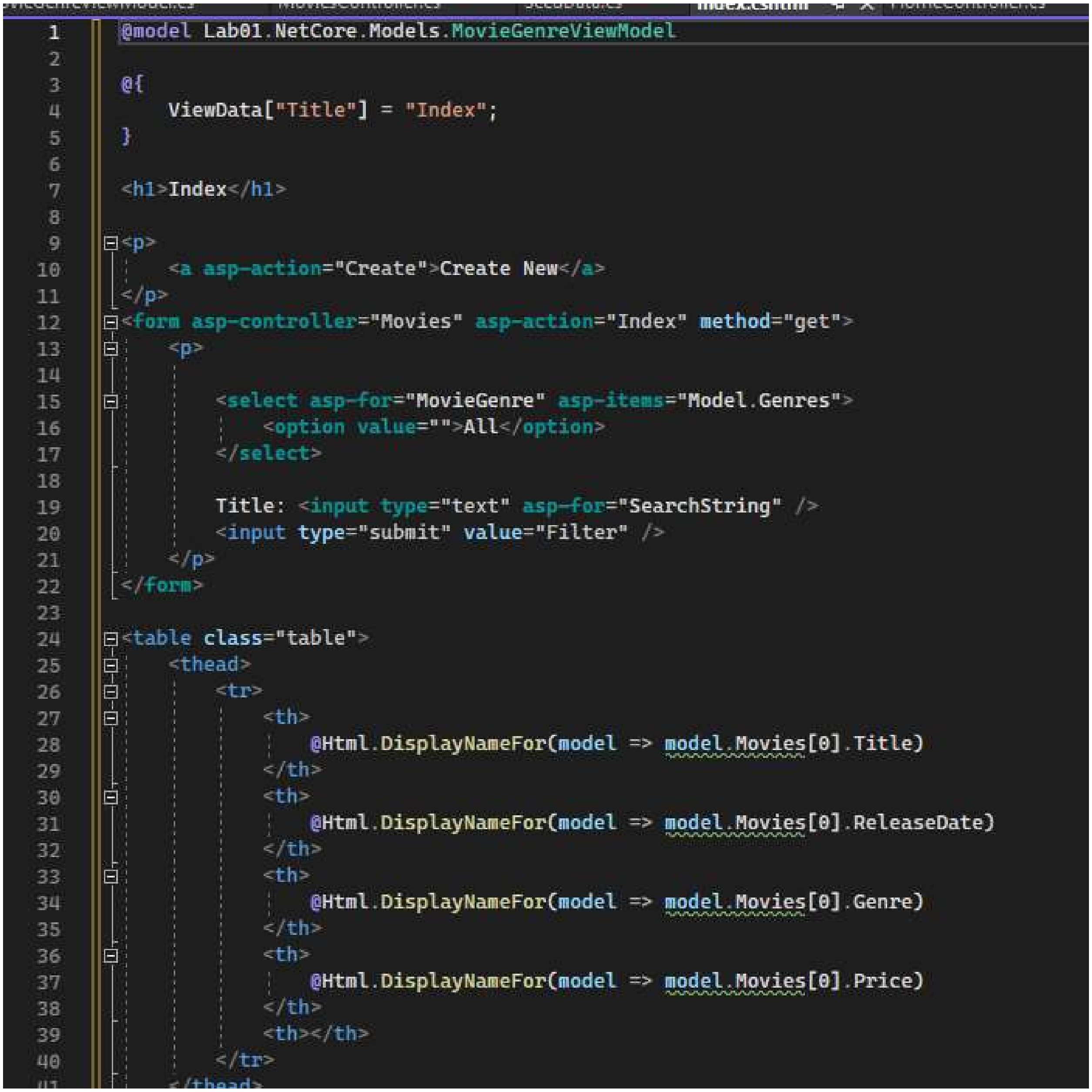
Title	Release Date	Genre	Price	
Rio Bravo	4/15/1959	Western	3.99	Edit | Details | Delete



