NATIONAL ECONOMICS UNIVERSITY

SCHOOL OF INFORMATION TECHNOLOGY AND DIGITAL ECONOMICS

CHAPTER IV

ASP.NET WEB APP (MVC)

PHAM THAO

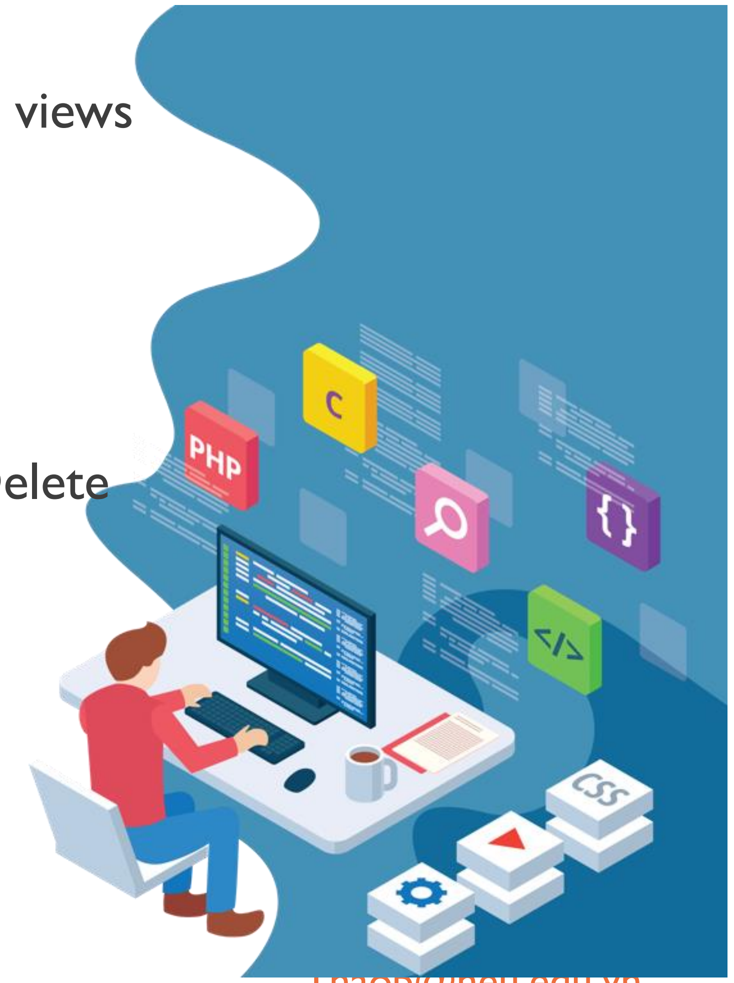
Thaop@neu.edu.vn

OUTLINE

ASP.NET Core Web App MVC

- The Model-View-Controller (MVC) architectural
- Add a controller
  - Change Method
- Add a view
  - Change Layout
  - Passing Data from the Controller to the View
- Add a model
  - Scaffolding movie Pages
  - Initial migration
- Run MovieWebsite
- appsettings.json

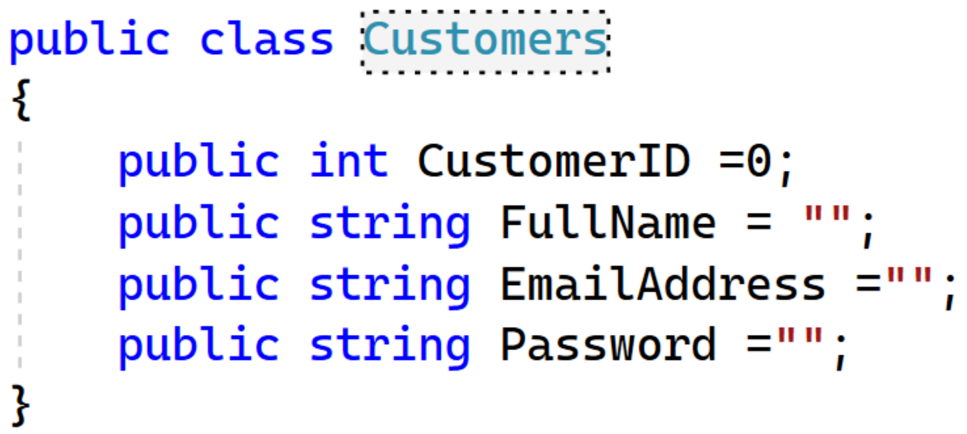
- Work with a database
- Controller actions and views
- Add search
- Add a new field
- Add validation
- Examine Details and Delete



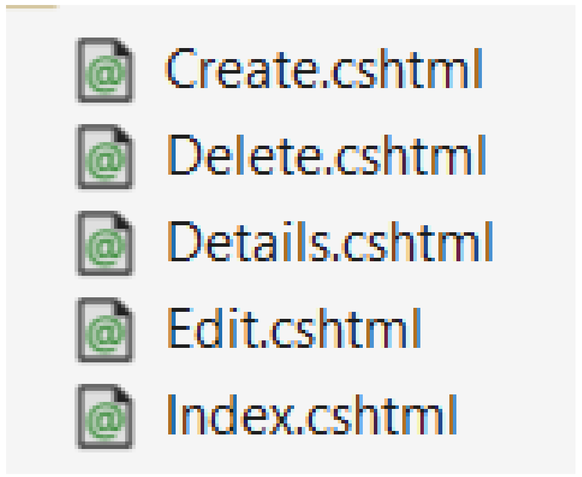
Thaop@neu.edu.vn

OBJECTIVES

- I. Model →
- II. Admin → MVC Model → Admin (CRUD)
- III. Member → Login→ EmailAddress and Password
  - → Check DB
  - Exists/Not
- IV. Guest→ Registration
  - Copy Template from Create.cshtml
  - Register → Add new to DB

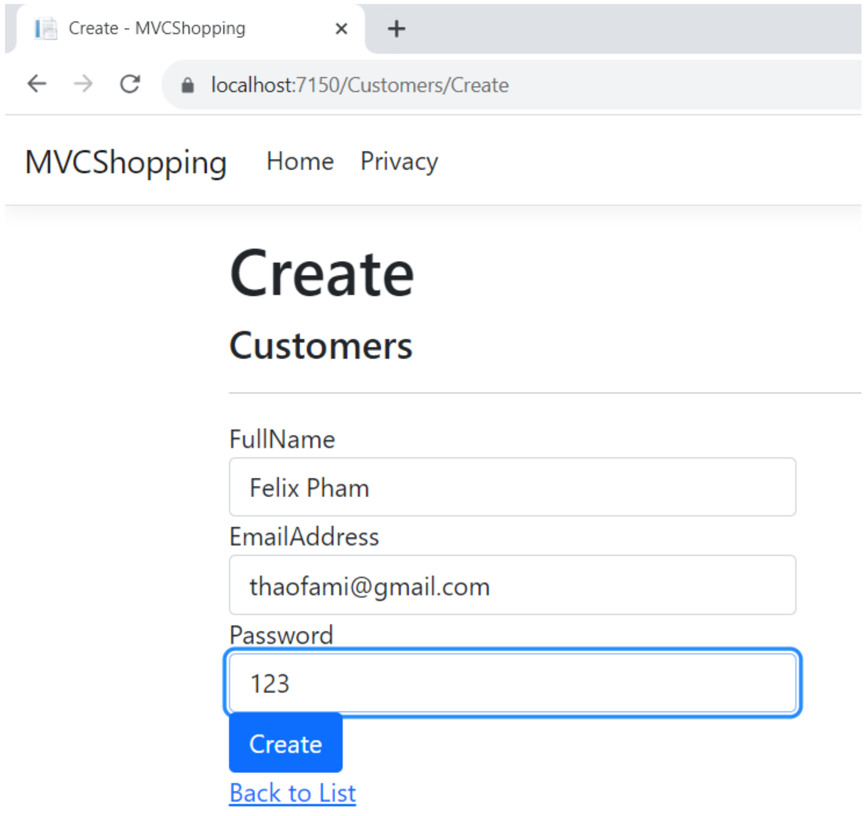


```
public class Customers
{
    public int CustomerID =0;
    public string FullName = "";
    public string EmailAddress ="";
    public string Password ="";
}
```



Create.cshtml
Delete.cshtml
Details.cshtml
Edit.cshtml
Index.cshtml

THE OBJECTIVES



Create - MVCShopping

localhost:7150/Customers/Create

MVCShopping Home Privacy

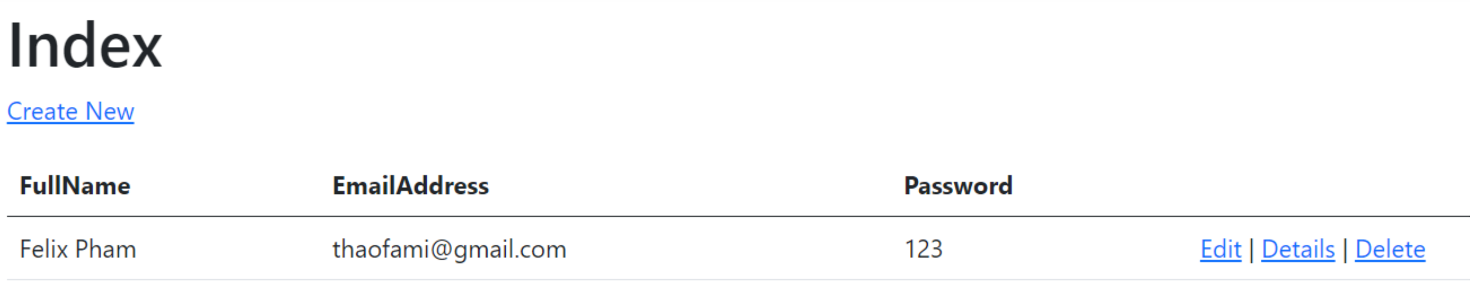
Create

Customers

FullName: Felix Pham

EmailAddress: thaofami@gmail.com

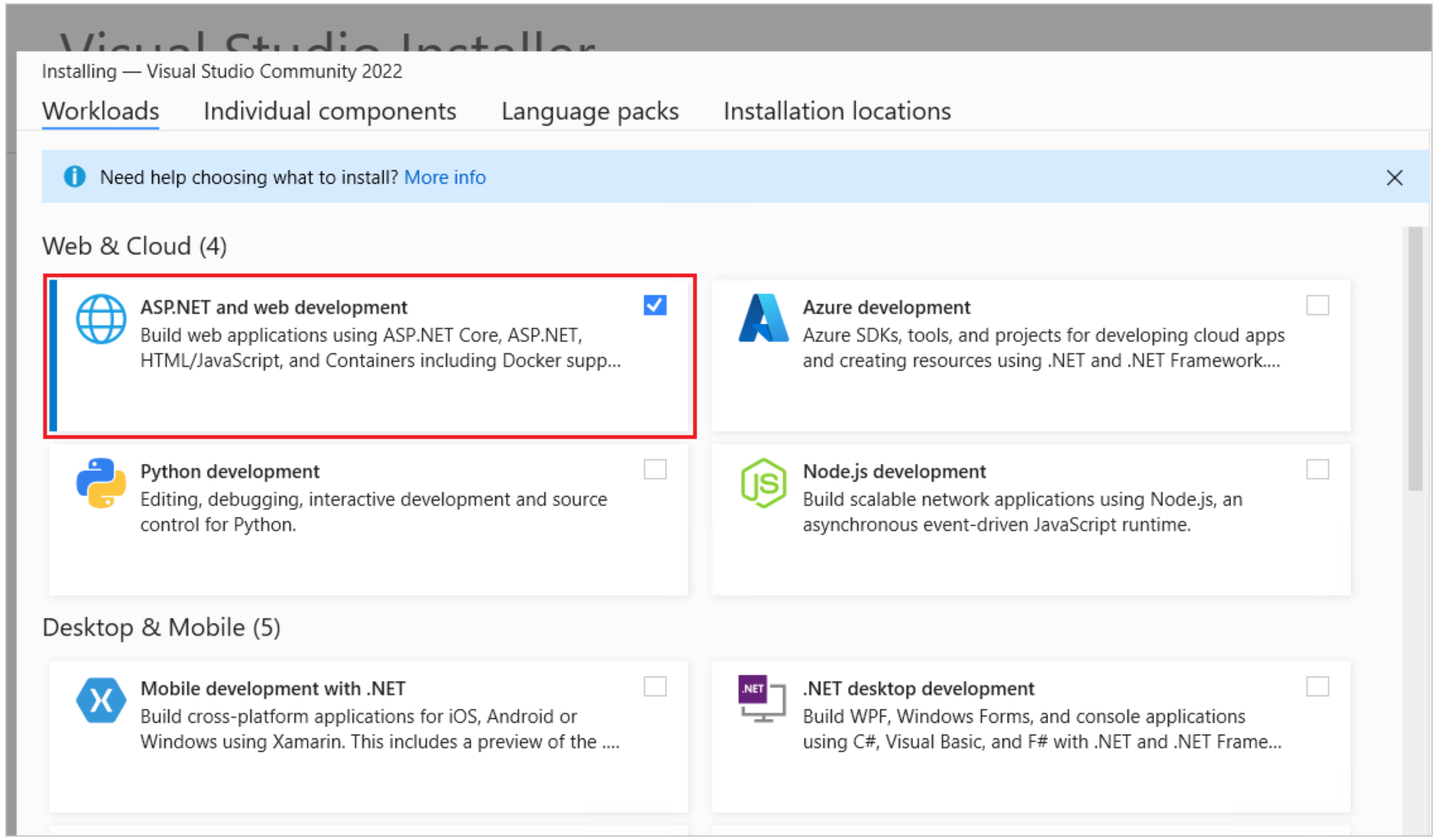
Password: 123



Index

FullName	EmailAddress	Password	
Felix Pham	thaofami@gmail.com	123	Edit | Details | Delete

SETUP ASP.NET AND WEB DEVELOPMENT


Installing — Visual Studio Community 2022

Workloads | Individual components | Language packs | Installation locations

Need help choosing what to install? More info

Web & Cloud (4)

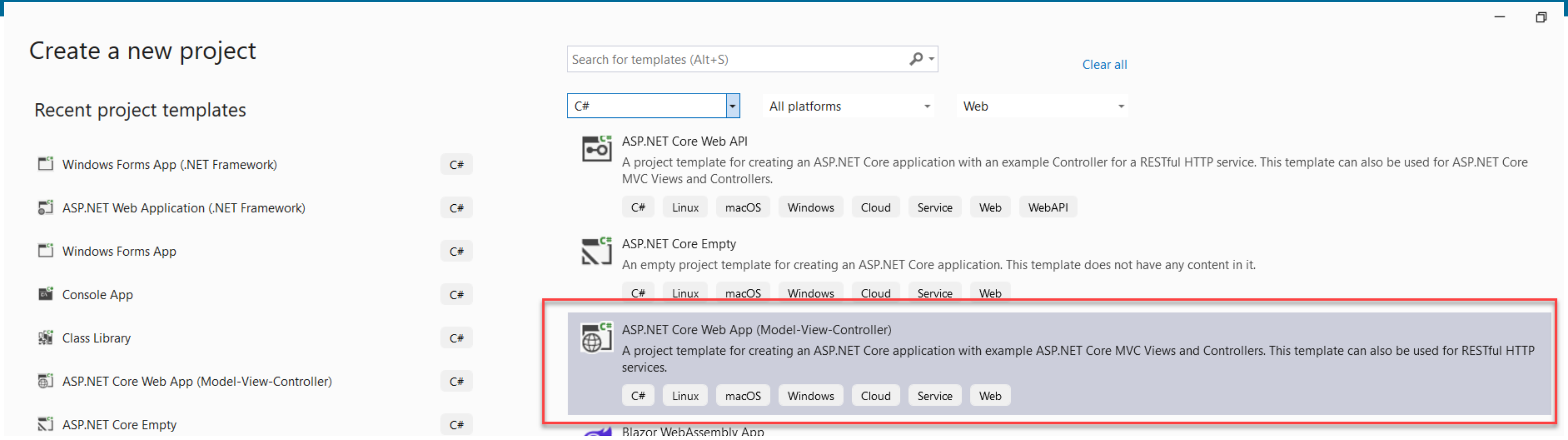
- [✓] **ASP.NET and web development** Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
- [ ] **Azure development** Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....
- [ ] **Python development** Editing, debugging, interactive development and source control for Python.
- [ ] **Node.js development** Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

- [ ] **Mobile development with .NET** Build cross-platform applications for iOS, Android or Windows using Xamarin. This includes a preview of the ....
- [ ] **.NET desktop development** Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-7.0&tabs=visual-studio

CREATE A NEW PROJECT



Create a new project

Recent project templates

Windows Forms App (.NET Framework)	C#
ASP.NET Web Application (.NET Framework)	C#
Windows Forms App	C#
Console App	C#
Class Library	C#
ASP.NET Core Web App (Model-View-Controller)	C#
ASP.NET Core Empty	C#

Search for templates (Alt+S)

Clear all

C# | All platforms | Web

ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
C# Linux macOS Windows Cloud Service Web WebAPI

ASP.NET Core Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
C# Linux macOS Windows Cloud Service Web

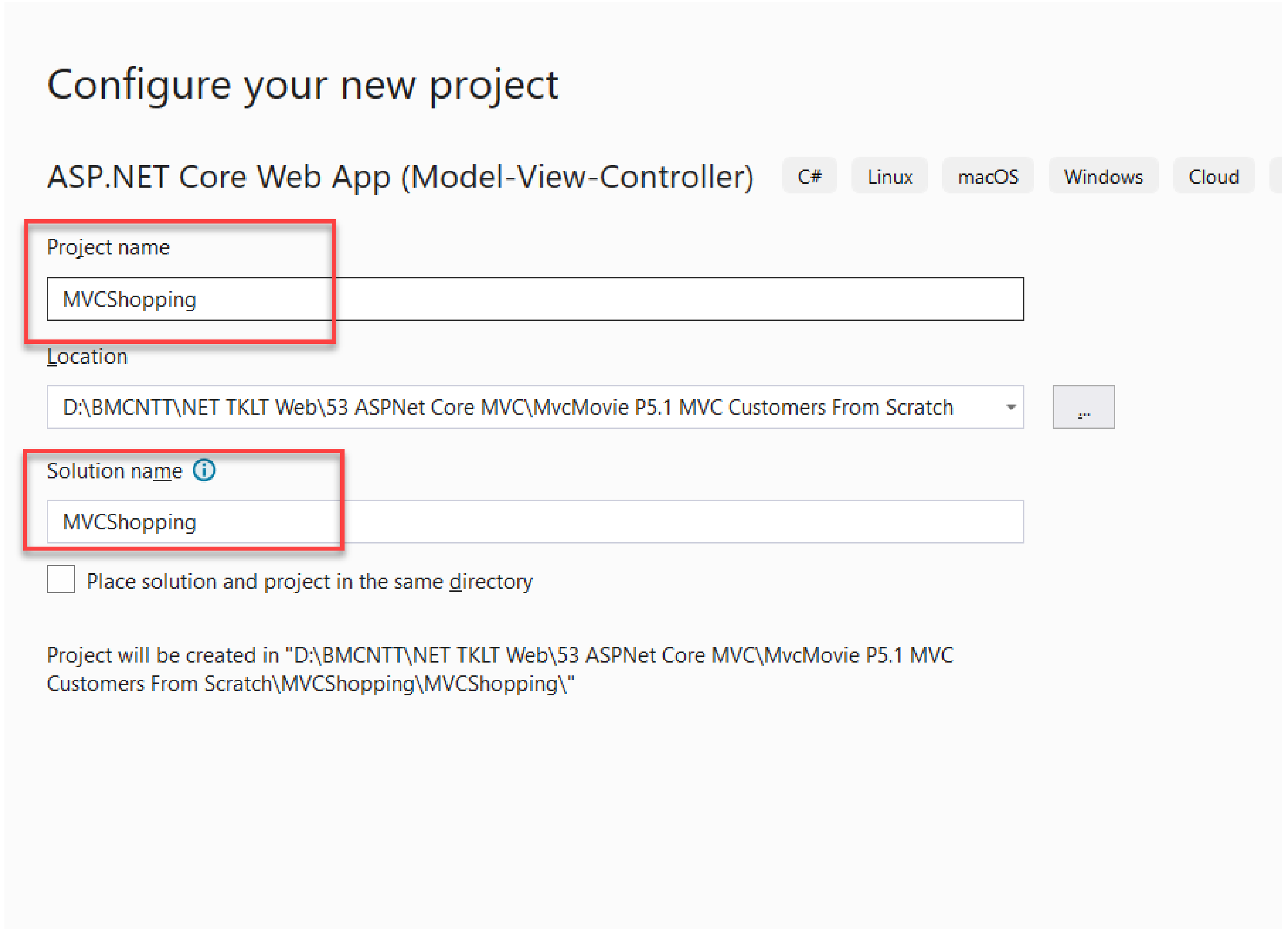
ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
C# Linux macOS Windows Cloud Service Web

Blazor WebAssembly App

- ASP.NET Core Web App (MVC):
  - Similar to the "ASP.NET Core Web App" template, but explicitly emphasizes the use of the Model-View-Controller (MVC) architectural pattern.
  - Provides a structured way to build web applications where data, presentation, and logic are separated into models, views, and controllers.
  - Suitable for developers familiar with MVC and who prefer to build applications following this pattern.
  - Offers a balanced approach between server-rendered views and API endpoints for data access.

