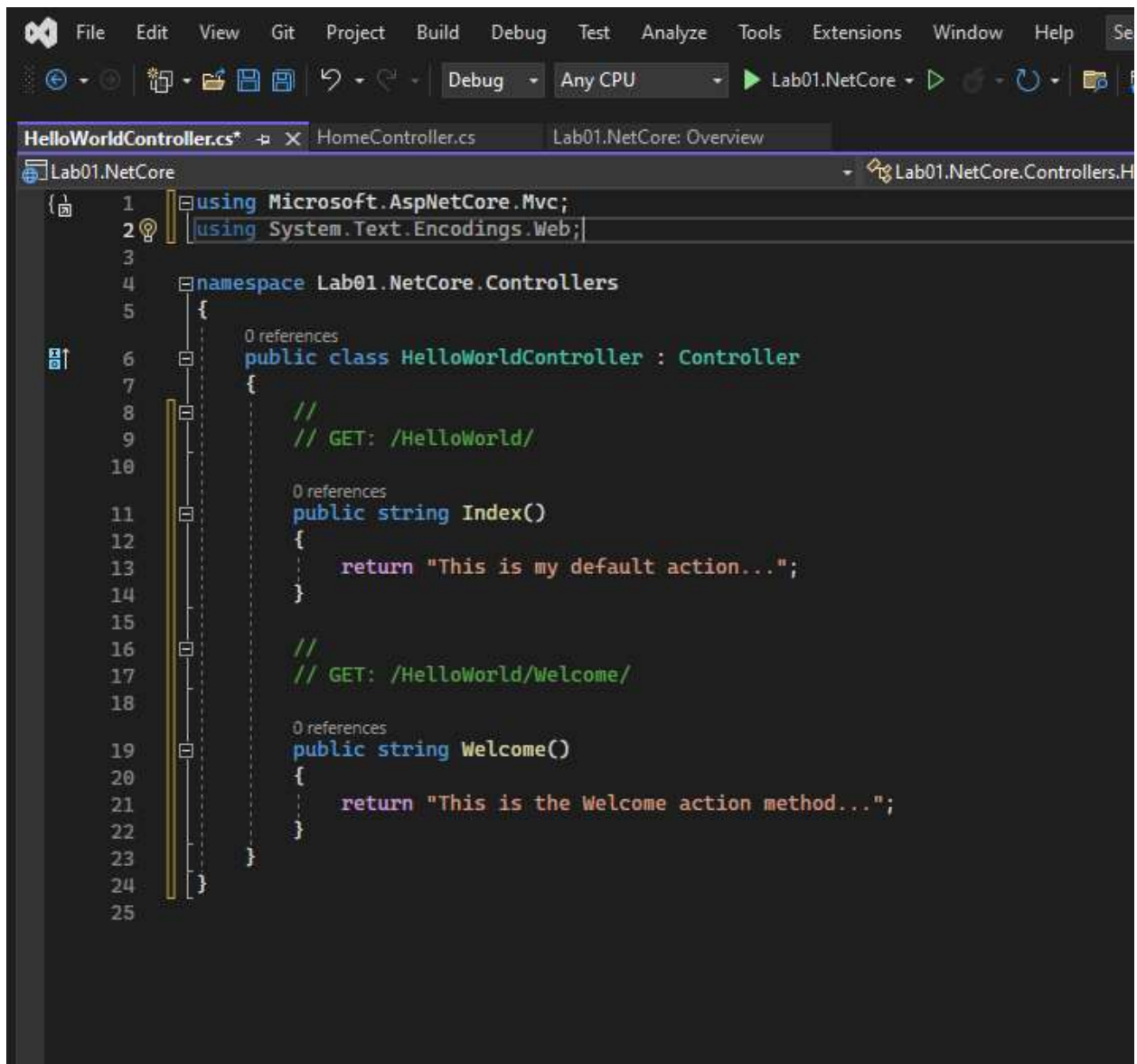


Câu 1

Lab01.NetCore [Home](#) [Privacy](#)

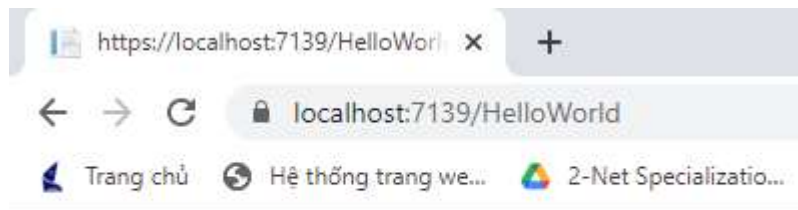
Welcome

Learn about [building Web apps with ASP.NET Core](#).