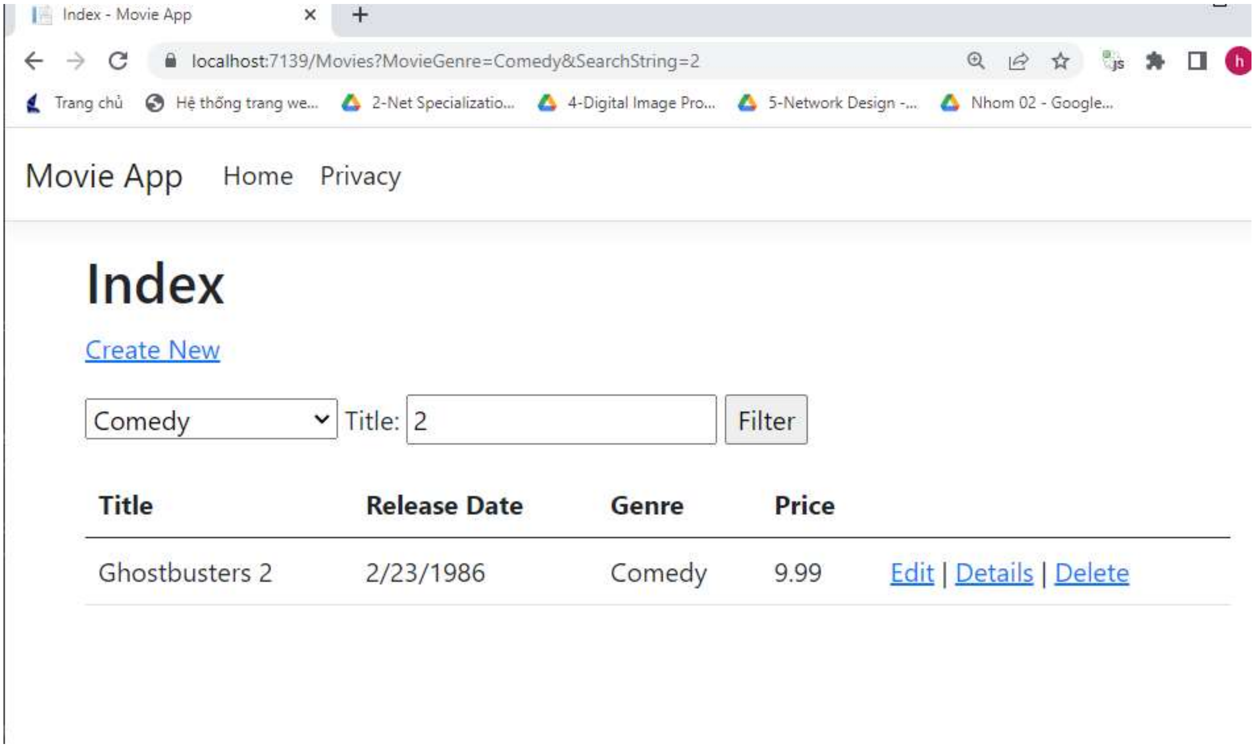
```
using Microsoft.AspNetCore.Mvc.Rendering;
using System.Collections.Generic;

namespace Lab01.NetCore.Models
{
    public class MovieGenreViewModel
    {
        public List<Movie>? Movies { get; set; }
        public SelectList? Genres { get; set; }
        public string? MovieGenre { get; set; }
        public string? SearchString { get; set; }
    }
}

```



```
{
    public class MoviesController : Controller
    {
        private readonly Lab01NetCoreContext _context;

        public MoviesController(Lab01NetCoreContext context)
        {
            _context = context;
        }

        // GET: Movies
        public async Task<IActionResult> Index(string movieGenre, string searchString)
        {
            IQueryable<string> genreQuery = from m in _context.Movie
                                            orderby m.Genre
                                            select m.Genre;
            var movies = from m in _context.Movie
                         select m;

            if (!string.IsNullOrEmpty(searchString))
            {
                movies = movies.Where(s => s.Title!.Contains(searchString));
            }

            if (!string.IsNullOrEmpty(movieGenre))
            {
                movies = movies.Where(x => x.Genre == movieGenre);
            }

            var movieGenreVM = new MovieGenreViewModel
            {
                Genres = new SelectList(await genreQuery.Distinct().ToListAsync()),
                Movies = await movies.ToListAsync()
            };

            return View(movieGenreVM);
        }

```



```
@model Lab01.NetCore.Models.MovieGenreViewModel

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>
<form asp-controller="Movies" asp-action="Index" method="get">
    <p>

        <select asp-for="MovieGenre" asp-items="Model.Genres">
            <option value="">All</option>
        </select>

        Title: <input type="text" asp-for="SearchString" />
        <input type="submit" value="Filter" />
    </p>
</form>

<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Movies[0].Title)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Movies[0].ReleaseDate)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Movies[0].Genre)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Movies[0].Price)
            </th>
            <th></th>
        </tr>
    </thead>
```