CREATE A NEW PROJECT

- Start Visual Studio and select Create a new project.
- In the Create a new project dialog, select ASP.NET Core Web App (Model-View-Controller) > Next.
- In the Configure your new project dialog, enter MvcShopping for Project name.
- Select Next.



Configure your new project

ASP.NET Core Web App (Model-View-Controller)

Project name: MVCShopping

Location: D:\BMCNTT\NET TKLT Web\53 ASPNet Core MVC\MvcMovie P5.1 MVC Customers From Scratch

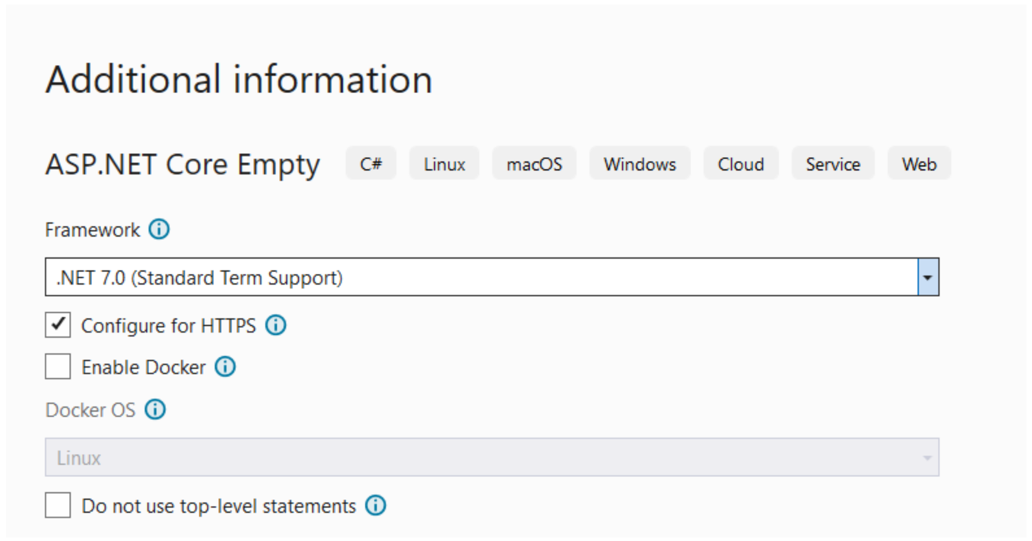
Solution name: MVCShopping

☐ Place solution and project in the same directory

Project will be created in "D:\BMCNTT\NET TKLT Web\53 ASPNet Core MVC\MvcMovie P5.1 MVC Customers From Scratch\MVCShopping\MVCShopping\"

CREATE A NEW PROJECT

- In the Additional information dialog:
- Select .NET 7.0.
- Verify that Do not use top-level statements is unchecked.
- Select Create.



Additional information

ASP.NET Core Empty   C#   Linux   macOS   Windows   Cloud   Service   Web

Framework ⓘ

.NET 7.0 (Standard Term Support)

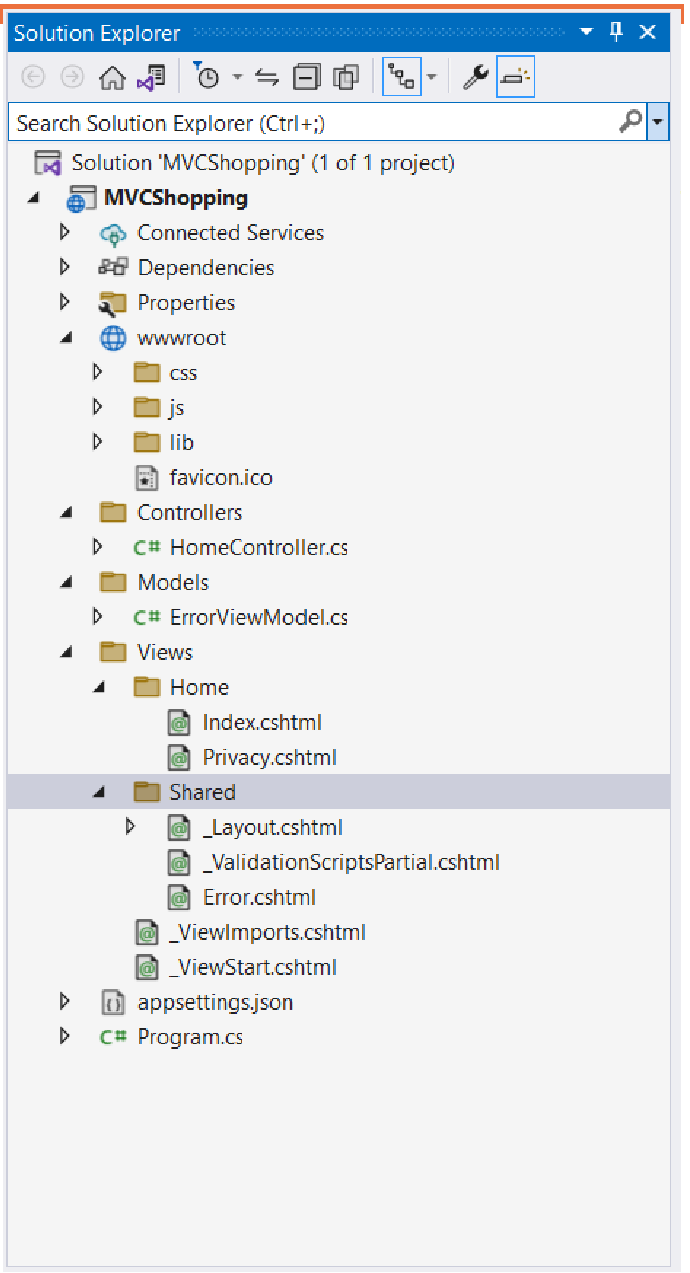
[✓] Configure for HTTPS ⓘ

[ ] Enable Docker ⓘ

Docker OS ⓘ

Linux

[ ] Do not use top-level statements ⓘ



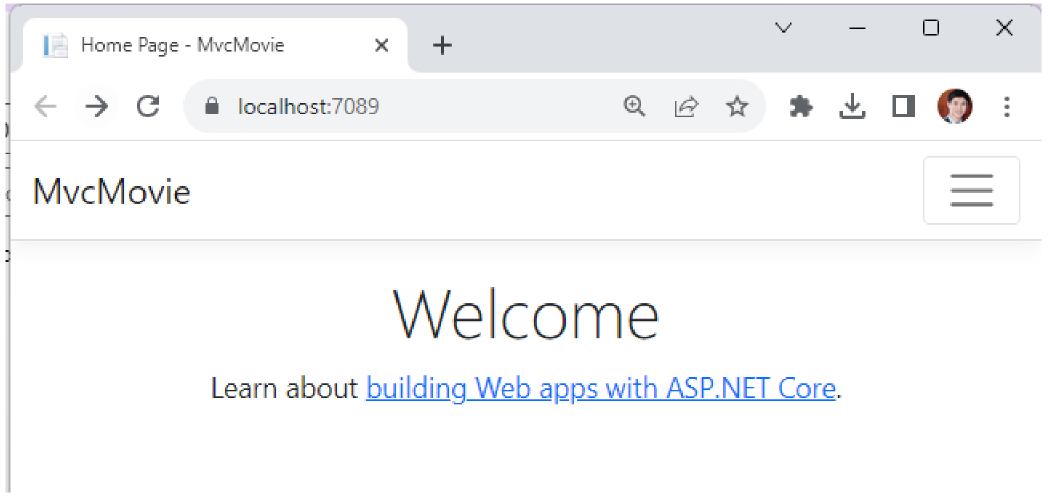
Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'MVCShopping' (1 of 1 project)
  MVCShopping
    Connected Services
    Dependencies
    Properties
    wwwroot
      css
      js
      lib
      favicon.ico
    Controllers
      HomeController.cs
    Models
      ErrorViewModel.cs
    Views
      Home
        Index.cshtml
        Privacy.cshtml
      Shared
        _Layout.cshtml
        _ValidationScriptsPartial.cshtml
        Error.cshtml
      _ViewImports.cshtml
      _ViewStart.cshtml
    appsettings.json
    Program.cs
```

CREATE A NEW PROJECT

- Select Ctrl+F5 to run the app without the debugger.
- Visual Studio displays the following dialog when a project is not yet configured to use SSL:

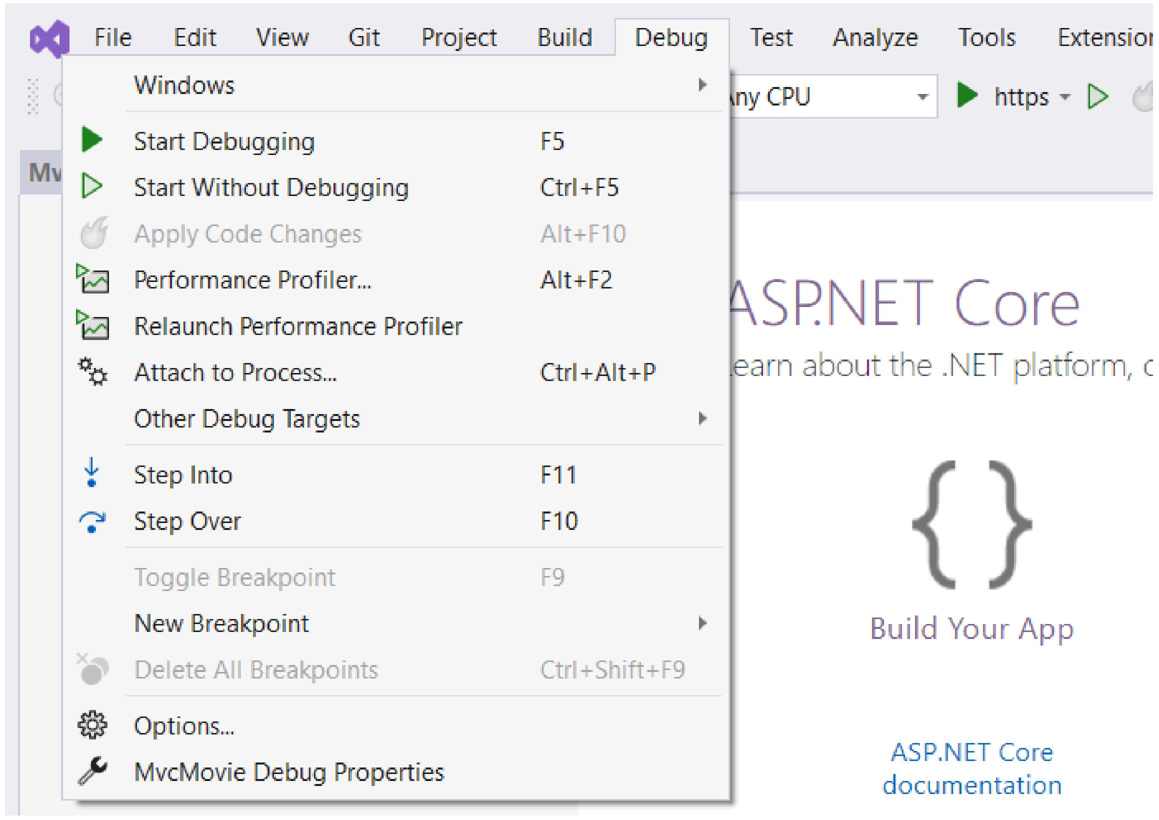

Home Page - MvcMovie

localhost:7089

MvcMovie

Welcome

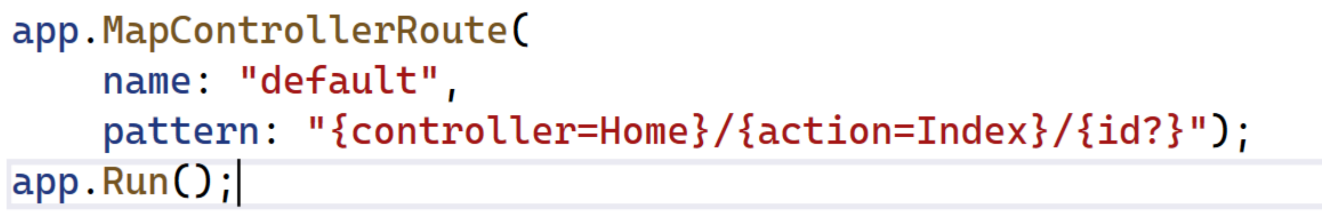
Learn about building Web apps with ASP.NET Core.


File Edit View Git Project Build Debug Test Analyze Tools Extensio

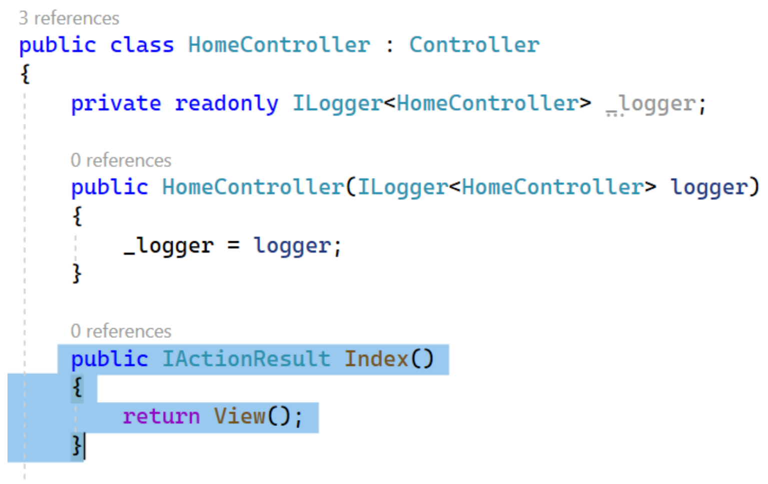
Windows	
Start Debugging	F5
Start Without Debugging	Ctrl+F5
Apply Code Changes	Alt+F10
Performance Profiler...	Alt+F2
Relaunch Performance Profiler	
Attach to Process...	Ctrl+Alt+P
Other Debug Targets	
Step Into	F11
Step Over	F10
Toggle Breakpoint	F9
New Breakpoint	
Delete All Breakpoints	Ctrl+Shift+F9
Options...	
MvcMovie Debug Properties	

CREATE A NEW PROJECT

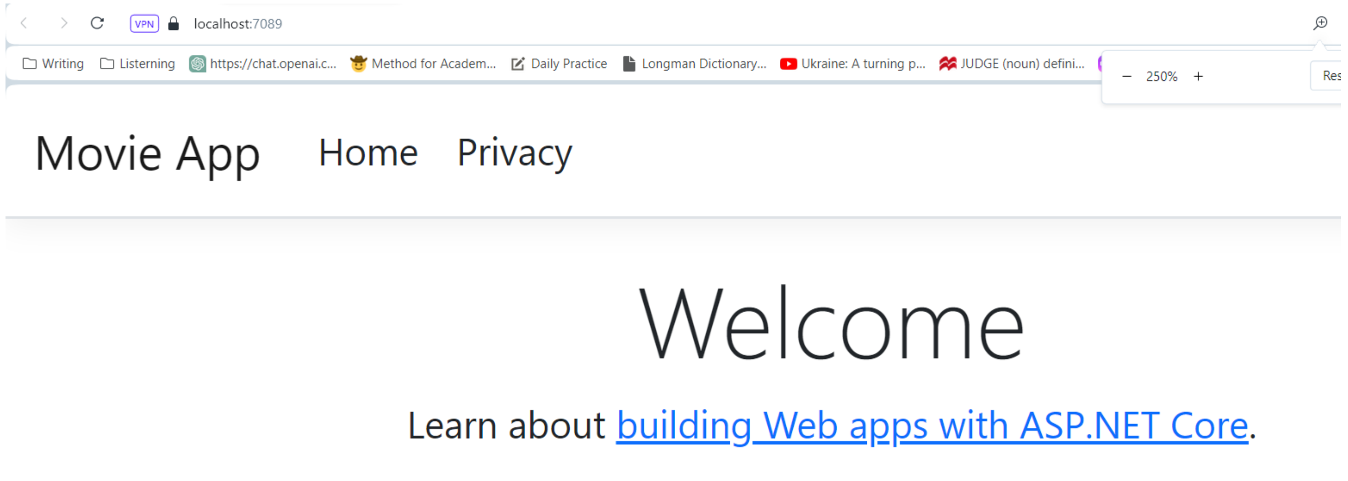
- Default
  - Home
  - Action: Index



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



```
3 references
public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;

    0 references
    public HomeController(ILogger<HomeController> logger)
    {
        _logger = logger;
    }