The image shows a screenshot of the Visual Studio IDE. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, and Help. Below the menu bar is a toolbar with icons for file operations and a dropdown menu showing 'Debug' and 'Any CPU'. The main window displays the 'HelloWorldController.cs' file, which is part of the 'Lab01.NetCore' project. The code is as follows:

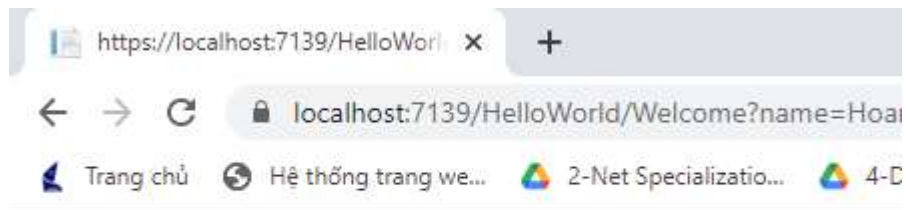
```
1 using Microsoft.AspNetCore.Mvc;
2 using System.Text.Encodings.Web;
3
4 namespace Lab01.NetCore.Controllers
5 {
6     0 references
7     public class HelloWorldController : Controller
8     {
9         //
10        // GET: /HelloWorld/
11
12        0 references
13        public string Index()
14        {
15            return "This is my default action...";
16        }
17
18        //
19        // GET: /HelloWorld/Welcome/
20
21        0 references
22        public string Welcome()
23        {
24            return "This is the Welcome action method...";
25        }
26    }
27 }
```



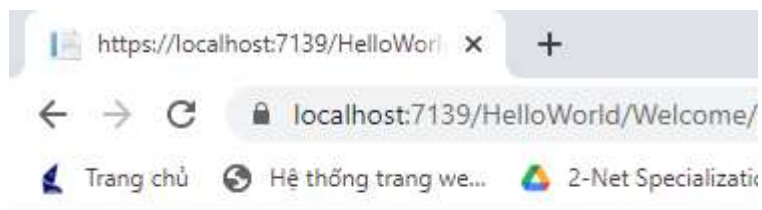
This is my default action...



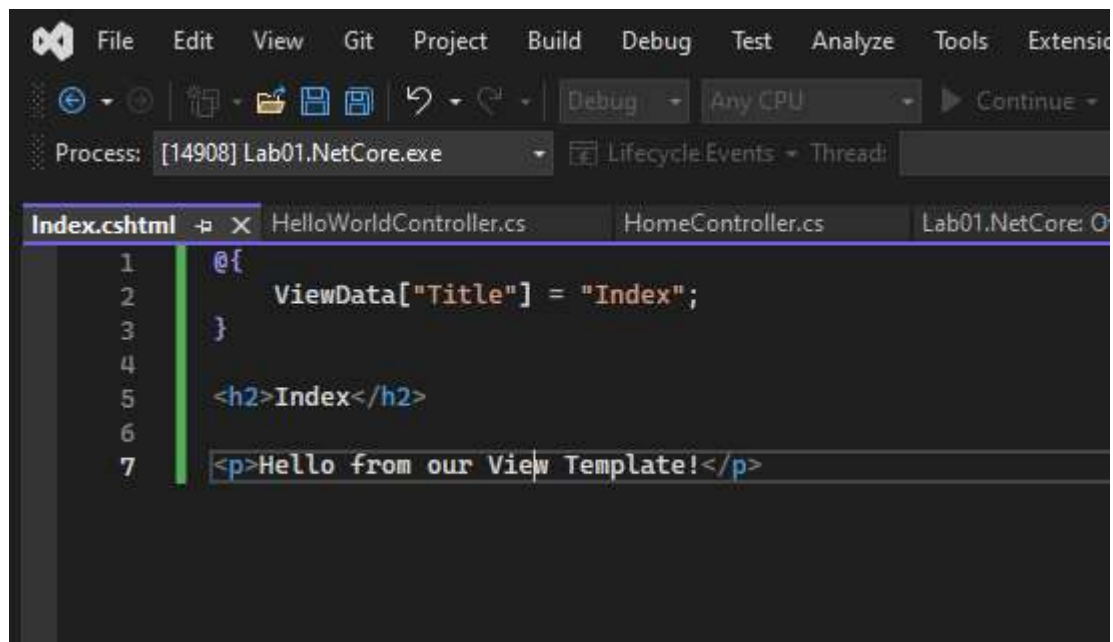
This is the Welcome action method...