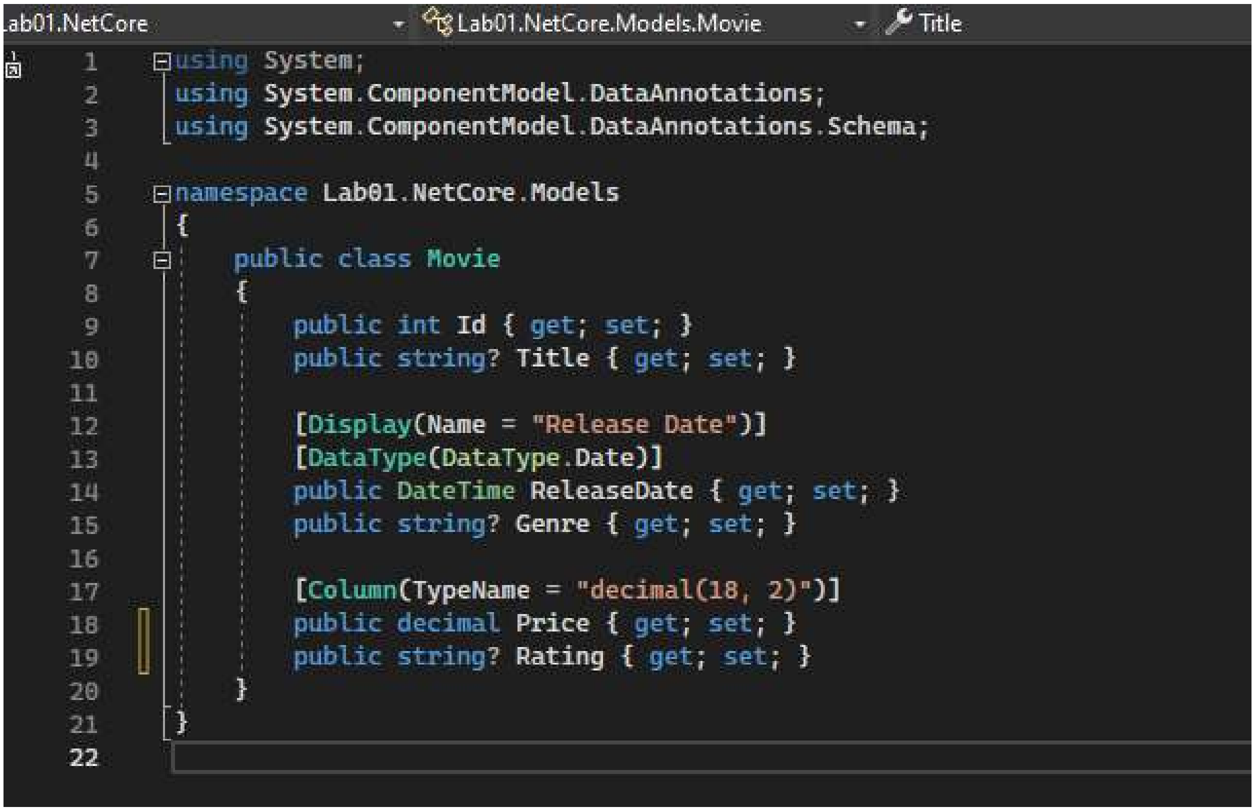


localhost:7139/Movies?MovieGenre=Comedy&SearchString=2

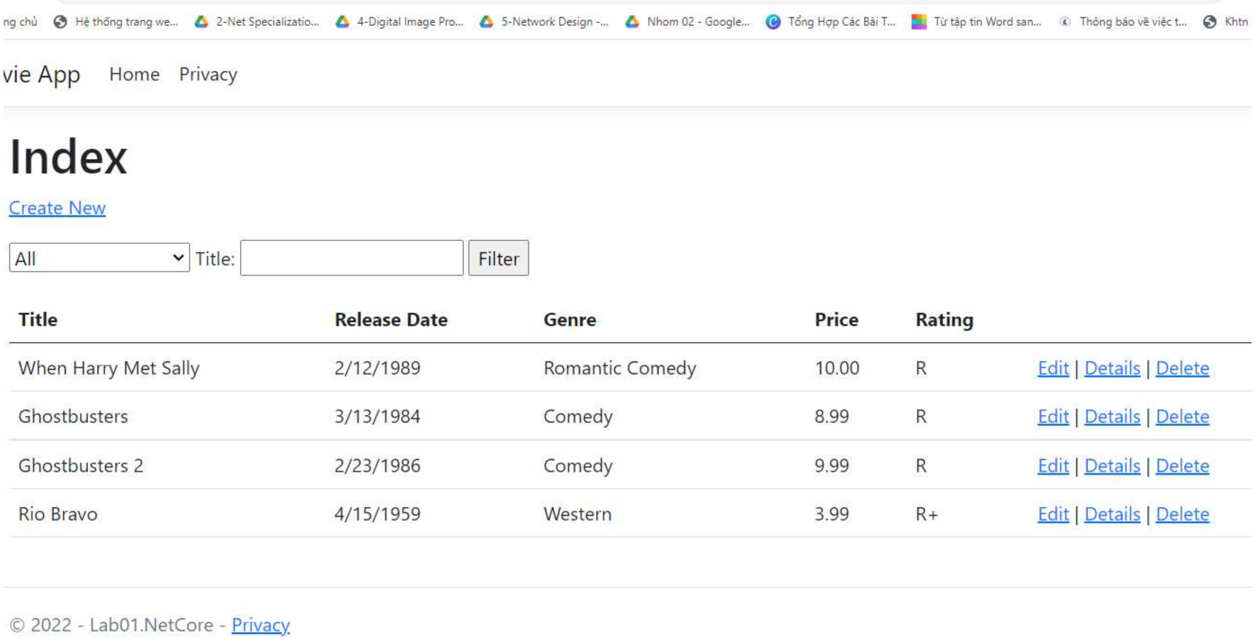
Index

Comedy Title: 2

Title	Release Date	Genre	Price
Ghostbusters 2	2/23/1986	Comedy	9.99



```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace Lab01.NetCore.Models
{
    public class Movie
    {
        public int Id { get; set; }
        public string? Title { get; set; }

        [Display(Name = "Release Date")]
        [DataType(DataType.Date)]
        public DateTime ReleaseDate { get; set; }
        public string? Genre { get; set; }

        [Column(TypeName = "decimal(18, 2)")]
        public decimal Price { get; set; }
        public string? Rating { get; set; }
    }
}
```