    0 references
    public IActionResult Index()
    {
        return View();
    }
```



Movie App Home Privacy

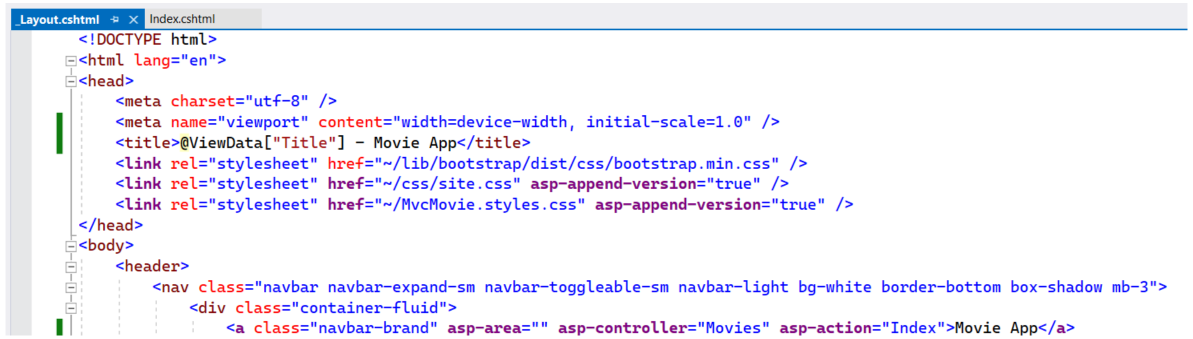
Welcome

Learn about building Web apps with ASP.NET Core.

Thaop@neu.edu.vn

ADD A VIEW - CHANGE VIEWS AND LAYOUT PAGES

- Default layout
- Open the Views/Shared/_Layout.cshtml file.

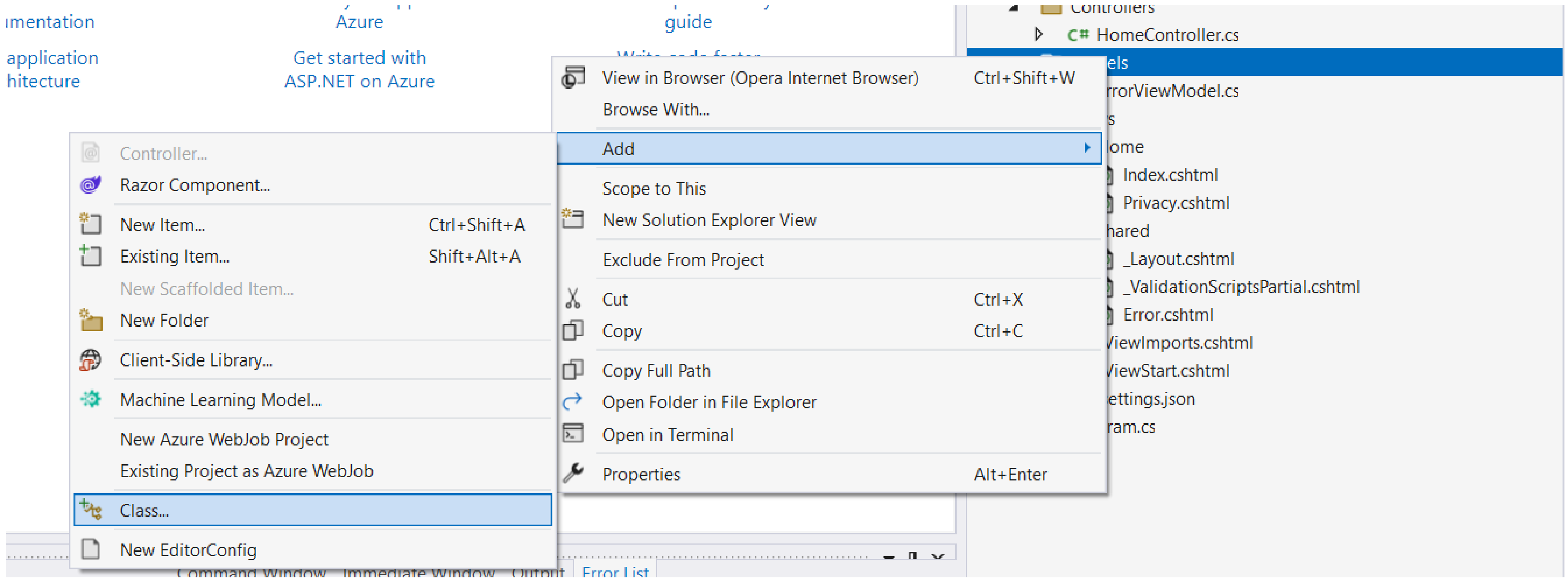


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - Movie App</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
    <link rel="stylesheet" href="~/MvcMovie.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>
```

- Layout templates allow:
  - Specifying the HTML container layout of a site in one place.
  - Applying the HTML container layout across multiple pages in the site.
  - Find the @RenderBody() line. RenderBody is a placeholder where all the view-specific pages you create show up, wrapped in the layout page. For example, if you select the Privacy link, the Views/Home/Privacy.cshtml view is rendered inside the RenderBody method.
- Change share layout

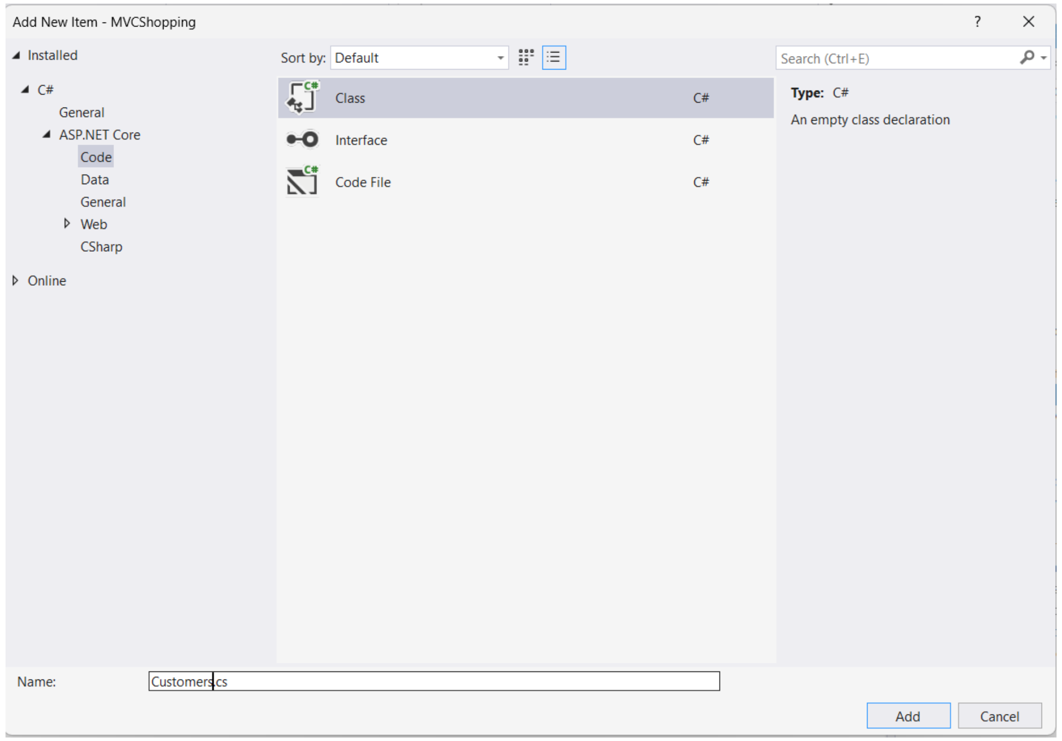
ADD MODEL

- Right-click the Models folder > Add > Class. Name the file Customers.cs.



ADD MODEL

- Right-click the Models folder > Add > Class. Name the file Customers.cs.



Add New Item - MVCShopping

Installed
- C#
  - General
  - ASP.NET Core
    - Code
    - Data
    - General
    - Web
    - CSharp
- Online

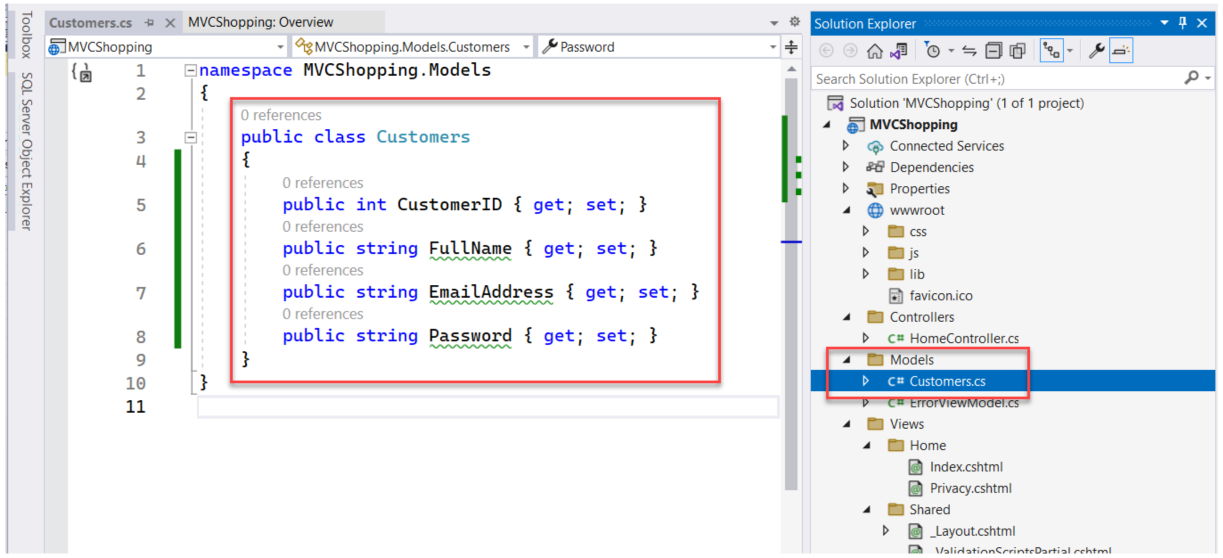
Sort by: Default

Class	C#
Interface	C#
Code File	C#

Type: C#
An empty class declaration

Name: Customers.cs

Add Cancel



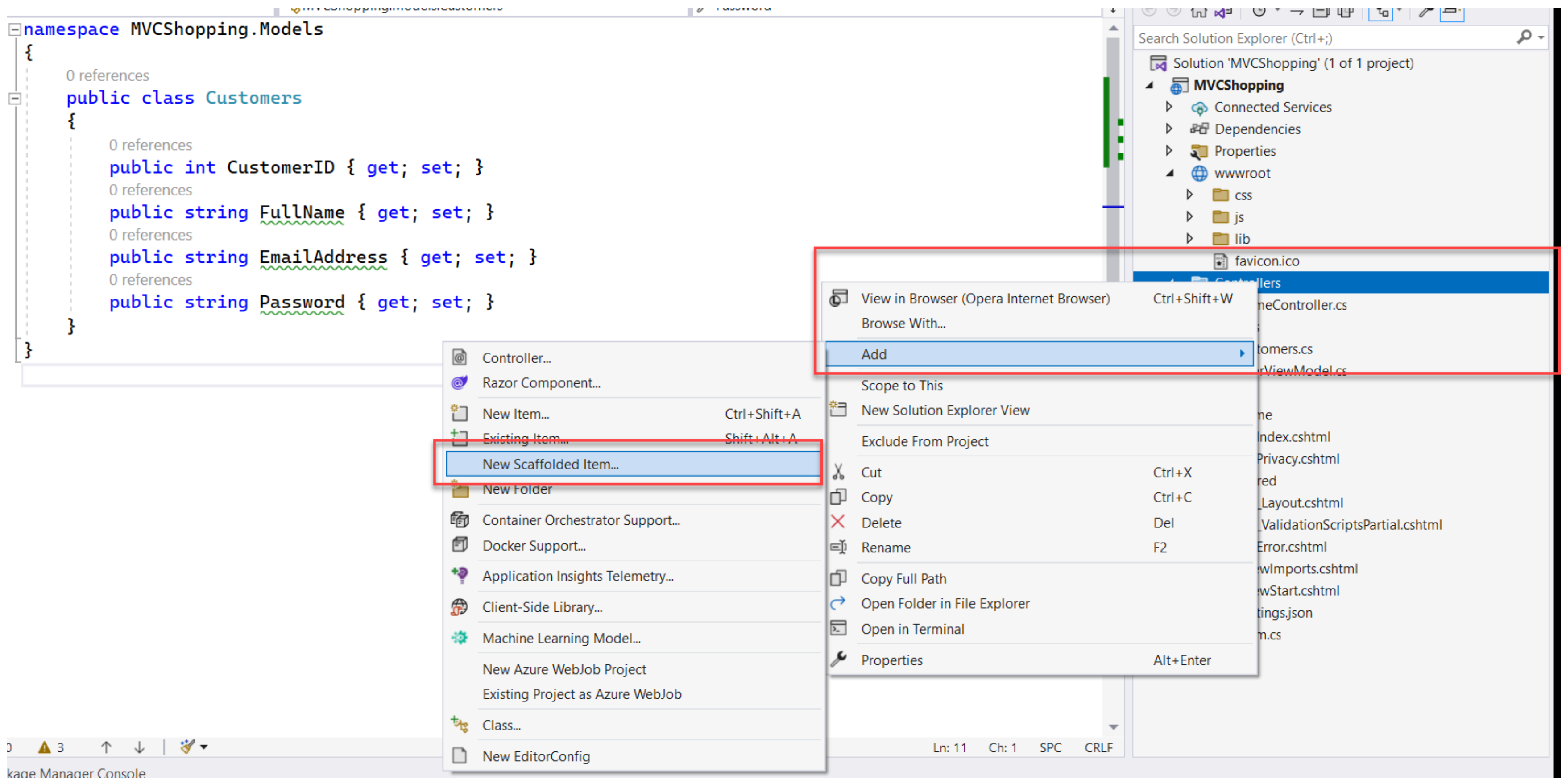
Customers.cs — MVCShopping: Overview
MVCShopping — MVCShopping.Models.Customers — Password

```
namespace MVCShopping.Models
{
    0 references
    public class Customers
    {
        0 references
        public int CustomerID { get; set; }
        0 references
        public string FullName { get; set; }
        0 references
        public string EmailAddress { get; set; }
        0 references
        public string Password { get; set; }
    }
}
```

Solution Explorer
Search Solution Explorer (Ctrl+;)
- Solution 'MVCShopping' (1 of 1 project)
  - MVCShopping
    - Connected Services
    - Dependencies
    - Properties
    - wwwroot
      - css
      - js
      - lib
      - favicon.ico
    - Controllers
      - HomeController.cs
    - Models
      - Customers.cs
      - ErrorViewModel.cs
    - Views
      - Home
        - Index.cshtml
        - Privacy.cshtml
      - Shared
        - _Layout.cshtml
        - _ValidationScriptsPartial.cshtml

SCAFFOLD CUSTOMER PAGES

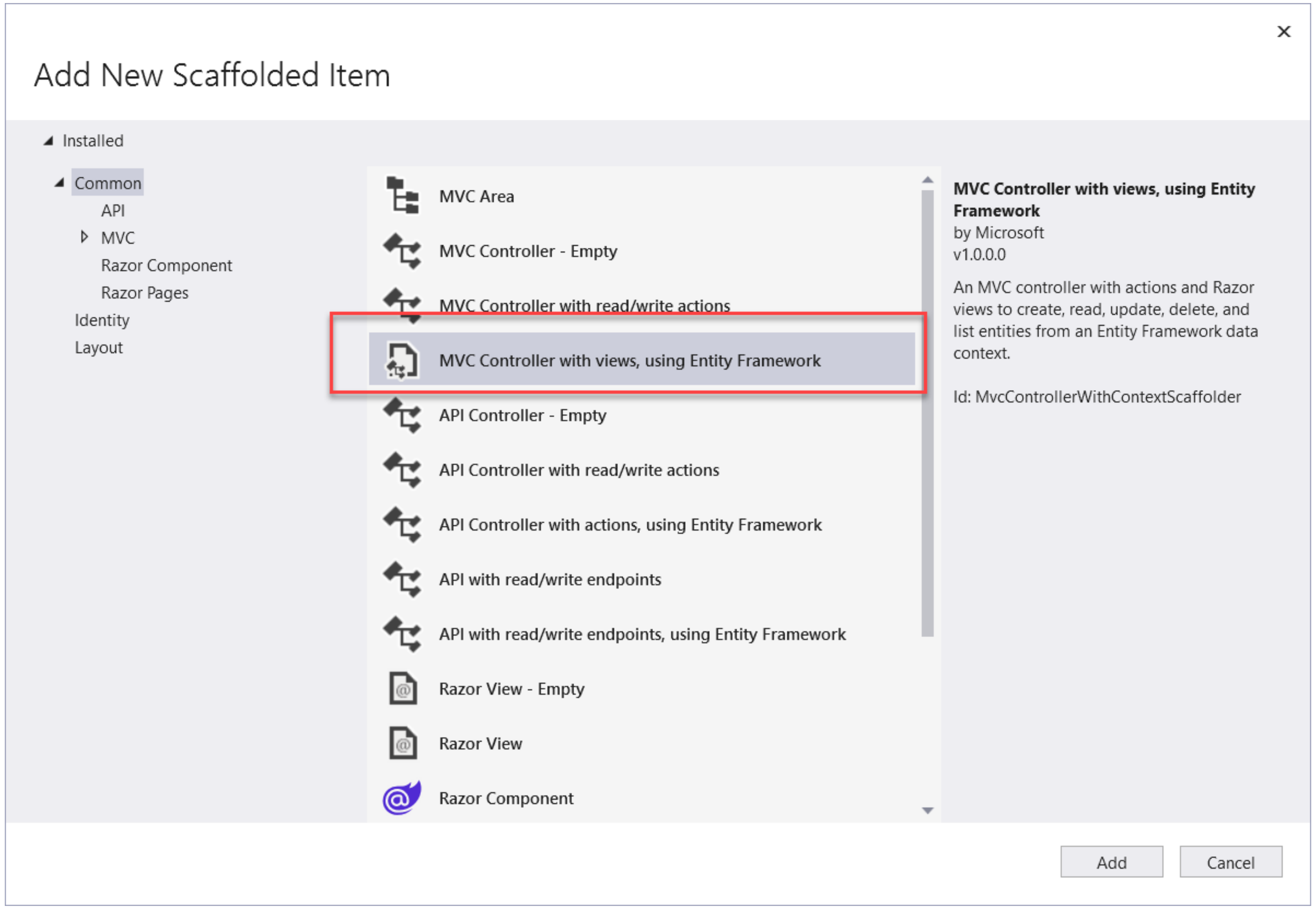
- Use the scaffolding tool to produce Create, Read, Update, and Delete (CRUD) pages for the movie model.
- In **Solution Explorer**, right-click the *Controllers* folder and select **Add > New Scaffolded Item**.



```
namespace MVCShopping.Models
{
    0 references
    public class Customers
    {
        0 references
        public int CustomerID { get; set; }
        0 references
        public string FullName { get; set; }
        0 references
        public string EmailAddress { get; set; }
        0 references
        public string Password { get; set; }
    }
}
```

SCAFFOLD CUSTOMER PAGES

- In the **Add New Scaffolded Item** dialog:
  - In the left pane, select **Installed** > **Com mon** > **MVC**.
  - Select **MVC Controller with views, using Entity Framework**.
  - Select **Add**.


Add New Scaffolded Item

- Installed
  - Common
    - API
    - MVC
    - Razor Component
    - Razor Pages
  - Identity
  - Layout

- MVC Area
- MVC Controller - Empty
- MVC Controller with read/write actions
- MVC Controller with views, using Entity Framework
- API Controller - Empty
- API Controller with read/write actions
- API Controller with actions, using Entity Framework
- API with read/write endpoints
- API with read/write endpoints, using Entity Framework
- Razor View - Empty
- Razor View
- Razor Component

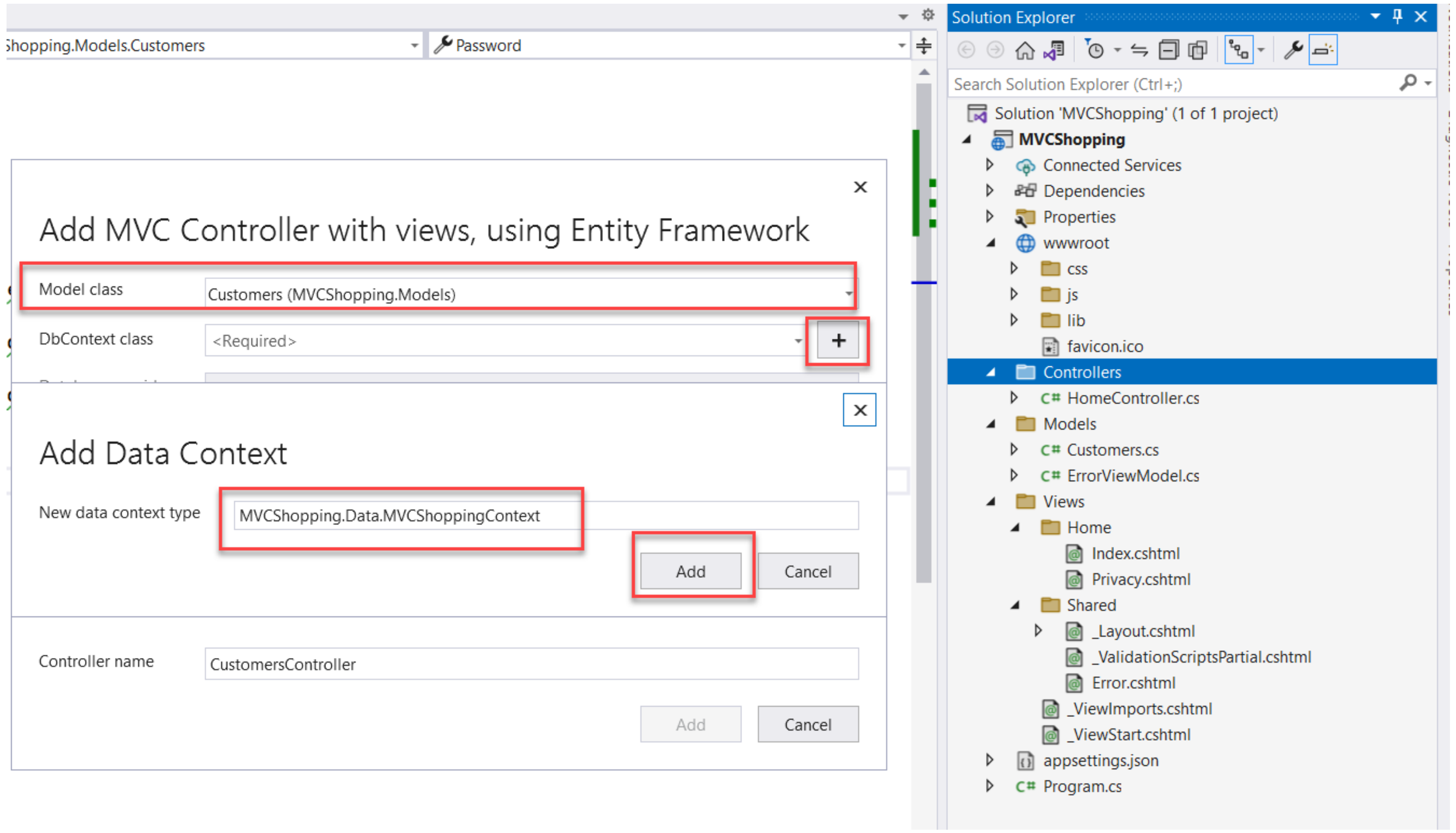
MVC Controller with views, using Entity Framework
by Microsoft
v1.0.0.0

An MVC controller with actions and Razor views to create, read, update, delete, and list entities from an Entity Framework data context.