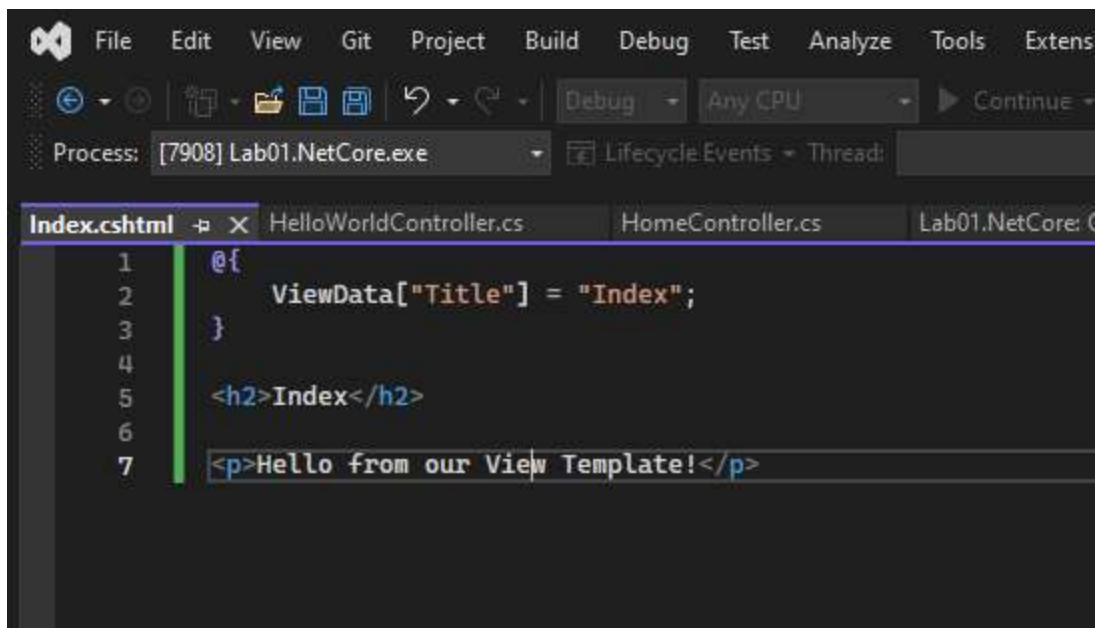


Hello Hoang, NumTimes is: 22



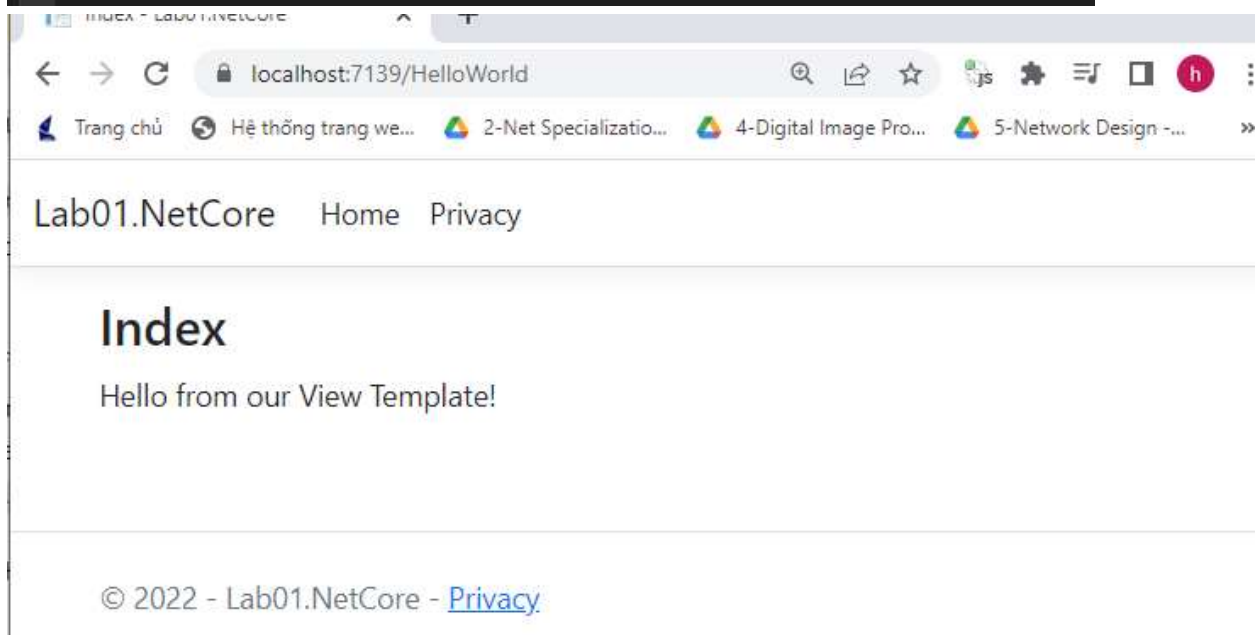
Hello Hoang, ID: 1554

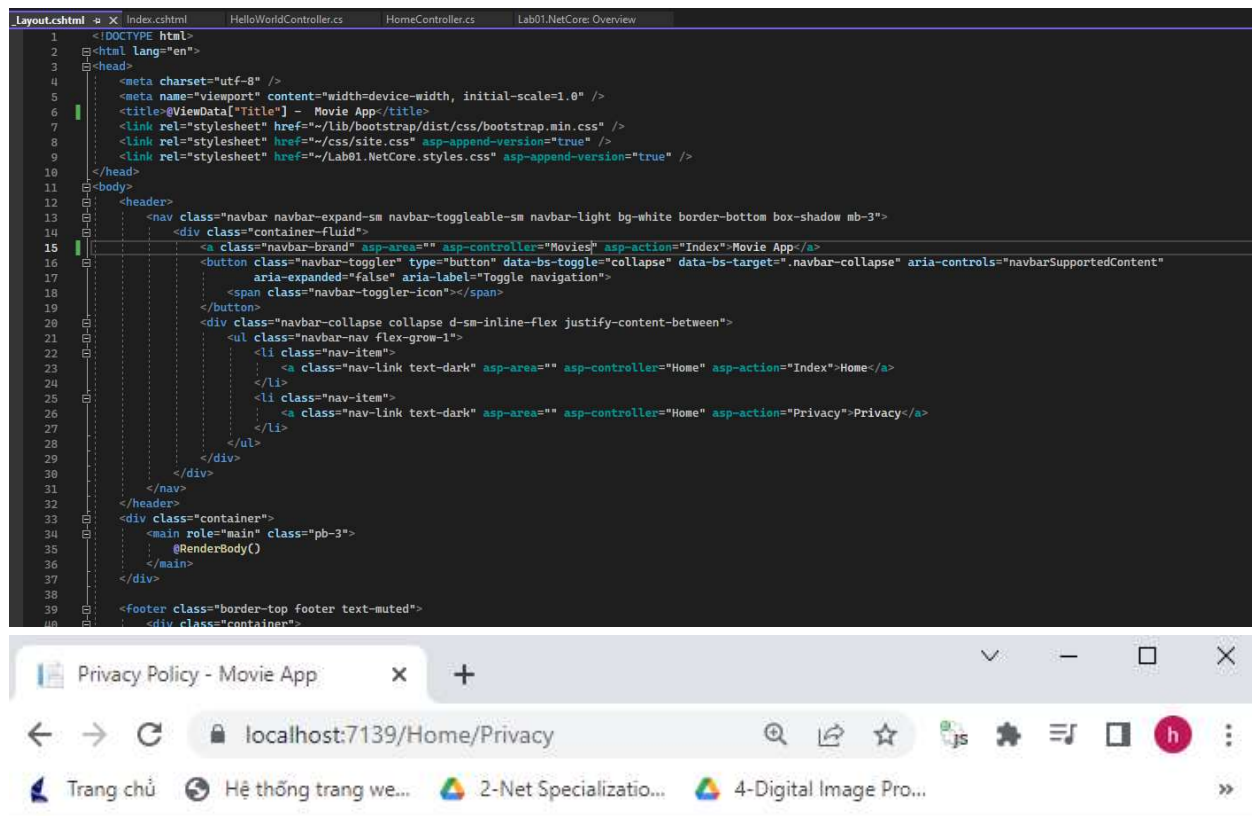




The screenshot shows the Visual Studio Code editor with the 'Index.cshtml' file open. The file content is as follows:

```
1 @{
2     ViewData["Title"] = "Index";
3 }
4
5 <h2>Index</h2>
6
7 <p>Hello from our View Template!</p>
```

