

Assignment 2

Book Store

Setting Up the Project
(Part 1)

Visual Studio 2019

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

[Continue without code →](#)

Visual Studio 2019

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

[Continue without code →](#)

Visual Studio 2019

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started



[Continue without code →](#)

Create a new project

Recent project templates

A list of your recently accessed templates will be displayed here.

Search for templates (Alt+S) 

All languages

All platforms

All project types



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C#

Linux

macOS

Windows

Console



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

Visual Basic

Windows

Linux

macOS

Console



ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C#

Linux

macOS

Windows

Cloud

Service

Web



Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

Back

Next

Create a new project

Recent project templates

A list of your recently accessed templates will be displayed here.

Search for templates (Alt+S) 

All languages

All platforms

All project types



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C#

Linux

macOS

Windows

Console



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

Visual Basic

Windows

Linux

macOS

Console



ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C#

Linux

macOS

Windows

Cloud

Service

Web



Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

Back

Next

Create a new project

Recent project templates

A list of your recently accessed templates will be displayed here.

Search for templates (Alt+S)

All languages All platforms All project types

Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
C# **Linux** **macOS** **Windows** **Console**

Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
Visual Basic **Windows** **Linux** **macOS** **Console**

ASP.NET Core Web Application
Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.
C# **Linux** **macOS** **Windows** **Cloud** **Service** **Web**

Blazor App
Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

Back Next

Create a new project

Recent project templates

A list of your recently accessed templates will be displayed here.

Search for templates (Alt+S) 

All languages

All platforms

All project types



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C#

Linux

macOS

Windows

Console



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

Visual Basic

Windows

Linux

macOS

Console



ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C#

Linux

macOS

Windows

Cloud

Service

Web



Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

Back

Next

Configure your new project

ASP.NET Core Web Application

C#

Linux

macOS

Windows

Cloud

Service

Web

Project name

AndrewsBookStore

Location

C:\Users\ASTEELE\source\repos



Solution name

AndrewsBookStore

Place solution and project in the same directory

Back

Create

Configure your new project

ASP.NET Core Web Application

C#

Linux

macOS

Windows

Cloud

Service

Web

Project name

AndrewsBookStore

Location

C:\Users\ASTEELE\source\repos



Solution name

AndrewsBookStore

Place solution and project in the same directory

Back

Create



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

No Authentication

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Enable [Razor runtime compilation](#)

Author: Microsoft

Source: Templates 5.0.2

Back

Create



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

No Authentication

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Enable [Razor runtime compilation](#)

Author: Microsoft

Source: Templates 5.0.2

Back

Create



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

No Authentication

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Enable [Razor runtime compilation](#)

Author: Microsoft

Source: Templates 5.0.2

Back

Create



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

No Authentication

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Enable [Razor runtime compilation](#)

Author: Microsoft

Source: Templates 5.0.2

Back

Create



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty
it.



ASP.NET

A project
This temp



ASP.NET

A project



ASP.NET

A project
Controller



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication



For applications that don't require any user authentication.

[Learn more](#)

[No Authentication](#)

[Individual User Accounts](#)

[Work or School Accounts](#)

[Windows Authentication](#)

[Learn more about third-party open source authentication options](#)

[OK](#)

[Cancel](#)

PS
port
esktop)

me compilation

Author: Microsoft

Source: Templates 5.0.2

[Back](#)

[Create](#)



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty
it.



ASP.NET

A project
This temp



ASP.NET

A project



ASP.NET

A project
Controller



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

Store user accounts in-app

[Learn more](#)

Select this option to create a project that includes a local user accounts store.

No Authentication

Individual User Accounts

Work or School Accounts

Windows Authentication

[Learn more about third-party open source authentication options](#)

OK

Cancel

me compilation

Author: Microsoft

Source: Templates 5.0.2

Back

Create



Create a new ASP.NET Core web application

.NET Core ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template.



ASP.NET Core Web API

A project template for building RESTful services.



ASP.NET Core MVC

A project template for building web applications.



ASP.NET Core Identity Controller

A project template for building identity controllers.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular.



ASP.NET Core with React.js

A project template for creating an ASP.NET Core application with React.js.

[Get additional project templates](#)

Authentication

Store user accounts in-app

[Learn more](#)

Select this option to create a project that includes a local user accounts store.

No Authentication

Individual User Accounts

Work or School Accounts

Windows Authentication

[Learn more about third-party open source authentication options](#)

[OK](#)

[Cancel](#)

me compilation

Author: Microsoft

Source: Templates 5.0.2

[Back](#)

[Create](#)



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty
it.



ASP.NET

A project
This temp



ASP.NET

A project



ASP.NET

A project
Controller



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication



Store user accounts in-app

[Learn more](#)

Select this option to create a project that includes a local user accounts store.

- No Authentication
- Individual User Accounts
- Work or School Accounts
- Windows Authentication

[Learn more about third-party open source authentication options](#)

[OK](#)

[Cancel](#)

PS
port
desktop)

me compilation

Author: Microsoft

Source: Templates 5.0.2

[Back](#)

[Create](#)



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty
it.



ASP.NET

A project
This temp



ASP.NET

A project i



ASP.NET

A project
Controller



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

[Back](#)

[Create](#)

Authentication

Store user accounts in-app

[Learn more](#)

Select this option to create a project that includes a local user accounts store.

- No Authentication
- Individual User Accounts
- Work or School Accounts
- Windows Authentication

[Learn more about third-party open source authentication options](#)

[OK](#)

[Cancel](#)

PS
port
esktop)

me compilation

Author: Microsoft

Source: Templates 5.0.2



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

Individual User Accounts

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Enable [Razor runtime compilation](#)

Author: Microsoft

Source: Templates 5.0.2

Back

Create



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

Individual User Accounts

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Enable Razor runtime compilation

Author: Microsoft

Source: Templates 5.0.2

Back

Create



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

Individual User Accounts

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Enable Razor runtime compilation

Author: Microsoft

Source: Templates 5.0.2

Back

Create



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

Individual User Accounts

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Enable [Razor runtime compilation](#)

Author: Microsoft

Source: Templates 5.0.2

Back

Create



Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



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.



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

Individual User Accounts

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Enable [Razor runtime compilation](#)

Author: Microsoft

Source: Templates 5.0.2

Back

Create





Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 5.0



ASP.NET Core Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.



ASP.NET Core Web API

A project template for creating an
This template can also be used for



ASP.NET Core Web App

A project template for creating an



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.



ASP.NET Core with Angular

A project template for creating an ASP.NET Core application with Angular



ASP.NET Core with React.js

[Get additional project templates](#)

Authentication

Individual User Accounts

[Change](#)

Advanced

Configure for HTTPS

Enable Docker Support

(Requires [Docker Desktop](#))

Linux

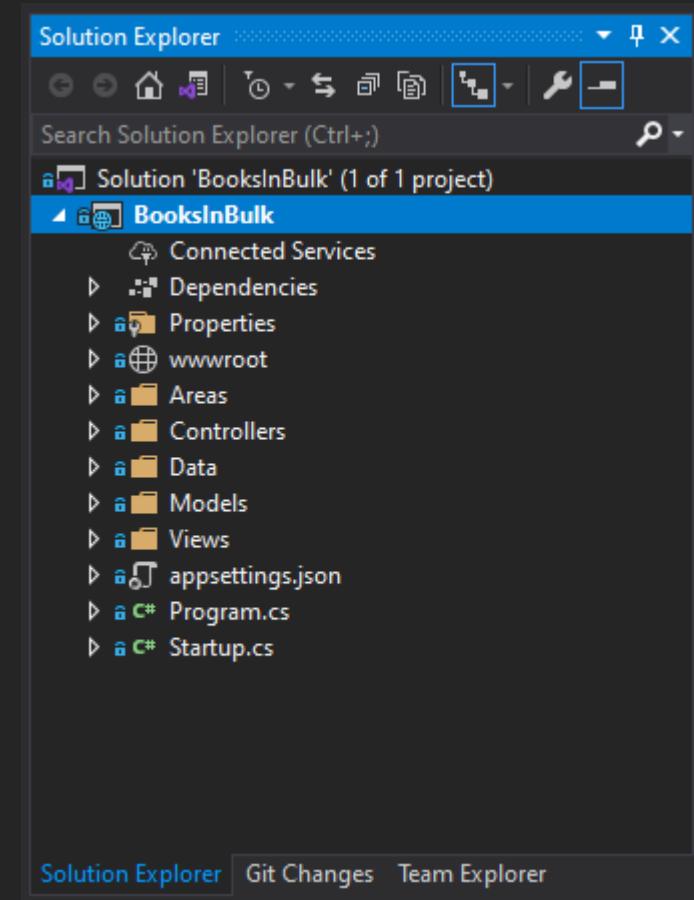
Enable [Razor runtime compilation](#)

Author: Microsoft

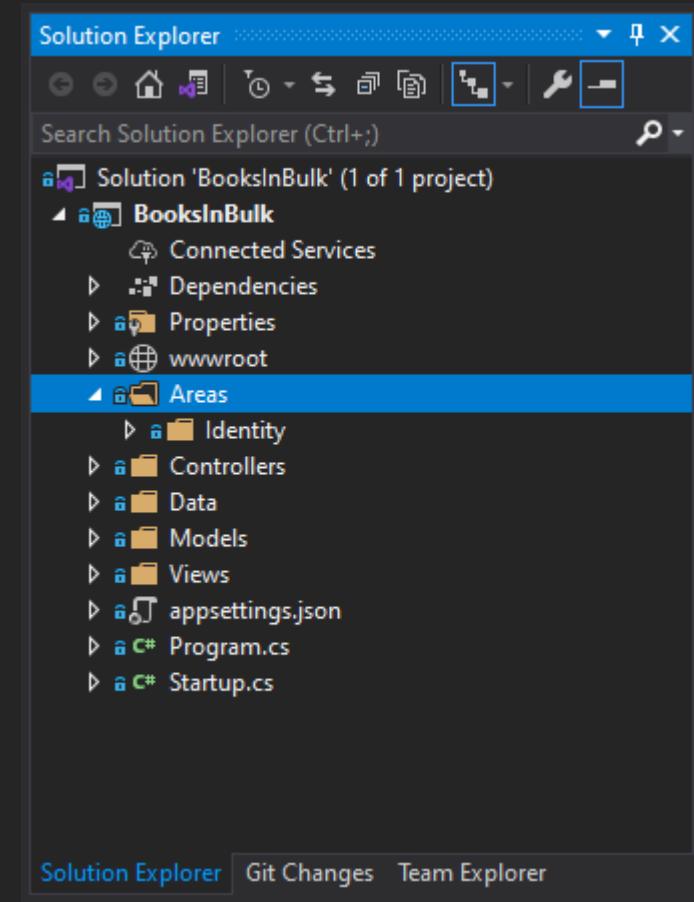
Source: Templates 5.0.2

Create

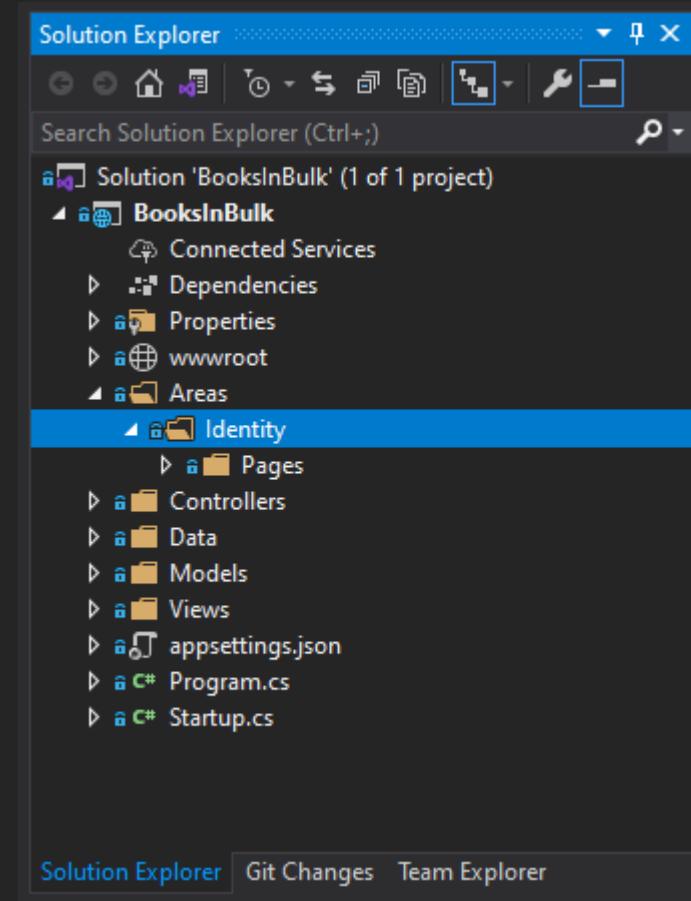
ASP.NET Core Areas



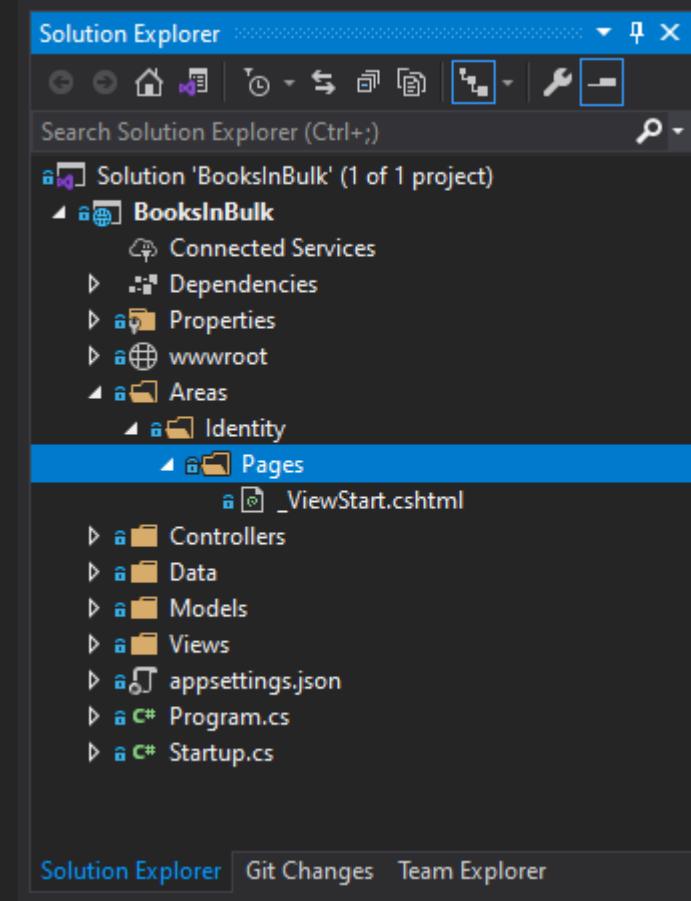
ASP.NET Core Areas



ASP.NET Core Areas



ASP.NET Core Areas



ASP.NET Core Areas

The screenshot shows the Visual Studio IDE interface for an ASP.NET Core project named "BooksInBulk".

Toolbar: File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), BooksInBulk.

Status Bar: BooksInBulk, _ViewStart.cshtml, Live Share.

Solution Explorer: Shows the project structure:

- BooksInBulk (solution)
- BooksInBulk (project)
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Identity
 - Pages
 - Controllers
 - Data
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs

Code Editor: Displays the content of _ViewStart.cshtml:1 @{
2 Layout = "/Views/Shared/_Layout.cshtml";
3 }
4

ASP.NET Core Areas

The screenshot shows the Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a search bar. The title bar displays "BooksInBulk". The toolbar below has icons for back, forward, search, and other common operations. The status bar at the bottom shows "BooksInBulk" and "Live Share".

The main code editor window contains the following C# code:

```
1 @{
2     Layout = "/Views/Shared/_Layout.cshtml";
3 }
4
```

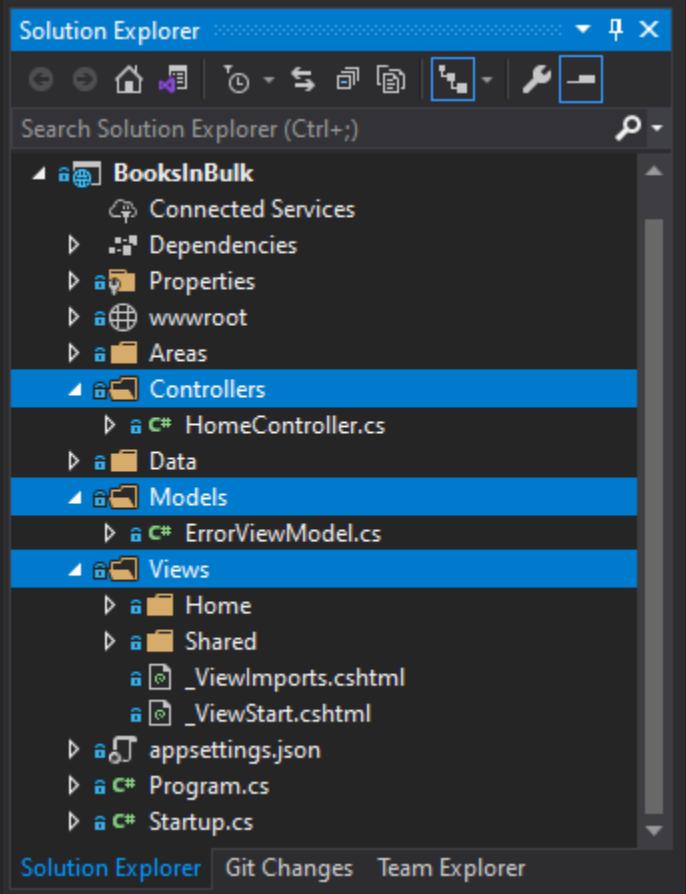
A red arrow points from the text "Razor class library linked to the _Layout in Views/Shared" to the line "Layout = "/Views/Shared/_Layout.cshtml";".

The Solution Explorer window on the right lists the project structure for "BooksInBulk":

- Solution 'BooksInBulk' (1 of 1 project)
 - BooksInBulk
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Identity
 - Pages
 - Controllers
 - Data
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs

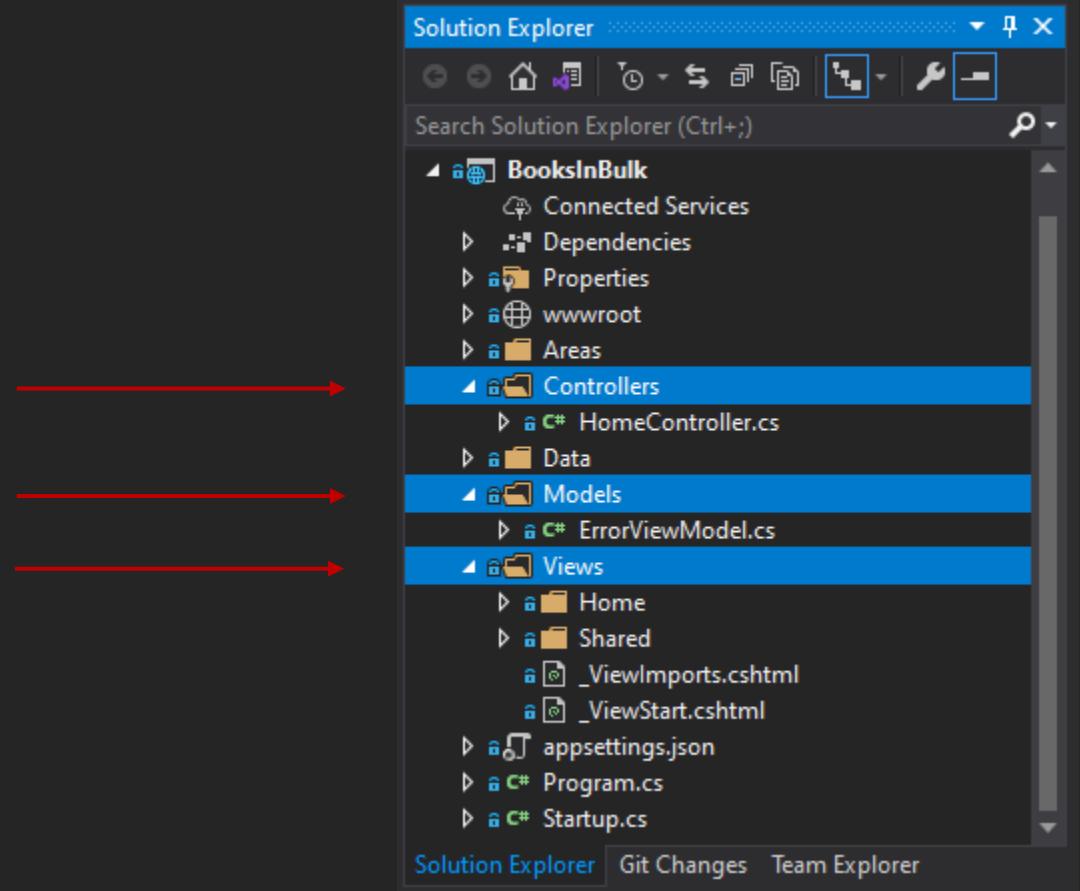
Razor class library linked to the _Layout in Views/Shared

ASP.NET Core MVC

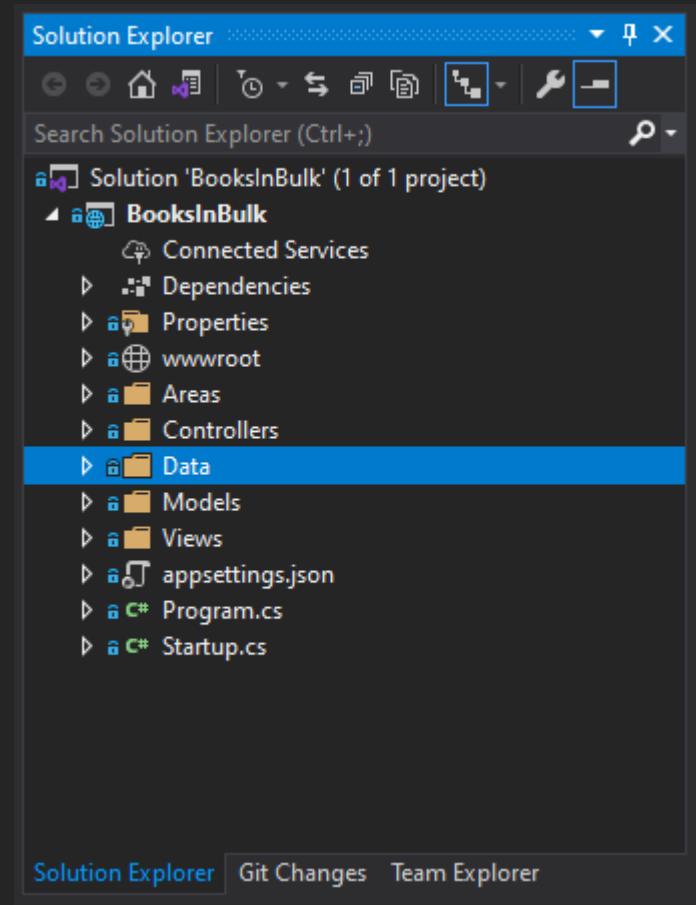


ASP.NET Core MVC

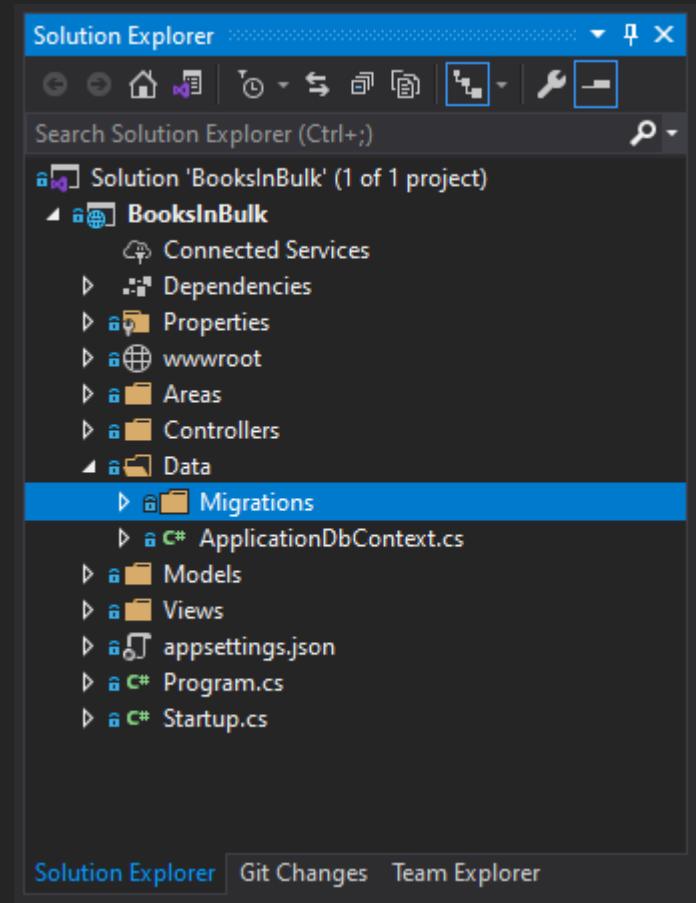
Default Controllers, Models, Views folders with
HomeController.cs, ErrorViewModel.cs and Home & Shared
View folders with _ViewImports.cshtml and _ViewStart.cshtml



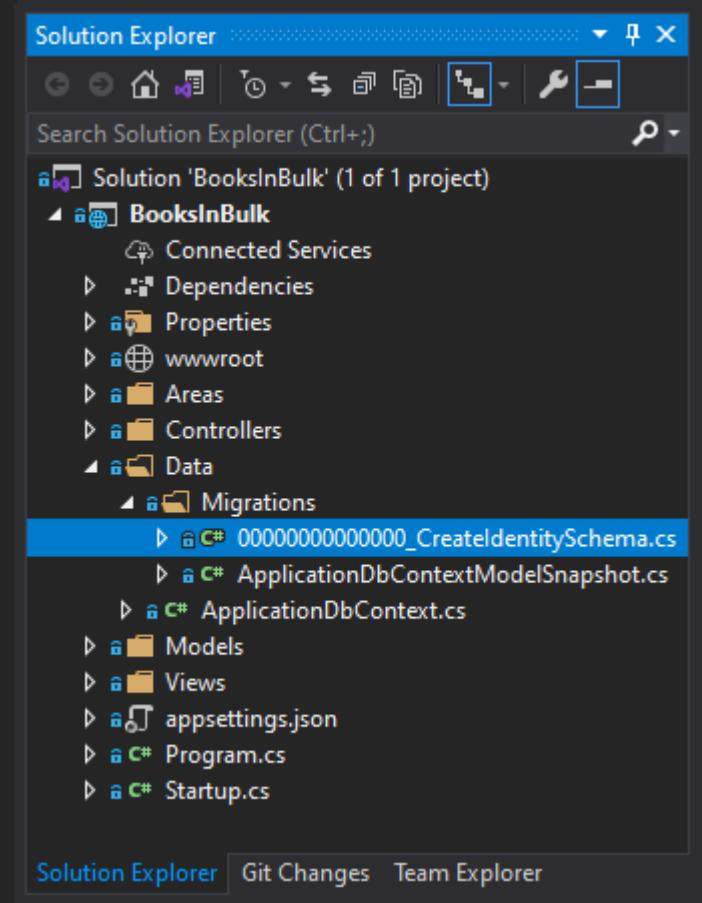
ASP.NET Core Data



ASP.NET Core Data

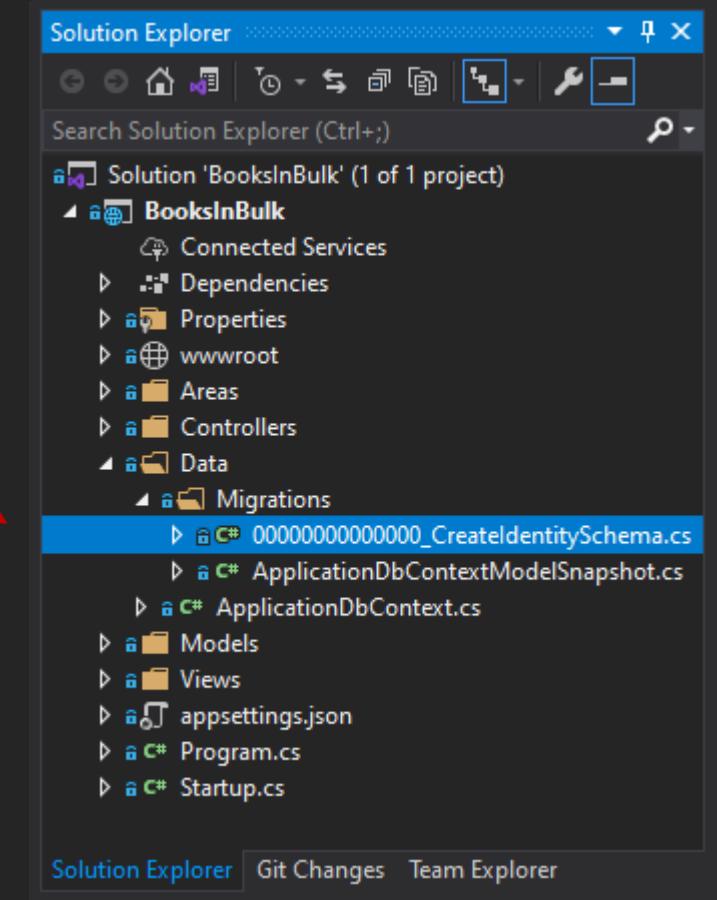


ASP.NET Core Data



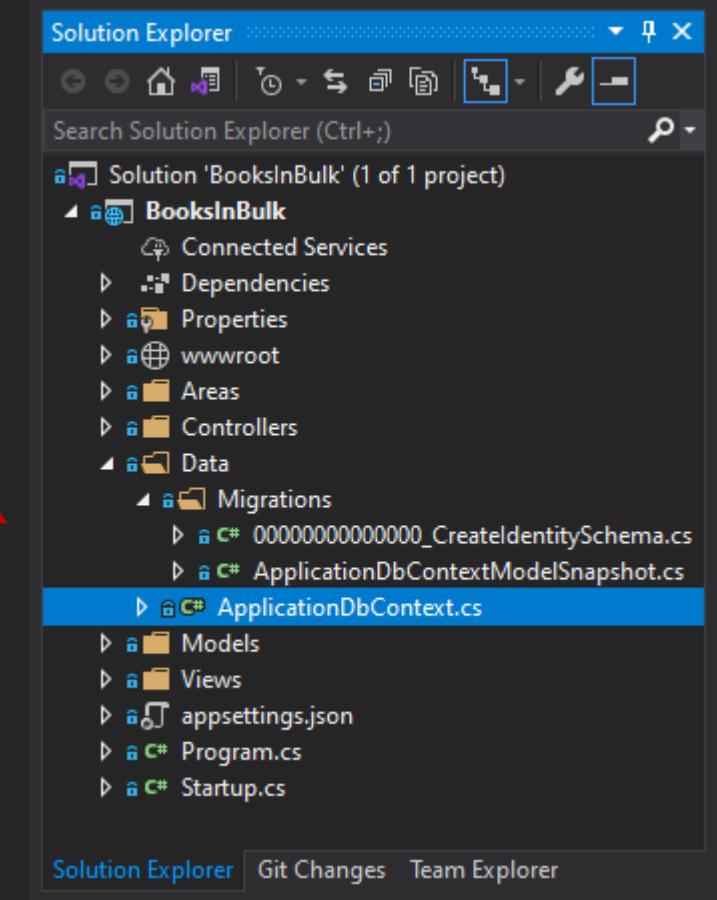
ASP.NET Core Data

00000000000000_CreatIdentitySchema.cs added on create
to build identity related tables for users/roles/security.



ASP.NET Core Data

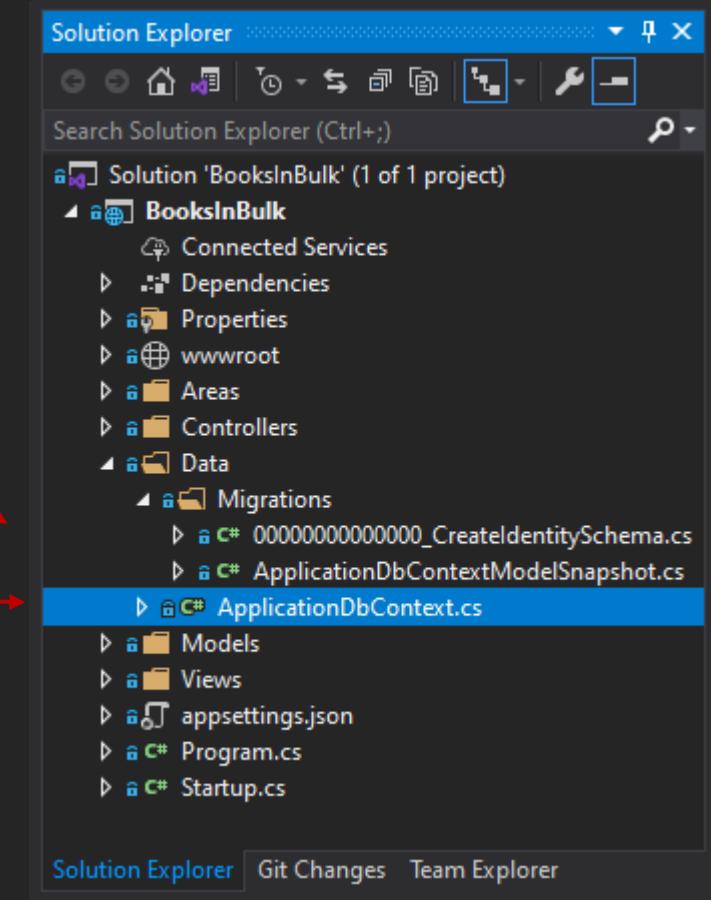
00000000000000_CreatIdentitySchema.cs added on create
to build identity related tables for users/roles/security.



ASP.NET Core Data

00000000000000_CreatIdentitySchema.cs added on create to build identity related tables for users/roles/security.