Id: MvcControllerWithContextScaffolder

SCAFFOLD CUSTOMER PAGES

- Complete the **Add MVC Controller with views, using Entity Framework** dialog:
- In the **Model class** drop down, select **Movie (MvcMovie.Models)**.
- In the **Data context class** row, select the **+** (plus) sign.
  - In the **Add Data Context** dialog, the class name *MvcMovie.Data.MvcMovieContext* is generated.
  - Select **Add**.


Shopping.Models.Customers | Password

Add MVC Controller with views, using Entity Framework

Model class: Customers (MVCShopping.Models)

DbContext class: <Required>

Add Data Context

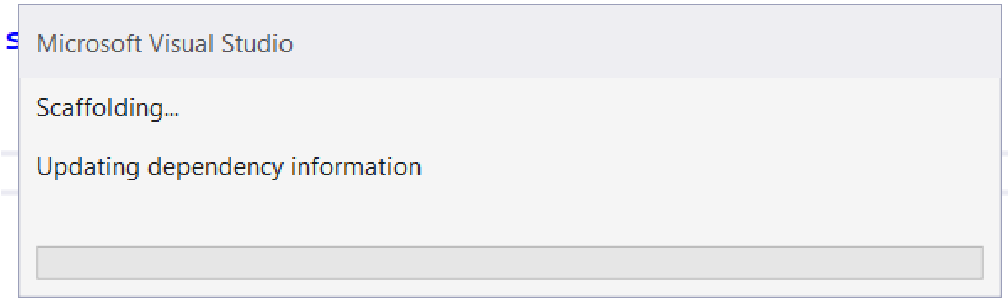
New data context type: MVCShopping.Data.MVCShoppingContext

Add | Cancel

Controller name: CustomersController

Add | Cancel

•In the **Database provider** drop down, select **SQL Server**.
•**Views** and **Controller name**: Keep the default.
•Select **Add**.

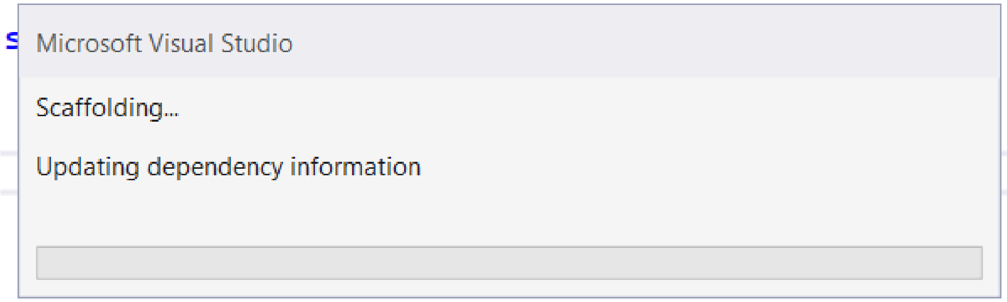

Microsoft Visual Studio

Scaffolding...

Updating dependency information

Thaop@neu.edu.vn

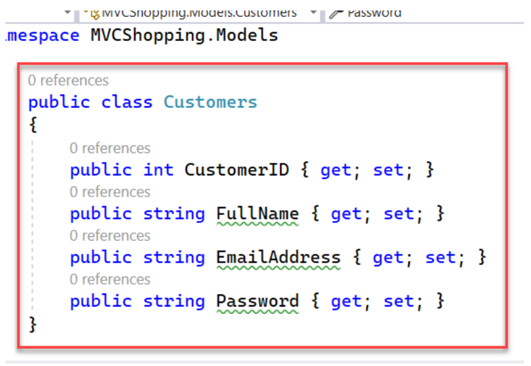
SCAFFOLD CUSTOMER PAGES



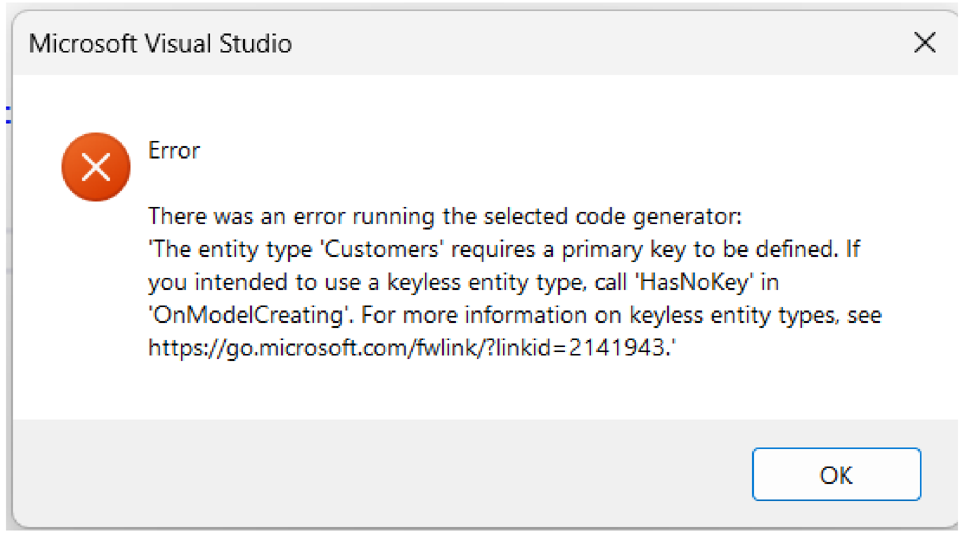
Microsoft Visual Studio

Scaffolding...

Updating dependency information



```
mespace MVCShopping.Models

0 references
public class Customers
{
    0 references
    public int CustomerID { get; set; }
    0 references
    public string FullName { get; set; }
    0 references
    public string EmailAddress { get; set; }
    0 references
    public string Password { get; set; }
}
```

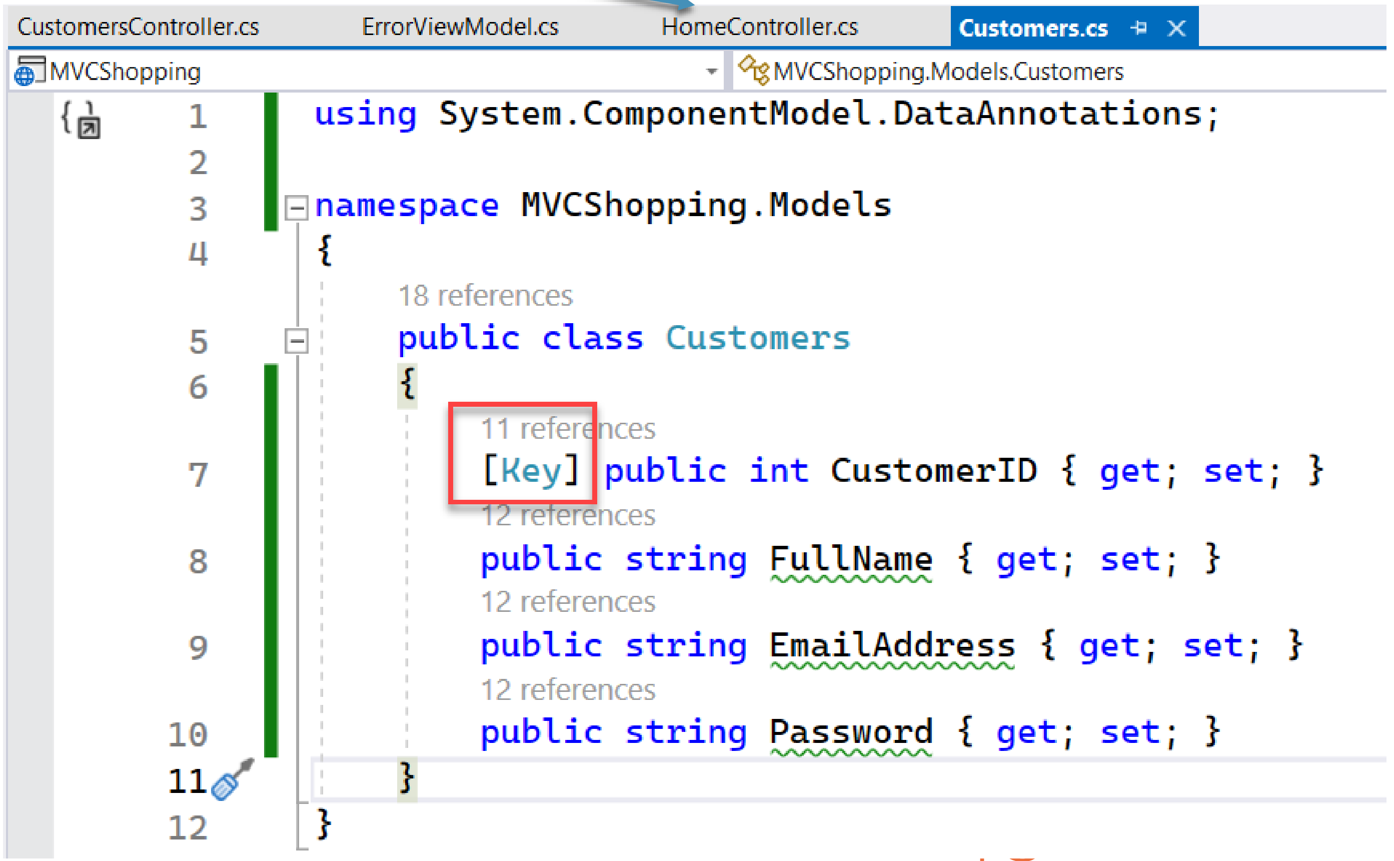


Microsoft Visual Studio

Error

There was an error running the selected code generator: 'The entity type 'Customers' requires a primary key to be defined. If you intended to use a keyless entity type, call 'HasNoKey' in 'OnModelCreating'. For more information on keyless entity types, see https://go.microsoft.com/fwlink/?linkid=2141943.'

OK

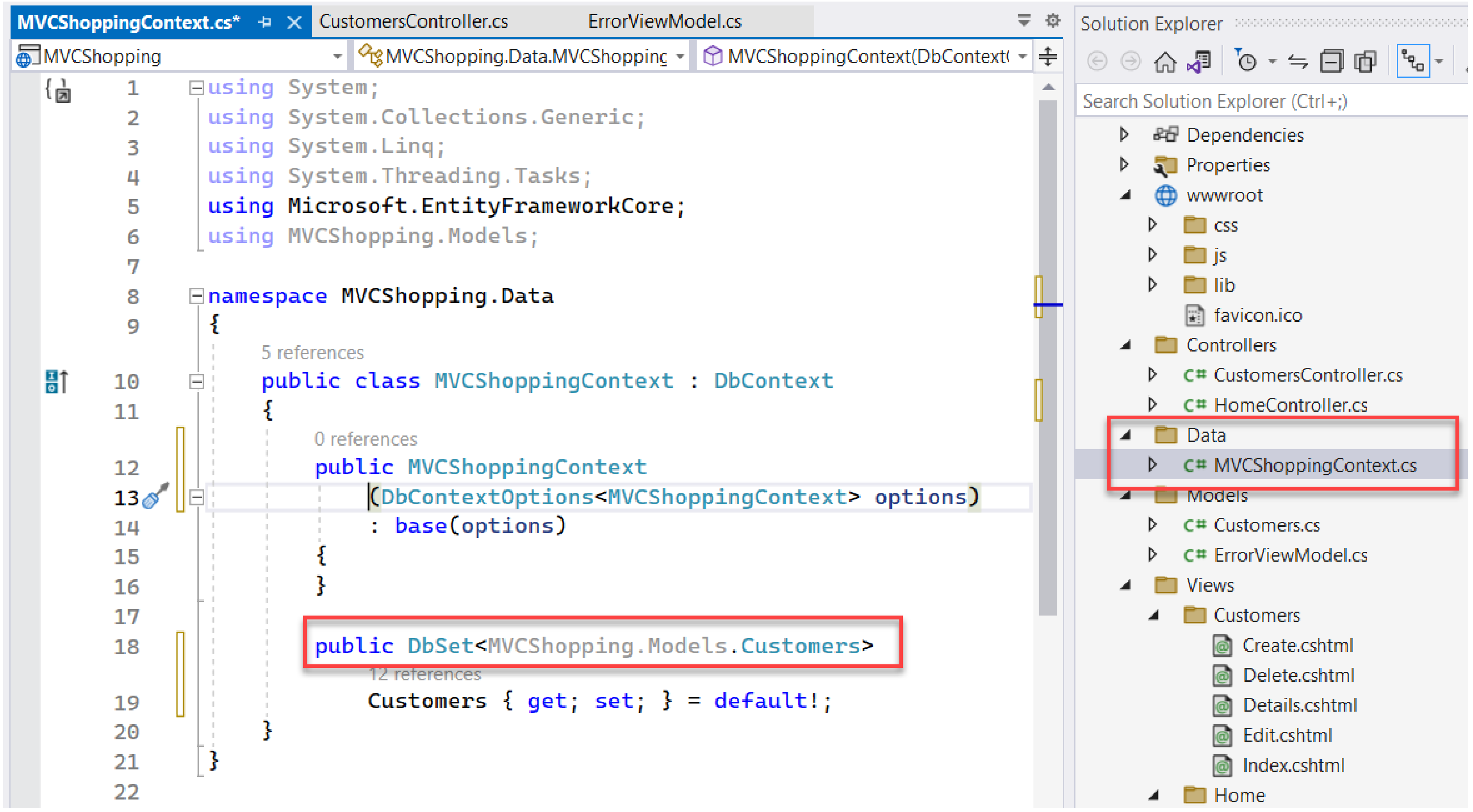


```
using System.ComponentModel.DataAnnotations;

namespace MVCShopping.Models
{
    18 references
    public class Customers
    {
        11 references
        [Key] public int CustomerID { get; set; }
        12 references
        public string FullName { get; set; }
        12 references
        public string EmailAddress { get; set; }
        12 references
        public string Password { get; set; }
    }
}
```

SCAFFOLD CUSTOMER PAGES

- Scaffolding adds the following **packages**:
  - Microsoft.EntityFrameworkCore.SqlServer
  - Microsoft.EntityFrameworkCore.Tools
  - Microsoft.VisualStudio.Web.CodeGeneration.Design
- Scaffolding creates the following:
  - A movies controller: Controllers/CustomersController.cs
  - Razor view files for Create, Delete, Details, Edit, and Index pages: Views/Customers/*.cshtml
  - A database context class: Data/MvcCustomersContext.cs
- Scaffolding updates the following:
  - Inserts required package references in the MvcMovie.csproj project file.
  - Registers the database context in the Program.cs file.
  - Adds a database connection string to the appsettings.json file.
  - The automatic creation of these files and file updates is known as scaffolding.
- However, the scaffolded pages can't be used yet because the database doesn't exist.
  - Running the app and selecting the Movie App link results in a Cannot open database or no such table: Movie error message.
  - Build the app to verify that there are no errors.

SCAFFOLD CUSTOMER PAGES

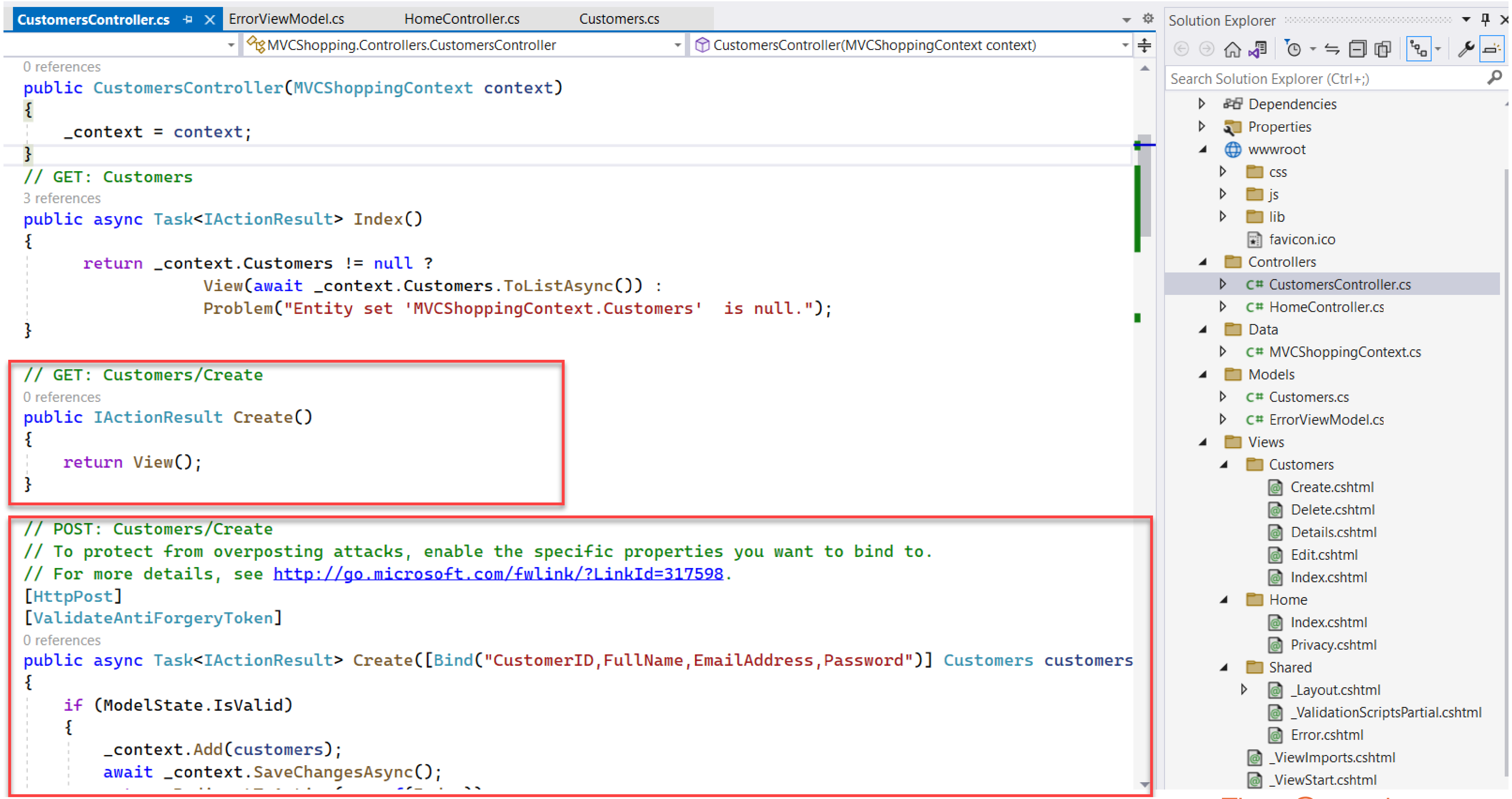


MVCShoppingContext.cs* | CustomersController.cs | ErrorViewModel.cs

MVCShopping | MVCShopping.Data.MVCShopping | MVCShoppingContext(DbContext

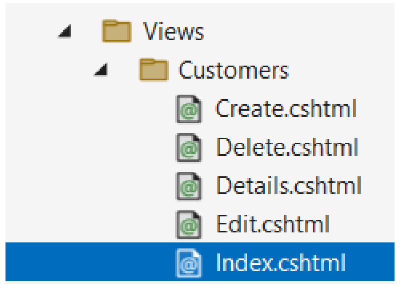
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.EntityFrameworkCore;
using MVCShopping.Models;