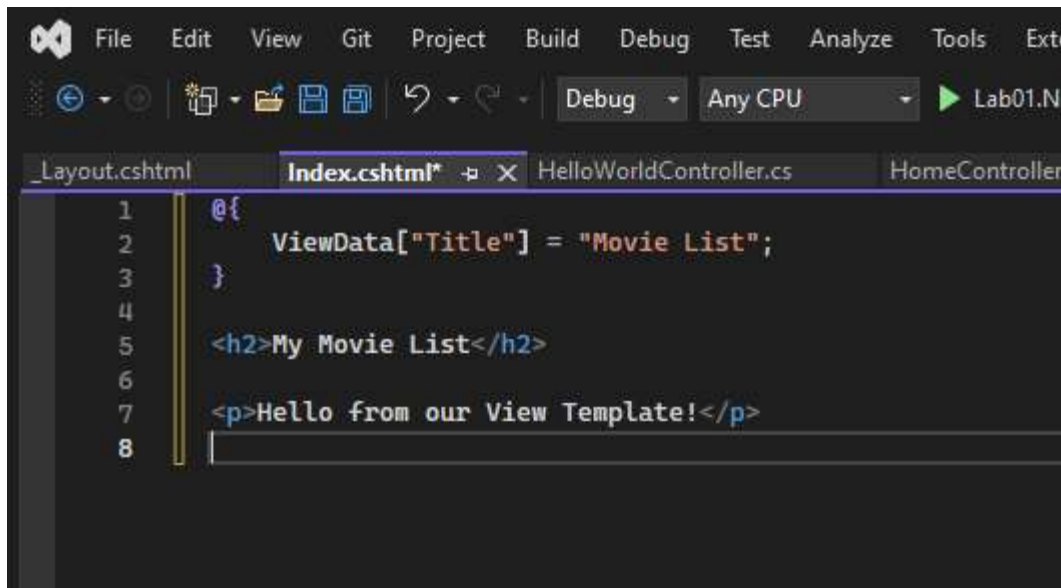




Movie App Home Privacy

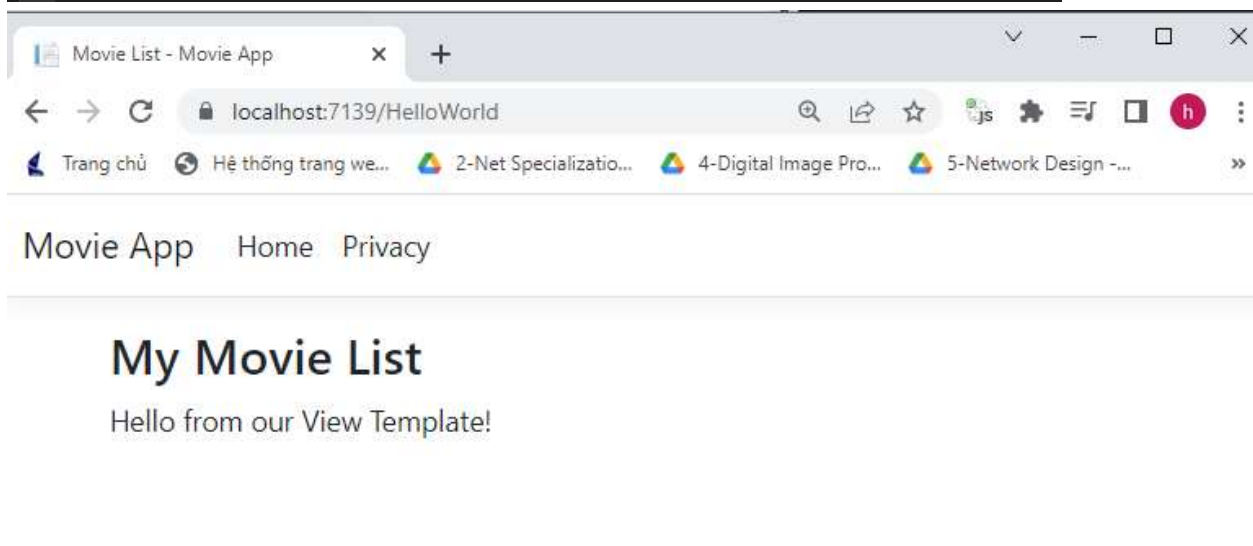
Privacy Policy

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



The screenshot shows the Visual Studio code editor with the 'Index.cshtml' file open. The code is as follows:

```
1 @{
2     ViewData["Title"] = "Movie List";
3 }
4
5 <h2>My Movie List</h2>
6
7 <p>Hello from our View Template!</p>
8
```



```

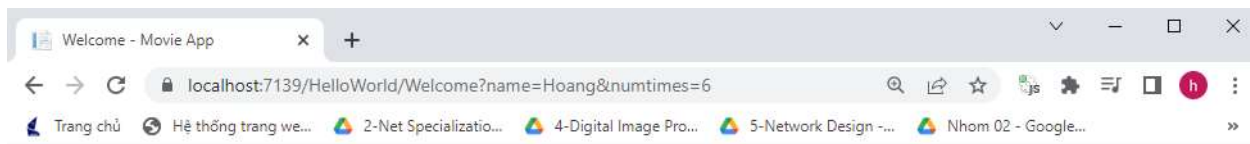
1  using Microsoft.AspNetCore.Mvc;
2  using System.Text.Encodings.Web;
3
4  namespace Lab01.NetCore.Controllers
5  {
6      0 references
7      public class HelloWorldController : Controller
8      {
9          //
10         // GET: /HelloWorld/
11
12         0 references
13         public IActionResult Index()
14         {
15             return View();
16         }
17
18         //
19         // GET: /HelloWorld/Welcome/
20         // Requires using System.Text.Encodings.Web;
21         0 references
22         public IActionResult Welcome(string name, int numTimes = 1)
23         {
24             ViewData["Message"] = "Hello " + name;
25             ViewData["NumTimes"] = numTimes;
26
27             return View();
28         }
29     }
30 }

```

```

1  @{
2      ViewData["Title"] = "Welcome";
3  }
4
5  <h2>Welcome</h2>
6
7  <ul>
8      @for (int i = 0; i < (int)ViewData["NumTimes"]!; i++)
9      {
10         <li>@ViewData["Message"]</li>
11     }
12 </ul>