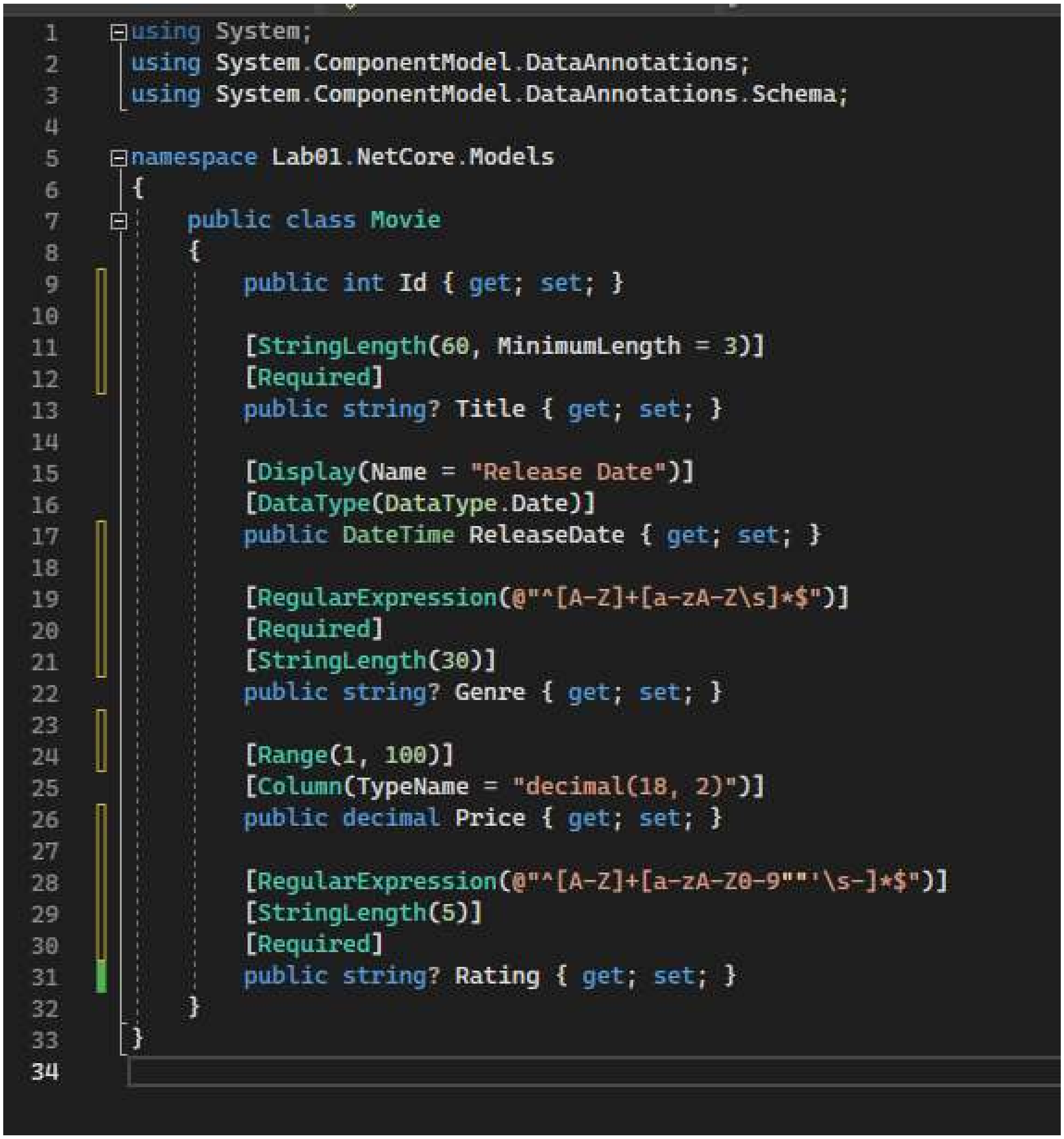


Index

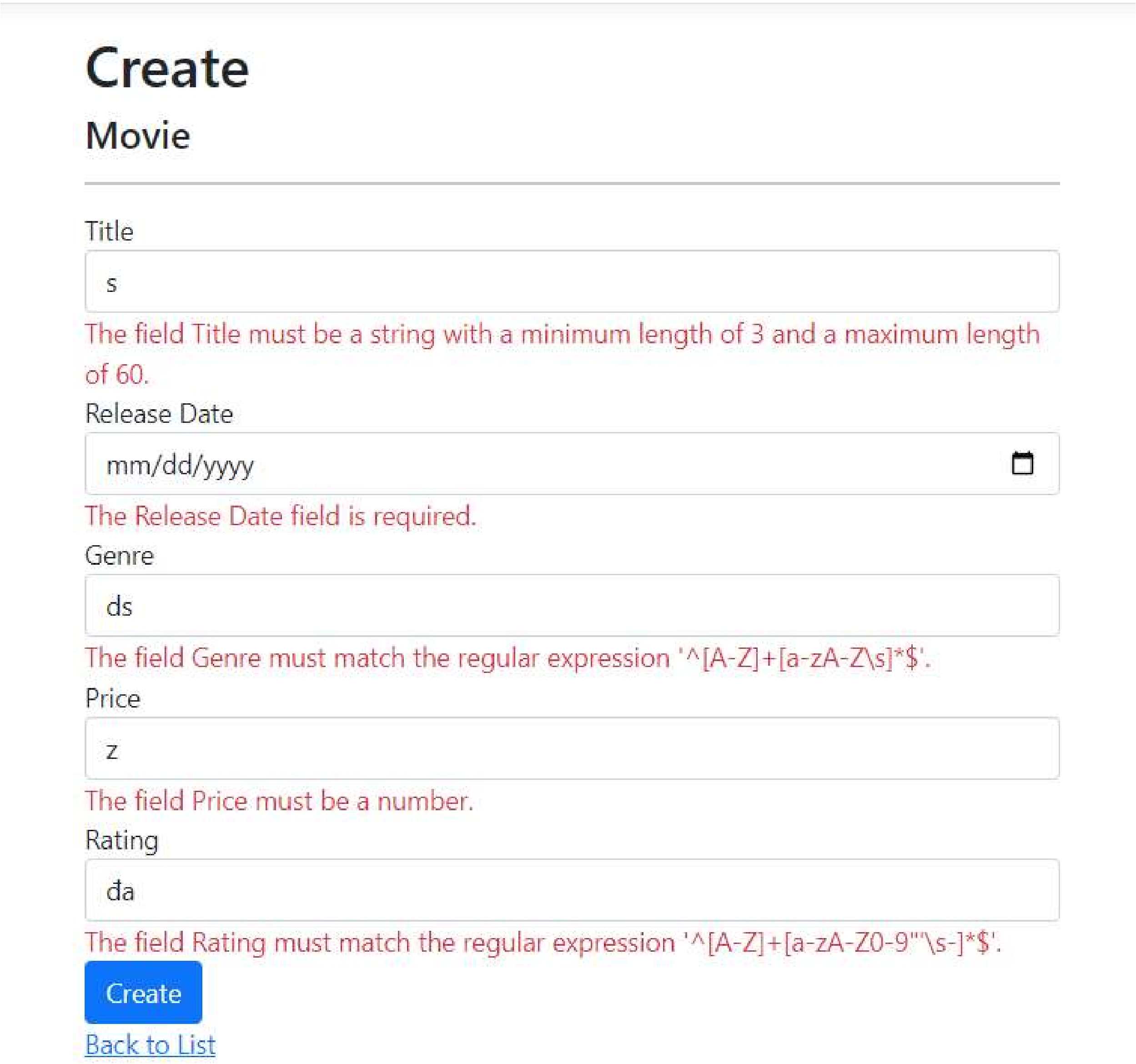
Create New

Title	Release Date	Genre	Price	Rating	
When Harry Met Sally	2/12/1989	Romantic Comedy	10.00	R	Edit | Details | Delete
Ghostbusters	3/13/1984	Comedy	8.99	R	Edit | Details | Delete
Ghostbusters 2	2/23/1986	Comedy	9.99	R	Edit | Details | Delete
Rio Bravo	4/15/1959	Western	3.99	R+	Edit | Details | Delete

© 2022 - Lab01.NetCore - Privacy



```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace Lab01.NetCore.Models
{
    public class Movie
    {
        public int Id { get; set; }

        [StringLength(60, MinimumLength = 3)]
        [Required]
        public string? Title { get; set; }

        [Display(Name = "Release Date")]
        [DataType(DataType.Date)]
        public DateTime ReleaseDate { get; set; }

        [RegularExpression(@"^[A-Z]+[a-zA-Z\s]*$")]
        [Required]
        [StringLength(30)]
        public string? Genre { get; set; }

        [Range(1, 100)]
        [Column(TypeName = "decimal(18, 2)")]
        public decimal Price { get; set; }

        [RegularExpression(@"^[A-Z]+[a-zA-Z0-9""'\s-]*$")]
        [StringLength(5)]
        [Required]
        public string? Rating { get; set; }
    }
}
```



Create

Movie

Title

s

The field Title must be a string with a minimum length of 3 and a maximum length of 60.

Release Date

mm/dd/yyyy

The Release Date field is required.

Genre

ds

The field Genre must match the regular expression '^[A-Z]+[a-zA-Z\s]*$'.