namespace MVCShopping.Data
{
    5 references
    public class MVCShoppingContext : DbContext
    {
        0 references
        public MVCShoppingContext
            (DbContextOptions<MVCShoppingContext> options)
            : base(options)
        {
        }

        public DbSet<MVCShopping.Models.Customers>
        12 references
            Customers { get; set; } = default!;
    }
}
```

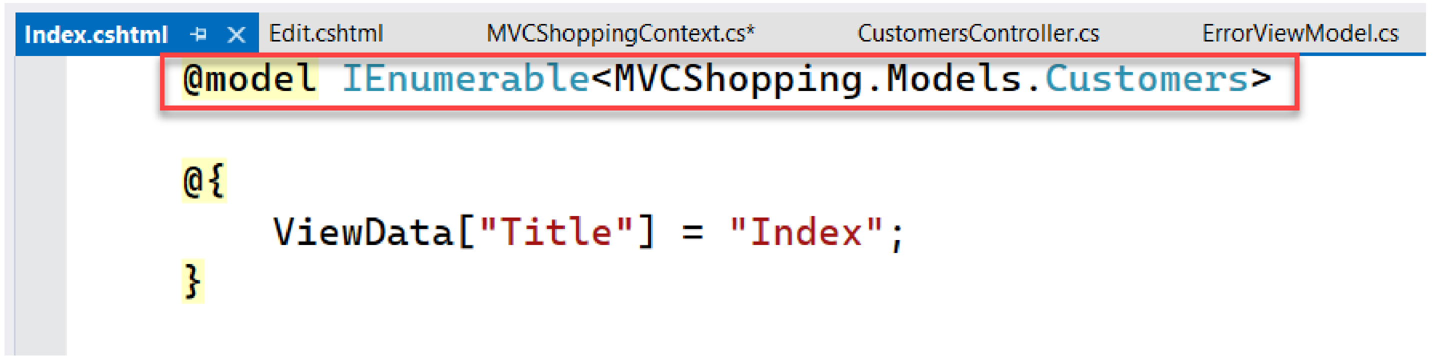
SCAFFOLD CUSTOMERS PAGES



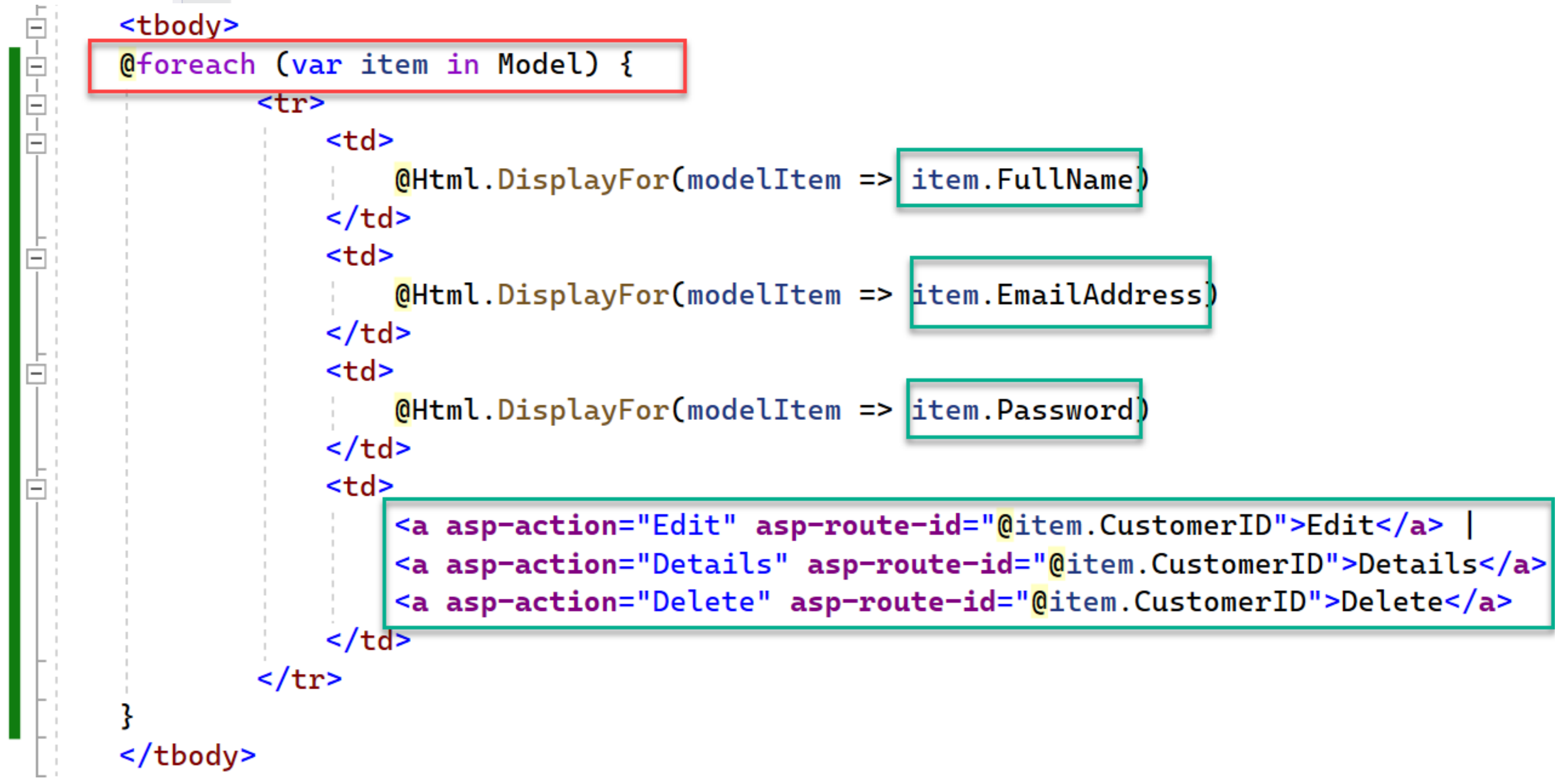
```
0 references
public CustomersController(MVCShoppingContext context)
{
    _context = context;
}
// GET: Customers
3 references
public async Task<IActionResult> Index()
{
      return _context.Customers != null ?
                  View(await _context.Customers.ToListAsync()) :
                  Problem("Entity set 'MVCShoppingContext.Customers'  is null.");
}

// GET: Customers/Create
0 references
public IActionResult Create()
{
    return View();
}

// POST: Customers/Create
// To protect from overposting attacks, enable the specific properties you want to bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create([Bind("CustomerID,FullName,EmailAddress,Password")] Customers customers
{
    if (ModelState.IsValid)
    {
        _context.Add(customers);
        await _context.SaveChangesAsync();
```

SCAFFOLD CUSTOMERS PAGES



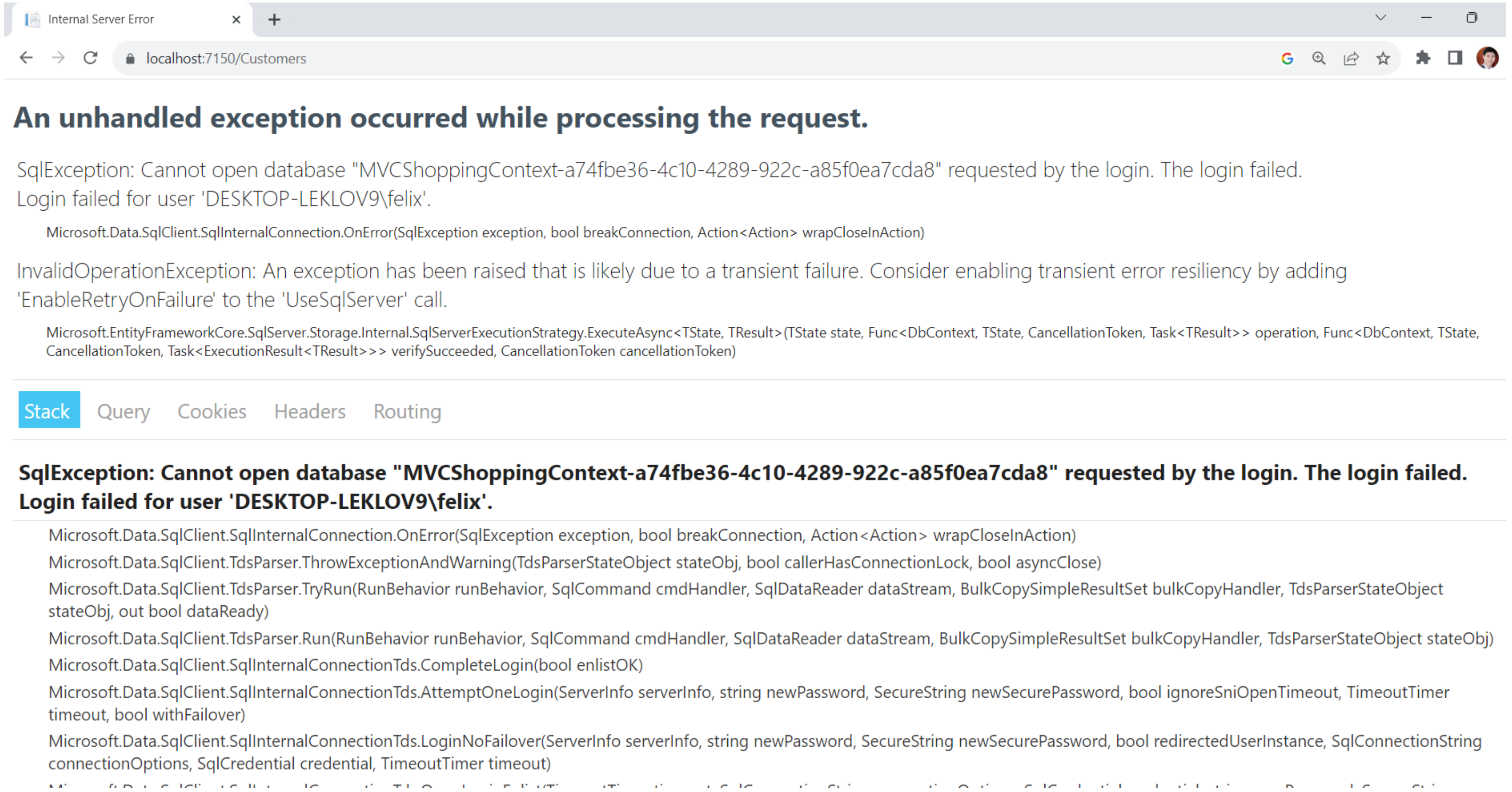


```
@model IEnumerable<MVCShopping.Models.Customers>

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



```
<tbody>
@foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.FullName)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.EmailAddress)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Password)
            </td>
            <td>
                <a asp-action="Edit" asp-route-id="@item.CustomerID">Edit</a> |
                <a asp-action="Details" asp-route-id="@item.CustomerID">Details</a>
                <a asp-action="Delete" asp-route-id="@item.CustomerID">Delete</a>
            </td>
        </tr>
}
</tbody>
```

TEST



An unhandled exception occurred while processing the request.

SqlException: Cannot open database "MVCShoppingContext-a74fbe36-4c10-4289-922c-a85f0ea7cda8" requested by the login. The login failed.
Login failed for user 'DESKTOP-LEKLOV9\felix'.

```
Microsoft.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, bool breakConnection, Action<Action> wrapCloseInAction)
```

InvalidOperationException: An exception has been raised that is likely due to a transient failure. Consider enabling transient error resiliency by adding 'EnableRetryOnFailure' to the 'UseSqlServer' call.

```
Microsoft.EntityFrameworkCore.SqlServer.Storage.Internal.SqlServerExecutionStrategy.ExecuteAsync<TState, TResult>(TState state, Func<DbContext, TState, CancellationToken, Task<TResult>> operation, Func<DbContext, TState, CancellationToken, Task<ExecutionResult<TResult>>> verifySucceeded, CancellationToken cancellationToken)
```

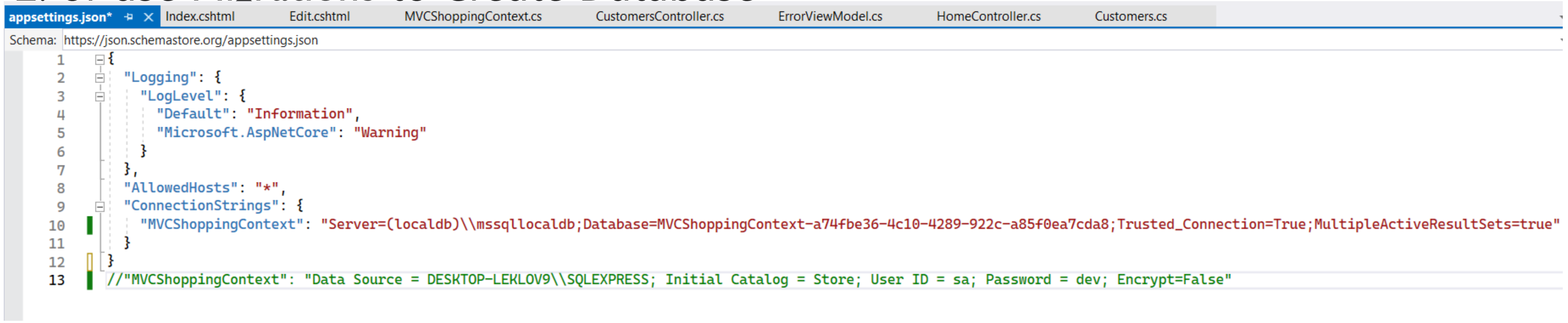
SqlException: Cannot open database "MVCShoppingContext-a74fbe36-4c10-4289-922c-a85f0ea7cda8" requested by the login. The login failed. Login failed for user 'DESKTOP-LEKLOV9\felix'.

```
Microsoft.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, bool breakConnection, Action<Action> wrapCloseInAction)
Microsoft.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, bool callerHasConnectionLock, bool asyncClose)
Microsoft.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj, out bool dataReady)
Microsoft.Data.SqlClient.TdsParser.Run(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj)
Microsoft.Data.SqlClient.SqlInternalConnectionTds.CompleteLogin(bool enlistOK)
Microsoft.Data.SqlClient.SqlInternalConnectionTds.AttemptOneLogin(ServerInfo serverInfo, string newPassword, SecureString newSecurePassword, bool ignoreSniOpenTimeout, TimeoutTimer timeout, bool withFailover)
Microsoft.Data.SqlClient.SqlInternalConnectionTds.LoginNoFailover(ServerInfo serverInfo, string newPassword, SecureString newSecurePassword, bool redirectedUserInstance, SqlConnectionString connectionOptions, SqlCredential credential, TimeoutTimer timeout)
```

- 1. Change Connect string to existing database or
- 2. use Migration to Create Database

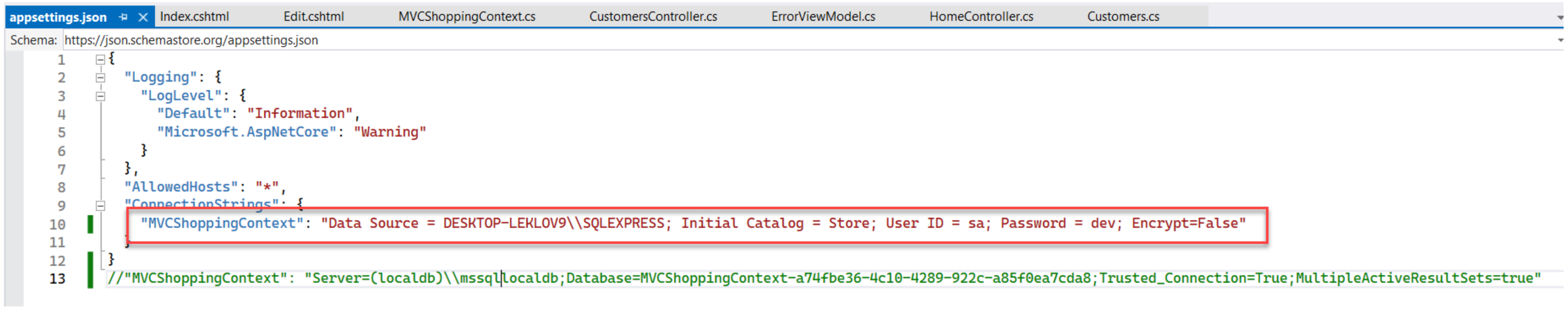
TEST

- 1. Change Connect string to existing database
  - `"Data Source = DESKTOP-LEKLOV9\\SQLEXPRESS; Initial Catalog = Store; User ID = sa; Password = dev; Encrypt=False"`
- 2. or use Migrations to Create Database