```

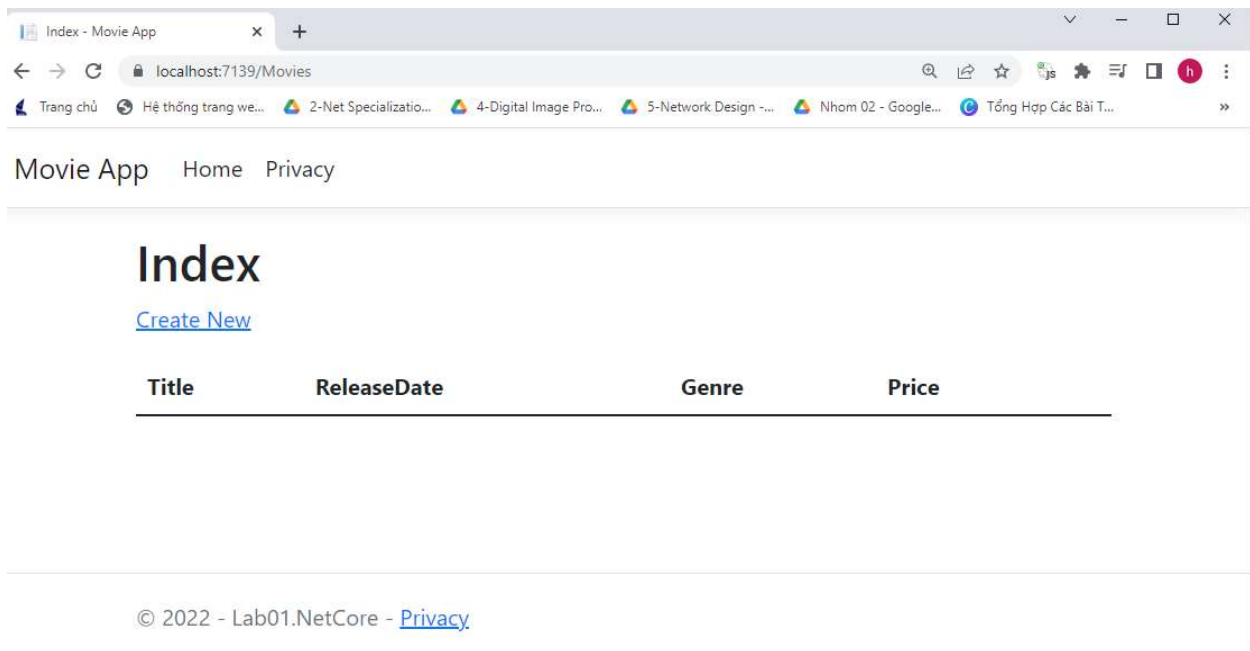
Movie App Home Privacy

Welcome

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

© 2022 - Lab01.NetCore - [Privacy](#)

```
1 using System;
2 using Microsoft.EntityFrameworkCore.Migrations;
3
4 #nullable disable
5
6 namespace Lab01.NetCore.Migrations
7 {
8     /// <inheritdoc />
9     public partial class InitialCreate : Migration
10     {
11         /// <inheritdoc />
12         protected override void Up(MigrationBuilder migrationBuilder)
13         {
14             migrationBuilder.CreateTable(
15                 name: "Movie",
16                 columns: table => new
17                 {
18                     Id = table.Column<int>(type: "int", nullable: false)
19                         .Annotation("SqlServer:Identity", "1, 1"),
20                     Title = table.Column<string>(type: "nvarchar(max)", nullable: true),
21                     ReleaseDate = table.Column<DateTime>(type: "datetime2", nullable: false),
22                     Genre = table.Column<string>(type: "nvarchar(max)", nullable: true),
23                     Price = table.Column<decimal>(type: "decimal(18,2)", nullable: false)
24                 },
25                 constraints: table =>
26                 {
27                     table.PrimaryKey("PK_Movie", x => x.Id);
28                 });
29         }
30
31         /// <inheritdoc />
32         protected override void Down(MigrationBuilder migrationBuilder)
33         {
34             migrationBuilder.DropTable(
35                 name: "Movie");
36         }
37     }
38 }
39
```



```

10     {
11         public static void Initialize(IServiceProvider serviceProvider)
12         {
13             using (var context = new Lab01NetCoreContext(
14                 serviceProvider.GetRequiredService<
15                     DbContextOptions<Lab01NetCoreContext>>()))
16             {
17                 // Look for any movies.
18                 if (context.Movie.Any())
19                 {
20                     return; // DB has been seeded
21                 }
22
23                 context.Movie.AddRange(
24                     new Movie
25                     {
26                         Title = "When Harry Met Sally",
27                         ReleaseDate = DateTime.Parse("1989-2-12"),
28                         Genre = "Romantic Comedy",
29                         Price = 7.99M
30                     },
31
32                     new Movie
33                     {
34                         Title = "Ghostbusters ",
35                         ReleaseDate = DateTime.Parse("1984-3-13"),
36                         Genre = "Comedy",
37                         Price = 8.99M
38                     },
39
40                     new Movie
41                     {
42                         Title = "Ghostbusters 2",
43                         ReleaseDate = DateTime.Parse("1986-2-23"),
44                         Genre = "Comedy",
45                         Price = 9.99M
46                     },
47
48                     new Movie
49                     {

```

```

1  using Microsoft.EntityFrameworkCore;
2  using Microsoft.Extensions.DependencyInjection;
3  using Lab01.NetCore.Data;
4  using Lab01.NetCore.Models;
5
6  var builder = WebApplication.CreateBuilder(args);
7
8  builder.Services.AddDbContext<Lab01NetCoreContext>(options =>
9      options.UseSqlServer(builder.Configuration.GetConnectionString("Lab01NetCoreContext") ?? throw new InvalidOperationException("Connection string 'Lab01NetCoreContext' not found.")));
10
11 // Add services to the container.
12 builder.Services.AddControllersWithViews();
13
14 var app = builder.Build();
15
16 using (var scope = app.Services.CreateScope())
17 {
18     var services = scope.ServiceProvider;
19     SeedData.Initialize(services);
20 }
21
22 // Configure the HTTP request pipeline.
23 if (app.Environment.IsDevelopment())
24 {
25     app.UseExceptionHandler("/Home/Error");
26     // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
27     app.UseHsts();
28 }
29
30 app.UseHttpsRedirection();
31 app.UseStaticFiles();
32
33 app.UseRouting();
34
35 app.UseAuthorization();
36
37 app.MapControllerRoute(
38     name: "default",
39     pattern: "{controller=Home}/{action=Index}/{id?}");
40

```

Index - Movie App

localhost:7139/Movies

Trang chủ | Hệ thống trang web... | 2-Net Specialization... | 4-Digital Image Pro... | 5-Network Design... | Nhóm 02 - Google... | Tổng Hợp Các Bài T... | Từ tập tin Word san... | Thông báo về việc t... | Khtn GPA | CSDL 20TTH - HK2...