ApplicationContext.cs is used to access database and confirm entities present. Inherits from IdentityDbContext - due to addition of authentication.



ASP.NET Core Startup.cs

The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** ASP - File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
- Search Bar:** Search (Ctrl+Q)
- Toolbars:** Standard toolbar with icons for New, Open, Save, Print, etc.
- Status Bar:** BooksInBulk
- Project Explorer:** BooksInBulk
- Code Editor:** Startup.cs (C#)
- Solution Explorer:** BooksInBulk (1 of 1 project) containing:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs
- Task List:** None
- Output Window:** None
- Properties Window:** None
- Toolbox:** None
- Help:** None

```
24
25     2 references
26     public IConfiguration Configuration { get; }
27
28     // This method gets called by the runtime. Use this method to add services to the container.
29     0 references
30     public void ConfigureServices(IServiceCollection services)
31     {
32         services.AddDbContext<ApplicationContext>(options =>
33             options.UseSqlServer(
34                 Configuration.GetConnectionString("DefaultConnection")));
35         services.AddDatabaseDeveloperPageExceptionFilter();
36
37         services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
38             .AddEntityFrameworkStores<ApplicationContext>();
39         services.AddControllersWithViews();
40     }
```

ASP.NET Core Startup.cs

The screenshot shows the Visual Studio IDE interface with the following details:

- Toolbar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Project Selector:** BooksInBulk (selected), BooksInBulk.Startup (active tab), Configuration (IConfiguration configuration).
- Toolbars:** Standard, Debug, Task List, Solution Explorer, Live Share.
- Solution Explorer:** Shows the project structure for 'BooksInBulk' (1 of 1 project). It includes:
 - BooksInBulk
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs
- Status Bar:** Solution Explorer, Git Changes, Team Explorer.

```
24
25     2 references
26     public IConfiguration Configuration { get; }
27
28     // This method gets called by the runtime. Use this method to add services to the container.
29     0 references
30     public void ConfigureServices(IServiceCollection services)
31     {
32         services.AddDbContext<ApplicationContext>(options =>
33             options.UseSqlServer(
34                 Configuration.GetConnectionString("DefaultConnection")));
35         services.AddDatabaseDeveloperPageExceptionFilter();
36
37         services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
38             .AddEntityFrameworkStores<ApplicationContext>();
39         services.AddControllersWithViews();
40     }
```

ASP.NET Core Startup.cs

The screenshot shows the Visual Studio IDE interface with the following details:

- Toolbar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Project Status:** BooksInBulk (highlighted), BooksInBulk.Startup, Startup(IConfiguration configuration).
- Build Configuration:** Any CPU, IIS Express.
- Solution Explorer:** Shows the project structure for 'BooksInBulk' with files like Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers, Data, Models, Views, appsettings.json, Program.cs, and Startup.cs.
- Code Editor:** Displays the contents of the Startup.cs file.

```
24
25     2 references
26     public IConfiguration Configuration { get; }
27
28     // This method gets called by the runtime. Use this method to add services to the container.
29     0 references
30     public void ConfigureServices(IServiceCollection services)
31     {
32         services.AddDbContext<ApplicationContext>(options =>
33             options.UseSqlServer(
34                 Configuration.GetConnectionString("DefaultConnection")));
35         services.AddDatabaseDeveloperPageExceptionFilter();
36
37         services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
38             .AddEntityFrameworkStores<ApplicationContext>();
39         services.AddControllersWithViews();
40     }
41 }
```

ASP.NET Core Startup.cs

The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** ASP - File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
- Search Bar:** Search (Ctrl+Q)
- Toolbars:** Standard toolbar with icons for New, Open, Save, Print, etc.
- Status Bar:** BooksInBulk
- Project Explorer:** BooksInBulk
- Code Editor:** Startup.cs (C#)
- Solution Explorer:** BooksInBulk (1 of 1 project) containing:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs
- Task List:** None
- Output Window:** None
- Properties Window:** None
- Toolbox:** None
- Help:** None

```
24
25     2 references
26     public IConfiguration Configuration { get; }
27
28     // This method gets called by the runtime. Use this method to add services to the container.
29     0 references
30     public void ConfigureServices(IServiceCollection services)
31     {
32         services.AddDbContext<ApplicationContext>(options =>
33             options.UseSqlServer(
34                 Configuration.GetConnectionString("DefaultConnection")));
35         services.AddDatabaseDeveloperPageExceptionFilter();
36
37         services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
38             .AddEntityFrameworkStores<ApplicationContext>();
39         services.AddControllersWithViews();
40     }
```

ASP.NET Core Startup.cs

The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** ASP - File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
- Search Bar:** Search (Ctrl+Q)
- Toolbars:** Standard toolbar with icons for New, Open, Save, Print, etc.
- Status Bar:** BooksInBulk
- Project Explorer:** BooksInBulk
- Code Editor:** Startup.cs (C#)
- Solution Explorer:** Solution 'BooksInBulk' (1 of 1 project) containing BooksInBulk, Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers, Data, Models, Views, appsettings.json, Program.cs, and Startup.cs (selected).
- Task List:** BooksInBulk
- Output Window:** BooksInBulk
- Properties Window:** BooksInBulk
- Tool Windows:** Task List, Output, Properties, Solution Explorer, Git Changes, Team Explorer

```
24
25     2 references
26     public IConfiguration Configuration { get; }
27
28     // This method gets called by the runtime. Use this method to add services to the container.
29     0 references
30     public void ConfigureServices(IServiceCollection services)
31     {
32         services.AddDbContext<ApplicationContext>(options =>
33             options.UseSqlServer(
34                 Configuration.GetConnectionString("DefaultConnection")));
35         services.AddDatabaseDeveloperPageExceptionFilter();
36
37         services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
38             .AddEntityFrameworkStores<ApplicationContext>();
39         services.AddControllersWithViews();
40     }
```

ASP.NET Core Startup.cs

The screenshot shows the Visual Studio IDE interface with the following details:

- Toolbar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Project Explorer:** BooksInBulk (selected), BooksInBulk.Startup, ConfigureServices(IServiceCollection services).
- Solution Explorer:** Solution 'BooksInBulk' (1 of 1 project) contains BooksInBulk (Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers, Data, Models, Views, appsettings.json, Program.cs, Startup.cs).
- Code Editor:** Startup.cs file open, showing the ConfigureServices method. A red arrow points from the explanatory text below to the line where the IdentityUser service is registered.
- Status Bar:** Live Share, AS.

```
24
25     2 references
26     public IConfiguration Configuration { get; }
27
28     // This method gets called by the runtime. Use this method to add services to the container.
29     0 references
30     public void ConfigureServices(IServiceCollection services)
31     {
32         services.AddDbContext<ApplicationContext>(options =>
33             options.UseSqlServer(
34                 Configuration.GetConnectionString("DefaultConnection")));
35         services.AddDatabaseDeveloperPageExceptionFilter();
36
37         services.AddDefaultIdentity<IdentityUser>()
38             .AddEntityFrameworkStores<ApplicationContext>();
39         services.AddControllersWithViews();
40     }
```

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

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

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseMigrationsEndPoint();
    }
    else
    {
        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.UseAuthentication();
    app.UseAuthorization();

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


Solution Explorer

AndrewsBookStore

ASP.NET Core

Build Your App

Connect To The Cloud

Learn Your IDE

ASP.NET Core documentation

.NET application architecture

Output

Show output from: Package Manager

Time Elapsed: 00:00:00.8507838

===== Finished =====

Error List ... Web Publish Activity Package Manager Console Output

Ready

Add to Source Control

Solution Explorer

AndrewsBookStore

ASP.NET Core

Build Your App

Connect To The Cloud

Learn Your IDE

ASP.NET Core documentation

.NET application architecture

Output

Show output from: Package Manager

Time Elapsed: 00:00:00.8507838

===== Finished =====

Error List ... Web Publish Activity Package Manager Console Output

Ready

Add to Source Control

SPI in abo

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AndrewsBookStore AS — □ X

Clone Repository... Create Git Repository... Local Repositories Commit or Stash... Ctrl+0, G Fetch Pull Push New Branch... View Branch History Manage Branches Open Repository in Manage Remotes... Settings

Debug Any CPU IIS Express Live Share

Create a Git repository

Push to a new remote Initialize a local Git repository

GitHub Local path C:\Users\ASTEEL\source\repos\AndrewsBookStore ...

Other Create a new GitHub repository

Existing remote Account asteеле-14 (GitHub)

Local only Owner asteèle-14

Repository Name AndrewsBookStore

Description Enter the description of the repository <Optional>

Private

Push your code to GitHub
<https://github.com/asteèle-14/AndrewsBookStore>

Create and Push Cancel

Output

Show output from: Package Manager Time Elapsed: 00:00:00.8507838 ====== Finished ======

aspnet-AndrewsBookStore-4AA32BD9-959B- AndrewsBookStore.csproj C:\Users\ASTEEL\source\repos\AndrewsBookSt C:\Users\ASTEEL\source\repos\AndrewsBookSt

Error List ... Web Publish Activity Package Manager Console Output

Ready Add to Source Control 2

SPI in abo

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AndrewsBookStore AS — □ X

Clone Repository... Create Git Repository... Local Repositories Commit or Stash... Ctrl+0, G Fetch Pull Push New Branch... View Branch History Manage Branches Open Repository in Manage Remotes... Settings

Debug Any CPU IIS Express Live Share

Create a Git repository

Push to a new remote Initialize a local Git repository

GitHub Local path C:\Users\ASTEEL\source\repos\AndrewsBookStore ...

Other Create a new GitHub repository

Existing remote Account asteеле-14 (GitHub)

Local only Owner asteèle-14

Repository Name AndrewsBookStore

Description Enter the description of the repository <Optional>

Private

Push your code to GitHub
<https://github.com/asteèle-14/AndrewsBookStore>

Create and Push Cancel

Output

Show output from: Package Manager Time Elapsed: 00:00:00.8507838 ====== Finished ======

aspnet-AndrewsBookStore-4AA32BD9-959B- AndrewsBookStore.csproj C:\Users\ASTEEL\source\repos\AndrewsBookSt C:\Users\ASTEEL\source\repos\AndrewsBookSt

Error List ... Web Publish Activity Package Manager Console Output

Ready Add to Source Control 2

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)  AndrewsBookStore AS

Clone Repository...  Create Git Repository...  Local Repositories  Commit or Stash... Ctrl+0, G Fetch Pull Push New Branch... View Branch History Manage Branches Open Repository in  Manage Remotes... Settings

Debug Any CPU IIS Express    

Create a Git repository

Push to a new remote  Initialize a local Git repository

Local path C:\Users\ASTEEL\source\repos\AndrewsBookStore 

Other  Create a new GitHub repository

Account asteеле-14 (GitHub) 

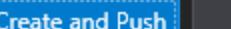
Owner asteèle-14 

Repository Name AndrewsBookStore

Description Enter the description of the repository <Optional>

Private

 Push your code to GitHub
<https://github.com/asteèle-14/AndrewsBookStore>

  Cancel

Error List ... Web Publish Activity Package Manager Console Output  Ready  

Live Share  f 1 project)

Explorer  aspnet-AndrewsBookStore-4AA32BD9-959B-
AndrewsBookStore.csproj
C:\Users\ASTEEL\source\repos\AndrewsBookSt
C:\Users\ASTEEL\source\repos\AndrewsBookSt

Solution Explorer

AndrewsBookStore

ASP.NET Core

Build Your App

Connect To The Cloud

Learn Your IDE

Output

Error List ... Web Publish Activity Package Manager Console Output

Ready

Solution Explorer

AndrewsBookStore

ASP.NET Core

Build Your App

Connect To The Cloud

Learn Your IDE

Output

Error List ... Web Publish Activity Package Manager Console Output

Ready

Setting Up the Project (Part 1)

1.1 Review

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AndrewsBookStore AS Live Share

AndrewsBookStore

Debug Any CPU IIS Express

Overview Connected Services Publish

ASP.NET Core

Learn about the .NET platform, create your first application and extend it to the cloud.

 Build Your App

 Connect To The Cloud

 Learn Your IDE

[ASP.NET Core documentation](#)

[.NET application architecture](#)

[Publish your app to Azure](#)

[Get started with ASP.NET on Azure](#)

[See our productivity guide](#)

[Write code faster](#)

Output

Show output from: Package Manager

Time Elapsed: 00:00:00.8507838

===== Finished =====

Error List ... Web Publish Activity Package Manager Console Output

This item does not support previewing

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'AndrewsBookStore' (1 of 1 project)

AndrewsBookStore

- Connected Services
- Dependencies
- Properties
- wwwroot
- Areas
- Controllers
- Data
- Models
- Views
- appsettings.json
- Program.cs
- Startup.cs

Solution Explorer Team Explorer

Properties

Controllers Folder Properties

Misc

Folder Name	Full Path
Controllers	C:\Users\ASTEEL\source\repos\And

Folder Name
Name of this folder.

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), and a red circular icon with 'AS'.
- Toolbar:** Includes icons for Undo, Redo, Save, Build, Run, and Publish.
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore'. The 'Controllers' folder contains 'HomeController.cs', which is currently selected.
- Properties:** Shows file properties for 'HomeController.cs' under the 'Advanced' tab, including Build Action (C# compiler), Copy to Output Directory (Do not copy), and File Name (HomeController.cs).
- Code Editor:** Displays the 'HomeController.cs' code, which includes imports for various .NET namespaces and a constructor that injects an ILogger<HomeController> dependency.
- Output:** Shows build output from 'Package Manager' indicating a successful build with a duration of 00:00:00.8507838.
- Status Bar:** Shows 'Ready' status at the bottom left and build statistics (0 errors, 0 warnings) at the bottom right.

```
1 [!] using AndrewsBookStore.Models;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.Extensions.Logging;
4 using System;
5 using System.Collections.Generic;
6 using System.Diagnostics;
7 using System.Linq;
8 using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Controllers
11 {
12     public class HomeController : Controller
13     {
14         private readonly ILogger<HomeController> _logger;
15
16         public HomeController(ILogger<HomeController> logger)
```

No issues found

146% | No issues found | Ln: 1 Ch: 1 SPC CRLF

Show output from: Package Manager

Time Elapsed: 00:00:00.8507838

===== Finished =====

Error List ... Web Publish Activity Package Manager Console Output

Ready ↑ 0 ✎ 0 ↗ AndrewsBookStore ↗ master ↗ 2

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), and a red circular icon with 'AS'.
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore'. The 'Controllers' folder contains the selected file, 'HomeController.cs'. Other files include 'Program.cs' and 'Startup.cs'.
- Properties Window:** Opened for 'HomeController.cs', showing settings like Build Action (C# compiler), Copy to Output Directory (Do not copy), and File Name (HomeController.cs).
- Code Editor:** Displays the 'HomeController.cs' file. The class definition is highlighted with a yellow background. The code includes imports for various .NET namespaces and the class definition itself.
- Output Window:** Shows build logs, indicating no issues found and a total time of 00:00:00.8507838.
- Status Bar:** Shows 'Ready' status at the bottom left and navigation icons at the bottom right.

```
1 using AndrewsBookStore.Models;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.Extensions.Logging;
4 using System;
5 using System.Collections.Generic;
6 using System.Diagnostics;
7 using System.Linq;
8 using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Controllers
11 {
12     public class HomeController : Controller
13     {
14         private readonly ILogger<HomeController> _logger;
15
16         public HomeController(ILogger<HomeController> logger)
```

No issues found

Time Elapsed: 00:00:00.8507838
===== Finished =====

Ln: 1 Ch: 1 SPC CRLF

Error List ... Web Publish Activity Package Manager Console Output

Ready ↑ 0 ⚪ 0 ↗ AndrewsBookStore ↗ master ↗ 2

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AndrewBookStore AS Live Share

AndrewsBookStore AndrewsBookStore.Controllers.HomeController _logger HomeController.cs

```
1 using AndrewsBookStore.Models;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.Extensions.Logging;
4 using System;
5 using System.Collections.Generic;
6 using System.Diagnostics;
7 using System.Linq;
8 using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Controllers
11 {
12     public class HomeController : Controller
13     {
14         private readonly ILogger<HomeController> _logger;
15
16         public HomeController(ILogger<HomeController> logger)
```

Home Controller “extends” from the Controller base class and applies action methods

No issues found

Output

Show output from: Package Manager

Time Elapsed: 00:00:00.8507838

===== Finished =====

Solution Explorer

Search Solution Explorer (Ctrl+.)

Solution 'AndrewsBookStore' (1 of 1 project)

- AndrewsBookStore
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - HomeController.cs
 - Data
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs

Solution Explorer Team Explorer

Properties

HomeController.cs File Properties

Advanced

Build Action	C# compiler
Copy to Output Directory	Do not copy
Custom Tool	
Custom Tool Namespace	

Misc

File Name	HomeController.cs
Full Path	C:\Users\ASTEEL\source\repos\And

Build Action

How the file relates to the build and deployment processes.

Error List ... Web Publish Activity Package Manager Console Output

Ready 0 0 AndrewsBookStore master 2

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), a search icon, and AndrewsBookStore.
- Toolbar:** Includes icons for Undo, Redo, Save, Cut, Copy, Paste, Find, Replace, and others.
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore'. It includes Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers (HomeController.cs is selected), Data, Models, Views, appsettings.json, Program.cs, and Startup.cs.
- Editor:** Displays the code for HomeController.cs. The code includes three action methods: Index, Privacy, and Error. The Error method contains a ResponseCache attribute with Duration = 0, Location = ResponseCacheLocation.None, and NoStore = true.
- Status Bar:** Shows 'Ready' status, file counts (0 files, 0 changes), the project name 'AndrewsBookStore', and the current branch 'master'.

```
21     0 references
22     public IActionResult Index()
23     {
24         return View();
25     }
26     0 references
27     public IActionResult Privacy()
28     {
29         return View();
30     }
31     [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
32     0 references
33     public IActionResult Error()
34     {
35         return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceId })
```

No issues found

Output

Show output from: Package Manager

Time Elapsed: 00:00:00.8507838

===== Finished =====

Error List ... Web Publish Activity Package Manager Console Output

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), a search icon, and AndrewsBookStore.
- Toolbar:** Includes icons for Undo, Redo, Save, Open, Copy, Paste, Find, Replace, and others.
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore'. The 'Controllers' folder contains 'HomeController.cs', which is currently selected. Other files include 'Data', 'Models', 'Views', 'appsettings.json', 'Program.cs', and 'Startup.cs'. The 'Properties' and 'wwwroot' folders are also listed.
- Code Editor:** Displays the 'HomeController.cs' file with three action methods: Index(), Privacy(), and Error(). The 'Index()' method has a note: '0 references public IActionResult Index() { return View(); }'. The 'Privacy()' method has a note: '0 references public IActionResult Privacy() { return View(); }'. The 'Error()' method has a note: '0 references [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)] public IActionResult Error() { return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceId }); }'.
- Status Bar:** Shows '146 %' zoom, 'No issues found', 'Ln: 1 Ch: 1 SPC CRLF', and a progress bar.
- Output Window:** Shows 'Show output from: Package Manager' and 'Time Elapsed: 00:00:00.8507838' followed by '===== Finished ====='.
- Bottom Navigation:** Error List, Web Publish Activity, Package Manager Console, Output, and a 'Ready' status indicator.

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), AndrewBookStore, AS.
- Solution Explorer:** Shows the project 'AndrewsBookStore' with its structure:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - HomeController.cs (selected)
 - Data
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs
- Code Editor:** Displays the 'HomeController.cs' file with three highlighted action methods:
 - `public IActionResult Index()`
 - `public IActionResult Privacy()`
 - `public IActionResult Error()`Red arrows point from the explanatory text below to each of these three highlighted methods.
- Output:** Shows the results of a package manager command:

```
Show output from: Package Manager
Time Elapsed: 00:00:00.8507838
===== Finished =====
```
- Bottom Status Bar:** Ready, 0, 0, AndrewBookStore, master, 2.

The explanatory text in the center-right area describes the `IActionResult` interface and its usage in controllers:

The `IActionResult` interface defines what will be the result of an action e.g. Redirect > when you type a URL it hits the action method within the controller (Index action) and process the action ,returning the view for Index found in 'Views > Home > Index.cshtml'

* Every controller has an action. If not defined it will be a GET action method calling those from the controllers GET / POST, and if an API, can be more.

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AndrewsBookStore AS Live Share

AndrewsBookStore

Debug Any CPU IIS Express

Overview Connected Services Publish

ASP.NET Core

Learn about the .NET platform, create your first application and extend it to the cloud.

 Build Your App
[ASP.NET Core documentation](#)
[.NET application architecture](#)

 Connect To The Cloud
[Publish your app to Azure](#)
[Get started with ASP.NET on Azure](#)

 Learn Your IDE
[See our productivity guide](#)
[Write code faster](#)

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

Solution 'AndrewsBookStore' (1 of 1 project)

- AndrewsBookStore
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs

Output

Show output from: Package Manager

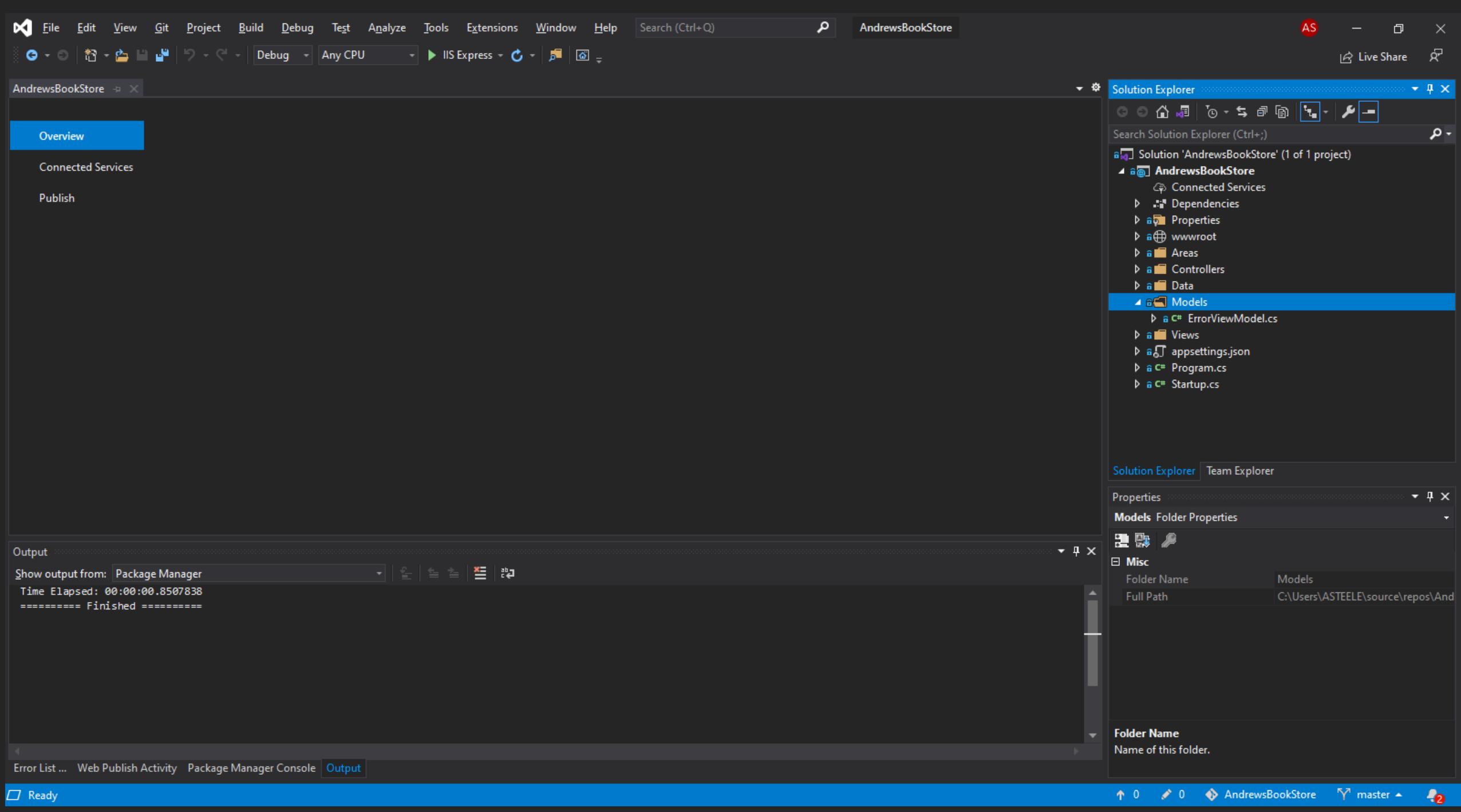
Time Elapsed: 00:00:00.8507838

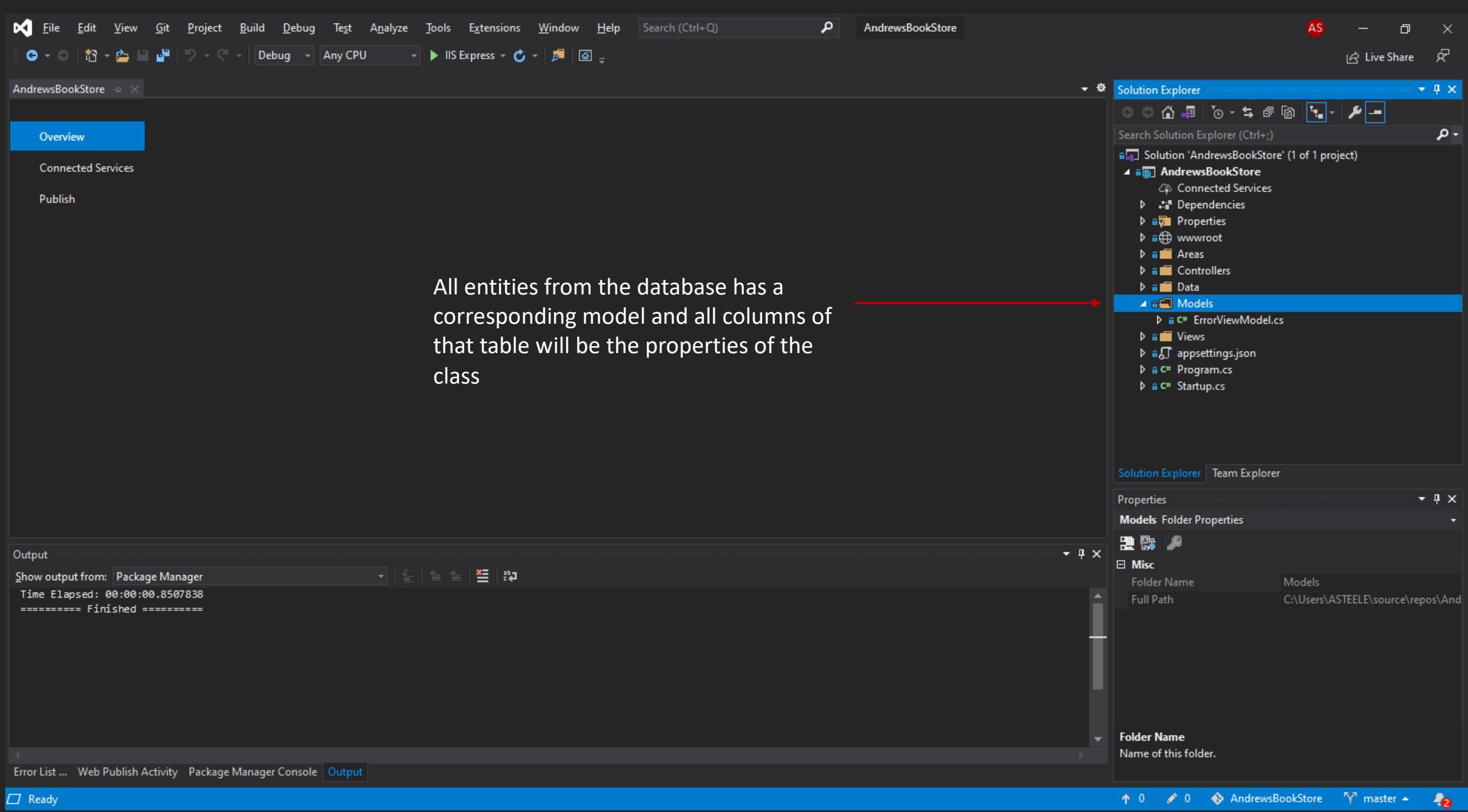
===== Finished =====

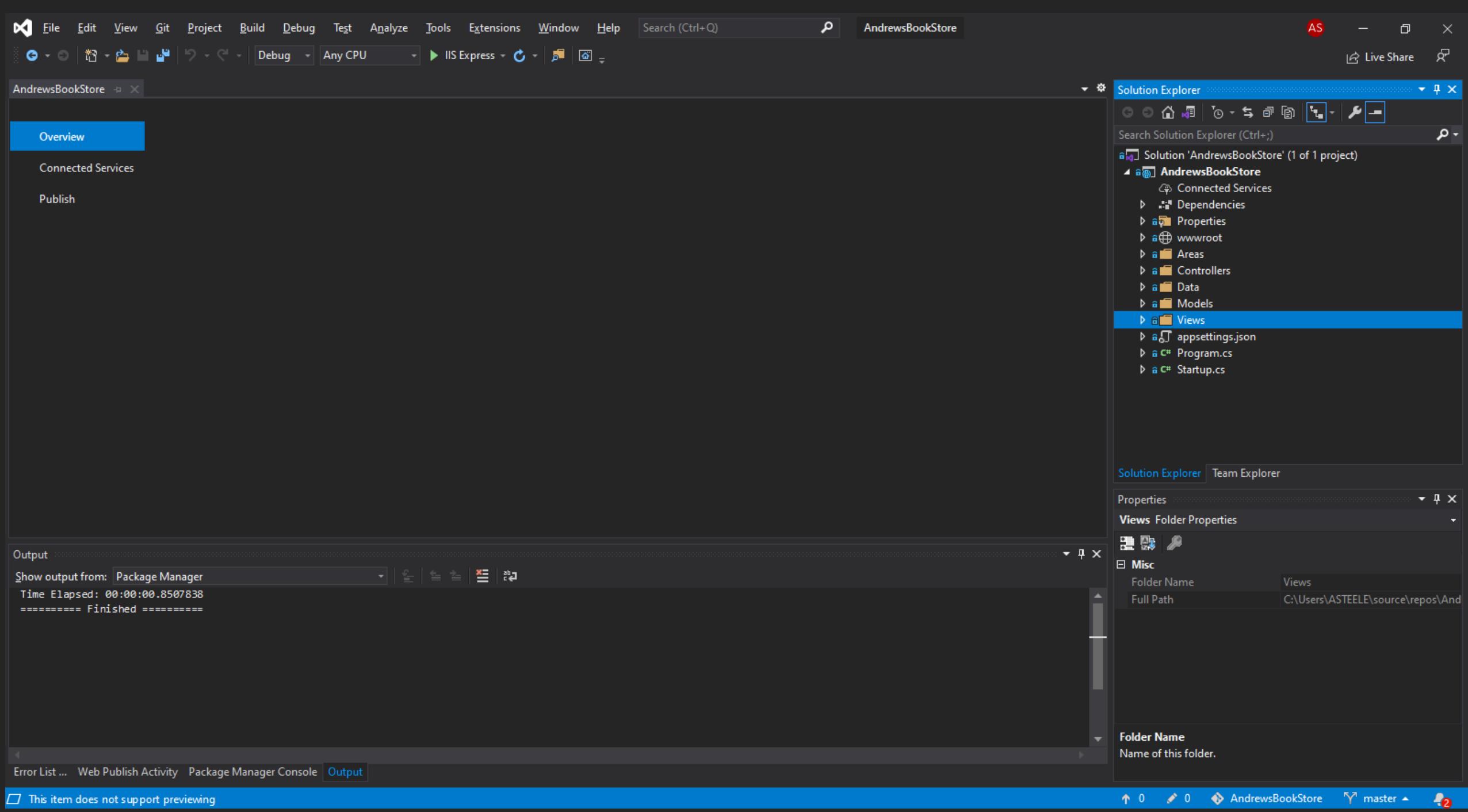
Error List ... Web Publish Activity Package Manager Console Output

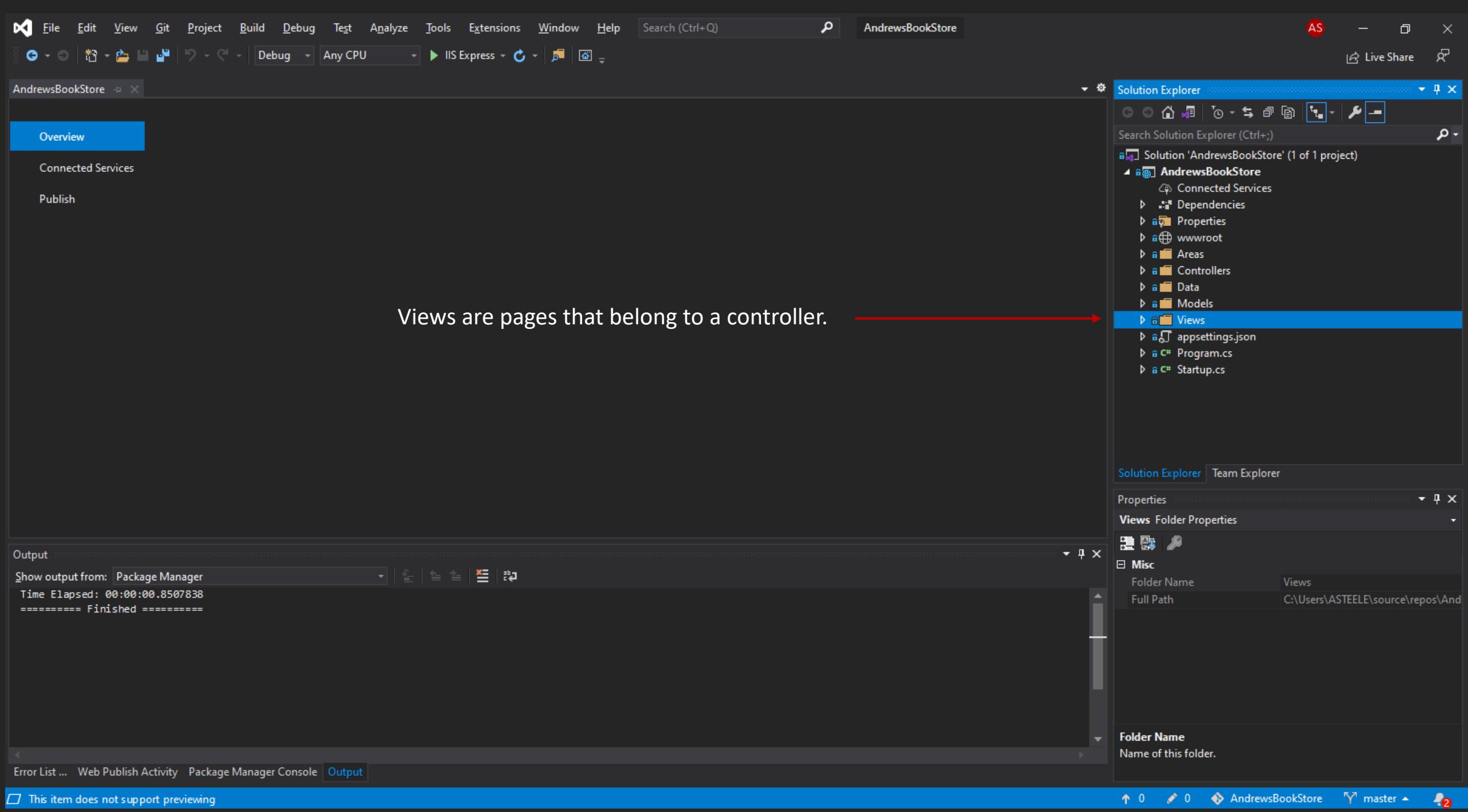
This item does not support previewing

↑ 0 ↖ 0 ↘ AndrewsBookStore ↗ master ↙ 2









File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AndrewsBookStore AS Live Share

AndrewsBookStore

Debug Any CPU IIS Express

Overview Connected Services Publish

ASP.NET Core

Learn about the .NET platform, create your first application and extend it to the cloud.