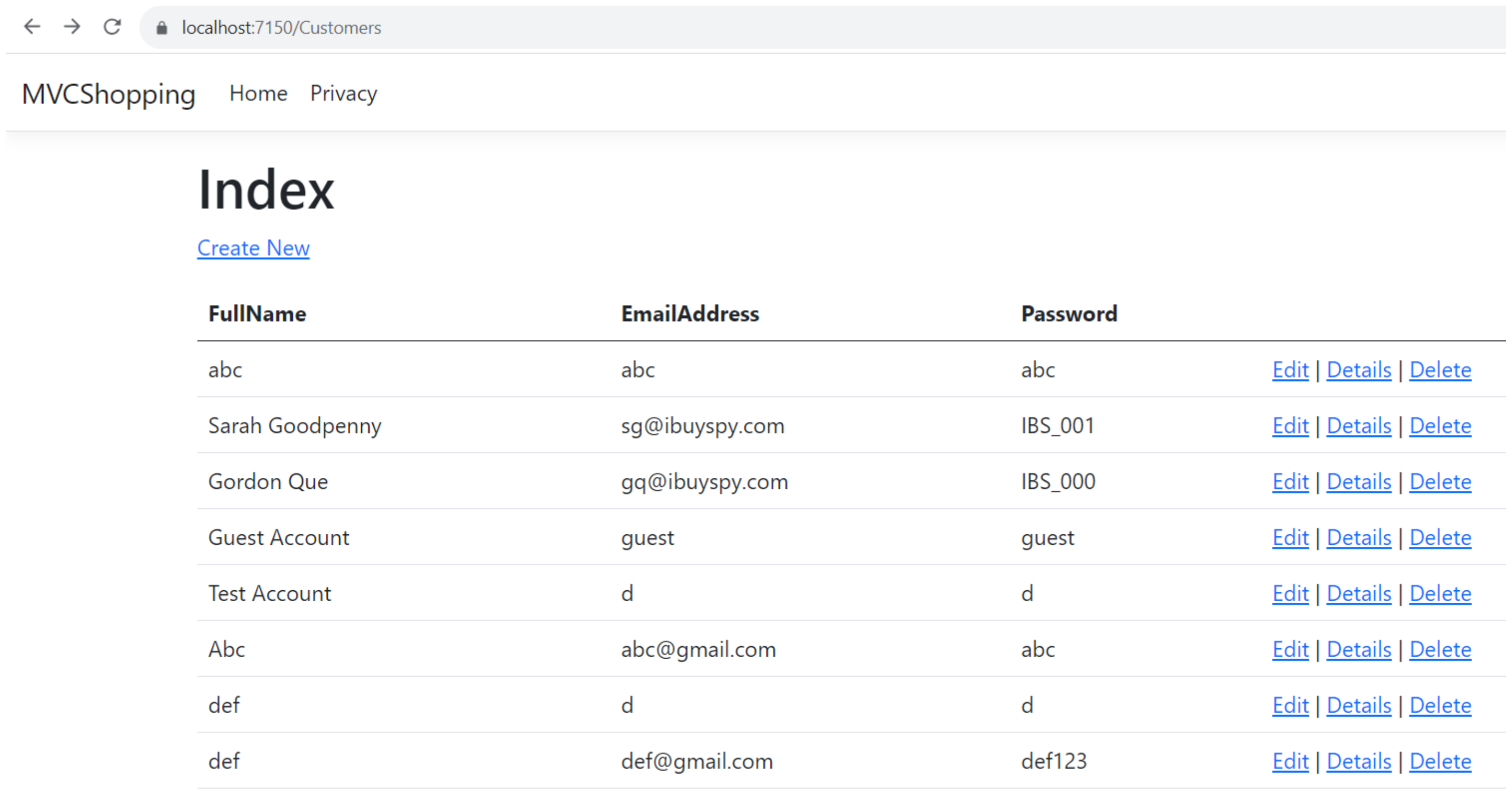
Schema: https://json.schemastore.org/appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "MVCShoppingContext": "Server=(localdb)\\mssqllocaldb;Database=MVCShoppingContext-a74fbe36-4c10-4289-922c-a85f0ea7cda8;Trusted_Connection=True;MultipleActiveResultSets=true"
  }
}
//"MVCShoppingContext": "Data Source = DESKTOP-LEKLOV9\\SQLEXPRESS; Initial Catalog = Store; User ID = sa; Password = dev; Encrypt=False"
```



Schema: https://json.schemastore.org/appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "MVCShoppingContext": "Data Source = DESKTOP-LEKLOV9\\SQLEXPRESS; Initial Catalog = Store; User ID = sa; Password = dev; Encrypt=False"
  }
}
//"MVCShoppingContext": "Server=(localdb)\\mssqllocaldb;Database=MVCShoppingContext-a74fbe36-4c10-4289-922c-a85f0ea7cda8;Trusted_Connection=True;MultipleActiveResultSets=true"
```



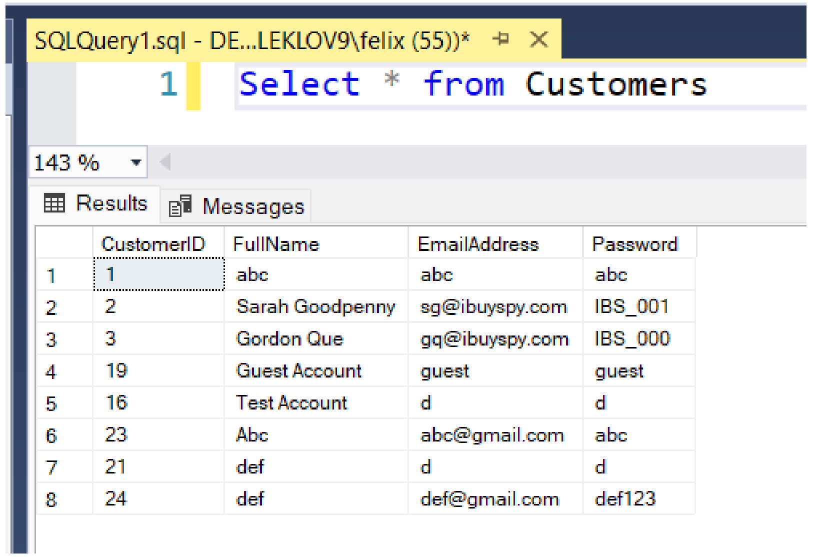
localhost:7150/Customers

MVCShopping Home Privacy

Index

Create New

FullName	EmailAddress	Password	
abc	abc	abc	Edit | Details | Delete
Sarah Goodpenny	sg@ibuyspy.com	IBS_001	Edit | Details | Delete
Gordon Que	gq@ibuyspy.com	IBS_000	Edit | Details | Delete
Guest Account	guest	guest	Edit | Details | Delete
Test Account	d	d	Edit | Details | Delete
Abc	abc@gmail.com	abc	Edit | Details | Delete
def	d	d	Edit | Details | Delete
def	def@gmail.com	def123	Edit | Details | Delete



SQLQuery1.sql - DE...LEKLOV9\felix (55))*

```sql
Select * from Customers
```

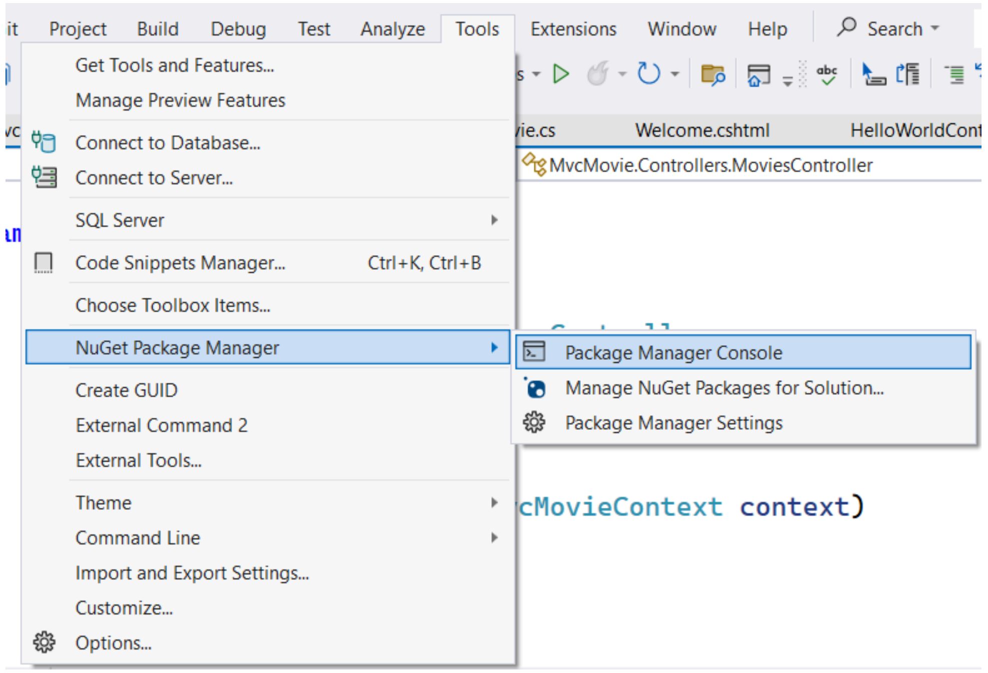
143 %

Results Messages

	CustomerID	FullName	EmailAddress	Password
1	1	abc	abc	abc
2	2	Sarah Goodpenny	sg@ibuyspy.com	IBS_001
3	3	Gordon Que	gq@ibuyspy.com	IBS_000
4	19	Guest Account	guest	guest
5	16	Test Account	d	d
6	23	Abc	abc@gmail.com	abc
7	21	def	d	d
8	24	def	def@gmail.com	def123

INITIAL MIGRATIONS

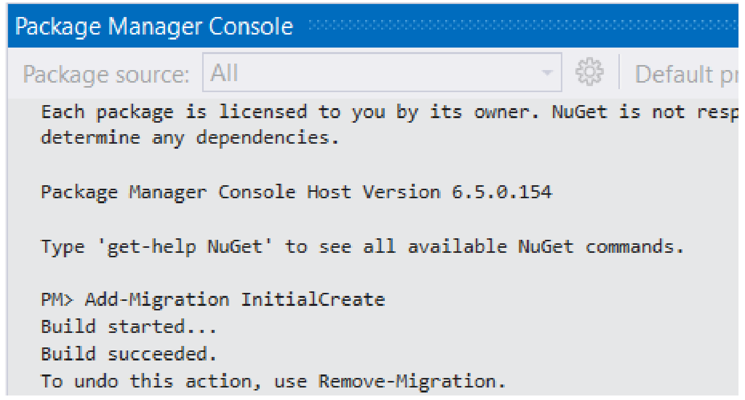
- Use the EF Core Migrations feature to create the database. *Migrations* is a set of tools that create and update a database to match the data model.
- From the Tools menu, select **NuGet Package Manager** > **Package Manager Console** .



In the Package Manager Console (PMC), enter the following commands:

Add-Migration InitialCreate

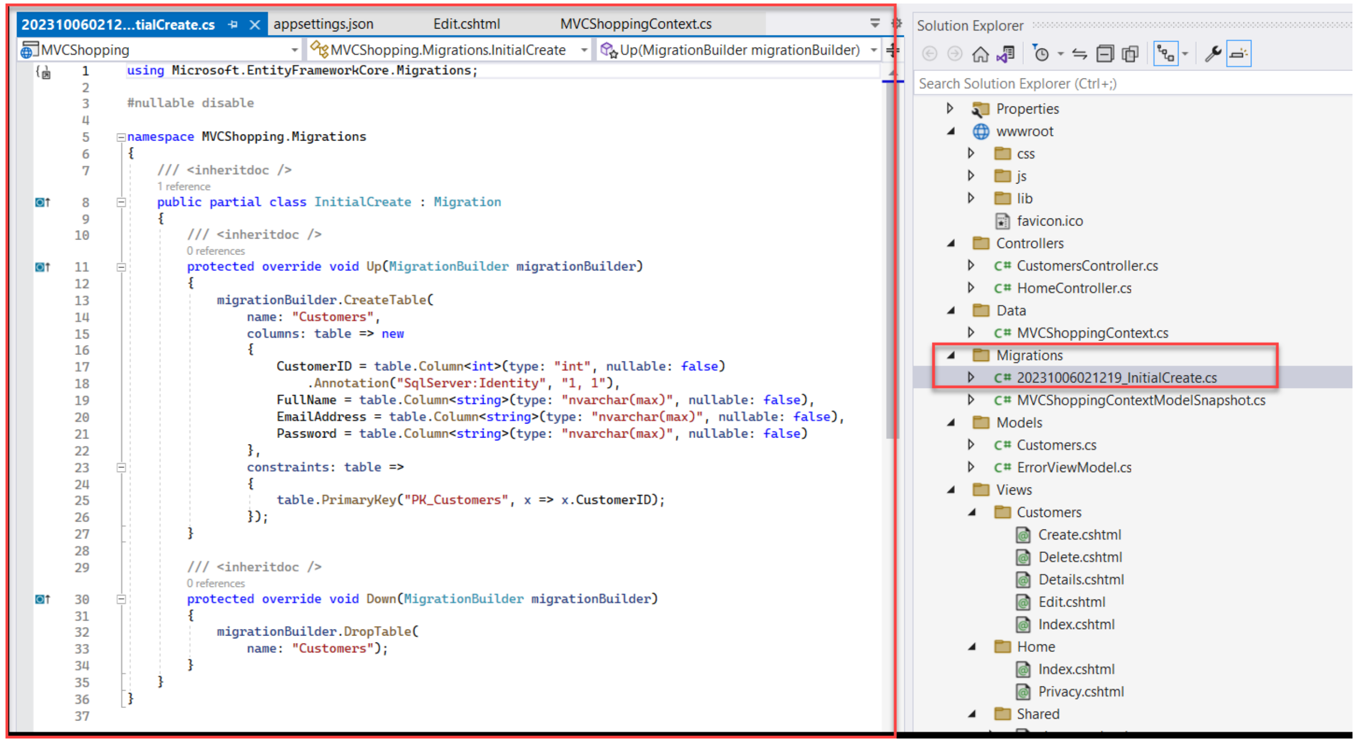
Update-Database



```
Each package is licensed to you by its owner. NuGet is not resp
determine any dependencies.

Package Manager Console Host Version 6.5.0.154

Type 'get-help NuGet' to see all available NuGet commands.

PM> Add-Migration InitialCreate
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
```

INITIAL MIGRATIONS

- **Add-Migration InitialCreate:**
  - Generates a Migrations/{timestamp}_InitialCreate.cs migration file.
  - The InitialCreate argument is the migration name. Any name can be used, but by convention, a name is selected that describes the migration.
  - Because this is the first migration, the generated class contains code to create the database schema.
- The database schema is based on the model specified in the MvcCustomerContext class.

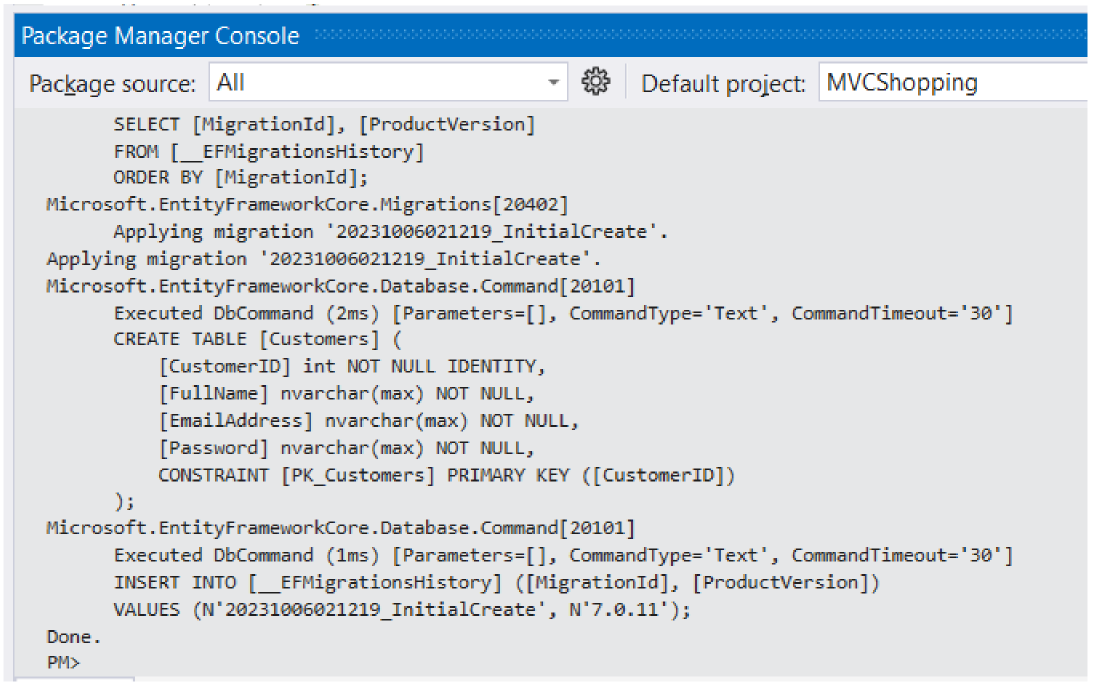


```
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace MVCShopping.Migrations
{
    /// <inheritdoc />
    1 reference
    public partial class InitialCreate : Migration
    {
        /// <inheritdoc />
        0 references
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "Customers",
                columns: table => new
                {
                    CustomerID = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    FullName = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    EmailAddress = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Password = table.Column<string>(type: "nvarchar(max)", nullable: false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Customers", x => x.CustomerID);
                });
        }

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

INITIAL MIGRATIONS

Update-Database:

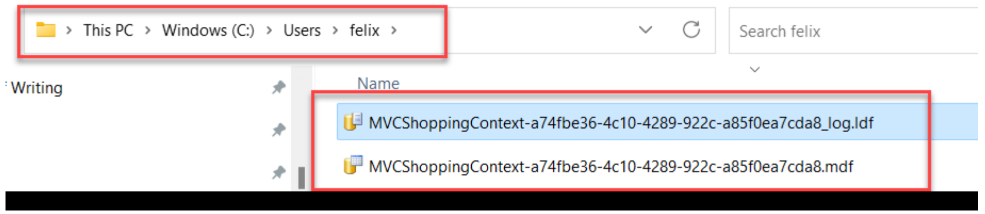
Updates the database to the latest migration, which the previous command created.
This command runs the Up method in the Migrations/{time-stamp}_InitialCreate.cs file, which creates the database.



Package Manager Console

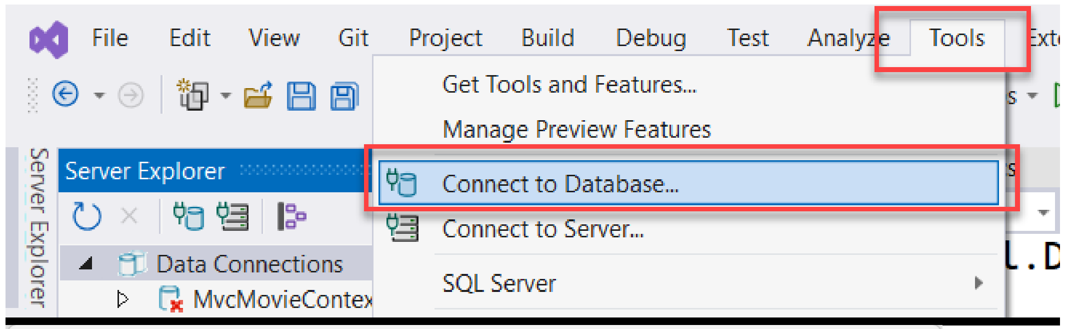
Package source: All   Default project: MVCShopping

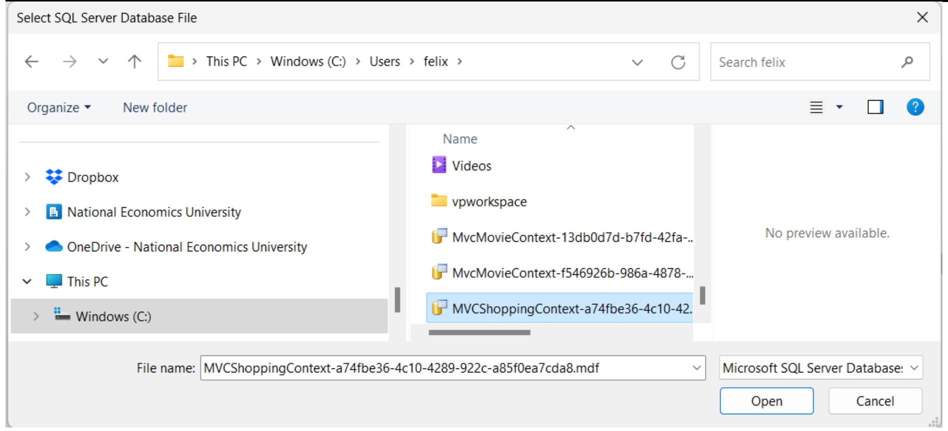
```
      SELECT [MigrationId], [ProductVersion]
      FROM [__EFMigrationsHistory]
      ORDER BY [MigrationId];
Microsoft.EntityFrameworkCore.Migrations[20402]
      Applying migration '20231006021219_InitialCreate'.
Applying migration '20231006021219_InitialCreate'.
Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (2ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      CREATE TABLE [Customers] (
          [CustomerID] int NOT NULL IDENTITY,
          [FullName] nvarchar(max) NOT NULL,
          [EmailAddress] nvarchar(max) NOT NULL,
          [Password] nvarchar(max) NOT NULL,
          CONSTRAINT [PK_Customers] PRIMARY KEY ([CustomerID])
      );
Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (1ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      INSERT INTO [__EFMigrationsHistory] ([MigrationId], [ProductVersion])
      VALUES (N'20231006021219_InitialCreate', N'7.0.11');
Done.
PM>
```



This PC › Windows (C:) › Users › felix ›

MVCShoppingContext-a74fbe36-4c10-4289-922c-a85f0ea7cda8_log.ldf
MVCShoppingContext-a74fbe36-4c10-4289-922c-a85f0ea7cda8.mdf





Select SQL Server Database File

This PC › Windows (C:) › Users › felix ›

File name: MVCShoppingContext-a74fbe36-4c10-4289-922c-a85f0ea7cda8.mdf   Microsoft SQL Server Database

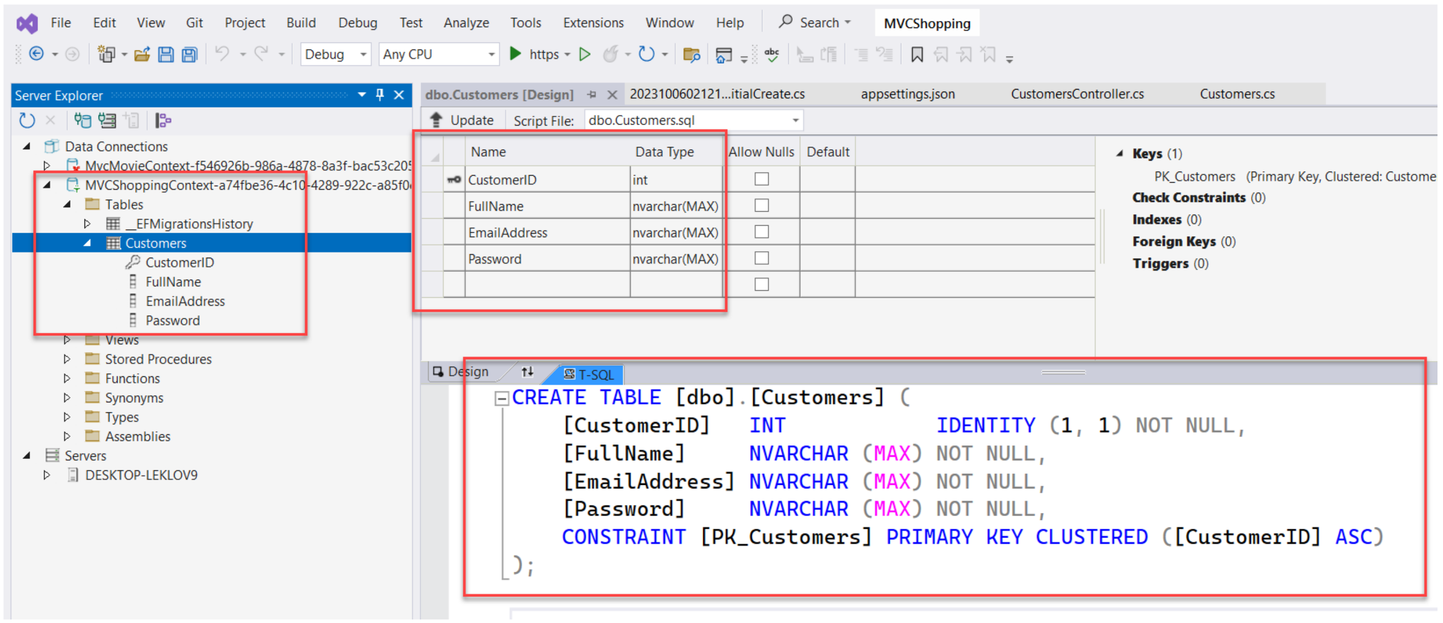
Search felix
Organize
New folder
Dropbox
National Economics University
OneDrive - National Economics University
This PC
Windows (C:)
Name
Videos
vpworkspace
MvcMovieContext-13db0d7d-b7fd-42fa-...
MvcMovieContext-f546926b-986a-4878-...
MVCShoppingContext-a74fbe36-4c10-42...
No preview available.
Open
Cancel

INITIAL MIGRATIONS

Update-Database:

Updates the database to the latest migration, which the previous command created.

This command runs the Up method in the Migrations/{time-stamp}_InitialCreate.cs file, which creates the database.



Server Explorer

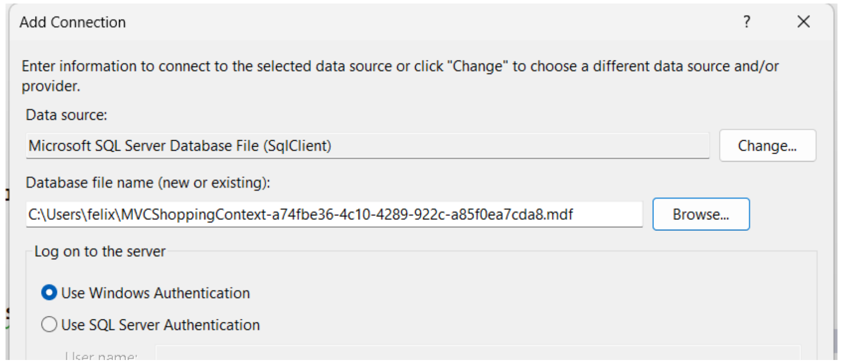
- Data Connections
  - MvcMovieContext-f546926b-986a-4878-8a3f-bac53c205
  - MVCShoppingContext-a74fbe36-4c10-4289-922c-a85f0
    - Tables
      - __EFMigrationsHistory
      - Customers
        - CustomerID
        - FullName
        - EmailAddress
        - Password
    - Views
    - Stored Procedures
    - Functions
    - Synonyms
    - Types
    - Assemblies
- Servers
  - DESKTOP-LEKLOV9

dbo.Customers [Design] — Script File: dbo.Customers.sql

	Name	Data Type	Allow Nulls	Default
	CustomerID	int	☐	
	FullName	nvarchar(MAX)	☐	
	EmailAddress	nvarchar(MAX)	☐	
	Password	nvarchar(MAX)	☐	
			☐	

Keys (1)
PK_Customers (Primary Key, Clustered: Custome
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```sql
CREATE TABLE [dbo].[Customers] (
    [CustomerID]   INT            IDENTITY (1, 1) NOT NULL,
    [FullName]     NVARCHAR (MAX) NOT NULL,
    [EmailAddress] NVARCHAR (MAX) NOT NULL,
    [Password]     NVARCHAR (MAX) NOT NULL,
    CONSTRAINT [PK_Customers] PRIMARY KEY CLUSTERED ([CustomerID] ASC)
);
```