Price

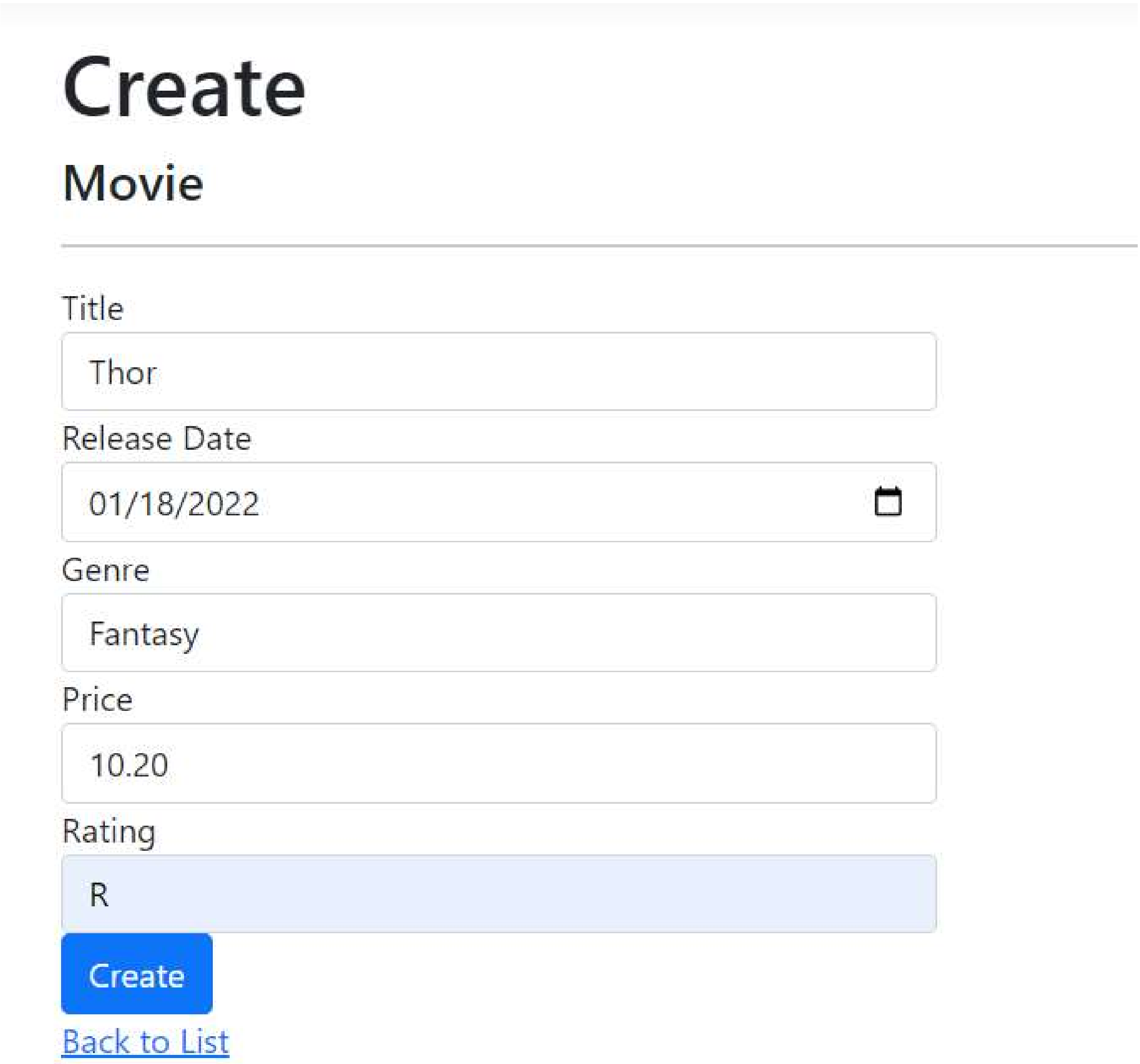
z

The field Price must be a number.

Rating

đa

The field Rating must match the regular expression '^[A-Z]+[a-zA-Z0-9"'\s-]*$'.