 Build Your App

 Connect To The Cloud

 Learn Your IDE

[ASP.NET Core documentation](#)

[.NET application architecture](#)

[Publish your app to Azure](#)

[Get started with ASP.NET on Azure](#)

[See our productivity guide](#)

[Write code faster](#)

Output

Show output from: Package Manager

Time Elapsed: 00:00:00.8507838

===== Finished =====

Error List ... Web Publish Activity Package Manager Console Output

This item does not support previewing

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'AndrewsBookStore' (1 of 1 project)

AndrewsBookStore

- Connected Services
- Dependencies
- Properties
- wwwroot
- Areas
- Controllers
- Data
- Models
- Views
- Home
- Shared
- _ViewImports.cshtml
- _ViewStart.cshtml
- appsettings.json
- Program.cs
- Startup.cs

Solution Explorer Team Explorer

Properties Shared Folder Properties

Misc

Folder Name	Shared
Full Path	C:\Users\ASTEEL\source\repos\And

Folder Name
Name of this folder.

AndrewsBookStore

_Layout.cshtml

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>@ViewData["Title"] - AndrewsBookStore</title>
        <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
        <link rel="stylesheet" href="~/css/site.css" />
    </head>
    <body>
        <header>
            <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom" style="border-bottom: 1px solid #ccc; margin-bottom: 10px;">
                <div class="container">
                    <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AndrewsBookStore</a>
                    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                    <div class="navbar-collapse collapse collapse d-sm inline flex justify-content-between">
                        <ul class="navbar-nav" style="list-style-type: none; padding-left: 0; margin: 0; font-size: 0.9em; font-weight: bold;">
                            <li><a href="#">Home</a></li>
                            <li><a href="#">Books</a></li>
                            <li><a href="#">Authors</a></li>
                            <li><a href="#">About</a></li>
                        </ul>
                    </div>
                </div>
            </nav>
            <div class="container">
                <partial-view>
            </div>
        </header>
        <div class="container">
            <partial-view>
        </div>
    </body>

```

146% No issues found

Output

Show output from: IntelliSense

Using TypeScript 4.0 for IntelliSense.

[ProjectHost] Successfully installed TypeScript typings @types/jquery@ts4.0

Solution Explorer

Search Solution Explorer (Ctrl+.)

Solution 'AndrewsBookStore' (1 of 1 project)

- AndrewsBookStore
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml

Solution Explorer Team Explorer

Properties

_Layout.cshtml File Properties

Advanced

Build Action	Content
Copy to Output Directory	Do not copy
Custom Tool	
Custom Tool Namespace	

Misc

File Name	_Layout.cshtml
Full Path	C:\Users\ASTEEL\source\repos\And...

Build Action

How the file relates to the build and deployment processes.

Error List ... Web Publish Activity Package Manager Console Output

Ready ↑ 0 ⌂ 0 ↗ AndrewsBookStore ↗ master ↗ 2

The screenshot shows the Microsoft Visual Studio interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Solution Explorer:** Shows the project 'AndrewsBookStore' structure:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml
- Properties Window:** Shows file properties for '_Layout.cshtml'.
 - Advanced:** Build Action (Content), Copy to Output Directory (Do not copy), Custom Tool, Custom Tool Namespace.
 - Misc:** File Name (_Layout.cshtml), Full Path (C:\Users\ASTEEL\source\repos\And...).
- Output Window:** Shows TypeScript 4.0 IntelliSense and successful installation of jQuery @types/jQuery@ts4.0.
- Code Editor:** Displays the content of the '_Layout.cshtml' file, which includes the HTML structure for a master page with Bootstrap styling and navigation components.
- Status Bar:** Shows the status 'Ready' and other system information.

A red arrow points from the explanatory text at the bottom to the '_Layout.cshtml' item in the Solution Explorer.

_Layout.cshtml is the “master page” of the application, holding references to the CSS (Bootstrap) in the header...

The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore'. The file '_Layout.cshtml' is selected and highlighted in blue. A red arrow points from the explanatory text below to this selection.
- Properties Window:** Shows file properties for '_Layout.cshtml' under the 'Content' tab, indicating 'Do not copy'.
- Output Window:** Shows TypeScript 4.0 IntelliSense and a message about successfully installing @types/jquery@ts4.0.
- Code Editor:** Displays the content of '_Layout.cshtml'. The code includes Bootstrap CSS imports and a navigation bar.
- Status Bar:** Shows 'Ready' status and various deployment metrics.

_Layout.cshtml is the “master page” of the application, holding references to the CSS (Bootstrap) in the header...

The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q) with a magnifying glass icon.
- Project Status:** AndrewsBookStore (AS)
- Toolbars:** Standard toolbar with icons for Save, Undo, Redo, Cut, Copy, Paste, Find, etc.
- Debugging:** Debug dropdown set to "Any CPU", IIS Express selected.
- Code Editor:** The `_Layout.cshtml` file is open, showing ASP.NET Razor code. A red arrow points from the highlighted line `@await RenderSectionAsync("Scripts", required: false)` to the corresponding entry in the Solution Explorer.
- Solution Explorer:** Shows the project structure for "AndrewsBookStore". The `_Layout.cshtml` file is selected and highlighted in blue.
- Properties:** The properties for `_Layout.cshtml` are displayed, including:
 - Advanced:** Build Action: Content, Copy to Output Directory: Do not copy.
 - Misc:** File Name: `_Layout.cshtml`, Full Path: `C:\Users\ASTEEL\source\repos\And...`
- Output:** Shows TypeScript 4.0 IntelliSense and successful installation of @types/jquery@ts4.0.
- Bottom Navigation:** Error List, Web Publish Activity, Package Manager Console, Output.
- Bottom Status:** Ready, 0 errors, 0 warnings, AndrewsBookStore, master branch, 2 notifications.

The screenshot shows the Visual Studio IDE interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), AndrewBookStore, AS.
- Toolbar:** Debug, Any CPU, IIS Express.
- Code Editor:** The file `_Layout.cshtml` is open. The code includes a header section, a main container with a render body, and a footer with a copyright notice and links. Below the footer, there are several script tags for jQuery, Bootstrap, and a site-specific JavaScript file. A callout arrow points from the text "...and JavaScript scripts below the footer." to the footer section of the code.
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore'. The `_Layout.cshtml` file is selected and highlighted in blue.
- Properties:** The properties for `_Layout.cshtml` are displayed, showing it has a Content build action and is set to do not copy.
- Output:** Shows TypeScript 4.0 IntelliSense and a message about successfully installing @types/jquery@ts4.0.
- Bottom Bar:** Error List..., Web Publish Activity, Package Manager Console, Output, Ready.

...and JavaScript scripts below the footer.

The screenshot shows a Microsoft Visual Studio interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q) with a magnifying glass icon.
- Solution Explorer:** Shows the project 'AndrewsBookStore' structure:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml**
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml
- Properties Window:** Shows file properties for '_LoginPartial.cshtml'.
 - Advanced:** Build Action: Content, Copy to Output Directory: Do not copy.
 - Misc:** File Name: _LoginPartial.cshtml, Full Path: C:\Users\ASTEEL\source\repos\And...
- Code Editor:** Displays the '_LoginPartial.cshtml' file content:

```
1  @using Microsoft.AspNetCore.Identity
2  @inject SignInManager<IdentityUser> SignInManager
3  @inject UserManager<IdentityUser> UserManager
4
5  <ul class="navbar-nav">
6      @if (SignInManager.IsSignedIn(User))
7      {
8          <li class="nav-item">
9              <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Manage/Index" title="Manage" href="#">@UserManager.GetUserName(User)</a>
10         </li>
11         <li class="nav-item">
12             <form class="form-inline" asp-area="Identity" asp-page="/Account/Logout" asp-route-returnUrl="#">
13                 <button type="submit" class="nav-link btn btn-link text-dark">Logout</button>
14             </form>
15         </li>
16     }
17     else
18     {
19         <li class="nav-item">
20             <a class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Login" title="Log in" href="#">Log in</a>
21         </li>
22     }
23 </ul>
```
- Output Window:** Shows TypeScript 4.0 IntelliSense and a message about successfully installing jQuery @types/jQuery@ts4.0.
- Status Bar:** Ready, 0 errors, 0 warnings, AndrewsBookStore, master branch, 2 notifications.

The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore'. The file '_LoginPartial.cshtml' is selected and highlighted with a red arrow.
- Code Editor:** Displays the C# code for '_LoginPartial.cshtml'. The code includes imports for Microsoft.AspNetCore.Identity, and injects SignInManager<IdentityUser> and UserManager<IdentityUser>. It contains logic for signed-in users, including a form with a 'Logout' button, and a section for non-signed-in users.
- Output Window:** Shows TypeScript 4.0 IntelliSense and a message about successfully installing @types/jquery@ts4.0.
- Status Bar:** Shows 'Ready' status.

A red arrow points from the text below to the '_LoginPartial.cshtml' file in the Solution Explorer.

_LoginPartial.cshtml is used for buttons to login and register

The screenshot shows a Microsoft Visual Studio interface with a dark theme. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), and a power icon. The toolbar below has icons for Undo, Redo, Save, Cut, Copy, Paste, Find, Replace, and others. The title bar says "AndrewsBookStore". The main code editor window displays the file `_ValidationScriptsPartial.cshtml` with the following content:

```
1 <script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>
2 <script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>
3
```

The Solution Explorer window on the right lists the project structure for "AndrewsBookStore":

- Connected Services
- Dependencies
- Properties
- wwwroot
- Areas
- Controllers
- Data
- Models
- Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml**
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml

The "Properties" window shows the file properties for `_ValidationScriptsPartial.cshtml`:

Advanced	Build Action	Content
	Copy to Output Directory	Do not copy
	Custom Tool	
	Custom Tool Namespace	
Misc	File Name	_ValidationScriptsPartial.cshtml
	Full Path	C:\Users\ASTEEL\source\repos\And

At the bottom, the status bar shows "Ready" and various build counts: 0 errors, 0 warnings, 0 files, and 2 notifications.

The screenshot shows a Visual Studio interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Solution Explorer:** Shows the project 'AndrewsBookStore' structure:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml
- Properties Window:** Shows the file properties for '_ValidationScriptsPartial.cshtml'.
 - Advanced:** Build Action: Content, Copy to Output Directory: Do not copy.
 - Misc:** File Name: _ValidationScriptsPartial.cshtml, Full Path: C:\Users\ASTEEL\source\repos\And...
- Output Window:** Shows TypeScript 4.0 IntelliSense and successful installation of @types/jquery@ts4.0.
- Editor:** Displays the code for '_ValidationScriptsPartial.cshtml':

```
1 <script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>
2 <script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>
3
```
- Status Bar:** Ready, 0 errors, 0 warnings, AndrewsBookStore, master.

A red arrow points from the text "VlaidationScriptsPartial.cshtml for JQuery" to the file entry in the Solution Explorer.

_ValidationScriptsPartial.cshtml for JQuery
responses to validate the model states on
the client side

Screenshot of the Visual Studio IDE interface showing the AndrewsBookStore project.

Top Bar:

- File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help
- Search (Ctrl+Q)
- AndrewsBookStore
- AS
- Live Share

Toolbar:

- Debug (selected), Any CPU, IIS Express

Solution Explorer:

- Solution 'AndrewsBookStore' (1 of 1 project)
 - AndrewsBookStore
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml (selected)
 - _ViewImports.cshtml
 - _ViewStart.cshtml

Properties:

- Error.cshtml File Properties
 - Advanced
 - Build Action: Content
 - Copy to Output Directory: Do not copy
 - Custom Tool
 - Custom Tool Namespace
 - Misc
 - File Name: Error.cshtml
 - Full Path: C:\Users\ASTEEL\source\repos\And...

Output:

- Show output from: IntelliSense
- Using TypeScript 4.0 for IntelliSense.
- [ProjectHost] Successfully installed TypeScript typings @types/jquery@ts4.0

Status Bar:

- Ready
- 0 0 0
- AndrewsBookStore
- master
- 2

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AndrewBookStore AS Live Share

AndrewsBookStore Error.cshtml

```
1 @model ErrorViewModel
2 @{
3     ViewData["Title"] = "Error";
4 }
5
6 <h1 class="text-danger">Error.</h1>
7 <h2 class="text-danger">An error occurred while processing your request.</h2>
8
9 @if (Model.ShowRequestId)
10 {
11     <p>
12         <strong>Request ID:</strong> <code>@Model.RequestId</code>
13     </p>
14 }
15
16 <h3>Development Mode</h3>
17 <p>
18     Swapping to <strong>Development</strong> environment will display more detailed information about
19 /n<
```

146% No issues found

Output

Show output from: IntelliSense

Using TypeScript 4.0 for IntelliSense.

[ProjectHost] Successfully installed TypeScript typings @types/jquery@ts4.0

Error.cshtml creates a generic error view (404)

Solution Explorer

Search Solution Explorer (Ctrl+.)

Solution 'AndrewsBookStore' (1 of 1 project)

AndrewsBookStore

- Connected Services
- Dependencies
- Properties
- wwwroot
- Areas
- Controllers
- Data
- Models
- Views
- Home
- Shared
- _Layout.cshtml
- _LoginPartial.cshtml
- _ValidationScriptsPartial.cshtml
- Error.cshtml
- _ViewImports.cshtml
- _ViewStart.cshtml

Properties

Error.cshtml File Properties

Advanced

Build Action	Content
Copy to Output Directory	Do not copy
Custom Tool	
Custom Tool Namespace	

Misc

File Name	Error.cshtml
Full Path	C:\Users\ASTEEL\source\repos\And

Build Action

How the file relates to the build and deployment processes.

Ready 0 0 AndrewsBookStore master 2

The screenshot shows the Visual Studio IDE interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), Andrew's Book Store, AS.
- Toolbar:** Debug (selected), Any CPU, IIS Express, Live Share.
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore'. The 'Views' folder contains '_Layout.cshtml', '_LoginPartial.cshtml', '_ValidationScriptsPartial.cshtml', 'Error.cshtml', '_ViewImports.cshtml', and '_ViewStart.cshtml'. The '_ViewImports.cshtml' file is currently selected.
- Properties Window:** Shows file properties for '_ViewImports.cshtml'. Under 'Advanced', 'Build Action' is set to 'Content' and 'Copy to Output Directory' is set to 'Do not copy'. Under 'Misc', 'File Name' is '_ViewImports.cshtml' and 'Full Path' is 'C:\Users\ASTEEL\source\repos\And...'.
- Output Window:** Shows TypeScript 4.0 IntelliSense and a message about successfully installing @types/jquery@ts4.0.
- Status Bar:** Ready, 0 errors, 0 warnings, Andrew's Book Store, master, 2 notifications.

```
1 @using AndrewsBookStore
2 @using AndrewsBookStore.Models
3 @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
4
```

The screenshot shows the Visual Studio IDE interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), AndrewBookStore, AS.
- Toolbar:** Debug, Any CPU, IIS Express.
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore'. The file '_ViewImports.cshtml' is selected and highlighted in blue.
- Properties Window:** Shows the properties for '_ViewImports.cshtml'. Under the 'Advanced' section, 'Build Action' is set to 'Content' and 'Copy to Output Directory' is set to 'Do not copy'. Under the 'Misc' section, 'File Name' is '_ViewImports.cshtml' and 'Full Path' is 'C:\Users\ASTEEL\source\repos\And...'.
- Output Window:** Shows 'No issues found' and output from 'IntelliSense' and 'TypeScript 4.0'.
- Code Editor:** Displays the content of '_ViewImports.cshtml':

```
1 @using AndrewsBookStore
2 @using AndrewsBookStore.Models
3 @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
4
```
- Bottom Status Bar:** Ready, 0 errors, 0 warnings, AndrewBookStore, master, 2 notifications.

A red arrow points from the text in the bottom left towards the '_ViewImports.cshtml' file in the Solution Explorer.

_ViewImports.cshtml for tag helpers
and where custom tag helpers can be
created

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), Andrew's Book Store, AS.
- Solution Explorer:** Shows the 'AndrewsBookStore' project structure:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml
- Code Editor:** Displays the content of _ViewStart.cshtml:

```
1  @{
2      Layout = "_Layout";
3  }
4
```
- Properties Window:** Shows the properties for _ViewStart.cshtml:

Advanced	Build Action	Content
	Copy to Output Directory	Do not copy
	Custom Tool	
	Custom Tool Namespace	
Misc	File Name	_ViewStart.cshtml
	Full Path	C:\Users\ASTEEL\source\repos\And

Build Action: How the file relates to the build and deployment processes.
- Output Window:** Shows TypeScript 4.0 IntelliSense and a message about successfully installing jQuery @types/jQuery@ts4.0.
- Status Bar:** Ready, 0 errors, 0 warnings, Andrew's Book Store, master, 2 notifications.

The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q) with a magnifying glass icon.
- Solution Explorer:** Shows the project 'AndrewsBookStore' structure:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models
 - Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml
- Code Editor:** Displays the content of the _ViewStart.cshtml file:

```
1  @{
2      Layout = "_Layout";
3  }
```
- Output:** Shows TypeScript 4.0 IntelliSense and a message about successfully installing jQuery @types/jQuery@ts4.0.
- Properties:** Shows the properties for _ViewStart.cshtml, including Build Action (Content), Copy to Output Directory (Do not copy), and File Name (_ViewStart.cshtml).
- Annotations:** A red arrow points from the text "will be the master page" to the _ViewStart.cshtml file in the Solution Explorer.
- Text at Bottom:** _ViewStart.cshtml will be the master page for all of the views within the project folder.

Setting Up the Project (Part 1)

1.2 Debugging

The screenshot shows a Microsoft Visual Studio interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Project Status:** AndrewsBookStore (AS) - Live Share.
- Code Editor:** The main window displays the `HomeController.cs` file from the `AndrewsBookStore.Controllers` namespace. The code includes imports for `AndrewsBookStore.Models`, `Microsoft.AspNetCore.Mvc`, `Microsoft.Extensions.Logging`, and various System namespaces. It defines a `HomeController` class that inherits from `Controller`. The constructor takes an `ILogger<HomeController>` dependency named `logger`.
- Solution Explorer:** Shows the project structure for `AndrewsBookStore`, including `Connected Services`, `Dependencies`, `Properties`, `wwwroot`, `Areas`, and `Controllers` (which contains `HomeController.cs`). Other files like `Data`, `Models`, `Views`, `appsettings.json`, `Program.cs`, and `Startup.cs` are also listed.
- Properties Window:** The `HomeController.cs` properties are shown under `File Properties`.
 - Advanced:** Build Action: C# compiler; Copy to Output Directory: Do not copy.
 - Misc:** File Name: `HomeController.cs`; Full Path: `C:\Users\ASTEEL\source\repos\And...`
- Status Bar:** Shows the status `Ready` and other build-related metrics.

In the HomeController.cs add two breakpoints on Index and Privacy and run the program

```
1 using AndrewsBookStore.Models;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.Extensions.Logging;
4 using System;
5 using System.Collections.Generic;
6 using System.Diagnostics;
7 using System.Linq;
8 using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Controllers
11 {
12     public class HomeController : Controller
13     {
14         private readonly ILogger<HomeController> _logger;
15
16         public HomeController(ILogger<HomeController> logger)
```

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), a search icon, and a red circular badge with 'AS'.
- Solution Explorer:** Shows the project 'AndrewsBookStore' with its structure:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - HomeController.cs (selected)
 - Data
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs
- Code Editor:** Displays the 'HomeController.cs' file with three action methods: Index, Privacy, and Error. The Error method includes a response cache attribute: `[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]`.
- Properties Window:** Shows the properties for 'HomeController.cs' under the 'Advanced' tab:

Build Action	C# compiler
Copy to Output Directory	Do not copy
Custom Tool	
Custom Tool Namespace	

Under the 'Misc' tab:

File Name	HomeController.cs
Full Path	C:\Users\ASTEEL\source\repos\And...

Under the 'Build Action' section:

How the file relates to the build and deployment processes.
- Output Window:** Shows 'No issues found'.
- Bottom Navigation:** Error List, Web Publish Activity, Package Manager Console, Output (selected).
- Status Bar:** Ready, 0 errors, 0 warnings, AndrewsBookStore, master, and a notifications icon with a red '2'.

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), Andrew's Book Store, AS.
- Toolbar:** Includes icons for Save, Undo, Redo, Debug (selected), Any CPU, IIS Express, Refresh, and various navigation and search tools.
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore' (1 of 1 project). It includes Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers (HomeController.cs is selected), Data, Models, Views, appsettings.json, Program.cs, and Startup.cs.
- Code Editor:** Displays the 'HomeController.cs' file with three methods: Index(), Privacy(), and Error(). The Index() and Privacy() methods return a View(). The Error() method returns a View() with a new ErrorViewModel model. A note above the Error() method indicates it uses ResponseCache attributes.
- Status Bar:** Shows 146%, No issues found, Ln: 27, Ch: 9, SPC, CRLF.
- Output Tab:** Shows output from Debug.
- Bottom Status Bar:** Error List ..., Web Publish Activity, Package Manager Console, Output, Ready, 0, 0, Andrew's Book Store, master, 2.

A screenshot of the Microsoft Visual Studio IDE interface. The main window shows the code editor with the file `HomeController.cs` open. The code defines three actions: `Index()`, `Privacy()`, and `Error()`. The `Privacy()` action includes a `[ResponseCache]` attribute with parameters `Duration = 0`, `Location = ResponseCacheLocation.None`, and `NoStore = true`. The `Error()` action returns a view for an `ErrorViewModel` with `RequestId` populated from `Activity.Current?.Id ?? HttpContext.TraceId`. The Solution Explorer panel on the right shows the project structure for 'AndrewsBookStore' with files like `Startup.cs`, `Program.cs`, and `HomeController.cs` under the `Controllers` folder. The status bar at the bottom indicates the repository is 'master' with 2 changes.

```
21     0 references
22     public IActionResult Index()
23     {
24         return View();
25     }
26     0 references
27     public IActionResult Privacy()
28     {
29         return View();
30     }
31     [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
32     0 references
33     public IActionResult Error()
34     {
35         return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceId });
36     }
```

Screenshot of the Visual Studio IDE interface showing a diagnostic session.

Top Bar: File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), AndrewsBookStore, AS, Live Share.

Toolbar: Process: [23956] iisexpress.exe, Lifecycle Events, Thread: [22688] Worker Thread, Stack Frame: AndrewsBookStore.Controllers.HomeCon.

Code Editor: HomeController.cs (AndrewsBookStore) - Index() method.

```
20
21     public IActionResult Index()
22     {
23         return View();
24     }
25
26     public IActionResult Privacy()
27     {
28         return View();
29     }
30
31     [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
32     public IActionResult Error()
33     {
34         return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
35     }
36
37 }
```

Diagnostic Tools: Diagnostics session: 2 seconds (2.183 s selected).

- Events:** Shows a timeline from 0s to 10s with one event marked.
- Process Memory (MB):** Shows memory usage starting at 0 MB and rising to approximately 94 MB.
- CPU (% of all processors):** Shows CPU usage at 100%.

Toolbars: Solution Explorer, Git Changes.

Bottom Status Bar: Watch 1, Call Stack, Breakpoints, Exception Settings, Command Window, Immediate Window, Output, Error List, Autos, Locals, Watch 1, Ready.

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AndrewsBookStore AS

Process: [23956] iisexpress.exe Lifecycle Events Thread: [22688] Worker Thread Stack Frame: AndrewsBookStore.Controllers.HomeCon

Live Share

HomeController.cs

AndrewsBookStore

AndrewsBookStore.Controllers.HomeController

Index()

```
20
21     0 references
22     public IActionResult Index()
23     {
24         return View();
25     }
26
27     0 references
28     public IActionResult Privacy()
29     {
30         return View();
31     }
32
33     [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
34     0 references
35     public IActionResult Error()
36     {
37         return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
38     }

```

Diagnostic Tools

Diagnostics session: 2 seconds (2.183 s selected)

Events

Process Memory (MB)

CPU (% of all processors)

Summary Events Memory Usage CPU Usage

Events

Show Events (1 of 1)

Application Insights Events (0 of 0)

Memory Usage

Take Snapshot

CPU Usage

Record CPU Profile

Watch 1

Search (Ctrl+E)

Search Depth: 3

Name Value Type

Add item to watch

Command Window

Call Stack Breakpoints Exception Settings Command Window Immediate Window Output Error List

Autos Locals Watch 1

Ready ↑ 0 ↪ 0 AndrewsBookStore ↑ master ▲ 2

Home Page - AndrewsBookStore X +

← → ⌂ https://localhost:44304

AndrewsBookStore Home Privacy Register Login

Welcome

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

© 2021 - AndrewsBookStore - [Privacy](#)

Home Page - AndrewsBookStore X +

← → ⌂ https://localhost:44304

AndrewsBookStore Home Privacy Register Login

>Welcome

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

© 2021 - AndrewsBookStore - [Privacy](#)



Screenshot of the Visual Studio IDE interface showing a diagnostic session.

Top Bar: File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), AndrewsBookStore, AS, Live Share.

Toolbar: Debug, Continue, Stack Frame, Application Insights.

Process: [23956] iisexpress.exe, Lifecycle Events, Thread: [25452] Worker Thread, Stack Frame: AndrewsBookStore.Controllers.HomeCon.

Diagnostic Tools: Shows a timeline from 1:40 to 1:54 minutes. It includes sections for Events, Process Memory (MB), and CPU (% of all processors). A summary table is also present.

Summary	Events	Memory Usage	CPU Usage
102 MB	1:40	0	0
102 MB	1:54	0	0

Code Editor: HomeController.cs

```
20
21     0 references
22     public IActionResult Index()
23     {
24         return View();
25     }
26
27     0 references
28     public IActionResult Privacy()
29     {
30         [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
31         0 references
32         public IActionResult Error()
33         {
34             return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
35         }
36     }
37 }
```

Watch 1: Search (Ctrl+E), Search Depth: 3, Add item to watch.

Command Window: >

Bottom Navigation: Autos, Locals, Watch 1, Call Stack, Breakpoints, Exception Settings, Command Window, Immediate Window, Output, Error List.

Status Bar: Ready, 0, 0, AndrewsBookStore, master, 2.

This screenshot shows a Microsoft Visual Studio interface during a diagnostic session. The top navigation bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), and AndrewsBookStore. The toolbar contains various icons for file operations like Open, Save, and Print.

The main window displays the code for `HomeController.cs` under the project `AndrewsBookStore`. The current method is `Privacy()`, which has a duration of `≤ 112,608ms elapsed`. A red arrow points to the `Continue` button in the toolbar.

The Diagnostic Tools window on the right shows a timeline from 1:40 to 1:54 minutes. It includes sections for Events, Process Memory (MB), and CPU (% of all processors). The Events section lists one event. The Process Memory section shows memory usage starting at 102 MB. The CPU Usage section shows CPU usage at 0%.

At the bottom, the Watch 1 window is open, showing a table with columns for Name, Value, and Type. The Command Window shows the prompt `>`. The status bar at the bottom indicates the session is `Ready`.

```
HomeController.cs
AndrewsBookStore.Controllers.HomeController
Privacy()