Add Connection

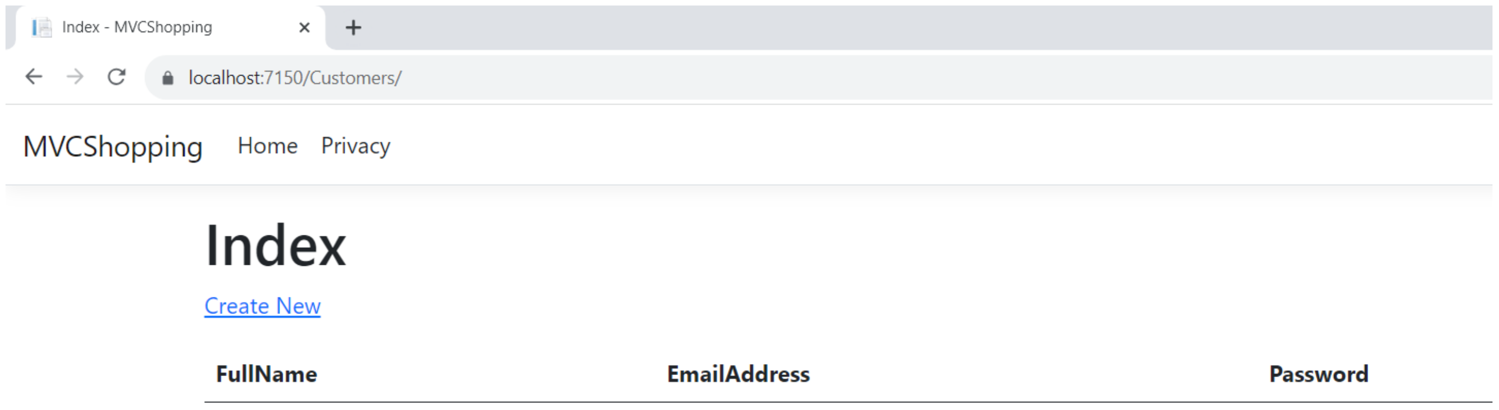
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server Database File (SqlClient)

Database file name (new or existing):
C:\Users\felix\MVCShoppingContext-a74fbe36-4c10-4289-922c-a85f0ea7cda8.mdf

Log on to the server
[✓ Use Windows Authentication]
Use SQL Server Authentication
User name:

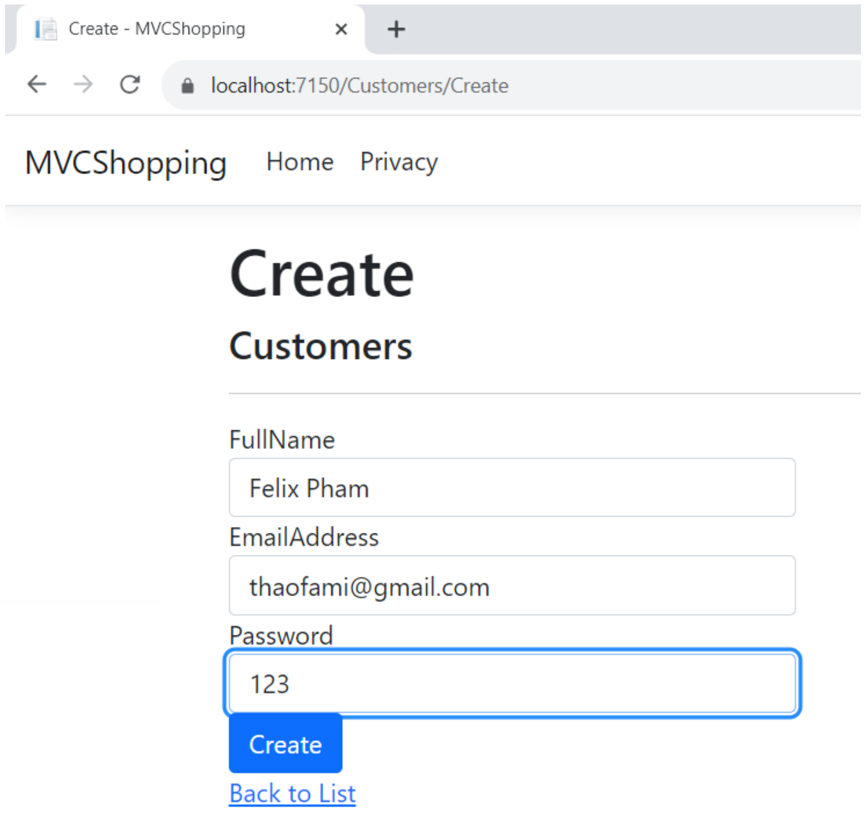
INITIAL MIGRATIONS



Index

Create New

FullName	EmailAddress	Password



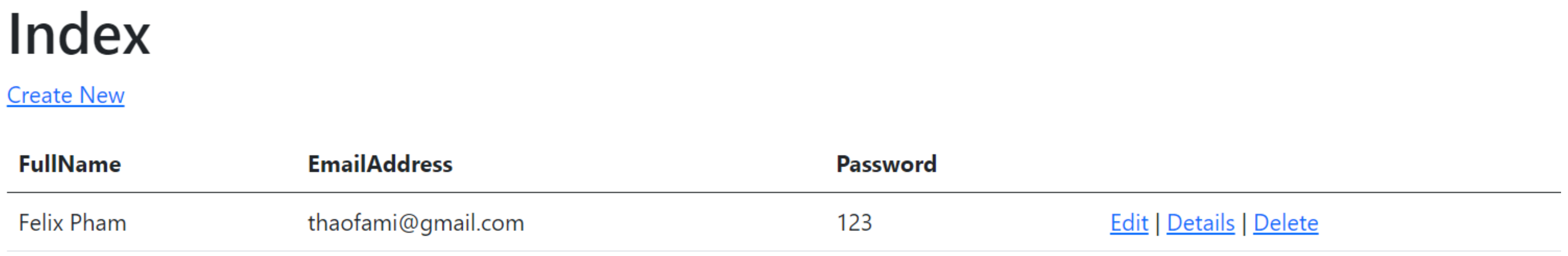
Create

Customers

FullName: Felix Pham

EmailAddress: thaofami@gmail.com

Password: 123

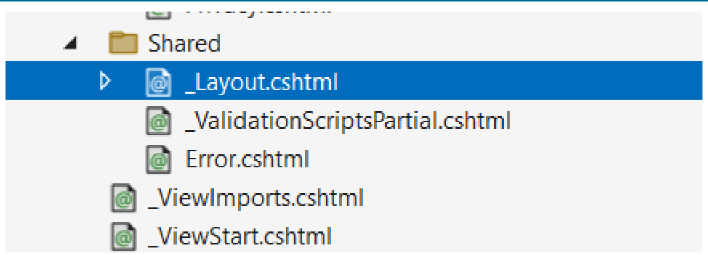


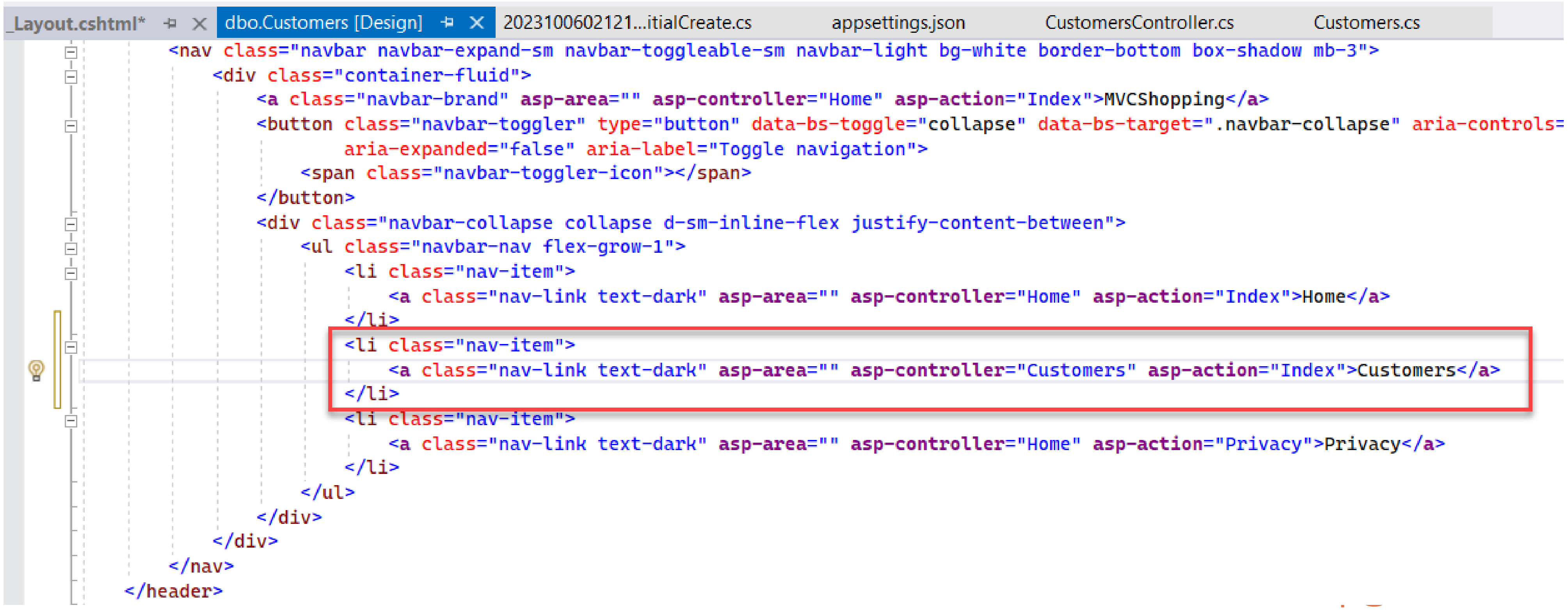
Index

Create New

FullName	EmailAddress	Password	
Felix Pham	thaofami@gmail.com	123	Edit | Details | Delete

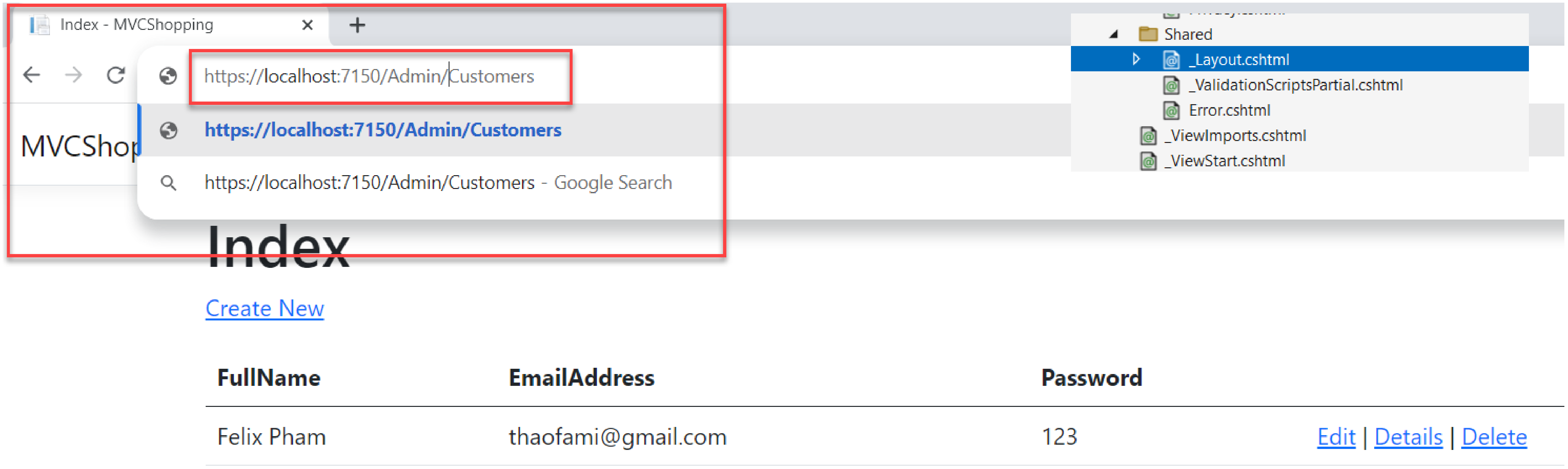
ADD A ‘CUSTOMERS' MENU





```
<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="Home" asp-action="Index">MVCShopping</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls=
                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="Customers" asp-action="Index">Customers</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>
```

ADD A 'CUSTOMERS' MENU



Index - MVCShopping

https://localhost:7150/Admin/Customers

https://localhost:7150/Admin/Customers

https://localhost:7150/Admin/Customers - Google Search

MVCShop

Shared
_Layout.cshtml
_ValidationScriptsPartial.cshtml
Error.cshtml
_ViewImports.cshtml
_ViewStart.cshtml