Movie App | Home | Privacy

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

```
1  using System.ComponentModel.DataAnnotations;
2  using System.ComponentModel.DataAnnotations;
3  using System.ComponentModel.DataAnnotations.Schema;
4
5  namespace Lab01.NetCore.Models
6  {
7      public class Movie
8      {
9          public int Id { get; set; }
10         public string? Title { get; set; }
11
12         [Display(Name = "Release Date")]
13         [DataType(DataType.Date)]
14         public DateTime ReleaseDate { get; set; }
15         public string? Genre { get; set; }
16
17         [Column(TypeName = "decimal(18, 2)")]
18         public decimal Price { get; set; }
19     }
20 }
21
```

Edit

Movie

Title

Release Date



Genre

Price

The field Price must be a number.

[Back to List](#)

```

1  @model IEnumerable<Lab01.NetCore.Models.Movie>
2
3  @{
4      ViewData["Title"] = "Index";
5  }
6
7  <h1>Index</h1>
8
9  <p>
10     <a asp-action="Create">Create New</a>
11 </p>
12
13 <form asp-controller="Movies" asp-action="Index" method="get">
14     <p>
15         Title: <input type="text" name="SearchString" />
16         <input type="submit" value="Filter" />
17     </p>
18 </form>
19
20 <table class="table">
21     <thead>
22         <tr>
23             <th>
24                 @Html.DisplayNameFor(model => model.Title)
25             </th>
26             <th>
27                 @Html.DisplayNameFor(model => model.ReleaseDate)
28             </th>
29             <th>
30                 @Html.DisplayNameFor(model => model.Genre)
31             </th>
32             <th>
33                 @Html.DisplayNameFor(model => model.Price)
34             </th>
35             <th></th>
36         </tr>
37     </thead>
38     <tbody>
39         @foreach (var item in Model) {
40             <tr>
41                 <td>

```



```

7  using Microsoft.EntityFrameworkCore;
8  using Lab01.NetCore.Data;
9  using Lab01.NetCore.Models;
10
11 namespace Lab01.NetCore.Controllers
12 {
13     public class MoviesController : Controller
14     {
15         private readonly Lab01NetCoreContext _context;
16
17         public MoviesController(Lab01NetCoreContext context)
18         {
19             _context = context;
20         }
21
22         // GET: Movies
23         public async Task<IActionResult> Index(string searchString)
24         {
25             var movies = from m in _context.Movie
26                         select m;
27
28             if (!String.IsNullOrEmpty(searchString))
29             {
30                 movies = movies.Where(s => s.Title!.Contains(searchString));
31             }
32
33             return View(await movies.ToListAsync());
34         }
35
36         // GET: Movies/Details/5
37         public async Task<IActionResult> Details(int? id)
38         {
39             if (id == null || _context.Movie == null)
40             {
41                 return NotFound();
42             }
43
44             var movie = await _context.Movie
45                 .FirstOrDefaultAsync(m => m.Id == id);
46             if (movie == null)

```

Index

[Create New](#)

Title:

Index

[Create New](#)

Title:

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

```
1  using Microsoft.AspNetCore.Mvc.Rendering;
2  using System.Collections.Generic;
3
4  namespace Lab01.NetCore.Models
5  {
6      public class MovieGenreViewModel
7      {
8          public List<Movie>? Movies { get; set; }
9          public SelectList? Genres { get; set; }
10         public string? MovieGenre { get; set; }
11         public string? SearchString { get; set; }
12     }
13 }
14
```



```
12 {
13     public class MoviesController : Controller
14     {
15         private readonly Lab01NetCoreContext _context;
16
17         public MoviesController(Lab01NetCoreContext context)
18         {
19             _context = context;
20         }
21
22         // GET: Movies
23         public async Task<IActionResult> Index(string movieGenre, string searchString)
24         {
25             IQueryable<string> genreQuery = from m in _context.Movie
26                                             orderby m.Genre
27                                             select m.Genre;
28
29             var movies = from m in _context.Movie
30                         select m;
31
32             if (!string.IsNullOrEmpty(searchString))
33             {
34                 movies = movies.Where(s => s.Title!.Contains(searchString));
35             }
36
37             if (!string.IsNullOrEmpty(movieGenre))
38             {
39                 movies = movies.Where(x => x.Genre == movieGenre);
40             }
41
42             var movieGenreVM = new MovieGenreViewModel
43             {
44                 Genres = new SelectList(await genreQuery.Distinct().ToListAsync()),
45                 Movies = await movies.ToListAsync()
46             };
47
48             return View(movieGenreVM);
49         }
50     }
51 }
```

```

1  @model Lab01.NetCore.Models.MovieGenreViewModel
2
3  @{
4      ViewData["Title"] = "Index";
5  }
6
7  <h1>Index</h1>
8
9  <p>
10     <a asp-action="Create">Create New</a>
11 </p>
12 <form asp-controller="Movies" asp-action="Index" method="get">
13     <p>
14         <select asp-for="MovieGenre" asp-items="Model.Genres">
15             <option value="">All</option>
16         </select>
17
18         Title: <input type="text" asp-for="SearchString" />
19         <input type="submit" value="Filter" />
20     </p>
21 </form>
22
23
24 <table class="table">
25     <thead>
26         <tr>
27             <th>
28                 @Html.DisplayNameFor(model => model.Movies[0].Title)
29             </th>
30             <th>
31                 @Html.DisplayNameFor(model => model.Movies[0].ReleaseDate)
32             </th>
33             <th>
34                 @Html.DisplayNameFor(model => model.Movies[0].Genre)
35             </th>
36             <th>
37                 @Html.DisplayNameFor(model => model.Movies[0].Price)
38             </th>
39         </tr>
40     </thead>