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
public IActionResult Privacy()
{
    return View();
}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
public IActionResult Error()
{
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
}
```

Privacy Policy - AndrewsBookSto X +

← → ⌂ https://localhost:44304/Home/Privacy

AndrewsBookStore Home Privacy Register Login

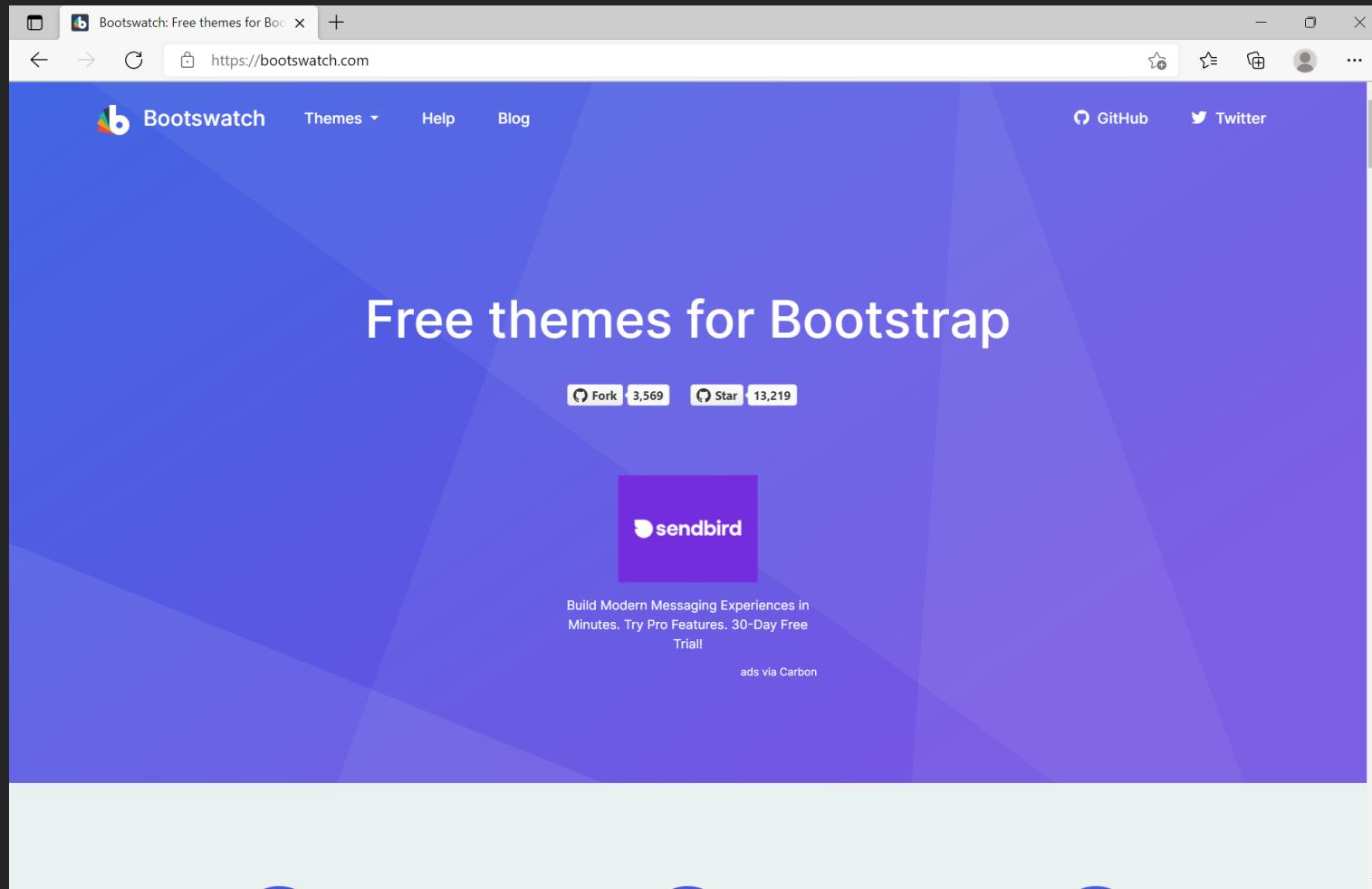
Privacy Policy

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

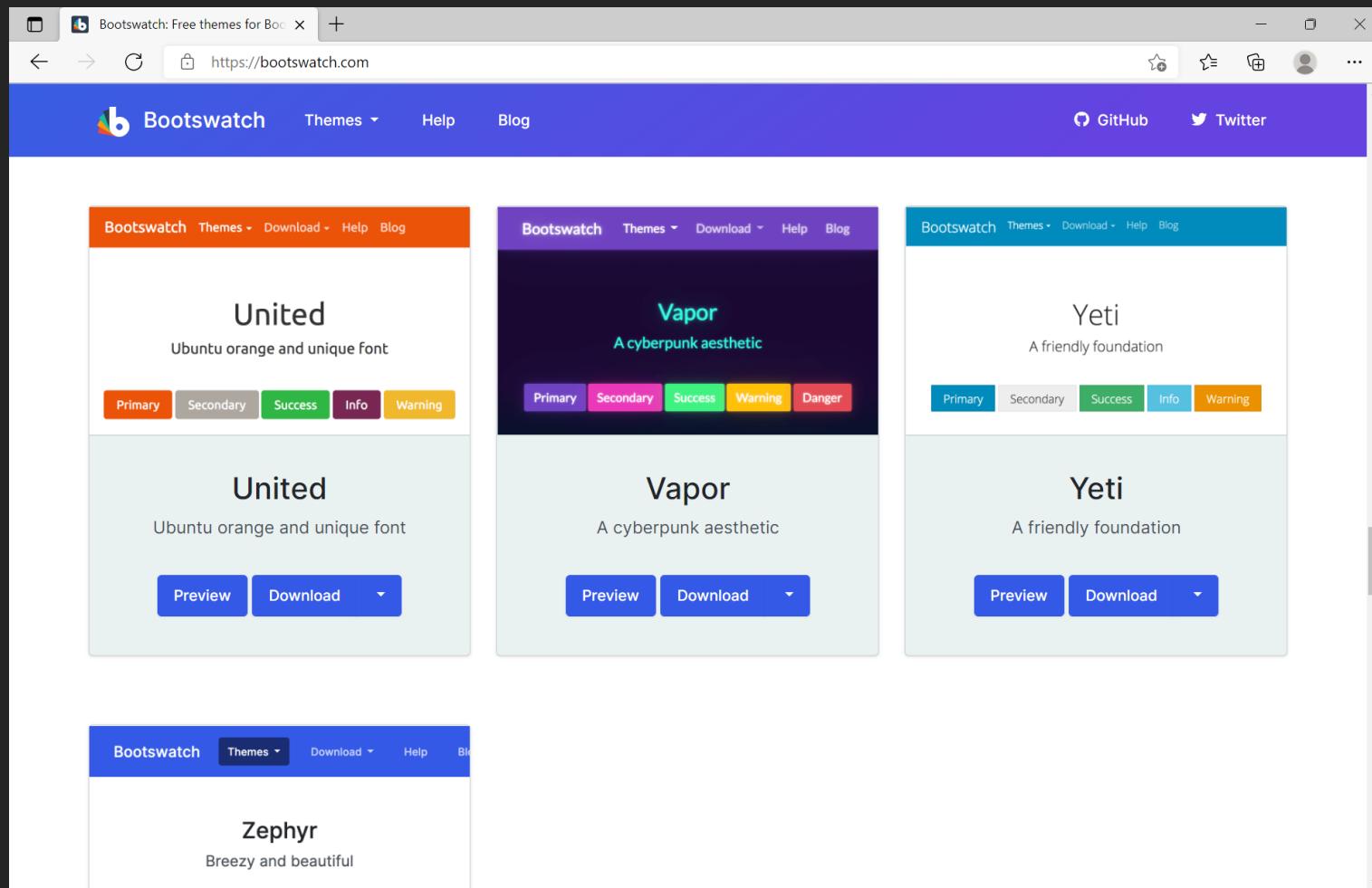
© 2021 - AndrewsBookStore - [Privacy](#)

Setting Up the Project (Part 1)

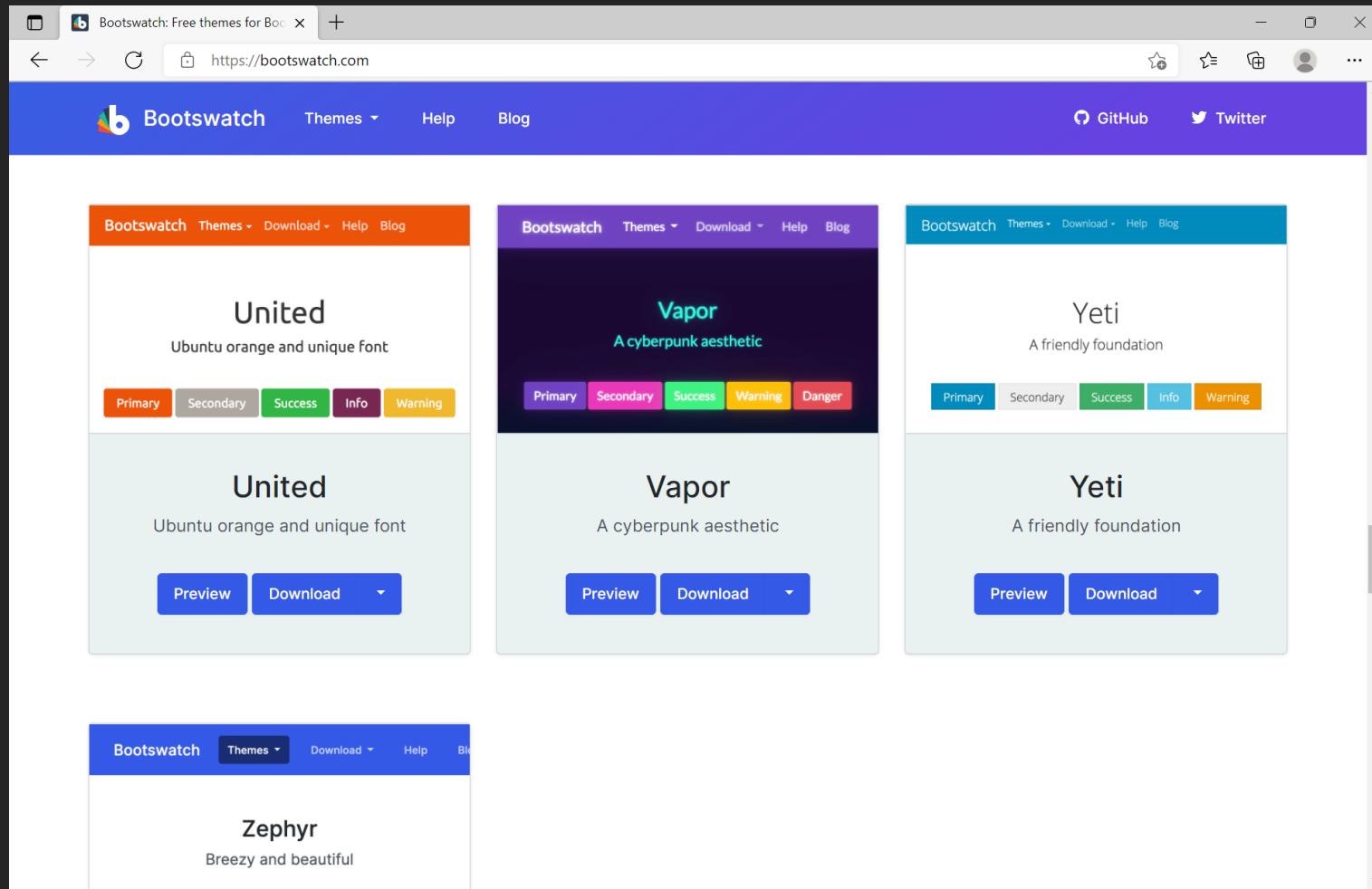
1.3 Bootstrap



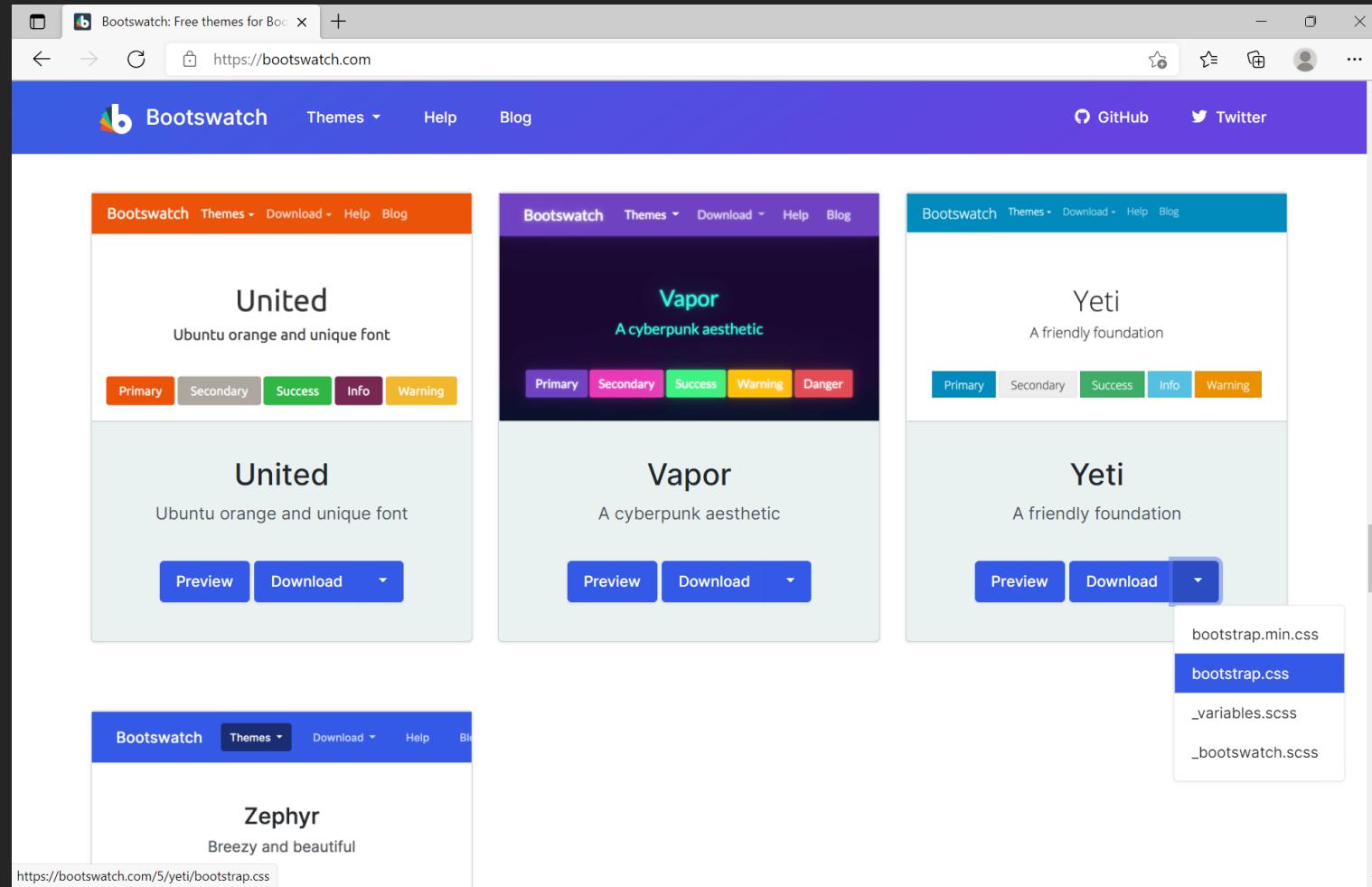
- Go to [Bootswatch.com](https://bootswatch.com)



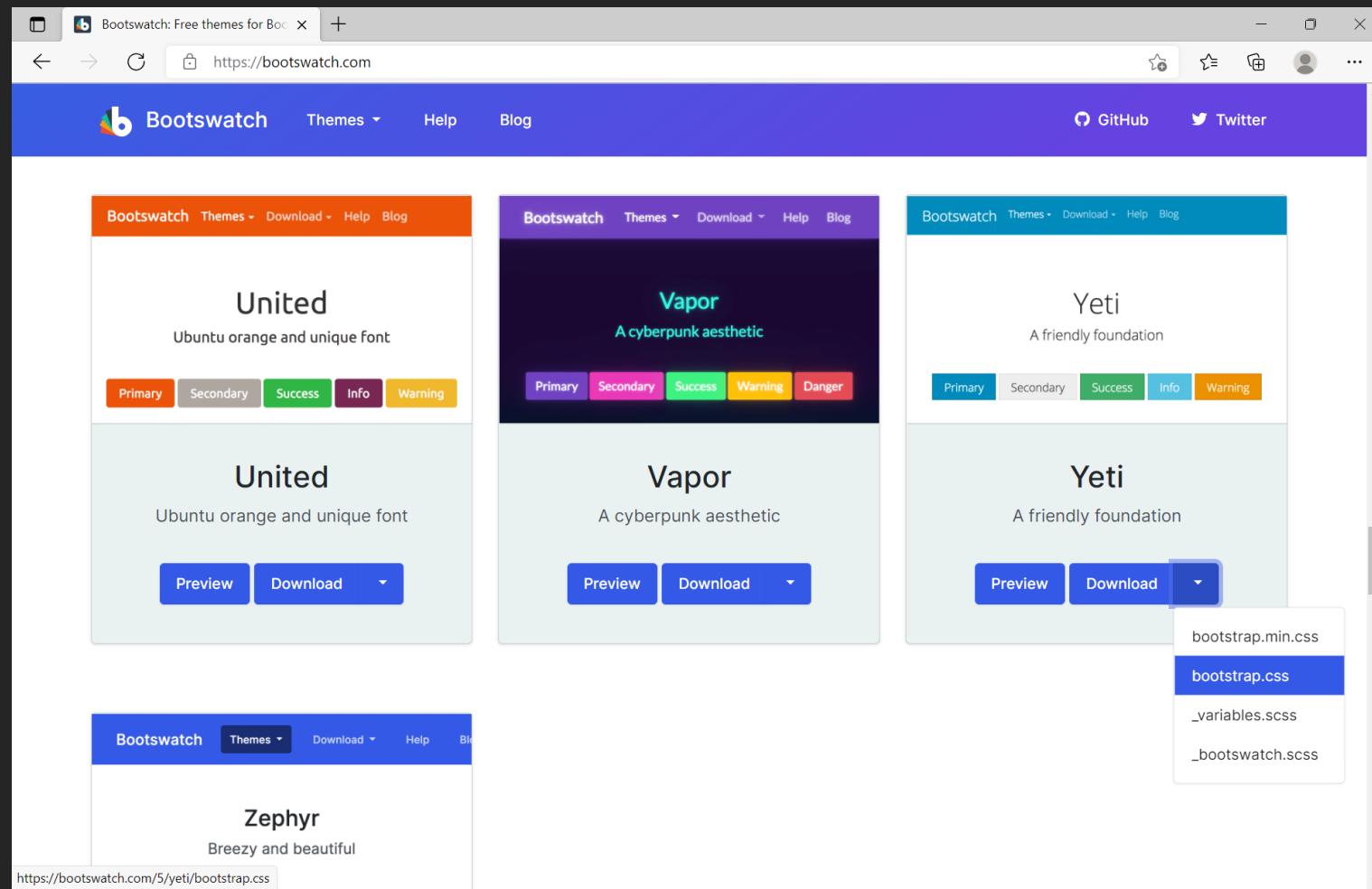
- Go to Bootswatch.com



- Go to [Bootswatch.com](https://bootswatch.com)
- Select a Theme (e.g. Yeti)



- Go to [Bootswatch.com](https://bootswatch.com)
- Select a Theme (e.g. Yeti)



- Go to [Bootswatch.com](https://bootswatch.com)
- Select a Theme (e.g. Yeti)
- Download bootstrap.css

- Go to wwwroot > replace the existing bootstrap.css found in:
lib > bootstrap > dist > css

The screenshot shows the Visual Studio IDE interface. The main window displays the contents of the `bootstrap.css` file, which includes Bootstrap's license information and some custom color definitions. The Solution Explorer on the right shows the project structure, with the `bootstrap.css` file selected. The Properties window is also visible, showing details like Build Action and File Name.

```
/*
 * Bootswatch v5.1.3
 * Homepage: https://bootswatch.com
 * Copyright 2012-2021 Thomas Park
 * Licensed under MIT
 */
/*
 * Bootstrap v5.1.3 (https://getbootstrap.com/)
 * Copyright 2011-2021 The Bootstrap Authors
 * Copyright 2011-2021 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 */
@import url("https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,700;1,300;1,400&display=swap");
:root {
    --bs-blue: #008cba;
    --bs-indigo: #6610f2;
    --bs-purplic: #6f42c1;
}
```

Solution Explorer:

- AndrewsBookStore
- Connected Services
- Dependencies
- Properties
- wwwroot
 - css
 - js
 - lib
 - bootstrap
 - dist
 - css
 - bootstrap.css
 - bootstrap-grid.css
 - bootstrap-reboot.css
 - js
 - LICENSE
 - jquery
 - jquery-validation

Properties:

- Advanced
 - Build Action: Content
 - Copy to Output Directory: Do not copy
 - Custom Tool
 - Custom Tool Namespace
- Misc
 - File Name: bootstrap.css
 - Full Path: C:\Users\ASTEEL\source\repos\And...

Build Action: How the file relates to the build and deployment processes.

- Go to wwwroot > replace the existing bootstrap.css found in:
lib > bootstrap > dist > css

- Go to wwwroot > replace the existing site.css file found in the main css folder

The screenshot shows the Visual Studio IDE interface. The main window displays the contents of the `bootstrap.css` file. The code includes Bootstrap v5.1.3 and Bootswatch v5.1.3 copyrights, along with a custom color palette at the bottom. The Solution Explorer on the right shows the project structure, including the `wwwroot` folder which contains the `css`, `js`, and `lib` folders. The `bootstrap` folder within `lib` is expanded, showing the `dist` folder and the `bootstrap.css` file, which is selected. The Properties window on the far right shows the file properties for `bootstrap.css`, including the build action set to "Content".

```
1 /*!
2 * Bootswatch v5.1.3
3 * Homepage: https://bootswatch.com
4 * Copyright 2012-2021 Thomas Park
5 * Licensed under MIT
6 * Based on Bootstrap
7 */
8 /*!
9 * Bootstrap v5.1.3 (https://getbootstrap.com/)
10 * Copyright 2011-2021 The Bootstrap Authors
11 * Copyright 2011-2021 Twitter, Inc.
12 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
13 */
14 @import url("https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,700;1,300;1,400&display=swap");
15
16 :root {
17   --bs-blue: #008cba;
18   --bs-indigo: #6610f2;
19   --bs-purple: #6f42c1;
```

- Go to wwwroot > replace the existing bootstrap.css found in:
lib > bootstrap > dist > css

- Go to wwwroot > replace the existing site.css file found in the main css folder

The screenshot shows the Microsoft Visual Studio interface for a .NET Core web application named "AndrewsBookStore".

- Solution Explorer:** Shows the project structure with "AndrewsBookStore" at the root, containing "Connected Services", "Dependencies", "Properties", "wwwroot", and "css". "site.css" is selected in the "css" folder.
- Properties Window:** Details for "site.css":
 - Advanced**: Build Action (Content), Copy to Output Directory (Do not copy), Custom Tool, Custom Tool Namespace.
 - Misc**: File Name (site.css), Full Path (C:\Users\ASTEEL\source\repos\And...).
 - Build Action**: Description: "How the file relates to the build and deployment processes."
- Code Editor (site.css):** Displays CSS code for the navbar brand, links, and primary button.

```
1 /* Please see documentation at https://docs.microsoft.com/aspnet/core/client-side/bundling-and-minification/enable-bundling-and-minification?view=aspnetcore-2.1 for details on configuring this project to bundle and minify static web assets. */
2 
3 a.navbar-brand {
4     white-space: normal;
5     text-align: center;
6     word-break: break-all;
7 }
8 
9 
10 /* Provide sufficient contrast against white background */
11 a {
12     color: #0366d6;
13 }
14 
15 /* remove bhrugen
16 .btn-primary {
17     color: #fff;
18     background-color: #2c3e50;
19     border-color: #1961a6;
```
- Output Window:** Shows output from "Debug".
- Bottom Status Bar:** Ready, 0 errors, 1 warning, AndrewsBookStore, master, 2 notifications.

- Go to wwwroot > replace the existing bootstrap.css found in:
lib > bootstrap > dist > css

- Go to wwwroot > replace the existing site.css file found in the main css folder
- Go to Views > Shared > _Layout.cshtml, change the file name from bootstrap.min.css to bootstrap.css

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** Displays the `site.css` file content. The code includes CSS rules for `a.navbar-brand`, `a`, and `.btn-primary`.
- Solution Explorer:** Shows the project structure for "AndrewsBookStore". The `wwwroot/css/site.css` file is selected.
- Properties:** The properties for `site.css` are displayed, including the file name, full path, and build action (Content).
- Output Window:** Shows the output from the Debug build.

- Go to wwwroot > replace the existing bootstrap.css found in:
lib > bootstrap > dist > css
 - Go to wwwroot > replace the existing site.css file found in the main css folder
 - Go to Views > Shared > _Layout.cshtml, change the file name from bootstrap.min.css to bootstrap.css

The screenshot shows the Microsoft Visual Studio IDE interface. The main area displays the `Layout.cshtml` file, which contains the following code:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>@ ViewData["Title"] - AndrewsBookStore</title>
        <!--link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" /-->
        <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.css" />
        <link rel="stylesheet" href="~/css/site.css" />
    </head>
    <body>
        <header>
            <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-primary border-bottom">
                <div class="container">
                    <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AndrewsBookStore</a>
                    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>
                </div>
            </nav>
        </header>
    </body>

```

The Solution Explorer window on the right shows the project structure for 'AndrewsBookStore' with files like `_Layout.cshtml`, `_LoginPartial.cshtml`, etc. The Properties window is also visible.

- Go to wwwroot > replace the existing bootstrap.css found in:
lib > bootstrap > dist > css

- Go to wwwroot > replace the existing site.css file found in the main css folder

- Go to Views > Shared > _Layout.cshtml, change the file name from bootstrap.min.css to bootstrap.css

- Change the nav class from navbar-light to navbar dark and bg-white to bg-primary

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ ViewData["Title"] - AndrewsBookStore</title>
    <!--link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" /-->
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.css" />
    <link rel="stylesheet" href="~/css/site.css" />
  </head>
  <body>
    <header>
      <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-primary border-bottom border-white" style="background-color: #002060; color: white; border-bottom: 1px solid black; border-top: 2px solid #002060; padding: 0; margin: 0; height: 50px;">
        <div class="container" style="padding: 0; margin: 0; height: 100%; position: relative;">
          <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AndrewsBookStore</a>
          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation" style="border: none; background-color: transparent; color: inherit; font-size: 1.2em; font-weight: bold; margin: 0; padding: 0; position: absolute; right: 0; top: 0; width: 40px; height: 100%; z-index: 1;">
            <span class="navbar-toggler-icon" style="font-size: 1.2em; font-weight: bold; margin: 0; padding: 0; position: absolute; left: 50%; top: 50%; transform: translate(-50%, -50%);"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarSupportedContent" style="margin: 0; padding: 0; position: absolute; left: 0; top: 0; width: 100%; height: 100%; background-color: #002060; color: white; overflow: hidden; transition: all 0.3s ease; z-index: 0;">
            <div style="position: absolute; left: 0; top: 0; width: 100%; height: 100%; background-color: #002060; color: white; opacity: 0.5; z-index: -1;"></div>
            <ul class="navbar-nav" style="list-style-type: none; padding: 0; margin: 0; position: relative; z-index: 1;">
              <li class="nav-item" style="margin: 0; padding: 0; position: relative; z-index: 1;">
                <a class="nav-link" href="#" style="color: white; text-decoration: none; font-size: 0.9em; font-weight: bold; margin: 0; padding: 0; position: relative; z-index: 1;">Home</a>
                <ul class="nav-item" style="list-style-type: none; padding: 0; margin: 0; position: absolute; left: 0; top: 0; width: 100%; height: 100%; background-color: #002060; color: white; opacity: 0.5; z-index: -1;">
                  <li class="nav-link" style="color: white; text-decoration: none; font-size: 0.8em; font-weight: bold; margin: 0; padding: 0; position: relative; z-index: 1;">Home</li>
                </ul>
              </li>
            </ul>
          </div>
        </div>
      </nav>
    </header>
    <div class="container" style="margin-top: 20px; margin-bottom: 20px;">
      <div style="border: 1px solid #ccc; padding: 10px; border-radius: 5px; background-color: #f9f9f9; margin-bottom: 10px;">
        <h2 style="margin: 0; padding: 0; font-size: 1.2em; font-weight: bold;">Welcome to AndrewsBookStore!
        <p style="margin: 0; padding: 0; font-size: 0.9em; margin-top: 5px;">This is a simple book store application built with ASP.NET Core. You can browse books, add them to your cart, and check out.
      </div>
      <div style="border: 1px solid #ccc; padding: 10px; border-radius: 5px; background-color: #f9f9f9; margin-bottom: 10px;">
        <h3 style="margin: 0; padding: 0; font-size: 1.1em; font-weight: bold;">Recent Books
        <table border="1" style="width: 100%; border-collapse: collapse; text-align: center; border: none; margin-bottom: 10px;">
          <thead>
            <tr>
              <th style="width: 10%;">#Book TitleAuthorPriceActions1The Great GatsbyF. Scott Fitzgerald$12.99View Details2To Kill a MockingbirdHarper Lee$14.99View Details3The Catcher in the RyeJ.D. Salinger$11.99View Details4The HobbitJ.R.R. Tolkien$16.99View Details5The Lord of the RingsJ.R.R. Tolkien$18.99View DetailsPopular Authors
        <table border="1" style="width: 100%; border-collapse: collapse; text-align: center; border: none; margin-bottom: 10px;">
          <thead>
            <tr>
              <th style="width: 10%;">#Author NameBooks PublishedFiction / Non-fictionActions1J.R.R. TolkienThe Hobbit, The Lord of the RingsFictionView Details2Harper LeeTo Kill a MockingbirdFictionView Details3F. Scott FitzgeraldThe Great GatsbyFictionView Details4J.D. SalingerThe Catcher in the RyeFictionView Details5William ShakespeareHamlet, Romeo and JulietFictionView Details
```

- Go to wwwroot > replace the existing bootstrap.css found in:
lib > bootstrap > dist > css

- Go to wwwroot > replace the existing site.css file found in the main css folder

- Go to Views > Shared > _Layout.cshtml, change the file name from bootstrap.min.css to bootstrap.css

- Change the nav class from navbar-light to navbar dark and bg-white to bg-primary

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `_Layout.cshtml` file. The code includes the following snippet:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ ViewData["Title"] - AndrewsBookStore</title>
    <!--link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" -->
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.css" />
    <link rel="stylesheet" href="~/css/site.css" />
  </head>
  <body>
    <header>
      <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-primary border-bottom">
        <div class="container">
          <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AndrewsBookStore</a>
          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
        </div>
      </nav>
    </header>
  </body>

```

The Solution Explorer on the right shows the project structure for 'AndrewsBookStore'. The `Views/Shared` folder contains files such as `_Layout.cshtml`, `LoginPartial.cshtml`, `ValidationScriptsPartial.cshtml`, `Error.cshtml`, `_ViewImports.cshtml`, and `_ViewStart.cshtml`.

- Go to wwwroot > replace the existing bootstrap.css found in: lib > bootstrap > dist > css

- Go to wwwroot > replace the existing site.css file found in the main css folder

- Go to Views > Shared > _Layout.cshtml, change the file name from bootstrap.min.css to bootstrap.css

- Change the nav class from navbar-light to navbar dark and bg-white to bg-primary

- Line 23 - remove text-dark

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `_Layout.cshtml` file in the editor. The code includes HTML tags for a header and body, with a navigation bar defined by a `<nav>` element. A tooltip is shown over the `navbar-dark` class, which is highlighted with a blue selection bar. The Solution Explorer on the right side of the interface shows the project structure for 'AndrewsBookStore', including the `Shared` folder which contains the `_Layout.cshtml` file. The status bar at the bottom indicates the file is ready.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>@ ViewData["Title"] - AndrewsBookStore</title>
7      <!--link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" -->
8      <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.css" />
9      <link rel="stylesheet" href="~/css/site.css" />
10 </head>
11 <body>
12     <header>
13         <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-primary border-bottom">
14             <div class="container">
15                 <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AndrewsBookStore</a>
16                 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
17                     <span class="navbar-toggler-icon"></span>
18                 </button>

```

- Go to wwwroot > replace the existing bootstrap.css found in: lib > bootstrap > dist > css
- Go to wwwroot > replace the existing site.css file found in the main css folder
- Go to Views > Shared > _Layout.cshtml, change the file name from bootstrap.min.css to bootstrap.css
- Change the nav class from navbar-light to navbar dark and bg-white to bg-primary
- Line 23 - remove text-dark

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `_Layout.cshtml` file, which contains the following code:

```

10  |</head>
11  |<body>
12  |  |<header>
13  |  |  |<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-primary border-bottom">
14  |  |  |  |<div class="container">
15  |  |  |  |  |<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AndrewsBookStore</a>
16  |  |  |  |  |<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-expanded="false" aria-label="Toggle navigation">
17  |  |  |  |  |  |<span class="navbar-toggler-icon"></span>
18  |  |  |  |</button>
19  |  |  |<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
20  |  |  |  |<ul class="navbar-nav flex-grow-1">
21  |  |  |  |  |<li class="nav-item">
22  |  |  |  |  |  |<a class="nav-link" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
23  |  |  |  |  |</li>
24  |  |  |  |  |<li class="nav-item">
25  |  |  |  |  |  |<a class="nav-link" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
26  |  |  |  |</li>
27  |</ul>

```

The Solution Explorer on the right shows the project structure for 'AndrewsBookStore' with files like `_Layout.cshtml`, `_LoginPartial.cshtml`, `_ValidationScriptsPartial.cshtml`, `Error.cshtml`, `_ViewImports.cshtml`, and `_ViewStart.cshtml`.

- Go to wwwroot > replace the existing bootstrap.css found in:
lib > bootstrap > dist > css

- Go to wwwroot > replace the existing site.css file found in the main css folder

- Go to Views > Shared > _Layout.cshtml, change the file name from bootstrap.min.css to bootstrap.css

- Change the nav class from navbar-light to navbar dark and bg-white to bg-primary

- Line 23 - remove text-dark
- Add additional properties to the footer class

```
10  |</head>
11  |<body>
12  |  |<header>
13  |  |  |<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-primary border-bottom">
14  |  |  |<div class="container">
15  |  |  |  |<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AndrewsBookStore</a>
16  |  |  |  |<button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
17  |  |  |  |  |<span class="navbar-toggler-icon"></span>
18  |  |  |</button>
19  |  |  |<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
20  |  |  |  |<ul class="navbar-nav flex-grow-1">
21  |  |  |  |  |<li class="nav-item">
22  |  |  |  |  |  |<a class="nav-link" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
23  |  |  |  |  |</li>
24  |  |  |  |<li class="nav-item">
25  |  |  |  |  |<a class="nav-link" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
26  |  |  |</li>
27  |</ul>
28  |</div>
29  |</header>
30  |<div class="container">
31  |<main role="main" class="pb-4">
32  |<partial\_view>
33  |</main>
34  |</div>
35  |<partial\_view>
36  |</div>
37  |</body>
```

- Go to wwwroot > replace the existing bootstrap.css found in:
lib > bootstrap > dist > css

- Go to wwwroot > replace the existing site.css file found in the main css folder

- Go to Views > Shared > _Layout.cshtml, change the file name from bootstrap.min.css to bootstrap.css

- Change the nav class from navbar-light to navbar dark and bg-white to bg-primary

- Line 23 - remove text-dark
- Add additional properties to the footer class