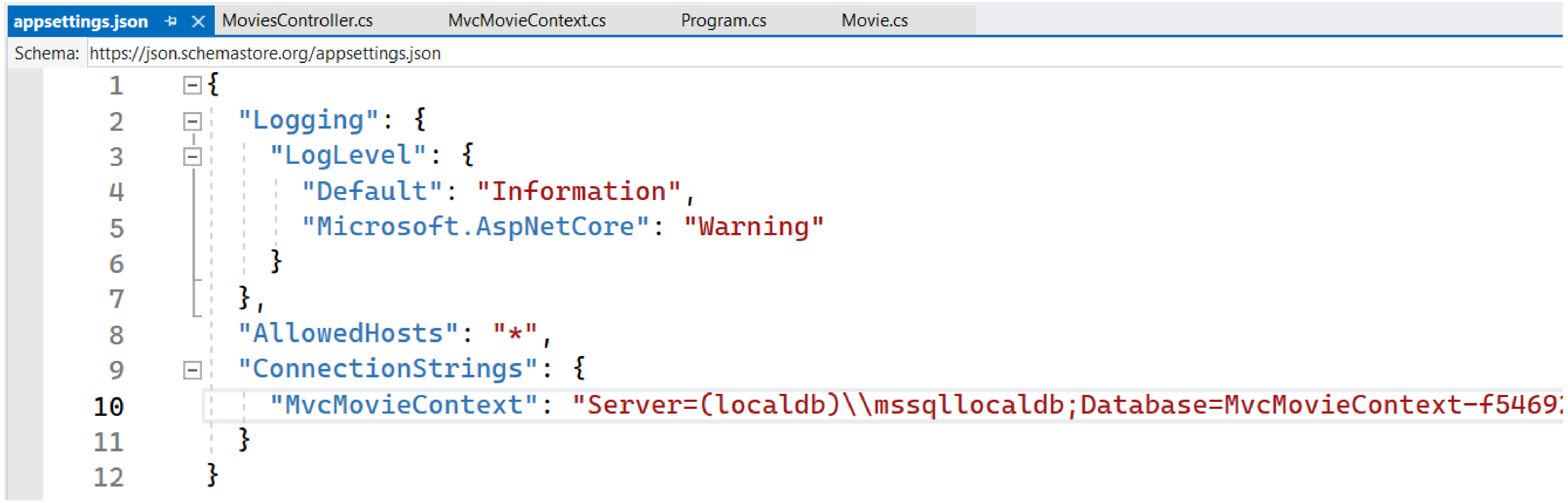
Index

Create New

FullName	EmailAddress	Password	
Felix Pham	thaofami@gmail.com	123	Edit | Details | Delete

CHECK THE CONFIGURATION

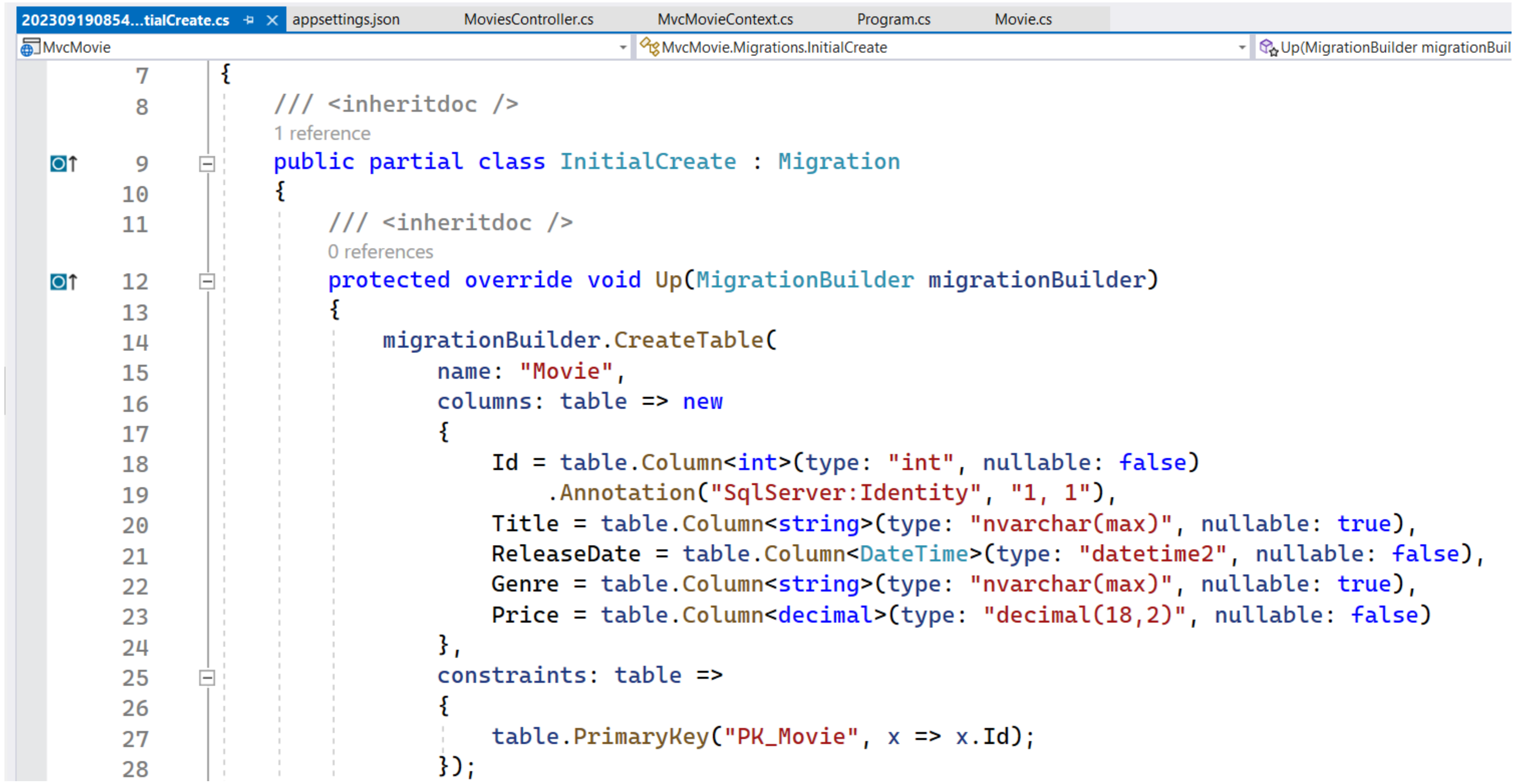
appsettings.json



```json
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "MvcMovieContext": "Server=(localdb)\\mssqllocaldb;Database=MvcMovieContext-f5469
  }
}
```

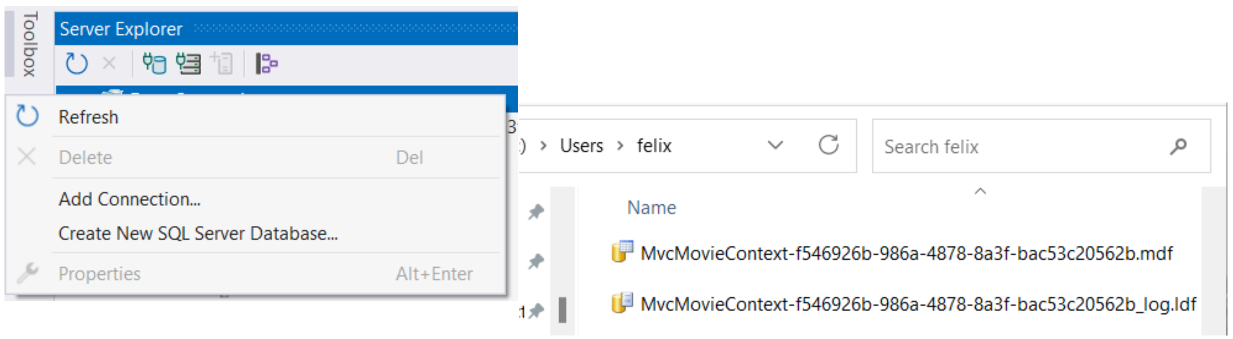
CHECK THE CONFIGURATION

The InitialCreate class


202309190854...tialCreate.cs
appsettings.json
MoviesController.cs
MvcMovieContext.cs
Program.cs
Movie.cs
MvcMovie
MvcMovie.Migrations.InitialCreate
Up(MigrationBuilder migrationBuil

```
{
    /// <inheritdoc />
    1 reference
    public partial class InitialCreate : Migration
    {
        /// <inheritdoc />
        0 references
        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);
                });
```

CONNECT TO DATABASE

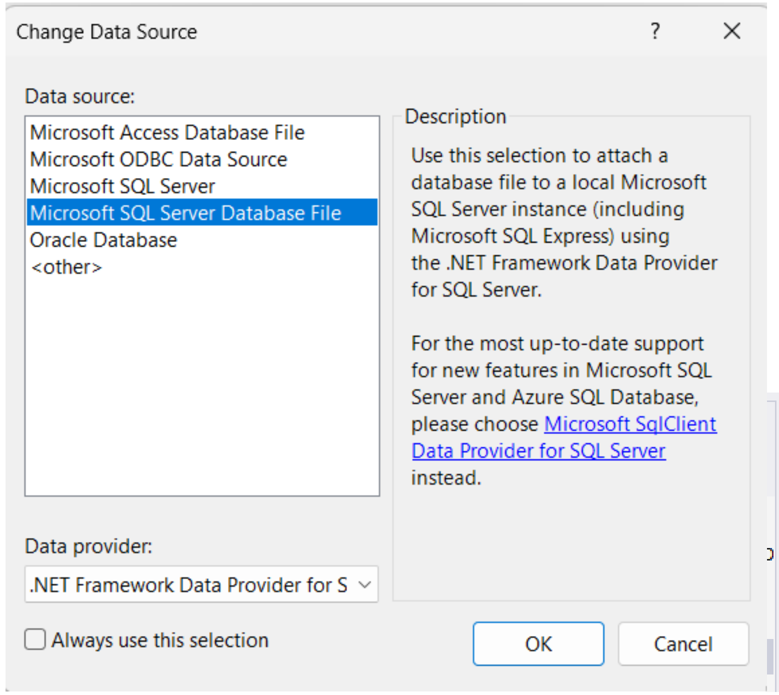


Server Explorer

- Refresh
- Delete — Del
- Add Connection...
- Create New SQL Server Database...
- Properties — Alt+Enter

Users › felix — Search felix

Name
- MvcMovieContext-f546926b-986a-4878-8a3f-bac53c20562b.mdf
- MvcMovieContext-f546926b-986a-4878-8a3f-bac53c20562b_log.ldf



Change Data Source

Data source:
- Microsoft Access Database File
- Microsoft ODBC Data Source
- Microsoft SQL Server
- Microsoft SQL Server Database File
- Oracle Database
- <other>

Description

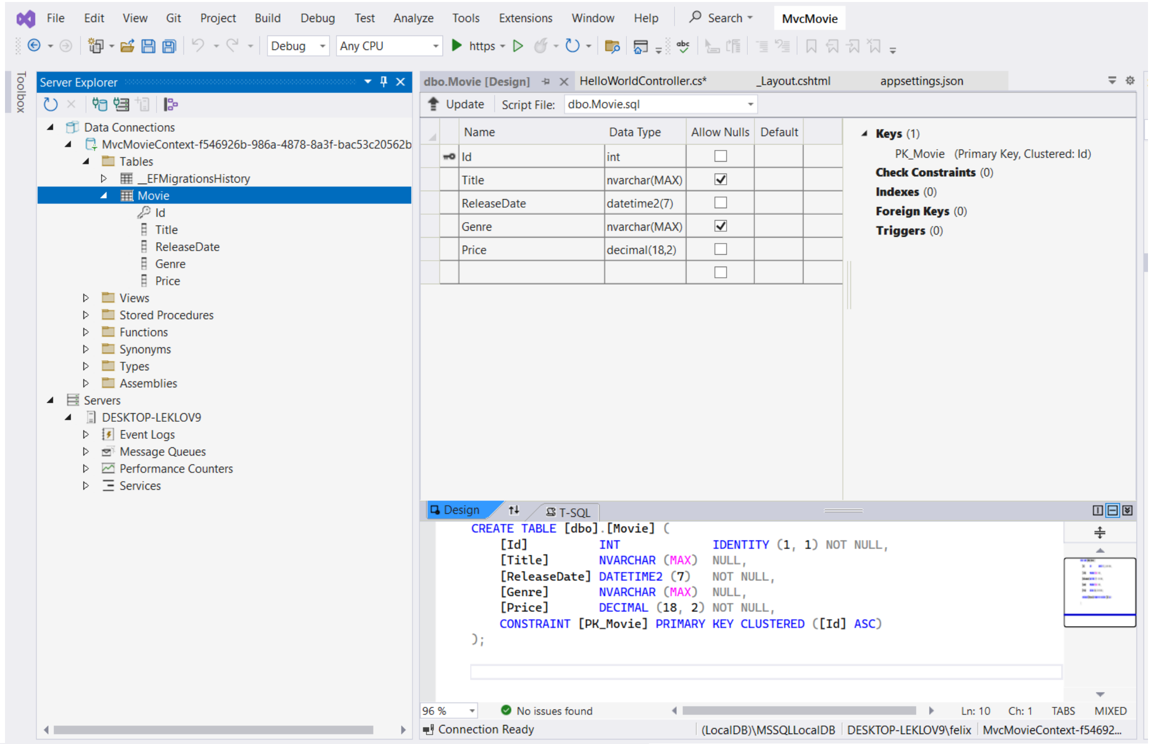
Use this selection to attach a database file to a local Microsoft SQL Server instance (including Microsoft SQL Express) using the .NET Framework Data Provider for SQL Server.

For the most up-to-date support for new features in Microsoft SQL Server and Azure SQL Database, please choose Microsoft SqlClient Data Provider for SQL Server instead.

Data provider: .NET Framework Data Provider for S

☐ Always use this selection

OK   Cancel



File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search MvcMovie

Server Explorer
- Data Connections
  - MvcMovieContext-f546926b-986a-4878-8a3f-bac53c20562b
    - Tables
      - __EFMigrationsHistory
      - Movie
        - Id
        - Title
        - ReleaseDate
        - Genre
        - Price
    - Views
    - Stored Procedures
    - Functions
    - Synonyms
    - Types
    - Assemblies
- Servers
  - DESKTOP-LEKLOV9
    - Event Logs
    - Message Queues
    - Performance Counters
    - Services

dbo.Movie [Design] | HelloWorldController.cs* | _Layout.cshtml | appsettings.json

Update Script File: dbo.Movie.sql

Name	Data Type	Allow Nulls	Default
Id	int	☐	
Title	nvarchar(MAX)	✓	
ReleaseDate	datetime2(7)	☐	
Genre	nvarchar(MAX)	✓	
Price	decimal(18,2)	☐	
		☐	

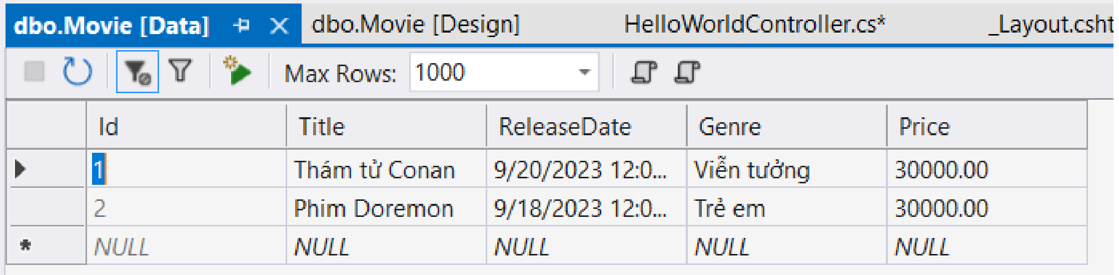
Keys (1)
PK_Movie (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Design | T-SQL

```
CREATE TABLE [dbo].[Movie] (
    [Id]          INT            IDENTITY (1, 1) NOT NULL,
    [Title]       NVARCHAR (MAX) NULL,
    [ReleaseDate] DATETIME2 (7)  NOT NULL,
    [Genre]       NVARCHAR (MAX) NULL,
    [Price]       DECIMAL (18, 2) NOT NULL,
    CONSTRAINT [PK_Movie] PRIMARY KEY CLUSTERED ([Id] ASC)
);
```

96 % No issues found Ln: 10 Ch: 1 TABS MIXED

Connection Ready (LocalDB)\MSSQLLocalDB DESKTOP-LEKLOV9\felix MvcMovieContext-f54692...



dbo.Movie [Data] | dbo.Movie [Design] | HelloWorldController.cs* | _Layout.csht

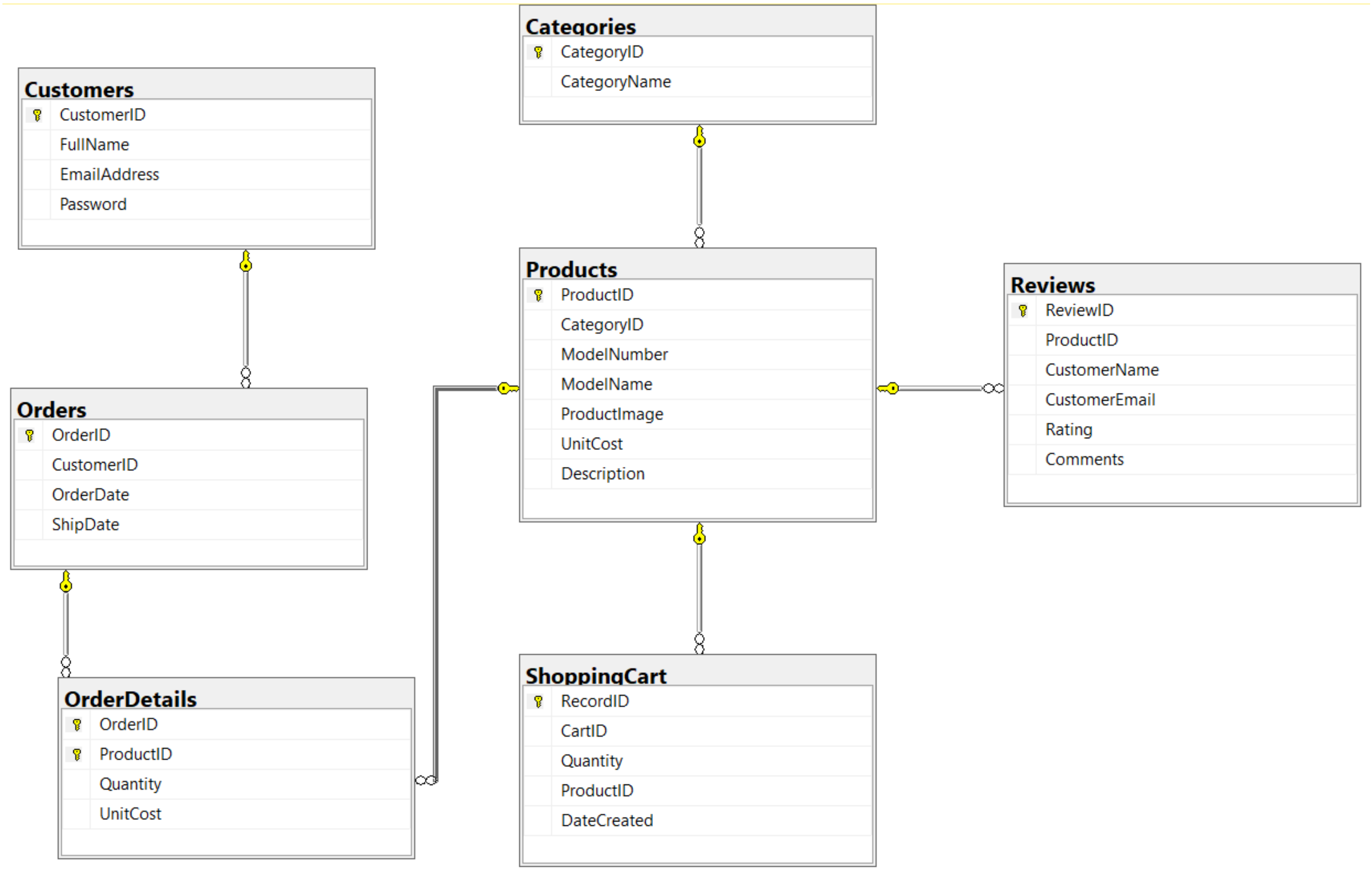
Max Rows: 1000

Id	Title	ReleaseDate	Genre	Price
1	Thám tử Conan	9/20/2023 12:0...	Viễn tưởng	30000.00
2	Phim Doremon	9/18/2023 12:0...	Trẻ em	30000.00
NULL	NULL	NULL	NULL	NULL

Thaop@neu.edu.vn

EXERCISE

- Create a database and its tables by running the script.
- For each table, adhere to the structure outlined in the diagram. Using Scaffold, let's create a corresponding Admin Page to manipulate data.
- Additionally, add a menu to the Admin Page. Run and test all pages



REFERENCES

- https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/controller-methods-views?view=aspnetcore-7.0