```

Index - Movie App

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

Trang chủ Hệ thống trang we... 2-Net Specializatio... 4-Digital Image Pro... 5-Network Design -... Nhom 02 - Google...

Movie App Home Privacy

Index

[Create New](#)

Comedy Title: 2 Filter

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

```
Lab01.NetCore
  Lab01.NetCore.Models.Movie
    Title
1  using System;
2  using System.ComponentModel.DataAnnotations;
3  using System.ComponentModel.DataAnnotations.Schema;
4
5  namespace Lab01.NetCore.Models
6  {
7      public class Movie
8      {
9          public int Id { get; set; }
10         public string? Title { get; set; }
11
12         [Display(Name = "Release Date")]
13         [DataType(DataType.Date)]
14         public DateTime ReleaseDate { get; set; }
15         public string? Genre { get; set; }
16
17         [Column(TypeName = "decimal(18, 2)")]
18         public decimal Price { get; set; }
19         public string? Rating { get; set; }
20     }
21 }
22
```

[vie App](#) [Home](#) [Privacy](#)

Index

[Create New](#)

Title:

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

```
1  using System;
2  using System.ComponentModel.DataAnnotations;
3  using System.ComponentModel.DataAnnotations.Schema;
4
5  namespace Lab01.NetCore.Models
6  {
7      public class Movie
8      {
9          public int Id { get; set; }
10
11          [StringLength(60, MinimumLength = 3)]
12          [Required]
13          public string? Title { get; set; }
14
15          [Display(Name = "Release Date")]
16          [DataType(DataType.Date)]
17          public DateTime ReleaseDate { get; set; }
18
19          [RegularExpression(@"^[A-Z]+[a-zA-Z\s]*$")]
20          [Required]
21          [StringLength(30)]
22          public string? Genre { get; set; }
23
24          [Range(1, 100)]
25          [Column(TypeName = "decimal(18, 2)")]
26          public decimal Price { get; set; }
27
28          [RegularExpression(@"^[A-Z]+[a-zA-Z0-9'\s-]*$")]
29          [StringLength(5)]
30          [Required]
31          public string? Rating { get; set; }
32      }
33  }
34
```

Create

Movie

Title

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

Release Date



The Release Date field is required.

Genre

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

Price

The field Price must be a number.

Rating

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

Create

[Back to List](#)

Create

Movie

Title

Thor

Release Date

01/18/2022



Genre

Fantasy

Price

10.20

Rating

R

Create

[Back to List](#)

Index

[Create New](#)

All Title:

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


```
136         return RedirectToAction(nameof(Index));
137     }
138     return View(movie);
139 }
140
141 // GET: Movies/Delete/5
142 public async Task<IActionResult> Delete(int? id)
143 {
144     if (id == null || _context.Movie == null)
145     {
146         return NotFound();
147     }
148
149     var movie = await _context.Movie
150         .FirstOrDefaultAsync(m => m.Id == id);
151     if (movie == null)
152     {
153         return NotFound();
154     }
155
156     return View(movie);
157 }
158
159 // POST: Movies/Delete/5
160 [HttpPost, ActionName("Delete")]
161 [ValidateAntiForgeryToken]
162 public async Task<IActionResult> DeleteConfirmed(int id)
163 {
164     if (_context.Movie == null)
165     {
166         return Problem("Entity set 'Lab01NetCoreContext.Movie' is null.");
167     }
168     var movie = await _context.Movie.FindAsync(id);
169     if (movie != null)
```



```

154     }
155
156     return View(movie);
157 }
158
159 // POST: Movies/Delete/5
160 [HttpPost, ActionName("Delete")]
161 [ValidateAntiForgeryToken]
162 public async Task<IActionResult> DeleteConfirmed(int id)
163 {
164     if (_context.Movie == null)
165     {
166         return Problem("Entity set 'Lab01NetCoreContext.Movie' is null.");
167     }
168     var movie = await _context.Movie.FindAsync(id);
169     if (movie != null)
170     {
171         _context.Movie.Remove(movie);
172     }
173
174     await _context.SaveChangesAsync();
175     return RedirectToAction(nameof(Index));
176 }
177
178 private bool MovieExists(int id)
179 {
180     return _context.Movie.Any(e => e.Id == id);
181 }
182 }
183 }
184

```

 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

Delete

| [Back to List](#)