```
31     </div>
32   </nav>
33 </header>
34 <div class="container">
35   <main role="main" class="pb-3">
36     @RenderBody()
37   </main>
38 </div>
39
40 <footer class="border-top footer text-white-50 bg-primary">
41   <div class="container">
42     &copy; 2021 - AndrewsBookStore
43   </div>
44 </footer>
45 <script src="~/lib/jquery/dist/jquery.min.js"></script>
46 <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
47 <script src="~/js/site.js" asp-append-version="true"></script>
48   @await RenderSectionAsync("Scripts", required: false)
49 </body>
```

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for the `_Layout.cshtml` file. The code has been modified to use Bootstrap's primary color scheme, specifically the "primary" variant, as indicated by the `bg-primary` class on the footer container. The Solution Explorer on the right shows the project structure for "AndrewsBookStore", including the `Views/Shared/_Layout.cshtml` file. The status bar at the bottom indicates the current branch is "master" with two changes.

- _LoginPartial.cshtml
 - remove references to 'text-dark'

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** Displays the file `_LoginPartial.cshtml` containing C# and Razor syntax. The code includes imports for `Microsoft.AspNetCore.Identity`, `SignInManager<IdentityUser>`, and `UserManager<IdentityUser>`. It features an `@if` block that checks if the user is signed in. If signed in, it displays a "Manage" link and a "Logout" button. If not signed in, it displays a "Log In" link.
- Solution Explorer:** Shows the project structure for "AndrewsBookStore". The `Views` folder contains `_Layout.cshtml`, `_LoginPartial.cshtml`, `_ValidationScriptsPartial.cshtml`, `Error.cshtml`, `_ViewImports.cshtml`, and `_ViewStart.cshtml`.
- Properties:** The properties for `_LoginPartial.cshtml` are shown, including:
 - Advanced:** Build Action: Content, Copy to Output Directory: Do not copy.
 - Misc:** File Name: `_LoginPartial.cshtml`, Full Path: `C:\Users\ASTEELE\source\repos\And...`.
- Status Bar:** Shows "Ready" and other status indicators.

- LoginPartial.cshtml
 - remove references to 'text-dark'
- Run project to review changes

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** Displays the file `_LoginPartial.cshtml` containing C# and Razor syntax. The code includes imports for `Microsoft.AspNetCore.Identity`, `SignInManager<IdentityUser>`, and `UserManager<IdentityUser>`. It features an `@if` block that checks if the user is signed in. If signed in, it displays two navigation items: "Manage" and "Logout". If not signed in, it displays a single "Login" item.
- Solution Explorer:** Shows the project structure for "AndrewsBookStore". The `Views` folder contains `_Layout.cshtml`, `_LoginPartial.cshtml`, `ValidationScriptsPartial.cshtml`, `Error.cshtml`, `_ViewImports.cshtml`, and `_ViewStart.cshtml`.
- Properties:** The properties for `_LoginPartial.cshtml` are displayed, including:
 - Advanced:** Build Action: Content, Copy to Output Directory: Do not copy.
 - Misc:** File Name: `_LoginPartial.cshtml`, Full Path: `C:\Users\ASTEELE\source\repos\And...`
- Output:** Shows the message "Using TypeScript 4.0 for IntelliSense."
- Status Bar:** Shows "Ready" and other status indicators.

Home Page - AndrewsBookStore

https://localhost:44304

AndrewsBookStore Home Privacy Register Login

Welcome

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

© 2021 - AndrewsBookStore

- _LoginPartial.cshtml
 - remove references to 'text-dark'
- Run project to review changes

- Additional 3rd party tools

The screenshot shows the homepage of the jQuery UI website (<https://jqueryui.com>). The page has a dark orange header with the jQuery logo and navigation links for Plugins, Contribute, Events, Support, and JS Foundation. A donation message and a 'SUPPORT THE PROJECT' button are visible. The main content area features sections for Interactions (Draggable, Droppable, Resizable, Selectable, Sortable) and Widgets (Accordion, Autocomplete, Button, Checkboxradio, Controlgroup, Datepicker, Dialog). A central column discusses the latest version (1.13.0) and its compatibility with recent jQuery versions. To the right, there's a 'Developer Links' sidebar with links to Source Code (GitHub), jQuery UI Git (WIP Build), Theme (WIP Build), Bug Tracker, Submit a New Bug Report, Discussion Forum, Using jQuery UI, and Developing jQuery UI.

jQuery UI

Your donations help fund the continued development and growth of **jQuery**.

SUPPORT THE PROJECT

Demos Download API Documentation Themes Development Support Blog About Search

Interactions

- Draggable
- Droppable
- Resizable
- Selectable
- Sortable

Widgets

- Accordion
- Autocomplete
- Button
- Checkboxradio
- Controlgroup
- Datepicker
- Dialog

jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library. Whether you're building highly interactive web applications or you just need to add a date picker to a form control, jQuery UI is the perfect choice.

Download jQuery UI 1.13.0

Custom Download

Quick Downloads:

Stable v1.13.0 jQuery 1.8+

Old/Unsupported v1.12.1 jQuery 1.7+

What's New in jQuery UI 1.13?

Compatibility with recent jQuery versions (up to 3.6): Usage of deprecated jQuery APIs have been removed. jQuery UI 1.13 triggers no jQuery Migrate warnings when running its test suite against jQuery 3.6.0 with jQuery Migrate 3.3.2, i.e. the latest versions at the moment of its release.

Interested in the full details of what changed? Check out the [1.13 upgrade guide](#), and [1.13.0 changelog](#).

Developer Links

[Source Code \(GitHub\)](#)
[jQuery UI Git \(WIP Build\)](#)
[Theme \(WIP Build\)](#)
[Bug Tracker](#)
[Submit a New Bug Report](#)
[Discussion Forum](#)
[Using jQuery UI](#)
[Developing jQuery UI](#)

- Additional 3rd party tools
 - JQuery UI (Datepicker)

The screenshot shows the homepage of the jQuery UI website (<https://jqueryui.com>). The page has a dark orange header with the jQuery logo and navigation links for Demos, Download, API Documentation, Themes, Development, Support, Blog, and About. A search bar is also present. The main content area features sections for Interactions (Draggable, Droppable, Resizable, Selectable, Sortable) and Widgets (Accordion, Autocomplete, Button, Checkboxradio, Controlgroup, Datepicker, Dialog). A central column highlights jQuery UI as a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library. It also features a "What's New in jQuery UI 1.13?" section and developer links for Source Code (GitHub), jQuery UI Git (WIP Build), Theme (WIP Build), Bug Tracker, Submit a New Bug Report, Discussion Forum, Using jQuery UI, and Developing jQuery UI. A "Download jQuery UI 1.13.0" section is also visible.

jQuery UI

Plugins Contribute Events Support JS Foundation

Your donations help fund the continued development and growth of **jQuery**.

SUPPORT THE PROJECT

Demos Download API Documentation Themes Development Support Blog About

Search

Interactions

- Draggable
- Droppable
- Resizable
- Selectable
- Sortable

Widgets

- Accordion
- Autocomplete
- Button
- Checkboxradio
- Controlgroup
- Datepicker
- Dialog

jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library. Whether you're building highly interactive web applications or you just need to add a date picker to a form control, jQuery UI is the perfect choice.

What's New in jQuery UI 1.13?

Compatibility with recent jQuery versions (up to 3.6): Usage of deprecated jQuery APIs have been removed. jQuery UI 1.13 triggers no jQuery Migrate warnings when running its test suite against jQuery 3.6.0 with jQuery Migrate 3.3.2, i.e. the latest versions at the moment of its release.

Interested in the full details of what changed? Check out the [1.13 upgrade guide](#), and [1.13.0 changelog](#).

Download jQuery UI 1.13.0

Custom Download

Quick Downloads:

Stable v1.13.0 jQuery 1.8+

Old/Unsupported v1.12.1 jQuery 1.7+

Source Code (GitHub)
jQuery UI Git (WIP Build)
Theme (WIP Build)
Bug Tracker
Submit a New Bug Report
Discussion Forum
Using jQuery UI
Developing jQuery UI

- Additional 3rd party tools
 - JQuery UI (Datepicker)

The screenshot shows a web browser window displaying the [jQuery UI](https://jqueryui.com/) website, specifically the Datepicker section. The URL in the address bar is <https://jqueryui.com/datepicker/#content>. The page has a dark orange header with the **jQuery user interface** logo. A sidebar on the left lists various interaction and widget demos. The main content area is titled **Datepicker** and contains a placeholder text "Select a date from a popup or inline calendar". Below it is a text input field labeled "Date:" with a small calendar icon to its right. To the right of the main content, there's a sidebar titled "Widgets" with a list of examples for the Datepicker widget, including "QuickNav Examples", "Options", "Methods", and "Example". At the bottom of this sidebar, there's a link to "Search by algolia". A vertical sidebar on the far right lists additional features: "Display month & year menus", "Display multiple months", "Format date", "Icon trigger", "Localize calendar", "Populate alternate field", and "Restrict date range".

Demos Download API Documentation Themes Development Support Blog About Datepicker

Widgets

Datepicker Examples

Datepicker Widget QuickNav Examples

Datepicker Widget Options

Datepicker Widget Methods

Datepicker Widget Example

Search by algolia

Display month & year menus

Display multiple months

Format date

Icon trigger

Localize calendar

Populate alternate field

Restrict date range

- Additional 3rd party tools
 - JQuery UI (Datepicker)
 - DataTables

The screenshot shows a web browser window displaying the [jQuery UI Datepicker](https://jqueryui.com/datepicker/#content) demo page. The page has a dark orange header with the **jQuery user interface** logo. The main content area features a large title **Datepicker** and a subtitle *Select a date from a popup or inline calendar*. Below this is a text input field labeled **Date:** followed by a small calendar icon. To the left, there are two sections: **Interactions** (Draggable, Droppable, Resizable, Selectable, Sortable) and **Widgets** (Accordion, Autocomplete, Button, Checkboxradio, Controlgroup, Datepicker, Dialog). On the right, a sidebar titled **Widgets** lists various Datepicker features: Examples, Datepicker Widget (QuickNav Examples, Options, Methods, Example), and a long list of additional options: Display inline, Display month & year menus, Display multiple months, Format date, Icon trigger, Localize calendar, Populate alternate field, and Restrict date range. The URL in the address bar is <https://jqueryui.com/datepicker/#content>.

- Additional 3rd party tools
 - JQuery UI (Datepicker)
 - DataTables

The screenshot shows a web browser window displaying the DataTables website at <https://datatables.net>. The page has a blue header with the CloudTables logo and a sub-header: "No-code DataTables, with full editing. Create database-driven applications and embed them anywhere." Below the header are three navigation links: "CloudTables", "DataTables", and "Editor". A search bar is also present. The main content area features a large heading: "Add advanced interaction controls to your HTML tables *the free & easy way*". Two numbered steps are provided: Step 1 shows code snippets for CSS and JS files to include, and Step 2 shows a single line of JavaScript to call the DataTable function. To the right, there is a live demonstration of a DataTables table with columns for Name, Position, Office, Age, and Start date. The table contains data for several employees.

1 - Include these two files ↴

CSS //cdn.datatables.net/1.11.3/css/jquery.dataTables.css

JS //cdn.datatables.net/1.11.3/js/jquery.dataTables.js

2 - Call this single function ↴

```
1 $(document).ready( function () {  
2     $('#myTable').DataTable();  
3 } );
```

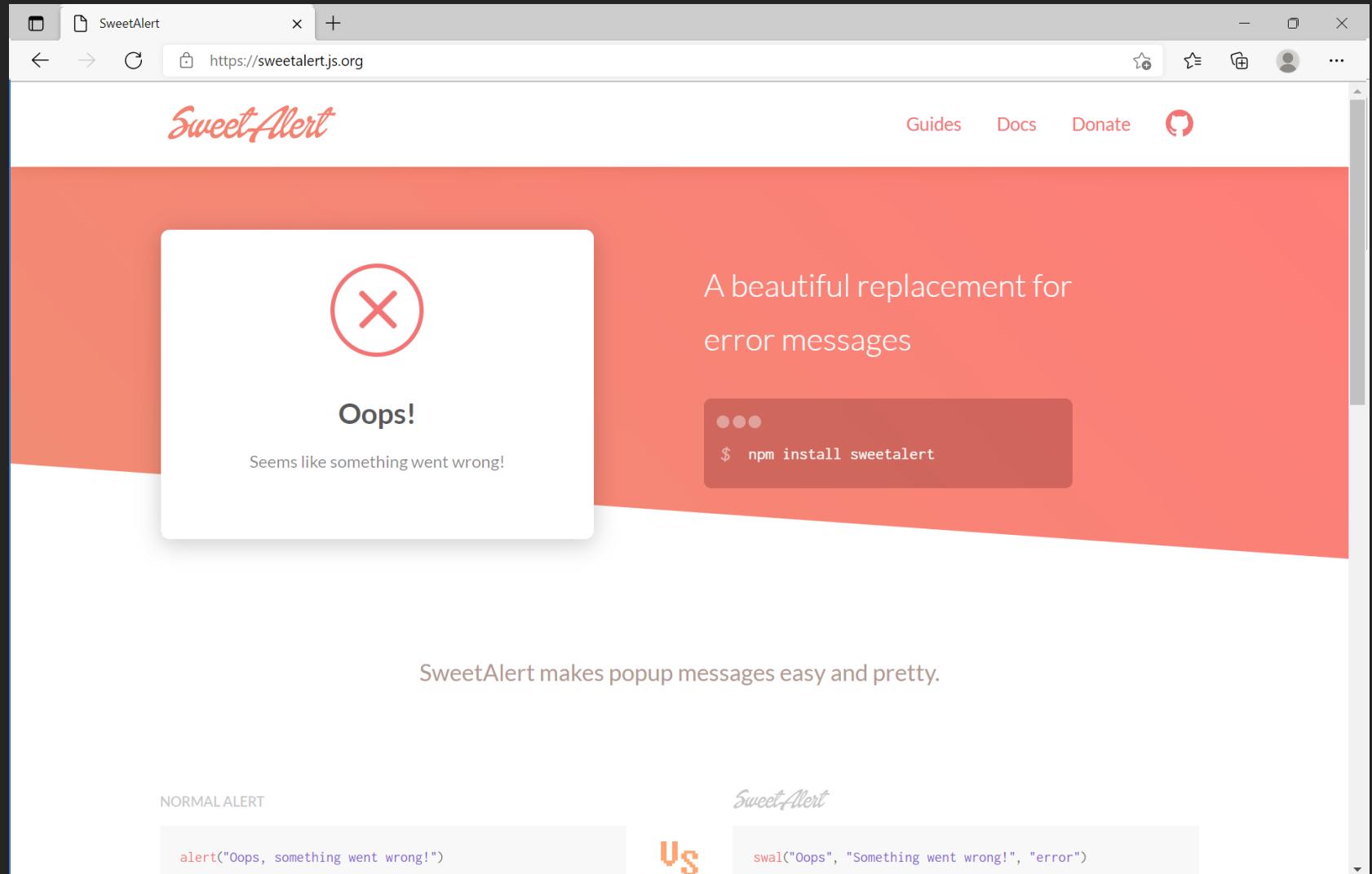
Name	Position	Office	Age	Start date
Airi Satou	Accountant	Tokyo	33	2008/11/28
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12
Bradley Greer	Software Engineer	London	41	2012/10/13
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07
Brielle Williamson	Integration Specialist	New York	61	2012/12/02

- Additional 3rd party tools
 - JQuery UI (Datepicker)
 - DataTables
 - Sweet Alert

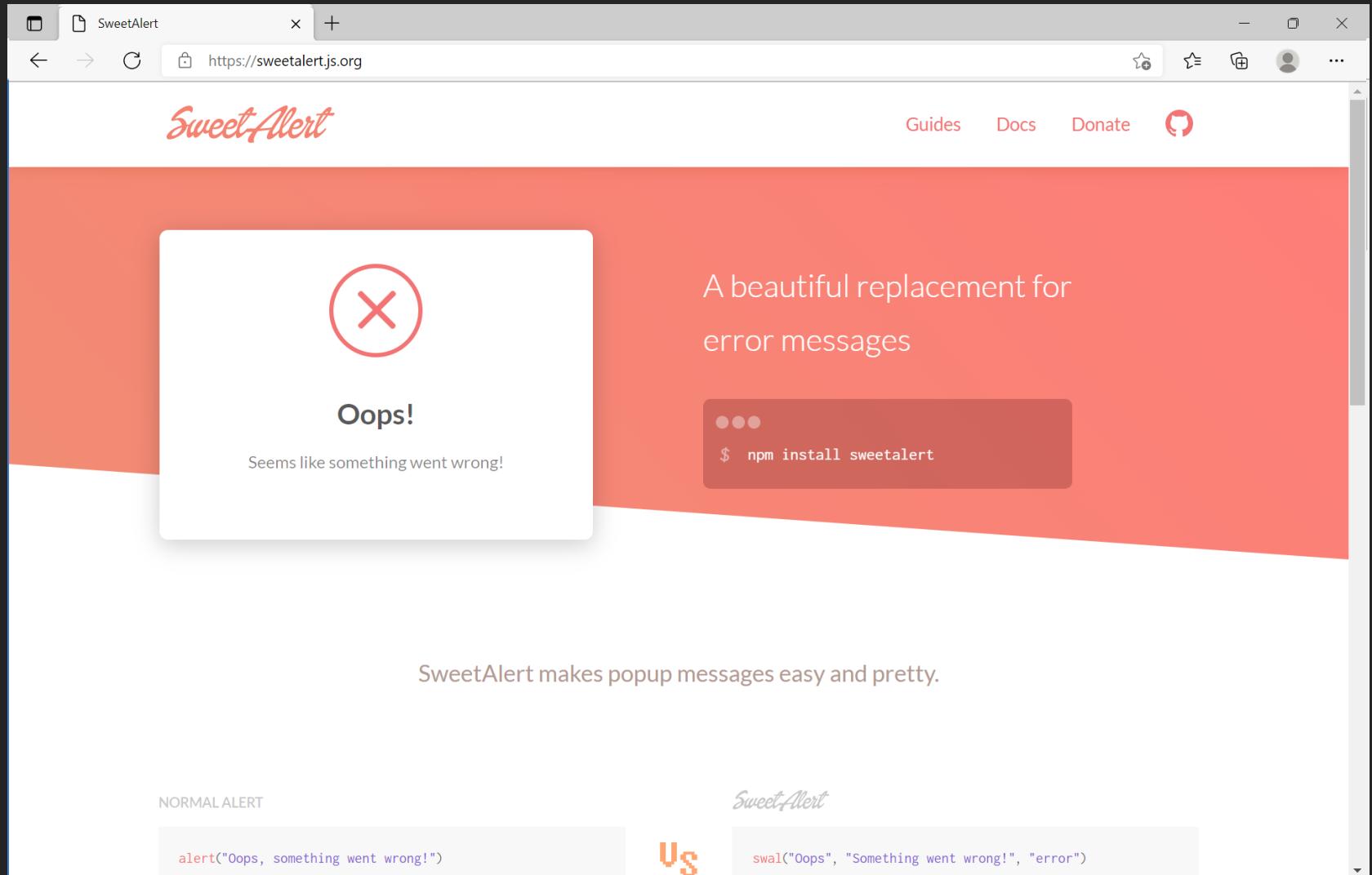
The screenshot shows a web browser displaying the DataTables homepage at <https://datatables.net>. The page has a blue header with the CloudTables logo and a sub-header: "No-code DataTables, with full editing. Create database-driven applications and embed them anywhere." Below the header are navigation links for "CloudTables", "DataTables", "Editor", "Manual", "Download", "Login / Register", and a search bar. The main content features a large heading: "Add advanced interaction controls to your HTML tables *the free & easy way*". Two numbered steps are shown: Step 1: "Include these two files" with CSS and JS code snippets pointing to CDN URLs. Step 2: "Call this single function" with a snippet of jQuery code. To the right, a live demonstration of a DataTables plugin is shown, displaying a table with columns for Name, Position, Office, Age, and Start date. The table contains 7 rows of employee data.

Name	Position	Office	Age	Start date
Airi Satou	Accountant	Tokyo	33	2008/11/28
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12
Bradley Greer	Software Engineer	London	41	2012/10/13
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07
Brielle Williamson	Integration Specialist	New York	61	2012/12/02

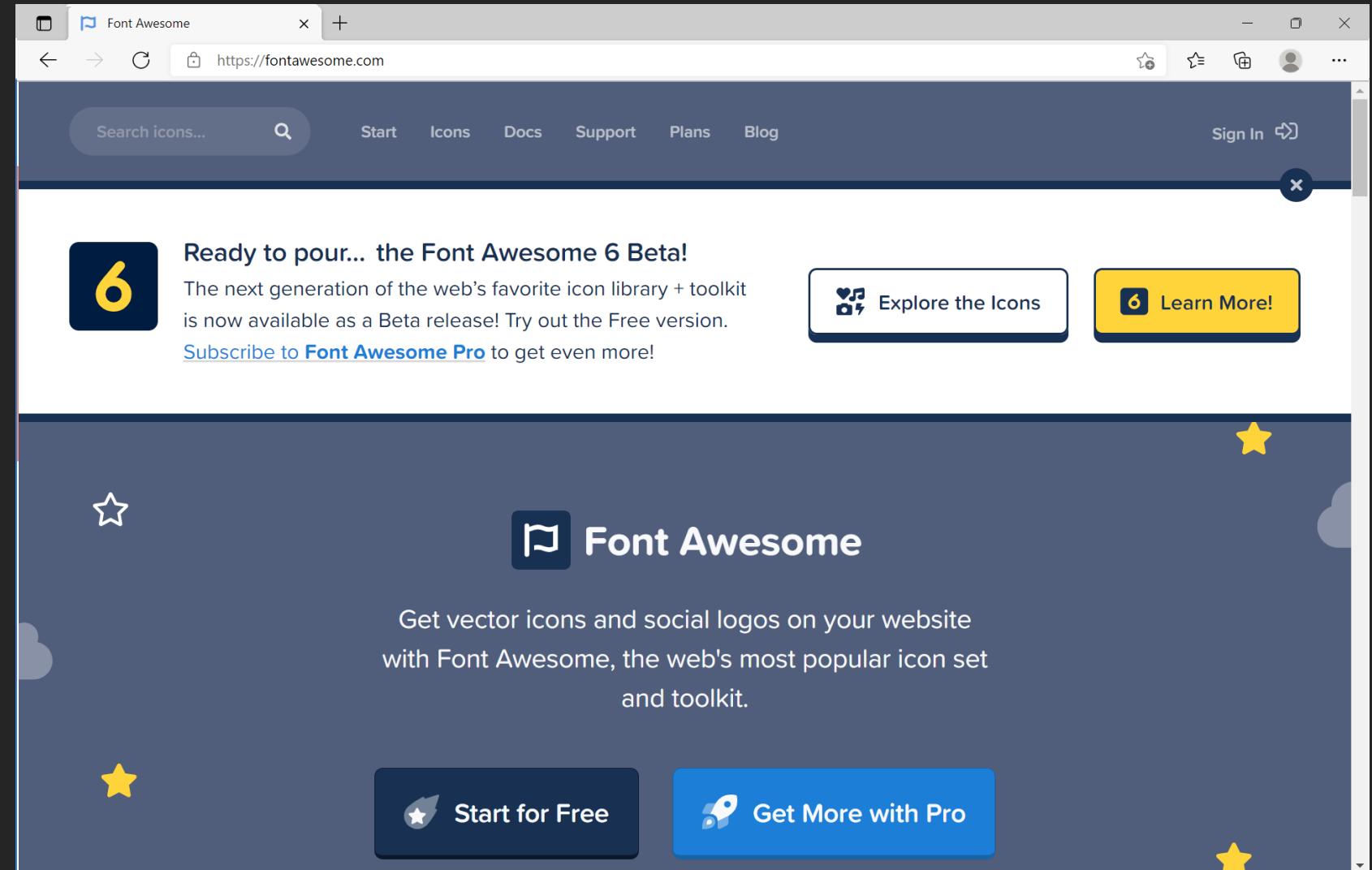
- Additional 3rd party tools
 - JQuery UI (Datepicker)
 - DataTables
 - Sweet Alert



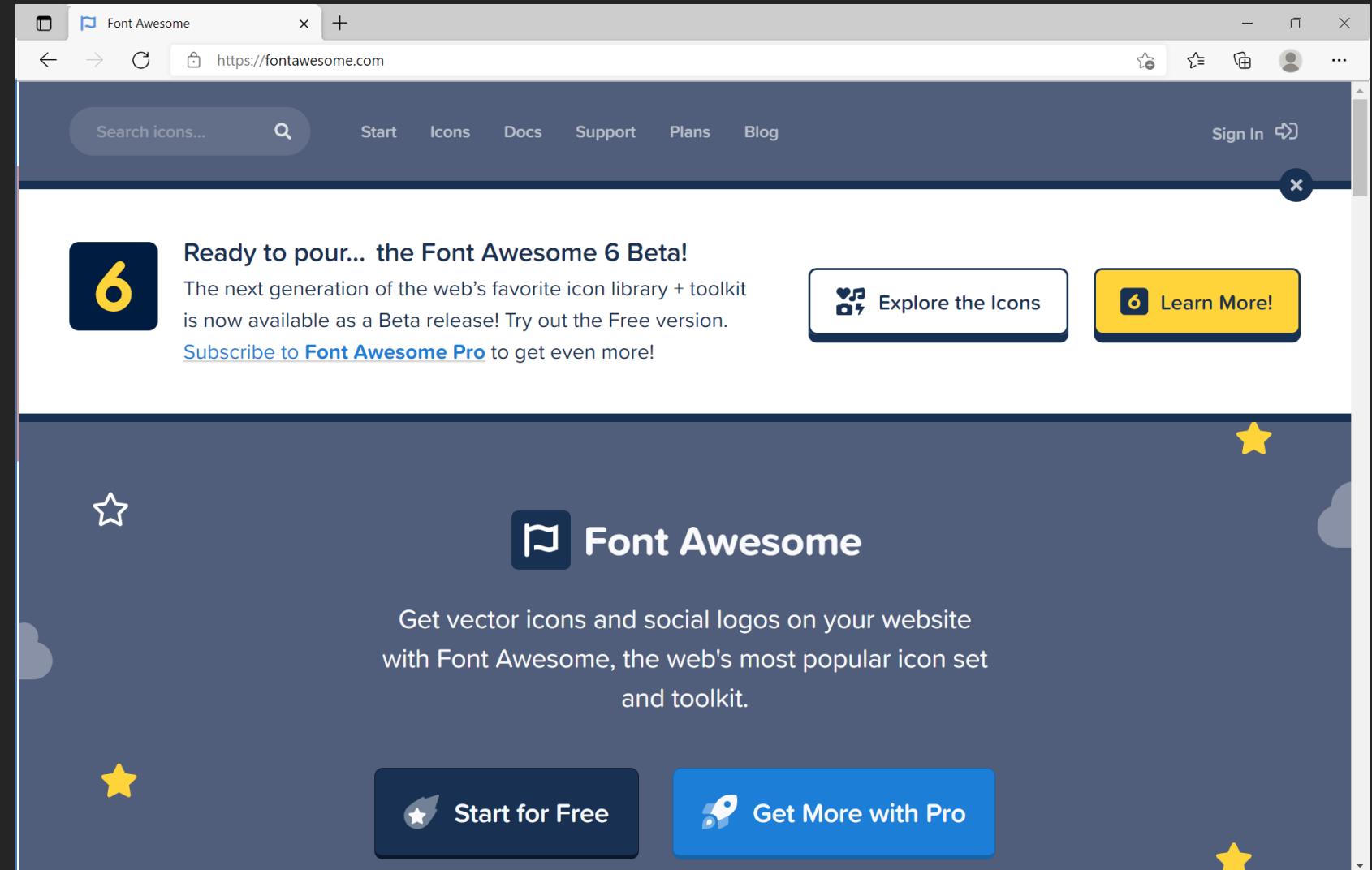
- Additional 3rd party tools
 - JQuery UI (Datepicker)
 - DataTables
 - Sweet Alert
 - Font Awesome



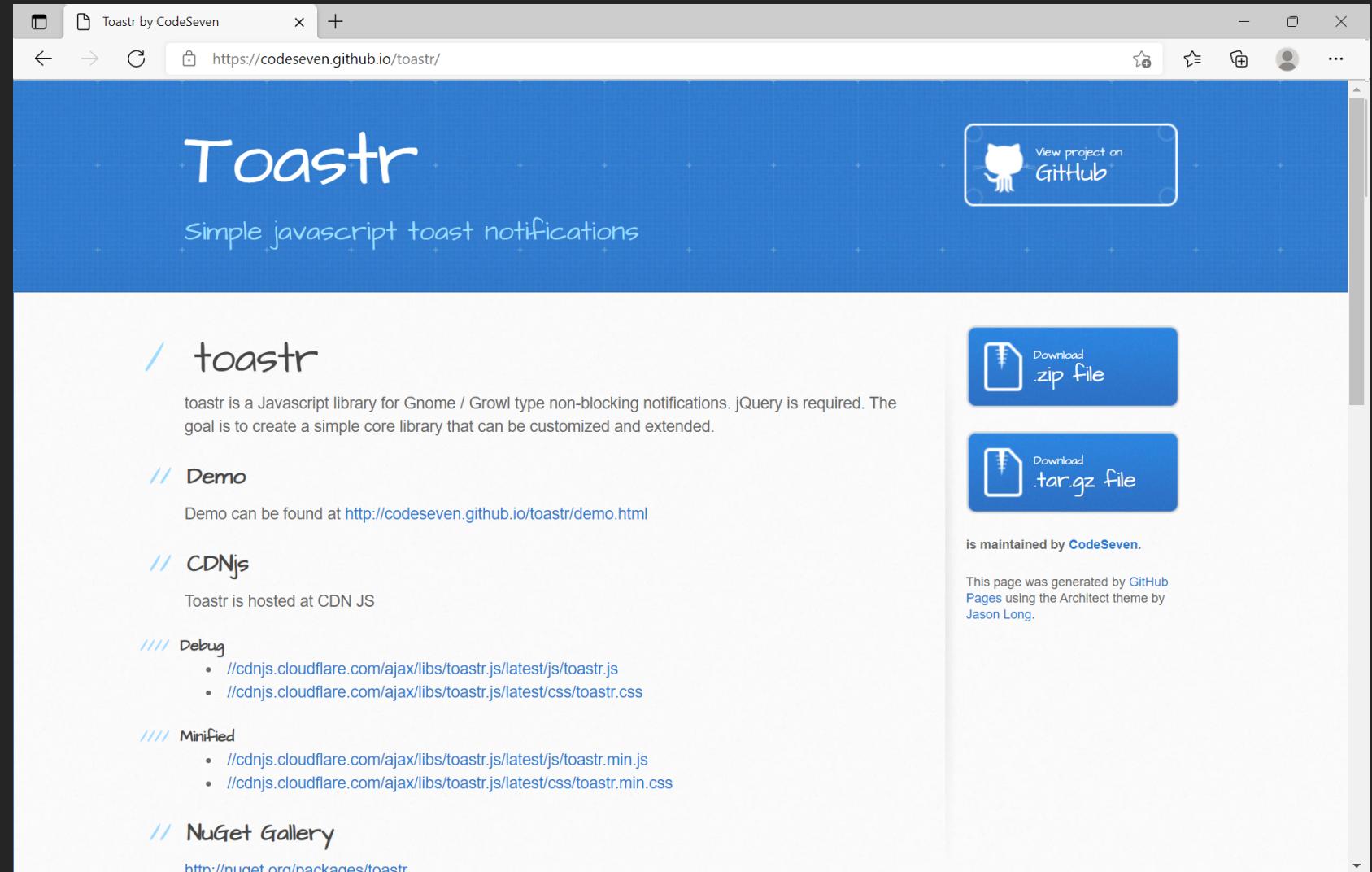
- Additional 3rd party tools
 - JQuery UI (Datepicker)
 - DataTables
 - Sweet Alert
 - Font Awesome



- Additional 3rd party tools
 - JQuery UI (Datepicker)
 - DataTables
 - Sweet Alert
 - Font Awesome
 - Toastr



- Additional 3rd party tools
 - JQuery UI (Datepicker)
 - DataTables
 - Sweet Alert
 - Font Awesome
 - Toastr



- Add to the _Layout.cshtml page the additional stylesheets and scripts from the CSS_JS.txt file

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `_Layout.cshtml` file with the following code:

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>@ ViewData["Title"] - AndrewsBookStore</title>
7      <!--link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" /-->
8      <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.css" />
9      <link rel="stylesheet" href="~/css/site.css" />
10     <!--additional stylesheets to use-->
11     <link rel="stylesheet" href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.min.css" />
12     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.css" />
13     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/css/toastr.min.css" />
14 </head>
15 <body>
16     <header>
17         <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-primary border-bottom box">
18             <div class="container">
19                 <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AndrewsBookS
20             </div>

```

The Diagnostic Tools window on the right shows a summary of the application's performance:

- Diagnostics session: 108:24 minutes
- Events: 108:20min
- Process Memory (MB): 101
- CPU (% of all processors): 100

The Watch 1 window at the bottom left shows the following table:

Name	Type
Add item to watch	

The status bar at the bottom right indicates the project name is `AndrewsBookStore` and the branch is `master`.

- Add to the _Layout.cshtml page the additional stylesheets and scripts from the CSS_JS.txt file

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `_Layout.cshtml` file with the following code:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  |<head>
4  ||<meta charset="utf-8" />
5  ||<meta name="viewport" content="width=device-width, initial-scale=1.0" />
6  ||<title>@ ViewData["Title"] - AndrewsBookStore</title>
7  ||<!--link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" /-->
8  ||<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.css" />
9  ||<link rel="stylesheet" href="~/css/site.css" />
10 |<!--additional stylesheets to use-->
11 |<link rel="stylesheet" href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.min.css" />
12 |<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.css" />
13 |<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/css/toastr.min.css" />
14 |</head>
15 |<body>
16 |<header>
17 |<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-primary border-bottom box">
18 |<div class="container">
19 |<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AndrewsBookS
20 |<button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar->
```

The Diagnostic Tools window on the right shows a summary of the application's performance, indicating a process memory usage of 101 MB and CPU usage at 100%.

- Add to the _Layout.cshtml page the additional stylesheets and scripts from the CSS_JS.txt file

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `_Layout.cshtml` file with the following code:

```
43<footer class="border-top footer text-white-50 bg-primary">
44    <div class="container">
45        © 2021 - AndrewsBookStore
46    </div>
47</footer>
48<script src="~/lib/jquery/dist/jquery.min.js"></script>
49<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
50<script src="~/js/site.js" asp-append-version="true"></script>
51<!--additional scripts to use-->
52<script src="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js"></script>
53<script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
54<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/js/toastr.min.js"></script>
55<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
56<script src="https://kit.fontawesome.com/e19c476714.js"></script>
57    @await RenderSectionAsync("Scripts", required: false)
58</body>
59</html>
```

A callout box points to the line `@await RenderSectionAsync("Scripts", required: false)`. The bottom right corner of the code editor has a small orange circle with the number "2".

The Diagnostic Tools window on the right side of the screen shows a timeline from 111:30min to 111:40min. It displays metrics for Process Memory (MB) and CPU (% of all processors). The summary indicates 101 MB of memory used and 100% CPU usage.

- Add to the _Layout.cshtml page the additional stylesheets and scripts from the CSS_JS.txt file

```
43<footer class="border-top footer text-white-50 bg-primary">
44    <div class="container">
45        © 2021 - AndrewsBookStore
46    </div>
47</footer>
48<script src="~/lib/jquery/dist/jquery.min.js"></script>
49<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
50<script src="~/js/site.js" asp-append-version="true"></script>
51<!--additional scripts to use-->
52<script src="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js"></script>
53<script src="https://cdn.datatables.net/1.10.16/js/dataTables.min.js"></script>
54<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/js/toastr.min.js"></script>
55<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
56<script src="https://kit.fontawesome.com/e19c476714.js"></script>
57    @await RenderSectionAsync("Scripts", required: false)
58</body>
59</html>
60
61
```

146% No issues found

Watch 1

Name	Type
Add item to watch	

Call Stack Breakpoints Exception Settings Command Window Immediate Window Output Error List

Item(s) Saved

0 2 AndrewsBookStore master 2

- In
Views/Shared/_Layout.cshtml,
add a dropdown to the NavBar

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for `_Layout.cshtml`. The code includes a navigation bar with a dropdown menu. The dropdown menu has three items: "Action", "Another action", and "Something else here". The code is written in C# and uses Bootstrap classes for styling.