Create

Movie

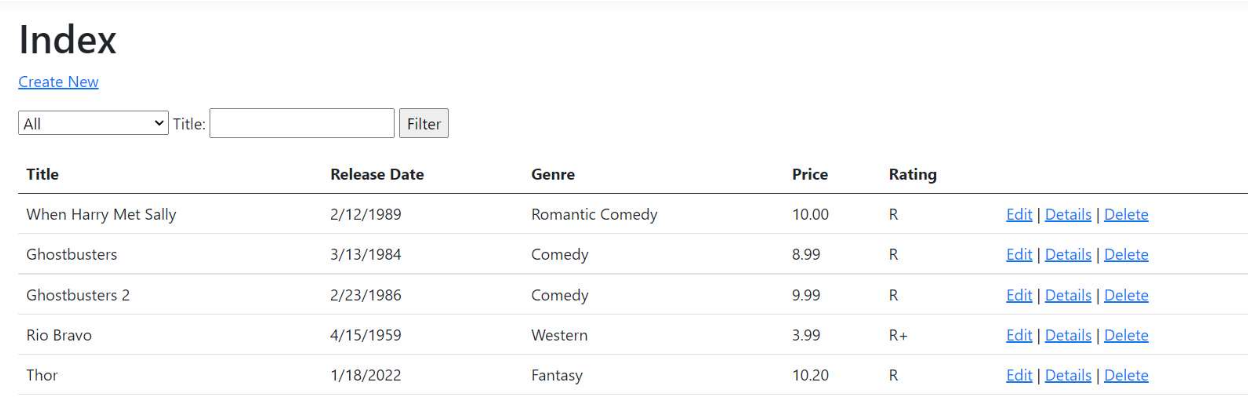
Title: Thor

Release Date: 01/18/2022

Genre: Fantasy

Price: 10.20

Rating: R

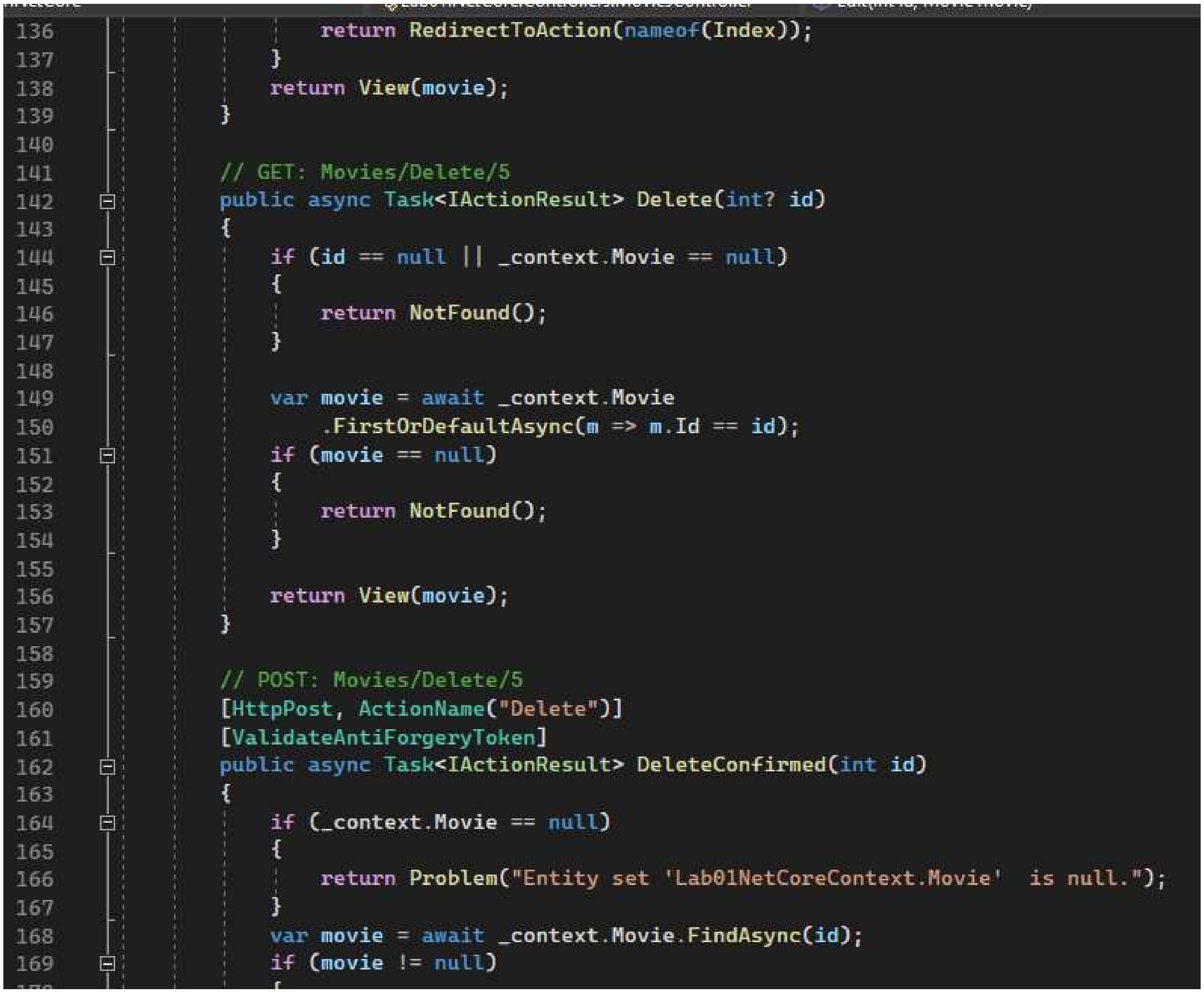


Index

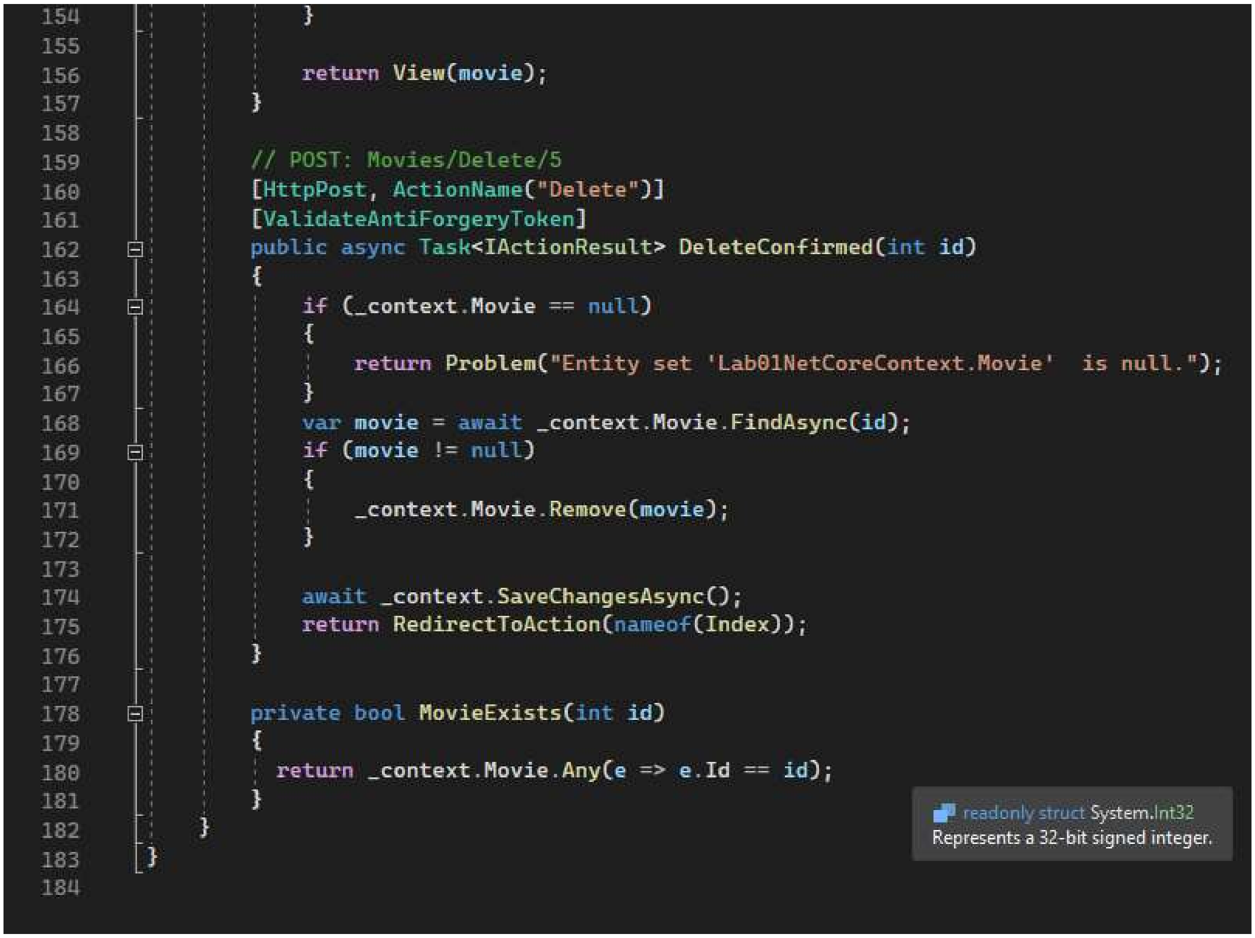
Create New

All Title: Filter

Title	Release Date	Genre	Price	Rating	
When Harry Met Sally	2/12/1989	Romantic Comedy	10.00	R	Edit | Details | Delete
Ghostbusters	3/13/1984	Comedy	8.99	R	Edit | Details | Delete
Ghostbusters 2	2/23/1986	Comedy	9.99	R	Edit | Details | Delete
Rio Bravo	4/15/1959	Western	3.99	R+	Edit | Details | Delete
Thor	1/18/2022	Fantasy	10.20	R	Edit | Details | Delete

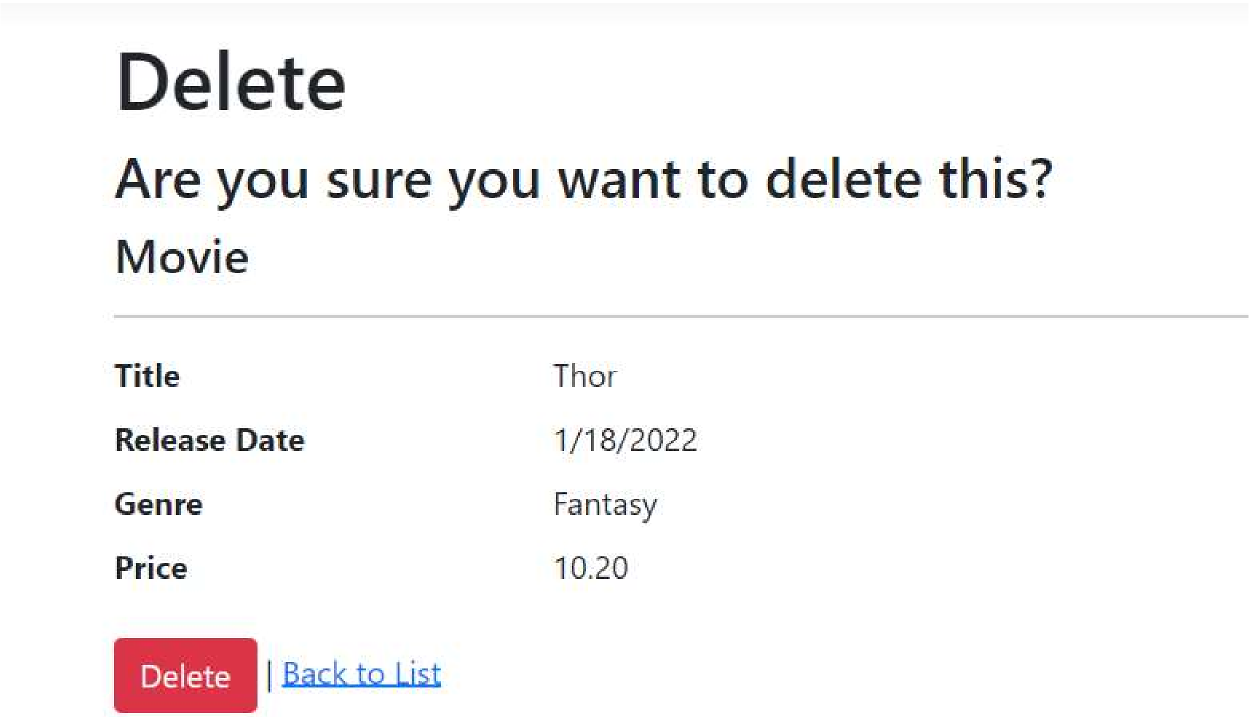


```
                return RedirectToAction(nameof(Index));
            }
            return View(movie);
        }

        // GET: Movies/Delete/5
        public async Task<IActionResult> Delete(int? id)
        {
            if (id == null || _context.Movie == null)
            {
                return NotFound();
            }

            var movie = await _context.Movie
                .FirstOrDefaultAsync(m => m.Id == id);
            if (movie == null)
            {
                return NotFound();
            }

            return View(movie);
        }

        // POST: Movies/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> DeleteConfirmed(int id)
        {
            if (_context.Movie == null)
            {
                return Problem("Entity set 'Lab01NetCoreContext.Movie'  is null.");
            }
            var movie = await _context.Movie.FindAsync(id);
            if (movie != null)
```



```
            }

            return View(movie);
        }

        // POST: Movies/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> DeleteConfirmed(int id)
        {
            if (_context.Movie == null)
            {
                return Problem("Entity set 'Lab01NetCoreContext.Movie'  is null.");
            }
            var movie = await _context.Movie.FindAsync(id);
            if (movie != null)
            {
                _context.Movie.Remove(movie);
            }

            await _context.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }

        private bool MovieExists(int id)
        {
          return _context.Movie.Any(e => e.Id == id);
        }
    }
}
```

readonly struct System.Int32
Represents a 32-bit signed integer.



Delete

Are you sure you want to delete this?

Movie

Title	Thor
Release Date	1/18/2022
Genre	Fantasy
Price	10.20