```
<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" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
        </li>
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" data-bs-display="static">Dropdown</a>
            <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                <a class="dropdown-item" href="#">Action</a>
                <a class="dropdown-item" href="#">Another action</a>
                <div class="dropdown-divider"></div>
                <a class="dropdown-item" href="#">Something else here</a>
            </div>
        </li>
    </ul>
</div>
```

The Visual Studio interface also includes the Diagnostic Tools window, which shows a summary of the application's performance, and the Watch 1 and Command Window windows at the bottom.



Bootstrap

- In
Views/Shared/_Layout.cshtml,
add a dropdown to the NavBar

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for `_Layout.cshtml`. A specific section of the code, which defines a dropdown menu item, is highlighted with a yellow background:

```
24 <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
25   <ul class="navbar-nav flex-grow-1">
26     <li class="nav-item">
27       <a class="nav-link" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
28     </li>
29     <li class="nav-item">
30       <a class="nav-link" asp-area="" asp-controller="Home" asp-action="Privacy">Pr</a>
31     </li>
32     <li class="nav-item dropdown">
33       <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button">
34         Dropdown
35       </a>
36       <div class="dropdown-menu" aria-labelledby="navbarDropdown">
37         <a class="dropdown-item" href="#">Action</a>
38         <a class="dropdown-item" href="#">Another action</a>
39         <div class="dropdown-divider"></div>
40         <a class="dropdown-item" href="#">Something else here</a>
41       </div>
42     </li>
43   </ul>
```

The code editor includes line numbers, syntax highlighting, and a status bar at the bottom indicating 146% zoom, no issues found, and other development details.

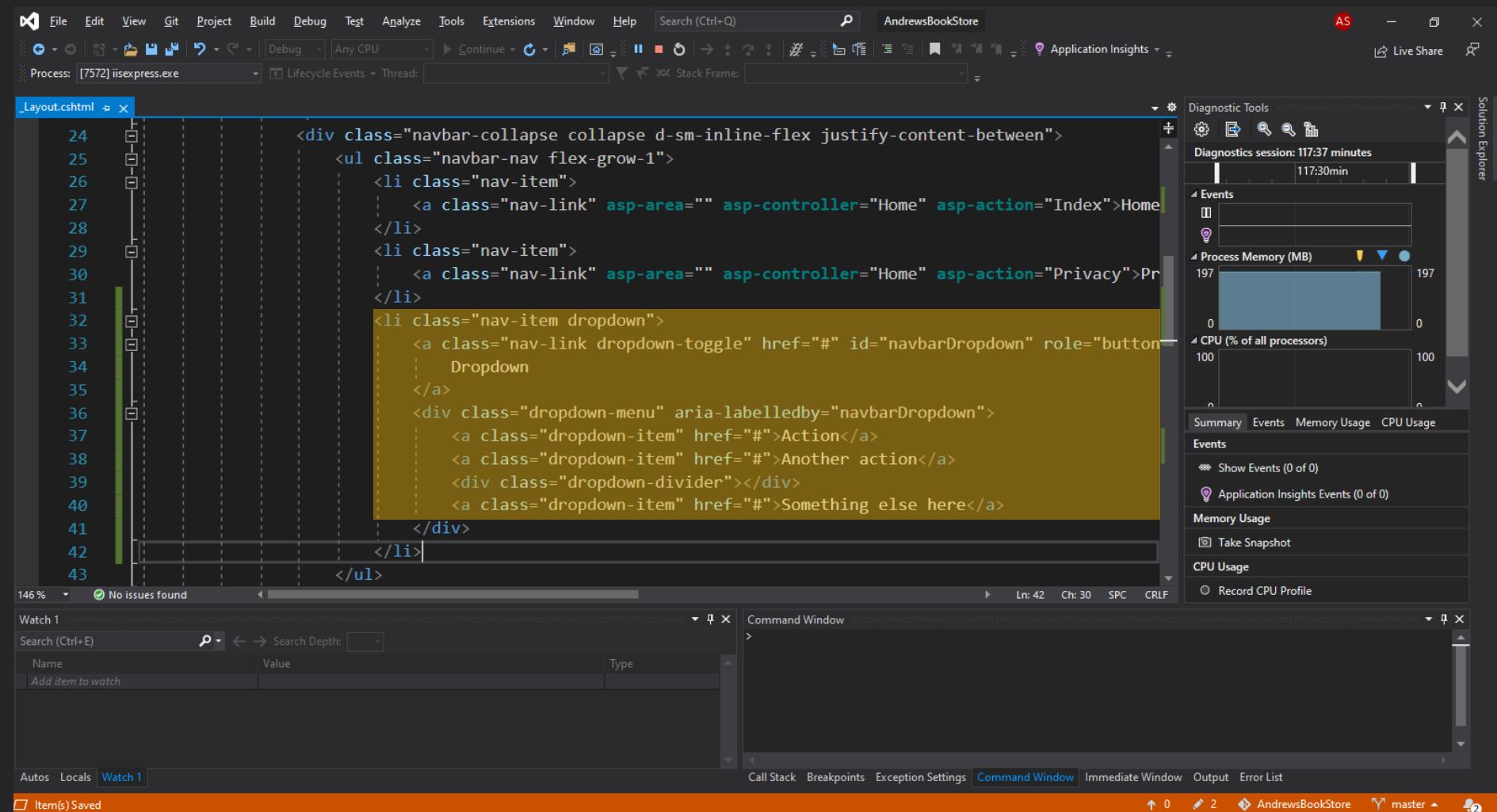
To the right of the code editor is the Diagnostic Tools window, which shows a summary of the application's performance over 117:37 minutes, including memory usage and CPU usage graphs.

At the bottom of the screen, there are several tabs and windows: Watch 1, Command Window, Call Stack, Breakpoints, Exception Settings, Immediate Window, Output, and Error List. The Command Window tab is currently selected.



Bootstrap

- In Views/Shared/_Layout.cshtml, add a dropdown to the NavBar
- Save, refresh and review the UI



The screenshot shows the Visual Studio IDE interface. The main window displays the code for `_Layout.cshtml`. A specific section of the code, which defines a dropdown menu item, is highlighted with a yellow background:

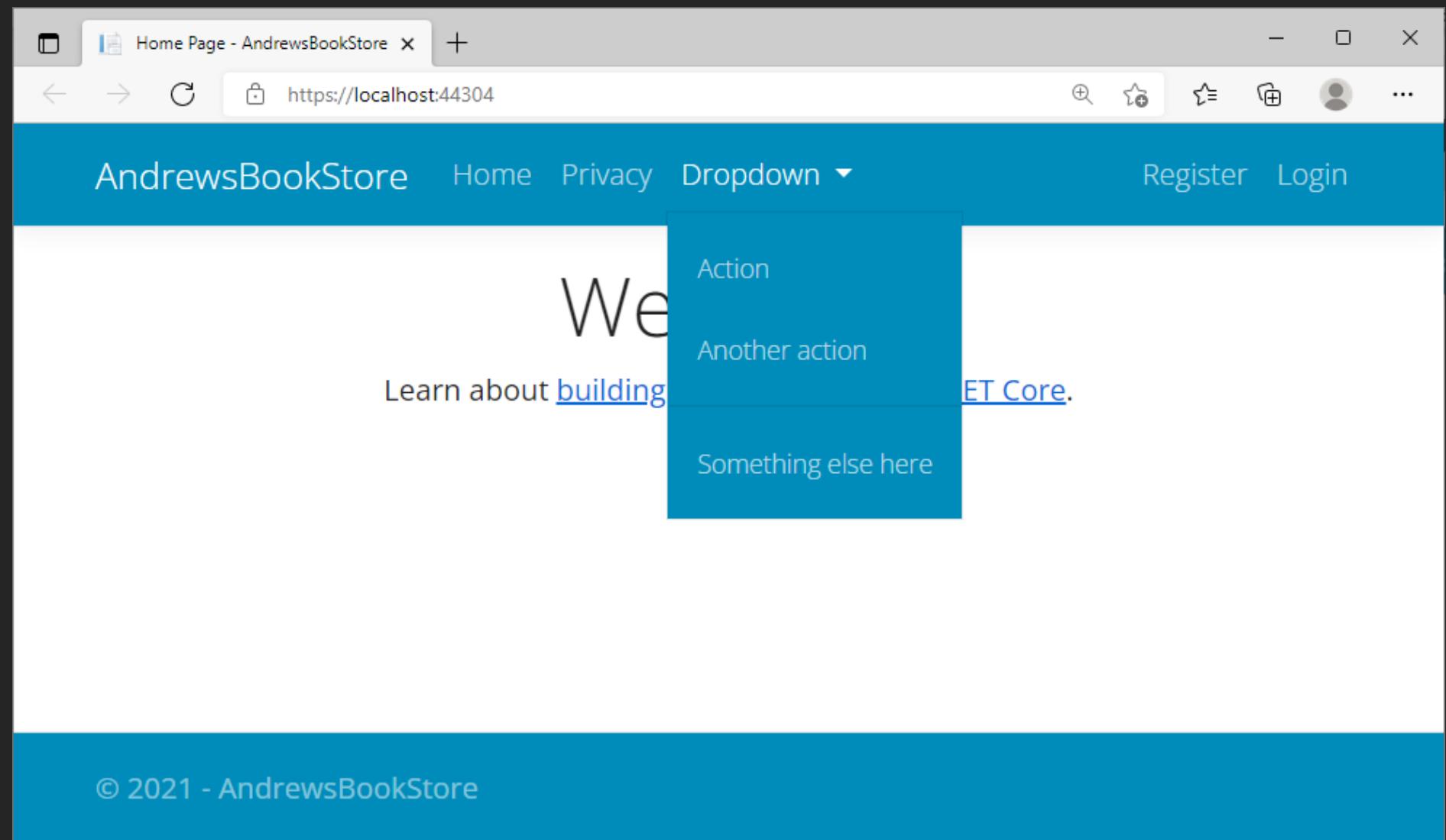
```
24 <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
25   <ul class="navbar-nav flex-grow-1">
26     <li class="nav-item">
27       <a class="nav-link" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
28     </li>
29     <li class="nav-item">
30       <a class="nav-link" asp-area="" asp-controller="Home" asp-action="Privacy">Pr</a>
31     </li>
32     <li class="nav-item dropdown">
33       <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button">
34         Dropdown
35       </a>
36       <div class="dropdown-menu" aria-labelledby="navbarDropdown">
37         <a class="dropdown-item" href="#">Action</a>
38         <a class="dropdown-item" href="#">Another action</a>
39         <div class="dropdown-divider"></div>
40         <a class="dropdown-item" href="#">Something else here</a>
41       </div>
42     </li>
43   </ul>
```

The code editor includes a status bar at the bottom showing "146 %", "No issues found", and line numbers 41-43. To the right of the code editor is the Diagnostic Tools window, which shows a summary of the application's performance over 117:37 minutes, including memory usage and CPU usage. Below the code editor are the Watch 1 and Command Windows.

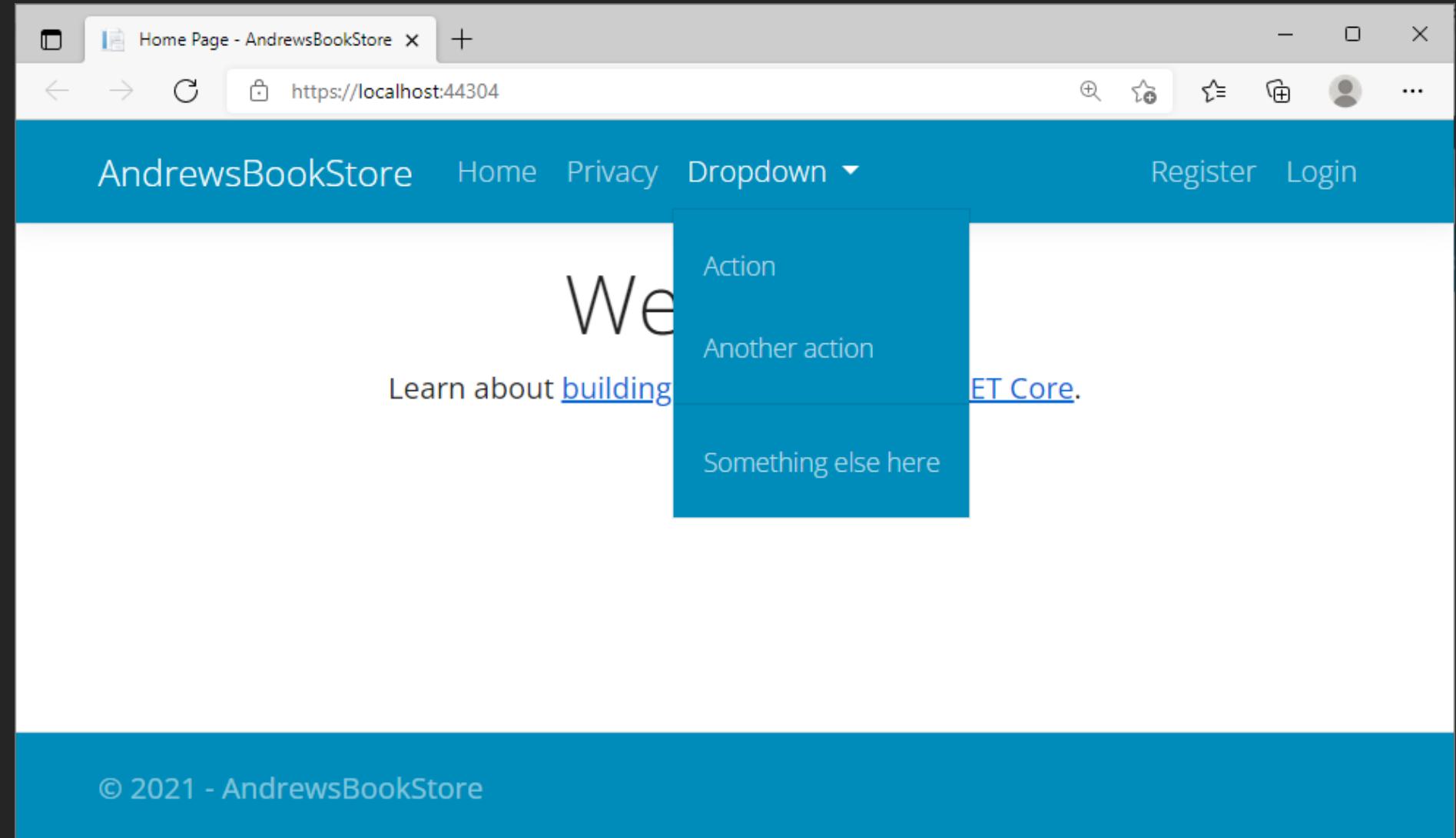
B

Bootstrap

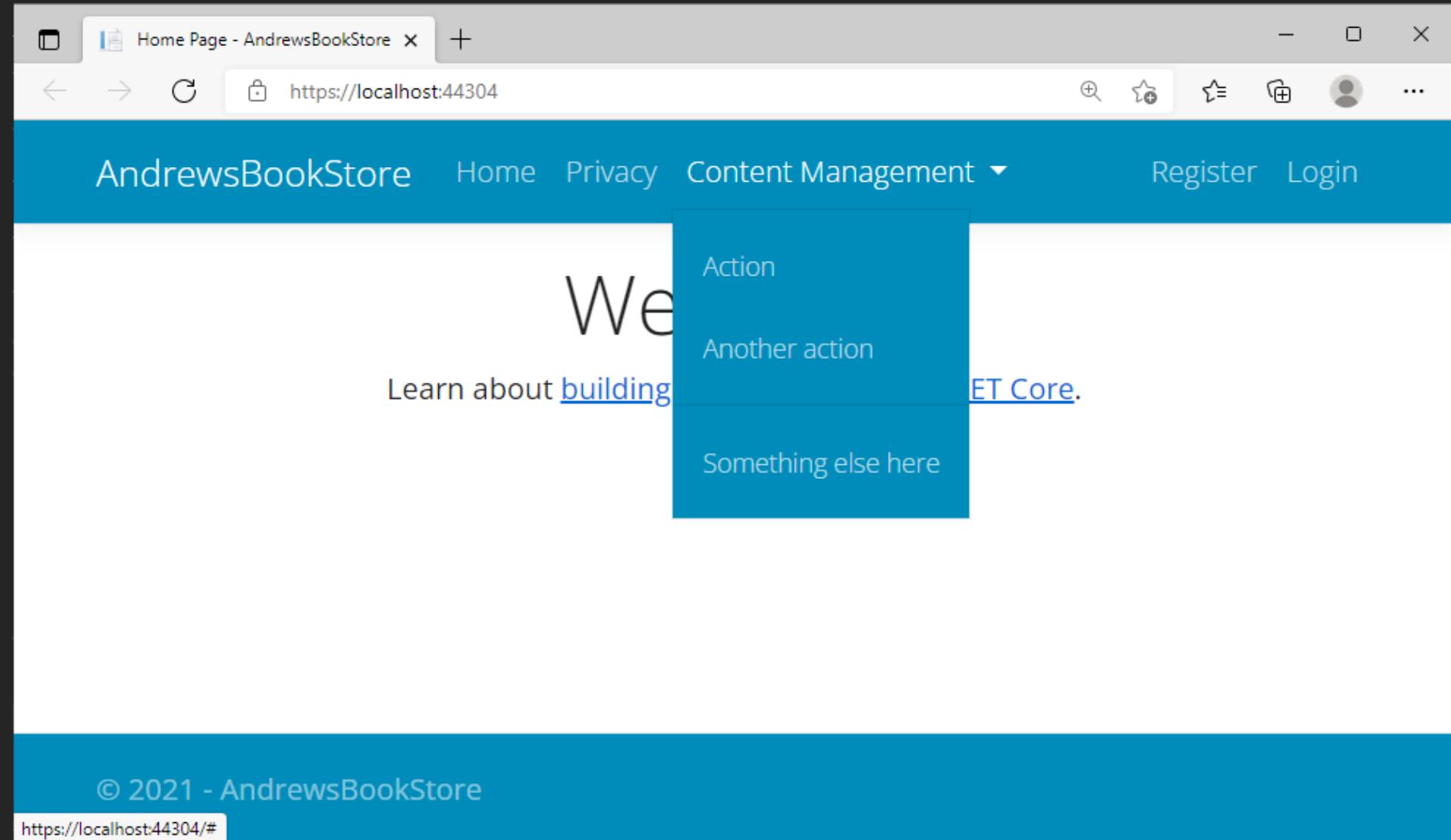
- In Views/Shared/_Layout.cshtml, add a dropdown to the NavBar
- Save, refresh and review the UI



- In Views/Shared/_Layout.cshtml, add a dropdown to the NavBar
- Save, refresh and review the UI
- Change 'Dropdown' to 'Content Management'



- In Views/Shared/_Layout.cshtml, add a dropdown to the NavBar
- Save, refresh and review the UI
- Change 'Dropdown' to 'Content Management'

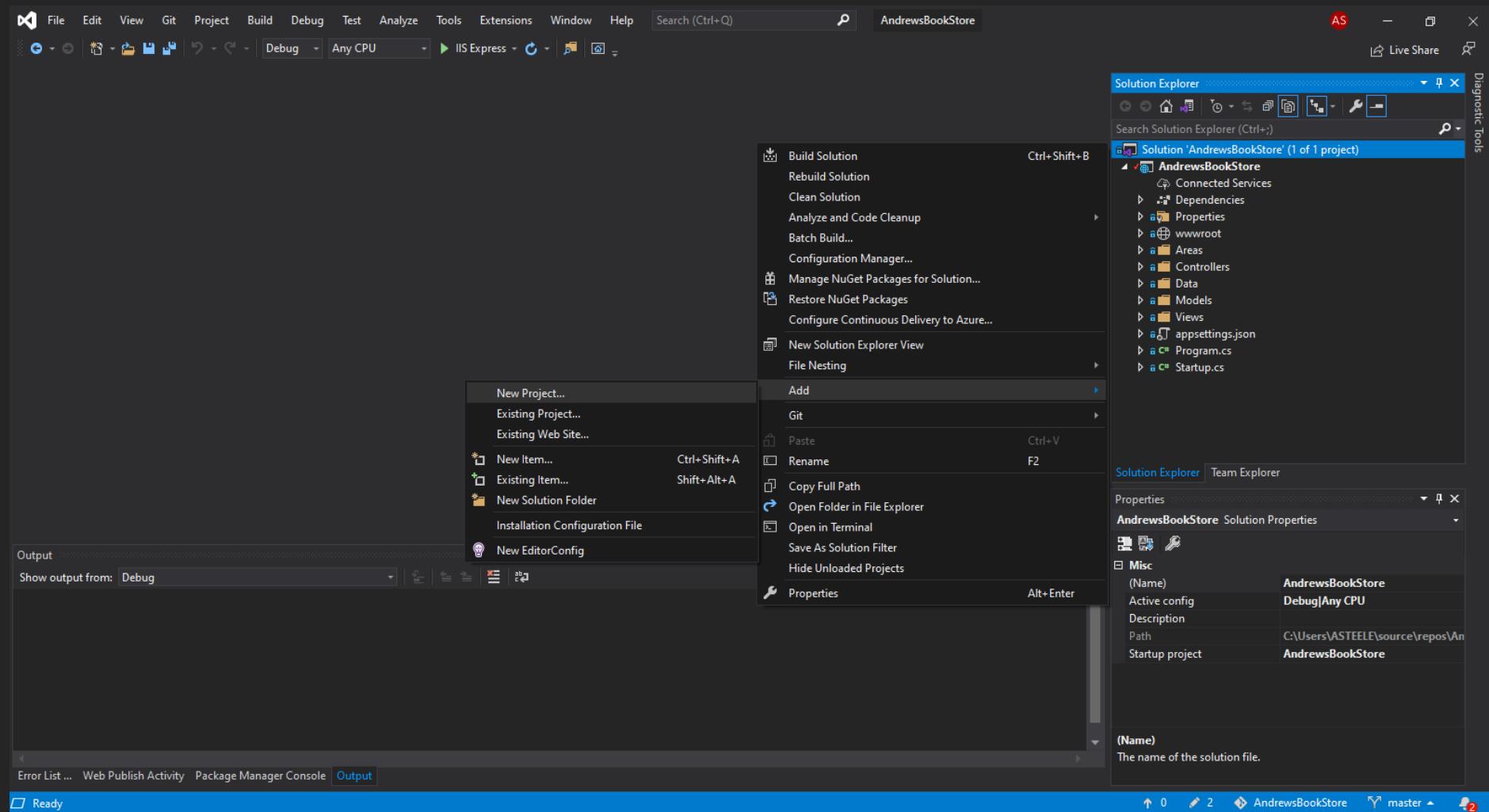


Setting Up the Project (Part 1)

1.4 Add Projects & Modify

- Add three (3) new projects (.NET Core class library) to the application:

Example:



- Add three (3) new projects (.NET Core class library) to the application:

Example:

The screenshot shows the Microsoft Visual Studio interface with a dark theme. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a Search bar. Below the menu is a toolbar with various icons. The main area displays the code for `AndrewsBooks.DataAccess.csproj`:

```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3     <PropertyGroup>
4         <TargetFramework>netcoreapp3.1</TargetFramework>
5     </PropertyGroup>
6
7 </Project>
8
```

The Solution Explorer on the right lists four projects under the solution 'AndrewsBookStore': `AndrewsBooks.DataAccess`, `AndrewsBooks.Models`, `AndrewsBooks.Utility`, and `AndrewsBookStore`. The `AndrewsBookStore` project is expanded, showing its structure: Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers, Data, and Models. The Properties window is also visible on the right.

- Add three (3) new projects (.NET Core class library) to the application:

Example:

- AndrewsBooks.DataAccess
- AndrewsBooks.Models
- AndrewsBooks.Utility

The screenshot shows the Microsoft Visual Studio interface with a dark theme. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search (Ctrl+Q). The toolbar below has icons for file operations like Open, Save, and Build. The status bar at the bottom shows 'Ready'.

The main area displays the code for `AndrewsBooks.DataAccess.csproj`:

```
1 <Project Sdk="Microsoft.NET.Sdk">
2
3     <PropertyGroup>
4         <TargetFramework>netcoreapp3.1</TargetFramework>
5     </PropertyGroup>
6
7 </Project>
8
```

The Solution Explorer on the right lists the solution structure:

- Solution 'AndrewsBookStore' (4 of 4 projects)
 - AndrewsBooks.DataAccess
 - Dependencies
 - Class1.cs
 - AndrewsBooks.Models
 - Dependencies
 - Class1.cs
 - AndrewsBooks.Utility
 - Dependencies
 - Class1.cs
 - AndrewsBookStore
 - Connected Services
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Data
 - Models

The Properties and Team Explorer panes are also visible on the right side of the interface.

- Add three (3) new projects (.NET Core class library) to the application:

Example:

- AndrewsBooks.DataAccess
- AndrewsBooks.Models
- AndrewsBooks.Utility

- Copy the 'Data' folder and paste to .DataAccess project (delete the original)

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Displays the solution 'AndrewsBookStore' containing four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore.
- Editor:** The code editor shows the contents of the `AndrewsBooks.DataAccess.csproj` file, which includes XML configuration for the .NET Core SDK and target framework.
- Output Window:** Shows the output from the 'Debug' configuration, indicating 'No issues found'.
- Status Bar:** Shows the status 'Ready'.

```

1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <TargetFramework>netcoreapp3.1</TargetFramework>
5   </PropertyGroup>
6
7 </Project>
8

```

- Add three (3) new projects (.NET Core class library) to the application:

Example:

- AndrewsBooks.DataAccess
- AndrewsBooks.Models
- AndrewsBooks.Utility

- Copy the 'Data' folder and paste to .DataAccess project (delete the original)

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows the solution 'AndrewsBookStore' containing four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore. The AndrewsBookStore project is selected. Inside AndrewsBookStore, there are folders for Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers, Data (which is highlighted), and Models.
- Properties Window:** Shows the properties for the selected 'Data' folder. The 'Folder Name' is set to 'Data' and the 'Full Path' is 'C:\Users\ASTEELE\source\repos\And'. The 'Misc' section shows the 'Folder Name' is 'Name of this folder.'
- Output Window:** Shows the output from the 'Debug' configuration. It displays 'No issues found' and has tabs for Error List, Web Publish Activity, Package Manager Console, and Output. The Output tab is active.
- Code Editor:** Displays the contents of the 'AndrewsBooks.DataAccess.csproj' file, which defines a .NET Core class library project with a target framework of netcoreapp3.1.

```

1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <TargetFramework>netcoreapp3.1</TargetFramework>
5   </PropertyGroup>
6
7 </Project>
8

```

- Add three (3) new projects (.NET Core class library) to the application:

Example:

- AndrewsBooks.DataAccess
- AndrewsBooks.Models
- AndrewsBooks.Utility

- Copy the 'Data' folder and paste to .DataAccess project (delete the original)

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a Search bar. The toolbar below has icons for file operations like Open, Save, and Build. The main code editor window displays the contents of the `AndrewsBooks.DataAccess.csproj` file:

```

1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <TargetFramework>netcoreapp3.1</TargetFramework>
5   </PropertyGroup>
6
7 </Project>
8

```

To the right of the code editor is the Solution Explorer, which lists the solution 'AndrewsBookStore' containing four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore. The AndrewsBooks.DataAccess project is expanded, showing its Data folder which contains a Class1.cs file. The Properties window is also visible, showing the properties for the Data folder.

- Add three (3) new projects (.NET Core class library) to the application:

Example:

- AndrewsBooks.DataAccess
- AndrewsBooks.Models
- AndrewsBooks.Utility

- Copy the 'Data' folder and paste to .DataAccess project (delete the original)

- Install Microsoft.EntityFrameworkCore.Core.Relational and Core.SqlServer packages

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Solution Explorer:** Shows a solution named 'AndrewsBookStore' containing four projects:
 - AndrewsBooks.DataAccess
 - AndrewsBooks.Models
 - AndrewsBooks.Utility
 - AndrewsBookStore
 The 'Data' folder under AndrewsBooks.DataAccess is selected, and its contents (Class1.cs) are highlighted in blue.
- Properties Panel:** Shows the properties for the 'Data' folder, including:
 - Folder Name: Data
 - Full Path: C:\Users\ASTEEL\source\repos\And...
- Output Window:** Shows 'No issues found'.
- Status Bar:** Shows the current branch as 'master'.

```

<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>netcoreapp3.1</TargetFramework>
  </PropertyGroup>
</Project>

```

- Add three (3) new projects (.NET Core class library) to the application:

Example:

- AndrewsBooks.DataAccess
- AndrewsBooks.Models
- AndrewsBooks.Utility

- Copy the 'Data' folder and paste to .DataAccess project (delete the original)

- Install Microsoft.EntityFrameworkCore.Core.Relational and Core.SqlServer packages

```

1 using Microsoft.EntityFrameworkCore.Metadata;
2 using Microsoft.EntityFrameworkCore.Migrations;
3 using System;
4
5 namespace AndrewsBookStore.Data.Migrations
6 {
7     public partial class CreateIdentitySchema : Migration
8     {
9         protected override void Up(MigrationBuilder migrationBuilder)
10        {
11            migrationBuilder.CreateTable(
12                name: "AspNetRoles",
13                columns: table => new
14                {
15                    Id = table.Column<string>(nullable: false),
16                    Name = table.Column<string>(maxLength: 256, nullable: true),
17                });
18        }
19    }

```

Solution Explorer:

- Solution 'AndrewsBookStore' (4 of 4 projects)
 - AndrewsBooks.DataAccess
 - AndrewsBooks.Models
 - AndrewsBooks.Utility
 - AndrewsBookStore

Properties for 000000000000_CreatedIdentitySchema.cs:

- Build Action: C# compiler
- File Name: 000000000000_CreatedIdentitySchema.cs
- Full Path: C:\Users\ASTEELE\source\repos\And...

- Add three (3) new projects (.NET Core class library) to the application:
 - Example:
 - AndrewsBooks.DataAccess
 - AndrewsBooks.Models
 - AndrewsBooks.Utility
- Copy the 'Data' folder and paste to .DataAccess project (delete the original)
- Install Microsoft.EntityFrameworkCore.Relational and Core.SqlServer packages
- Delete the Migrations folder

The screenshot shows the Visual Studio IDE interface with the following details:

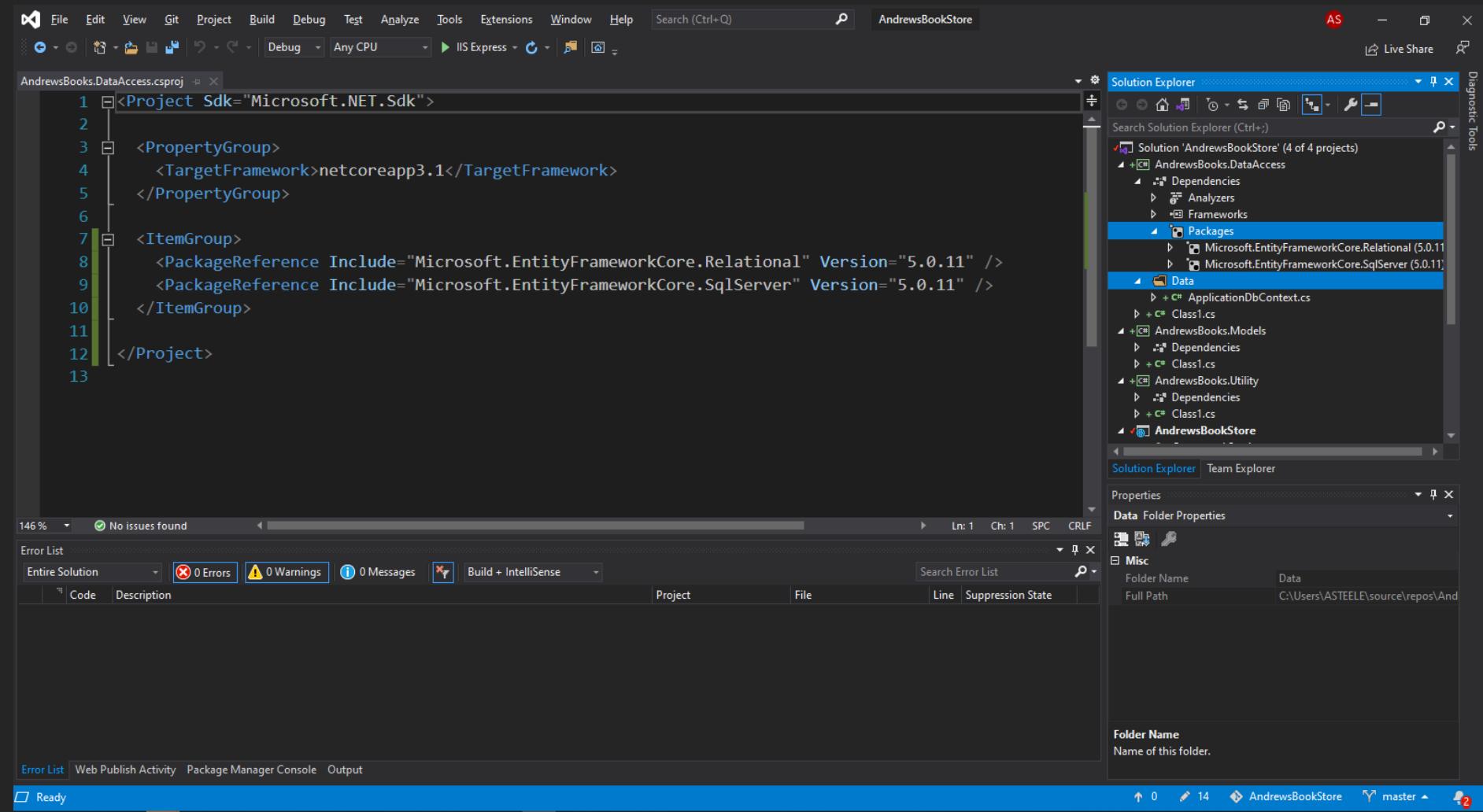
- Solution Explorer:** Shows the solution structure with four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore.
- Code Editor:** Displays the file `000000000000_CreatedIdentitySchema.cs` containing C# code for Entity Framework migrations. The code includes namespaces for `Microsoft.EntityFrameworkCore.Metadata` and `Microsoft.EntityFrameworkCore.Migrations`, and defines a partial class `CreateIdentitySchema` that overrides the `Up` method to create a table named `AspNetRoles`.
- Properties Window:** Opened for the file `000000000000_CreatedIdentitySchema.cs`. It shows the following settings:
 - Advanced:** Build Action set to `C# compiler`, Copy to Output Directory set to `Do not copy`.
 - Misc:** File Name is `000000000000_CreatedIdentitySchema.cs`, Full Path is `C:\Users\ASTEELE\source\repos\And...`.
 - Build Action:** Description: "How the file relates to the build and deployment processes."
- Status Bar:** Shows the status `Ready` and other build-related information.

- Add three (3) new projects (.NET Core class library) to the application:

Example:

- AndrewsBooks.DataAccess
- AndrewsBooks.Models
- AndrewsBooks.Utility

- Copy the 'Data' folder and paste to .DataAccess project (delete the original)
- Install Microsoft.EntityFrameworkCore.Core.Relational and Core.SqlServer packages
- Delete the Migrations folder



- Install NuGet package:

- Identity.EntityFrameworkCore
- Core

The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Solution Explorer:** Shows the solution structure with four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore.
- Toolbox:** Standard Visual Studio toolbox icons.
- Task List:** Live Share icon.
- Code Editor:** The `ApplicationDbContext.cs` file is open. The cursor is on the line `public class ApplicationDbContext : IdentityDbContext`. A context menu is open with the following options:
 - Generate class 'IdentityDbContext' in new file
 - Generate class 'IdentityDbContext'
 - Generate new type...
 - Install package 'Microsoft.AspNetCore.Identity.EntityFrameworkCore'
 - Install package 'Microsoft.AspNetCore.Identity.EntityFrameworkCore'
 - Use local version '5.0.2'
 - Find and install latest version
 - Install with package manager...
- Error List:** Shows 7 Errors, 3 Warnings, and 0 Messages. The errors include:
 - CS0234: The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)
 - CS0234: The type or namespace name 'AspNetCore' does not exist in the namespace 'Microsoft' (are you missing an assembly reference?)
 - CS0246: The type or namespace name 'IdentityDbContext' could not be found (are you missing a using directive or an assembly reference?)
 - CS0311: The type 'AndrewsBookStore.Data.ApplicationDbContext' cannot be used as type parameter 'TContext' in the generic type or method 'DbContextOptions<TContext>'. There is no implicit reference conversion from 'AndrewsBookStore.Data.ApplicationDbContext' to 'Microsoft.EntityFrameworkCore.DbContext'.
 - CS0246: The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)
- Properties:** Properties panel showing the selected project.
- Preview changes:** Preview changes button.
- Bottom Bar:** Ready, 0, 11, AndrewsBookStore, master, 2.

- Install NuGet package:
 - Identity.EntityFrameworkCore

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** Displays the `ApplicationDbContext.cs` file from the `AndrewsBooks.DataAccess` project. The code defines a `ApplicationContext` class that inherits from `IdentityDbContext`. The namespace is `AndrewsBookStore.DataAccess.Data`.
- Solution Explorer:** Shows the solution structure with four projects: `AndrewsBooks.DataAccess`, `AndrewsBooks.Models`, `AndrewsBooks.Utility`, and `AndrewsBookStore`. The `AndrewsBookStore` project is currently selected.
- Error List:** Contains the following errors:

Code	Description	Project	File	Line	Suppression State
CS0234	The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)	AndrewsBookStore	Startup.cs	1	Active
CS0246	The type or namespace name 'ApplicationContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	30	Active
CS0246	The type or namespace name 'ApplicationContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	37	Active
NU11701	Package 'EntityFramework 6.1.0' was restored using '.NETFramework,Version=v4.6.1, .NETFramework,Version=v4.6.2, .NETFramework,Version=v4.7, .NETFramework,Version=v4.7.1, .NETFramework,Version=v4.7.2, .NETFramework,Version=v4.8' AndrewsBooks.DataAccess AndrewsBooks.DataAccess instead of the project target framework '.NETCoreApp,Version=v3.1'. This package may not be compatible with your application.				
- Properties:** Shows the properties for the selected project.

- Install NuGet package:
 - Identity.EntityFrameworkCore
- Modify the namespace to reflect the project

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows the solution 'AndrewsBookStore' containing four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore.
- Properties Window:** Visible on the right side of the interface.
- Code Editor:** Displays the file `ApplicationDbContext.cs` with the following code:

```

1  using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Collections.Generic;
5  using System.Text;
6
7  namespace AndrewsBookStore.DataAccess.Data
8  {
9      public class ApplicationDbContext : IdentityDbContext
10     {
11         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
12             : base(options)
13         {
14         }
15     }
16 }

```

- Error List:** Shows three errors and one warning:

Code	Description	Project	File	Line	Suppression State
CS0234	The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)	AndrewsBookStore	Startup.cs	1	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	30	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	37	Active
NU1701	Package 'EntityFramework 6.1.0' was restored using '.NETFramework,Version=v4.6.1, .NETFramework,Version=v4.6.2, .NETFramework,Version=v4.7, .NETFramework,Version=v4.7.1, .NETFramework,Version=v4.7.2, .NETFramework,Version=v4.8' instead of the project target framework '.NETCoreApp,Version=v3.1'. This package may not be compatible with your application.				
- Status Bar:** Shows the status "Ready".

- Install NuGet package:
 - Identity.EntityFrameworkCore
- Modify the namespace to reflect the project

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** Displays the file `Class1.cs` from the `AndrewsBooks.DataAccess` project. The code defines a class `Class1` within the `AndrewsBooks.DataAccess` namespace.
- Solution Explorer:** Shows the solution structure with four projects:
 - `AndrewsBooks.DataAccess`: Contains `Dependencies`, `Packages` (with `Microsoft.AspNetCore.Identity.EntityFrameworkCore`, `Microsoft.EntityFrameworkCore.Relational`, and `Microsoft.EntityFrameworkCore.SqlServer`), and `Data` (containing `ApplicationContext.cs`).
 - `AndrewsBooks.Models`: Contains `Dependencies` and `Class1.cs`.
 - `AndrewsBooks.Utility`: Contains `Dependencies` and `Class1.cs`.
 - `AndrewsBookStore`: Contains `Dependencies` and `Class1.cs`.
- Properties Window:** Shows the properties for the selected `Class1.cs` file under the `AndrewsBookStore` project. It includes sections for `Advanced` (Build Action: `C# compiler`, Copy to Output Directory: `Do not copy`) and `Misc` (File Name: `Class1.cs`, Full Path).
- Output Window:** Displays the build logs for a rebuild operation, showing the compilation of four projects: `AndrewsBooks.Utility`, `AndrewsBooks.Models`, `AndrewsBooks.DataAccess`, and `AndrewsBookStore`. It indicates one error (CS0234) and one failed build step for `AndrewsBookStore`.
- Status Bar:** Shows the status "Ready" at the bottom left.

- Install NuGet package:
 - Identity.EntityFrameworkCore
- Modify the namespace to reflect the project
- Delete default Class1.cs file in all projects

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows the solution 'AndrewsBookStore' containing four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore.
- Properties Window:** For the 'Class1.cs' file in the 'AndrewsBookStore' project, the 'Build Action' is set to 'C# compiler' and 'Copy to Output Directory' is set to 'Do not copy'.
- Output Window:** Displays the build logs:


```
1>----- Rebuild All started: Project: AndrewsBookStore, Configuration: Debug Any CPU -----
2>----- Rebuild All started: Project: AndrewsBooks.DataAccess, Configuration: Debug Any CPU -----
3>----- Rebuild All started: Project: AndrewsBooks.Models, Configuration: Debug Any CPU -----
4>----- Rebuild All started: Project: AndrewsBooks.Utility, Configuration: Debug Any CPU -----
4>AndrewsBooks.Utility -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Utility\bin\Debug\netcoreapp3.1\AndrewsBooks.Utility.dll
3>AndrewsBooks.Models -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Models\bin\Debug\netcoreapp3.1\AndrewsBooks.Models.dll
2>AndrewsBooks.DataAccess -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.DataAccess\bin\Debug\netcoreapp3.1\AndrewsBooks.DataAccess.dll
1>C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\Startup.cs(1,24,1,28): error CS0234: The type or namespace name 'Data' does not exist in the
1>Done building project "AndrewsBookStore.csproj" -- FAILED.
===== Rebuild All: 3 succeeded, 1 failed, 0 skipped ======
```
- Error List:** Shows one error:


```
1>C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\Startup.cs(1,24,1,28): error CS0234: The type or namespace name 'Data' does not exist in the
```

- Install NuGet package:
 - Identity.EntityFrameworkCore
- Modify the namespace to reflect the project
- Delete default Class1.cs file in all projects

The screenshot shows the Visual Studio IDE interface with the following components:

- Code Editor:** Displays `Startup.cs` and `ApplicationDbContext.cs`. The `ApplicationDbContext` file contains the following code:

```

1  using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Collections.Generic;
5  using System.Text;
6
7  namespace AndrewsBookStore.DataAccess.Data
8  {
9      public class ApplicationDbContext : IdentityDbContext
10     {
11         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
12             : base(options)
13         {
14         }
15     }
16 }

```

- Solution Explorer:** Shows the solution structure for 'AndrewsBookStore' (4 projects):
 - AndrewsBooks.DataAccess
 - AndrewsBooks.Models
 - AndrewsBooks.Utility
 - AndrewsBookStore (selected)
 The `Packages` node under `AndrewsBookStore` lists:
 - Microsoft.AspNetCore.Identity.EntityFrameworkCore (5.0.2)
 - Microsoft.EntityFrameworkCore.Relational (5.0.11)
 - Microsoft.EntityFrameworkCore.SqlServer (5.0.11)
- Error List:** Shows three errors:

Code	Description	Project	File	Line	Suppression State
CS0234	The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)	AndrewsBookStore	Startup.cs	1	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	30	Active
CS0246	The type or namespace name ' ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	37	Active
- Properties:** Shows the properties for the `AndrewsBookStore` project, including Name: `AndrewsBookStore`, Active config: `Debug|Any CPU`, and Path: `C:\Users\ASTEEL\source\repos\AndrewsBookStore`.

- Install NuGet package:
 - Identity.EntityFrameworkCore
- Modify the namespace to reflect the project
- Delete default Class1.cs file in all projects
- Build the project

The screenshot shows the Microsoft Visual Studio interface with the following components:

- Code Editor:** Displays the `Startup.cs` file with the following code:

```

1  using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Collections.Generic;
5  using System.Text;
6
7  namespace AndrewsBookStore.DataAccess.Data
8  {
9      public class ApplicationDbContext : IdentityDbContext
10     {
11         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
12             : base(options)
13         {
14         }
15     }
16 }

```

- Solution Explorer:** Shows the solution structure for 'AndrewsBookStore' (4 projects):
 - AndrewsBooks.DataAccess
 - AndrewsBooks.Models
 - AndrewsBooks.Utility
 - AndrewsBookStore (selected)
 The `Packages` node under `AndrewsBookStore` lists:
 - Microsoft.AspNetCore.Identity.EntityFrameworkCore (5.0.2)
 - Microsoft.EntityFrameworkCore.Relational (5.0.11)
 - Microsoft.EntityFrameworkCore.SqlServer (5.0.11)
- Error List:** Shows three errors:

Code	Description	Project	File	Line	Suppression State
CS0234	The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)	AndrewsBookStore	Startup.cs	1	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	30	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	37	Active
- Properties:** Shows the properties for the `AndrewsBookStore` project:

(Name)	AndrewsBookStore
Active config	Debug Any CPU
Description	
Path	C:\Users\ASTEEL\source\repos\AndrewsBo
Startup project	AndrewsBookStore

- Move Models in to AndrewsBooks.Models (delete original)

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Code Editor:** Displays `Startup.cs` with the following code:

```
1  using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Collections.Generic;
5  using System.Text;
6
7  namespace AndrewsBookStore.DataAccess.Data
8  {
9      public class ApplicationDbContext : IdentityDbContext
10     {
11         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
12             : base(options)
13         {
14         }
15     }
16 }
```
- Solution Explorer:** Shows the solution structure for 'AndrewsBookStore' (4 projects). The 'Models' folder under 'AndrewsBookStore' is selected.
- Error List:** Shows three errors:

Code	Description	Project	File	Line	Suppression State
CS0234	The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)	AndrewsBookStore	Startup.cs	1	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	30	Active
CS0246	The type or namespace name ' ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	37	Active
- Properties:** Shows the 'Models' folder properties with the full path: `C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\AndrewsBookStore\AndrewsBookStore\Models`.

- Move Models in to AndrewsBooks.Models (delete original)

The screenshot shows the Visual Studio IDE interface with the following details:

- Code Editor:** Displays the `Startup.cs` file with the following code snippet:


```
1  using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Collections.Generic;
5  using System.Text;
6
7  namespace AndrewsBookStore.DataAccess.Data
8  {
9      public class ApplicationDbContext : IdentityDbContext
10     {
11         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
12             : base(options)
13         {
14         }
15     }
16 }
```
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore' (4 of 4 projects):
 - AndrewsBooks.DataAccess** (selected):
 - Dependencies
 - Data
 - AndrewsBooks.Models**:
 - Dependencies
 - Models
 - ErrorViewModel.cs
- AndrewsBookStore**:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Views
 - appsettings.json

- Error List:** Shows three errors:

Code	Description	Project	File	Line	Suppression State
CS0234	The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)	AndrewsBookStore	Startup.cs	1	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	30	Active
CS0246	The type or namespace name ' ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	37	Active
- Properties:** Shows the 'Models' folder properties with the following details:

Misc	Folder Name	Models
	Folder Name	C:\Users\ASTEEL\source\repos\AndrewsBo
	Full Path	C:\Users\ASTEEL\source\repos\AndrewsBo

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml

The screenshot shows the Visual Studio IDE interface with the following details:

- Code Editor:** Displays the `Startup.cs` file with the following code snippet:


```
1  using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Collections.Generic;
5  using System.Text;
6
7  namespace AndrewsBookStore.DataAccess.Data
8  {
9      public class ApplicationDbContext : IdentityDbContext
10     {
11         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
12             : base(options)
13         {
14         }
15     }
16 }
```
- Solution Explorer:** Shows the solution structure for 'AndrewsBookStore' (4 of 4 projects):
 - AndrewsBooks.DataAccess** (selected):
 - Dependencies
 - Data
 - AndrewsBooks.Models**:
 - Dependencies
 - Models
 - ErrorViewModel.cs
- AndrewsBookStore**:
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Views
 - appsettings.json

- Error List:** Shows three errors:

Code	Description	Project	File	Line	Suppression State
CS0234	The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)	AndrewsBookStore	Startup.cs	1	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	30	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	37	Active
- Properties:** Shows the properties for the 'Models' folder.

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models

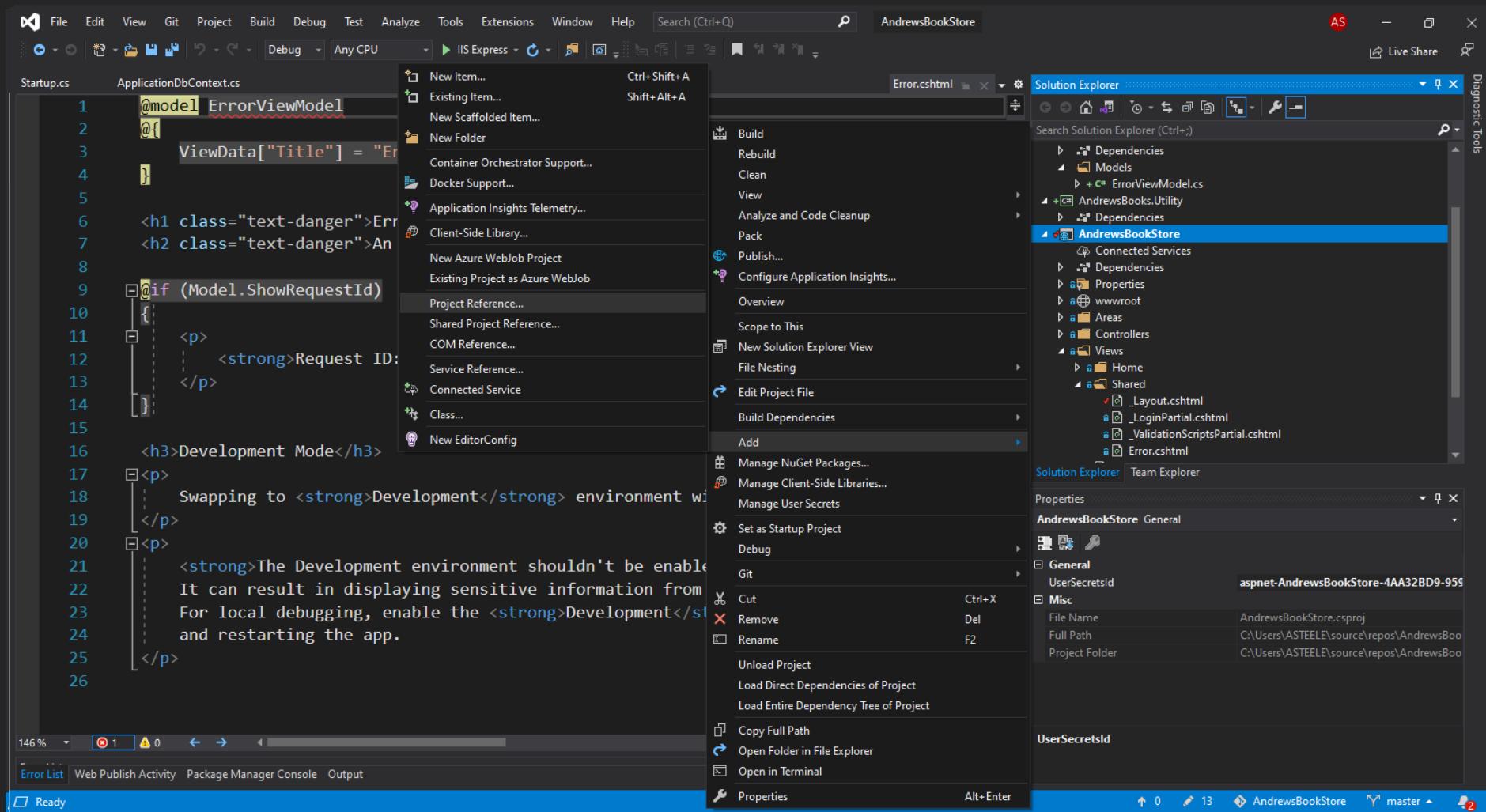
The screenshot shows the Visual Studio IDE interface with the following details:

- Code Editor:** Displays the `Startup.cs` file with the following code snippet:

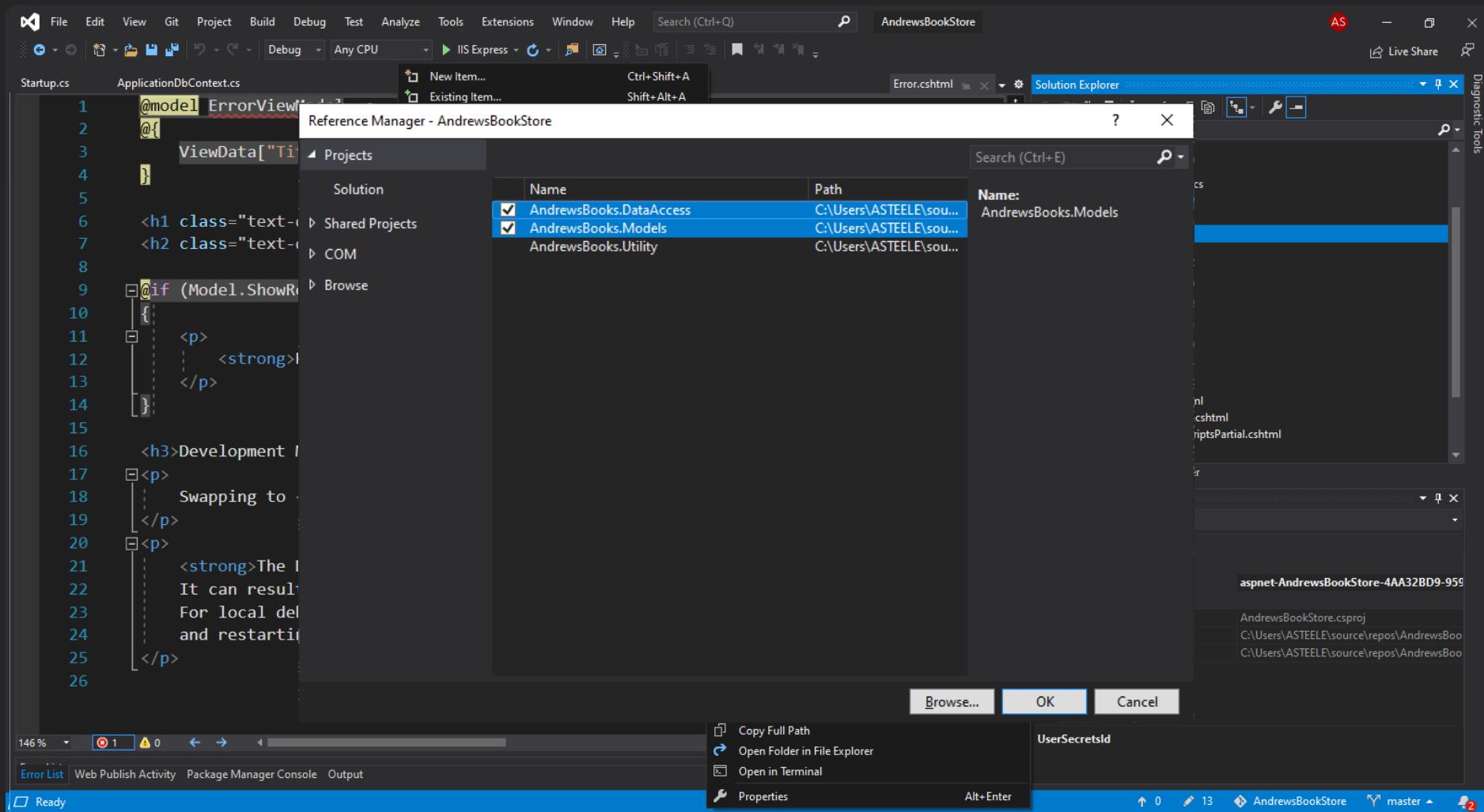

```
1  using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore;
3  using System;
4  using System.Collections.Generic;
5  using System.Text;
6
7  namespace AndrewsBookStore.DataAccess.Data
8  {
9      public class ApplicationDbContext : IdentityDbContext
10     {
11         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
12             : base(options)
13         {
14         }
15     }
16 }
```
- Solution Explorer:** Shows the solution structure for 'AndrewsBookStore' with four projects:
 - AndrewsBooks.DataAccess
 - AndrewsBooks.Models (selected)
 - AndrewsBooks.Utility
 - AndrewsBookStore
- Error List:** Shows three errors:

Code	Description	Project	File	Line	Suppression State
CS0234	The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)	AndrewsBookStore	Startup.cs	1	Active
CS0246	The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	30	Active
CS0246	The type or namespace name ' ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)	AndrewsBookStore	Startup.cs	37	Active
- Properties:** Shows the properties for the 'Models' folder.

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models



- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models



- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** AndrewsBookStore
- Toolbars:** Standard, Debug, Windows, Help
- Search Bar:** Search (Ctrl+Q)
- Startup.cs:** Shows code for setting up the application.
- Error.cshtml:** The current file being edited, containing the following code:

```

1  @model ErrorViewModel
2  @{
3      ViewData["Title"] = "Error";
4  }
5
6  <h1 class="text-danger">Error.</h1>
7  <h2 class="text-danger">An error occurred while processing your request.</h2>
8
9 @if (Model.ShowRequestId)
10 {
11     <p>
12         <strong>Request ID:</strong> <code>@Model.RequestId</code>
13     </p>
14 }
15
16 <h3>Development Mode</h3>
17 <p>
18     Swapping to <strong>Development</strong> environment will display more detailed information.
19 </p>
20 <p>
21     <strong>The Development environment shouldn't be enabled for deployed applications.</strong>
22     It can result in displaying sensitive information from exceptions to end users.
23     For local debugging, enable the <strong>Development</strong> environment by setting the
24     environment variable <code>ASPNETCORE_ENVIRONMENT</code> to <code>Development</code> and
25     restarting the app.
26

```

- Solution Explorer:** Shows the project structure:
 - AndrewsBooks.Models
 - AndrewsBooks.Utility
 - AndrewsBookStore (selected)
 - Connected Services
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
- Properties:** Shows file properties for selected files.
- Bottom Status Bar:** Ready, 0, 13, AndrewsBookStore, master, 2

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Code Editor:** Displays the `ErrorViewModel.cs` file under the `AndrewsBooks.Models` namespace. The code defines a class `ErrorViewModel` with properties `RequestId` and `ShowRequestId`.
- Solution Explorer:** Shows the solution structure for 'AndrewsBookStore' (4 projects):
 - `AndrewsBooks.DataAccess`
 - `AndrewsBooks.Models`
 - `AndrewsBooks.Utility`
 - `AndrewsBookStore` (selected)
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Views
 - Home
- Properties Window:** Shows the `Models` folder properties with the following settings:
 - Folder Name: `Models`
 - Full Path: `C:\Users\ASTEELE\source\repos\AndrewsBo`
- Status Bar:** Shows the current branch as `master` with a commit count of 13.

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models
- Rename Models folder to ViewModels

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** Displays the `ErrorViewModel.cs` file under the `AndrewsBooks.Models` namespace. The code defines a class `ErrorViewModel` with a property `RequestId` and a method `ShowRequestId`.
- Solution Explorer:** Shows the solution structure for 'AndrewsBookStore' with four projects: `AndrewsBooks.DataAccess`, `AndrewsBooks.Models`, `AndrewsBooks.Utility`, and `AndrewsBookStore`. The `ViewModels` folder under `AndrewsBooks.Models` is selected.
- Properties Window:** Shows the properties for the `ViewModels` folder, including the full path: `C:\Users\ASTEELE\source\repos\AndrewsBo`.
- Status Bar:** Shows the current branch as `master` with two changes.

```

1 using System;
2
3 namespace AndrewsBookStore.Models
4 {
5     public class ErrorViewModel
6     {
7         public string RequestId { get; set; }
8
9         public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
10    }
11}
12

```

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models
- Rename Models folder to ViewModels
- Change the ErrorViewModels.cs namespace .Models.ViewModels

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** Displays the file `ErrorViewModel.cs` containing the following C# code:


```

1 using System;
2
3 namespace AndrewsBookStore.Models
4 {
5     public class ErrorViewModel
6     {
7         public string RequestId { get; set; }
8
9         public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
10    }
11}
12
      
```
- Solution Explorer:** Shows the project structure for 'AndrewsBookStore' (4 projects):
 - AndrewsBooks.DataAccess**: Contains `Dependencies`, `Data`, and `Properties`.
 - AndrewsBooks.Models**: Contains `Dependencies` and **ViewModels** (selected).
 - AndrewsBooks.Utility**: Contains `Dependencies` and `Properties`.
 - AndrewsBookStore**: Contains `Connected Services`, `Dependencies`, `Properties`, `wwwroot`, `Areas`, `Controllers`, `Views` (with `Home` selected), and `Properties`.
- Properties:** Shows the properties for the `ViewModels` folder:
 - Folder Name**: `ViewModels`
 - Full Path**: `C:\Users\ASTEELE\source\repos\AndrewsBo`
- Status Bar:** Shows the commit count (2), branch name (`master`), and repository name (`AndrewsBookStore`).

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models
- Rename Models folder to ViewModels
- Change the ErrorViewModels.cs namespace .Models.ViewModels

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** The main window displays the `ErrorViewModel.cs` file. The code defines a class `ErrorViewModel` with properties `RequestId` and `ShowRequestId`.
- Solution Explorer:** On the right, the Solution Explorer shows the project structure for 'AndrewsBookStore' (4 of 4 projects). It includes:
 - AndrewsBooks.DataAccess**: Dependencies, Data
 - AndrewsBooks.Models**: Dependencies, ViewModels (containing `ErrorViewModel.cs`)
 - AndrewsBooks.Utility**: Dependencies
 - AndrewsBookStore**: Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers, Views (containing Home)
- Properties:** A small panel on the right side of the Solution Explorer.
- Status Bar:** At the bottom, it shows the status bar with 'Ready', 'Ln: 3 Ch: 45 SPC CRLF', and a GitHub icon indicating 2 changes.

```

1  using System;
2
3  namespace AndrewsBookStore.Models.ViewModels
4  {
5      public class ErrorViewModel
6      {
7          public string RequestId { get; set; }
8
9          public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
10     }
11 }
12

```

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models
- Rename Models folder to ViewModels
- Change the ErrorViewModels.cs namespace .Models.ViewModels
- Build Project

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** The main window displays the `ErrorViewModel.cs` file. The code defines a class `ErrorViewModel` with a property `RequestId` and a method `ShowRequestId`.
- Solution Explorer:** On the right, the Solution Explorer shows the project structure for 'AndrewsBookStore' (4 of 4 projects). It includes:
 - AndrewsBooks.DataAccess**: Contains `Dependencies`, `Data`, and `Properties`.
 - AndrewsBooks.Models**: Contains `Dependencies`, `ViewModels` (which contains `ErrorViewModel.cs`), and `Properties`.
 - AndrewsBooks.Utility**: Contains `Dependencies` and `Properties`.
 - AndrewsBookStore**: Contains `Connected Services`, `Dependencies`, `Properties`, `wwwroot`, `Areas`, `Controllers`, `Views` (which contains `Home`), and `Properties`.
- Properties:** A small properties panel is visible at the bottom of the Solution Explorer.
- Bottom Status Bar:** Shows the status bar with various icons and text like 'Ready', 'Ln: 3 Ch: 45 SPC CRLF', and a GitHub icon with 'master' and '2'.

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models
- Rename Models folder to ViewModels
- Change the ErrorViewModels.cs namespace .Models.ViewModels
- Build Project

The screenshot shows the Microsoft Visual Studio interface with the following details:

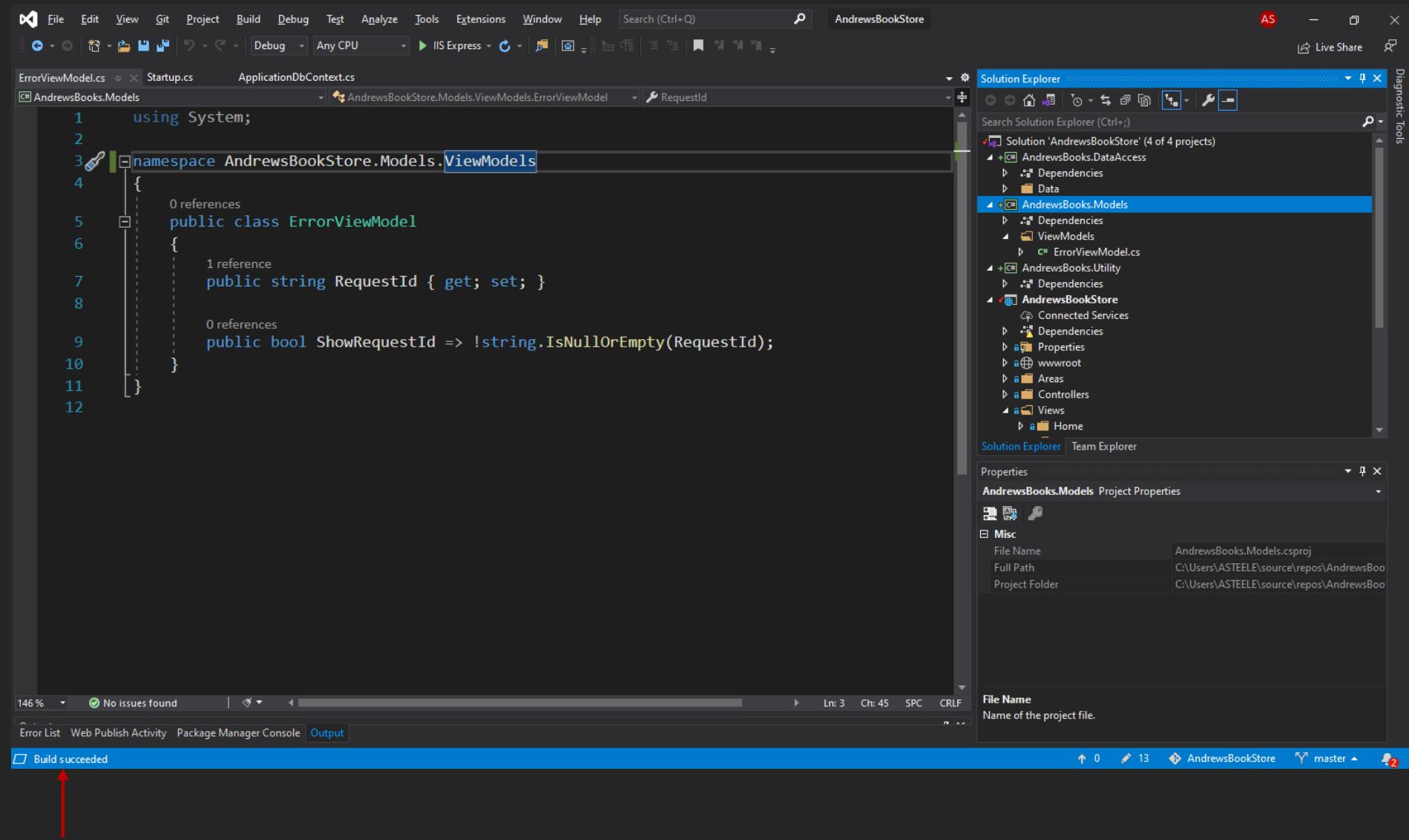
- Code Editor:** The main window displays the `ErrorViewModel.cs` file. The code defines a class `ErrorViewModel` with a property `RequestId`.
- Solution Explorer:** On the right, the Solution Explorer shows the project structure for 'AndrewsBookStore' (4 of 4 projects):
 - AndrewsBooks.DataAccess**: Contains Dependencies and Data.
 - AndrewsBooks.Models**: Contains Dependencies, ViewModels (with `ErrorViewModel.cs` checked), and Utilities.
 - AndrewsBooks.Utility**: Contains Dependencies.
 - AndrewsBookStore**: Contains Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers, Views (with Home checked).
- Properties:** A properties panel is open for the `AndrewsBooks.Models` project, showing the file name, full path, and project folder.
- Status Bar:** At the bottom, it shows 'Build succeeded'.

```

1  using System;
2
3  namespace AndrewsBookStore.Models.ViewModels
4  {
5      public class ErrorViewModel
6      {
7          public string RequestId { get; set; }
8
9          public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
10     }
11 }
12

```

- Move Models in to AndrewsBooks.Models (delete original)
- Modify Views > Shared > Error.cshtml
- Project - Add - Project Reference - .DataAccess and .Models
- Rename Models folder to ViewModels
- Change the ErrorViewModels.cs namespace .Models.ViewModels
- Build Project



The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** The main window displays the `ErrorViewModel.cs` file. The code defines a class `ErrorViewModel` with a property `RequestId`.
- Solution Explorer:** Shows the solution structure for 'AndrewsBookStore' (4 of 4 projects):
 - AndrewsBooks.DataAccess**: Dependencies, Data
 - AndrewsBooks.Models**: Dependencies, ViewModels (containing `ErrorViewModel.cs`), Utility
 - AndrewsBooks.Utility**: Dependencies
 - AndrewsBookStore**: Connected Services, Dependencies, Properties, wwwroot, Areas, Controllers, Views (containing Home)
- Properties:** The properties for **AndrewsBooks.Models** are shown, including File Name (`AndrewsBooks.Models.csproj`), Full Path (`C:\Users\ASTEEL\source\repos\AndrewsBo`), and Project Folder (`C:\Users\ASTEEL\source\repos\AndrewsBo`).
- Status Bar:** The status bar at the bottom indicates "Build succeeded".

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q) with a magnifying glass icon.
- Project Name:** AndrewsBookStore
- Toolbars:** Standard toolbar with icons for Open, Save, Print, etc.
- Startup.cs Editor:** The main code editor window displays the `Startup.cs` file. The code is as follows:

```
20 public Startup(IConfiguration configuration)
21 {
22     Configuration = configuration;
23 }
24
25 public IConfiguration Configuration { get; }
26
27 // This method gets called by the runtime. Use this method to add services to the
28 // services.AddDbContext<ApplicationDbContext>(options =>
29 //    options.UseSqlServer(
30 //        Configuration.GetConnectionString("DefaultConnection")));
31 //services.AddDatabaseDeveloperPageExceptionFilter();
32
33 // removed 'options => options.SignIn.RequireConfirmedAccount = true'
34 services.AddDefaultIdentity<IdentityUser>()
35     .AddEntityFrameworkStores<ApplicationDbContext>();
36
37 using AndrewsBookStore.DataAccess.Data; > CS0246 The type or namespace name 'ApplicationDbContext' could
38 // not be found (are you missing a using directive or an assembly reference?)
39 // Generate type 'ApplicationDbContext'
40 // Use discard '_'
41 // Wrapping
42 // Suppress or Configure issues
43 if (env.IsDevelopment())
44 {
45     if (env.IsDevelopment())
46     {
47         app.UseDeveloperExceptionPage();
48         app.UseMigrationsEndPoint();
49     }
50     else
51     {
52         app.UseExceptionHandler("/Error");
53         app.UseHsts();
54     }
55 }
56
57 app.UseHttpsRedirection();
58 app.UseStaticFiles();
59
60 app.UseRouting();
61
62 app.UseAuthorization();
63
64 app.MapControllerRoute(
65     name: "default",
66     pattern: "{controller=Home}/{action=Index}/{id?}");
67 }
```

A tooltip for the CS0246 error is displayed at the bottom left of the editor area:

CS0246 The type or namespace name 'ApplicationDbContext' could not be found (are you missing a using directive or an assembly reference?)

The Solution Explorer pane on the right shows the project structure:

- AndrewsBookStore
- Connected Services
- Dependencies
- Properties
- wwwroot
- Areas
- Controllers
- Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml
- appsettings.json
- Program.cs
- Startup.cs

The Properties and Team Explorer panes are also visible on the right side of the interface.

- Modify Startup.cs service
'AddContext' with the
using statement

```
AndrewsBookStore
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AndrewsBookStore
Debug Any CPU IIS Express
AS Live Share Diagnostic Tools
Startup.cs + x ApplicationDbContext.cs
AndrewsBookStore.Startup ConfigureServices(IServiceCollection services)
20 public Startup(IConfiguration configuration)
21 {
22     Configuration = configuration;
23 }
24
25 public IConfiguration Configuration { get; }
26
27 // This method gets called by the runtime. Use this method to add services to the
28 // services.AddDbContext<ApplicationDbContext>(options =>
29 //    options.UseSqlServer(
30 //        Configuration.GetConnectionString("DefaultConnection")));
31 //    services.AddDatabaseDeveloperPageExceptionFilter();
32
33 // removed 'options => options.SignIn.RequireConfirmedAccount = true'
34 services.AddDefaultIdentity<IdentityUser>()
35     .AddEntityFrameworkStores<ApplicationDbContext>();
36
37 using AndrewsBookStore.DataAccess.Data; > CS0246 The type or namespace name 'ApplicationDbContext' could
38 DataAccess.DataContext Generate type 'ApplicationDbContext' not be found (are you missing a using directive or an assembly reference?)
39 Use discard ' '
40 Wrapping
41 Suppress or Configure issues
42 Preview changes
43 if (env.IsDevelopment())
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146 % 0 13 AndrewsBookStore master 2
Ready
```

Solution Explorer

- AndrewsBookStore
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Areas
 - Controllers
 - Views
 - Home
 - Shared
 - _Layout.cshtml
 - _LoginPartial.cshtml
 - _ValidationScriptsPartial.cshtml
 - Error.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml
 - appsettings.json
 - Program.cs
 - Startup.cs

Properties

- Modify Startup.cs service 'AddContext' with the using statement
 - Run the application and review errors.

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `Startup.cs` file for the `AndrewsBookStore` project. The code is as follows:

```
public void ConfigureServices(IServiceCollection services)
{
    Configuration = configuration;

    services.AddDbContext<ApplicationContext>(options =>
        options.UseSqlServer(
            Configuration.GetConnectionString("DefaultConnection")));
    services.AddDatabaseDeveloperPageExceptionFilter();

    services.AddDefaultIdentity<IdentityUser>()
        .AddEntityFrameworkStores<ApplicationContext>();

    using AndrewsBookStore.DataAccess.Data;
    using Microsoft.AspNetCore.Builder;
    using Microsoft.Extensions.Configuration;
    using Microsoft.Extensions.DependencyInjection;
    using Microsoft.Extensions.Hosting;
}
```

A tooltip for the `CS0246` error is visible at the bottom of the code editor, pointing to the first `using` statement. The tooltip text is: "CS0246 The type or namespace name 'ApplicationContext' could not be found (are you missing a using directive or an assembly reference?)". Below the tooltip, there are several context menu options: "Generate type 'ApplicationContext'", "Use discard '_'", "Wrapping", "Suppress or Configure issues", and "Preview changes".

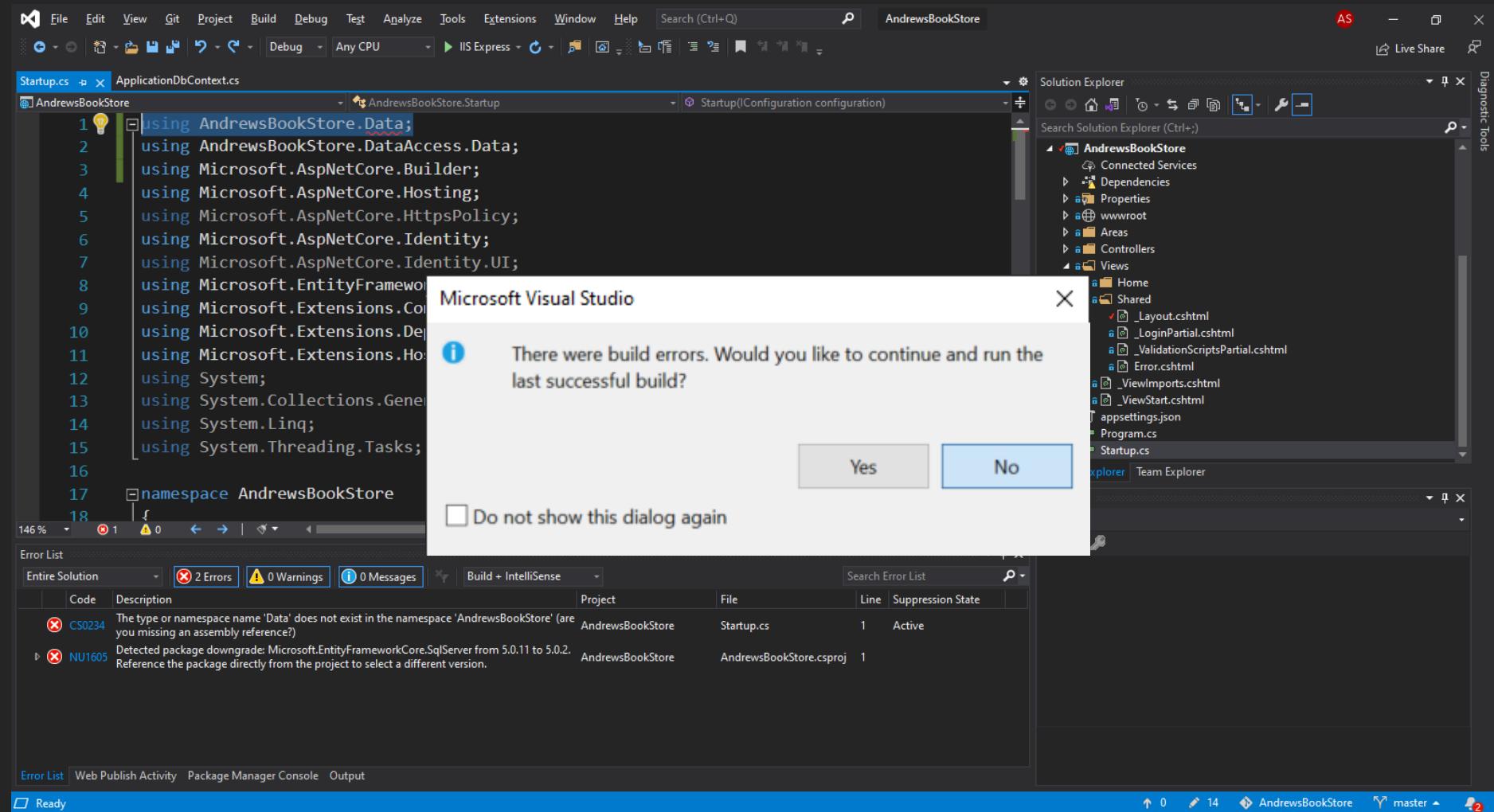
The Solution Explorer pane on the right shows the project structure for `AndrewsBookStore`, including files like `Program.cs` and `Startup.cs`. The Properties pane is also visible.

- Modify Startup.cs service 'AddContext' with the using statement
- Run the application and review errors.

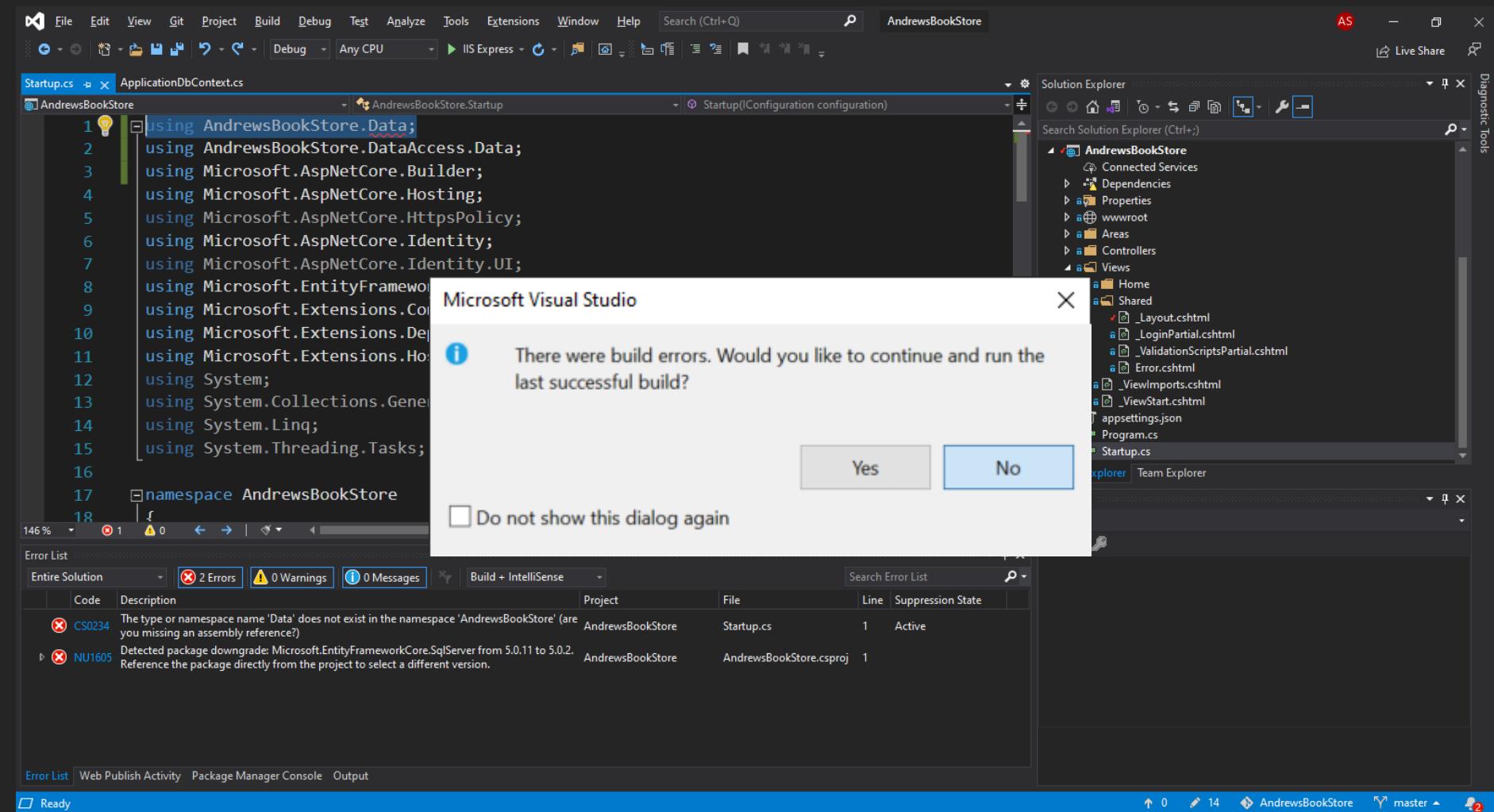
The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** The main window displays the `Startup.cs` file. The first line contains a warning icon and the code: `using AndrewsBookStore.Data;`. The rest of the file includes standard ASP.NET Core startup configurations.
- Solution Explorer:** Located on the right, it shows the project structure for "AndrewsBookStore". It includes files like `Program.cs`, `Startup.cs`, `appsettings.json`, and various views and controllers.
- Error List:** The bottom-left pane shows two errors:
 - CS0234:** The type or namespace name 'Data' does not exist in the namespace 'AndrewsBookStore' (are you missing an assembly reference?)
 - NU1605:** Detected package downgrade: Microsoft.EntityFrameworkCore.SqlServer from 5.0.11 to 5.0.2. Reference the package directly from the project to select a different version.
- Status Bar:** At the bottom, it shows "Ready" and other system information.

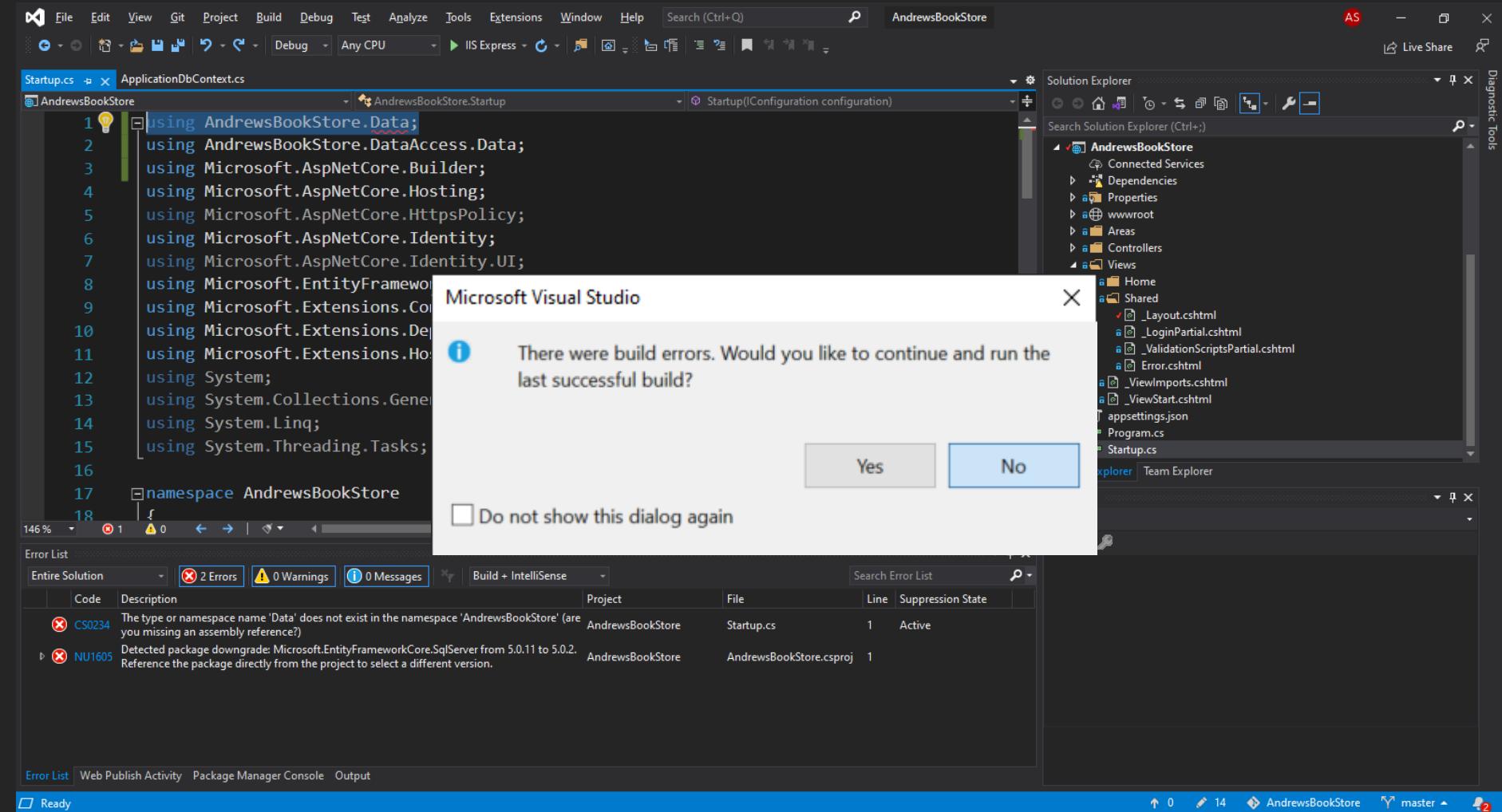
- Modify Startup.cs service 'AddContext' with the using statement
- Run the application and review errors.



- Modify Startup.cs service 'AddContext' with the using statement
- Run the application and review errors.
- Remove the using statement



- Modify Startup.cs service 'AddContext' with the using statement
- Run the application and review errors.
- Remove the using statement
- Correct any default reference to ErrorViewModel to the new .Models.ViewModels.ErrorViewModels



- Modify Startup.cs service 'AddContext' with the using statement
- Run the application and review errors.
- Remove the using statement
- Correct any default reference to ErrorViewModel to the new .Models.ViewModels.ErrorViewModels

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `Error.cshtml` file, which contains the following code:

```

1  @model AndrewsBookStore.Models.ViewModels.ErrorViewModel
2  @{
3      ViewData["Title"] = "Error";
4  }
5
6  <h1 class="text-danger">Error.</h1>
7  <h2 class="text-danger">An error occurred while processing your request.</h2>
8
9  @if (Model.ShowRequestId)
10 {
11     <p>
12         <strong>Request ID:</strong> <code>@Model.RequestId</code>
13     </p>
14 }
15
16 <h3>Development Mode</h3>
17 <p>
18     Swapping to <strong>Development</strong> environment will display more detailed information about the error.
19 </p>
20 <p>
21     <strong>The Development environment shouldn't be enabled for deployed applications.</strong>
22     It can result in displaying sensitive information from exceptions to end users.
23     For local debugging, enable the <strong>Development</strong> environment by setting the <strong>ASPNETCORE_ENVIRONMENT</strong> environment variable to 'Development' and restarting the app.
24 </p>
25
26

```

The Solution Explorer on the right shows the project structure:

- Solution 'AndrewsBookStore' (4 of 4 projects)
 - AndrewsBooks.DataAccess
 - AndrewsBooks.Models
 - AndrewsBooks.Utility
 - AndrewsBookStore
 - Connected Services
 - Dependencies
 - Imports
 - Properties
 - wwwroot
 - Areas
 - bin
 - Controllers
 - obj
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs**

The status bar at the bottom indicates the current branch is 'master'.

- Modify Startup.cs service 'AddContext' with the using statement
- Run the application and review errors.
- Remove the using statement
- Correct any default reference to ErrorViewModel to the new .Models.ViewModels.ErrorViewModels
- When all errors are resolved run the application and review the browser presentation.

```

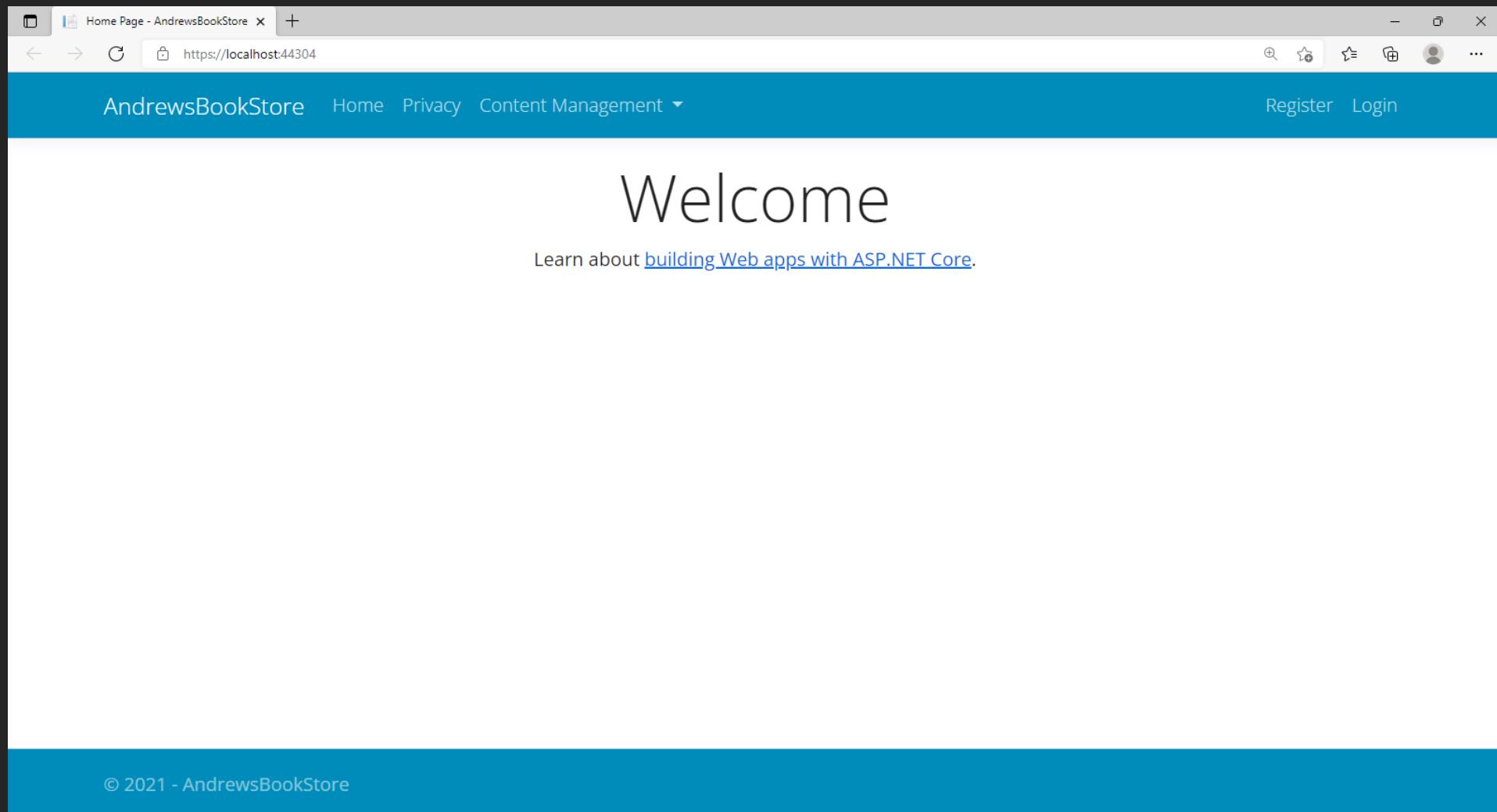
1  @model AndrewsBookStore.Models.ViewModels.ErrorViewModel
2  @{
3      ViewData["Title"] = "Error";
4  }
5
6  <h1 class="text-danger">Error.</h1>
7  <h2 class="text-danger">An error occurred while processing your request.</h2>
8
9  @if (Model.ShowRequestId)
10 {
11     <p>
12         <strong>Request ID:</strong> <code>@Model.RequestId</code>
13     </p>
14 }
15
16 <h3>Development Mode</h3>
17 <p>
18     Swapping to <strong>Development</strong> environment will display more detailed information about the error.
19 </p>
20 <p>
21     <strong>The Development environment shouldn't be enabled for deployed applications.</strong>
22     It can result in displaying sensitive information from exceptions to end users.
23     For local debugging, enable the <strong>Development</strong> environment by setting the <strong>ASPNETCORE_ENVIRONMENT</strong> environment variable to 'Development' and restarting the app.
24 </p>
25
26

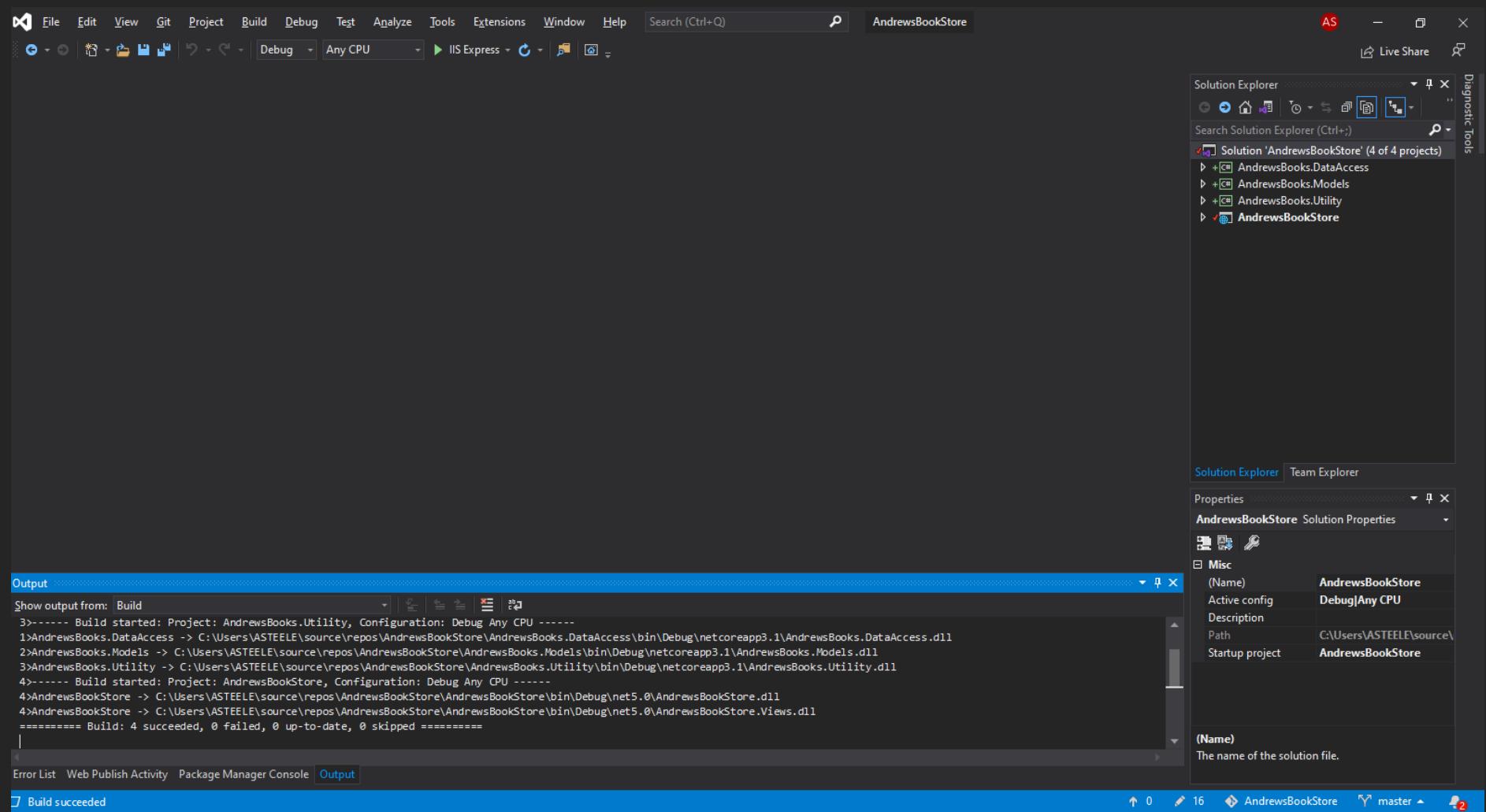
```

The screenshot shows the Visual Studio IDE interface with the following details:

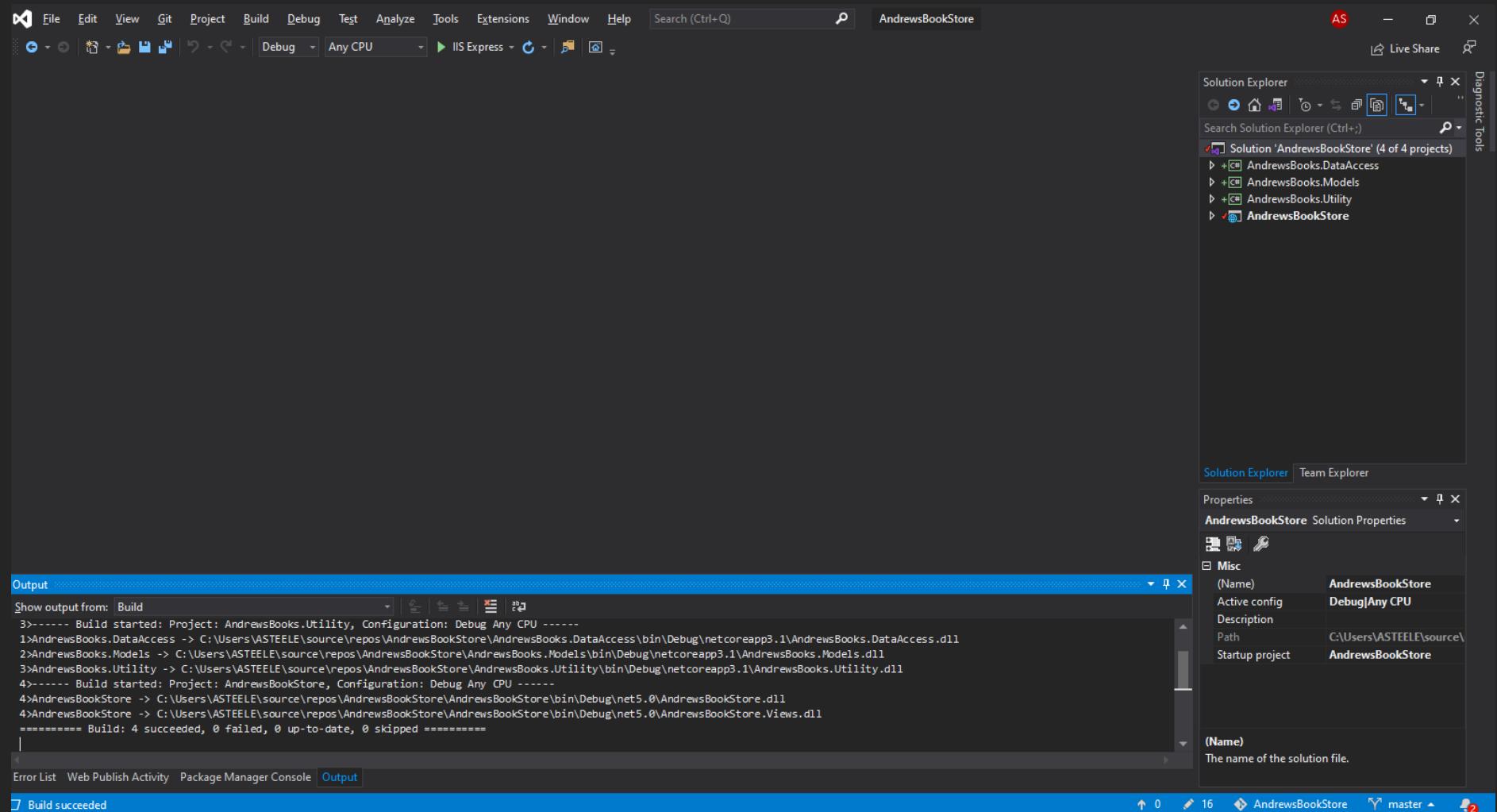
- Solution Explorer:** Shows the solution structure with four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore.
- Properties:** Shows the properties for the AndrewsBookStore project, including Connected Services, Dependencies, Imports, Properties, wwwroot, Areas, bin, Controllers, obj, Views, appsettings.json, Program.cs, and Startup.cs.
- Startup.cs:** The file is open in the editor, showing the configuration for the application's startup.
- Error.cshtml:** The file is also open in the editor, displaying the error page template.
- Status Bar:** Shows the current branch as 'master' and the number of changes as '2'.

- Modify Startup.cs service 'AddContext' with the using statement
- Run the application and review errors.
- Remove the using statement
- Correct any default reference to ErrorViewModel to the new .Models.ViewModels.Error ViewModels
- When all errors are resolved run the application and review the browser presentation.

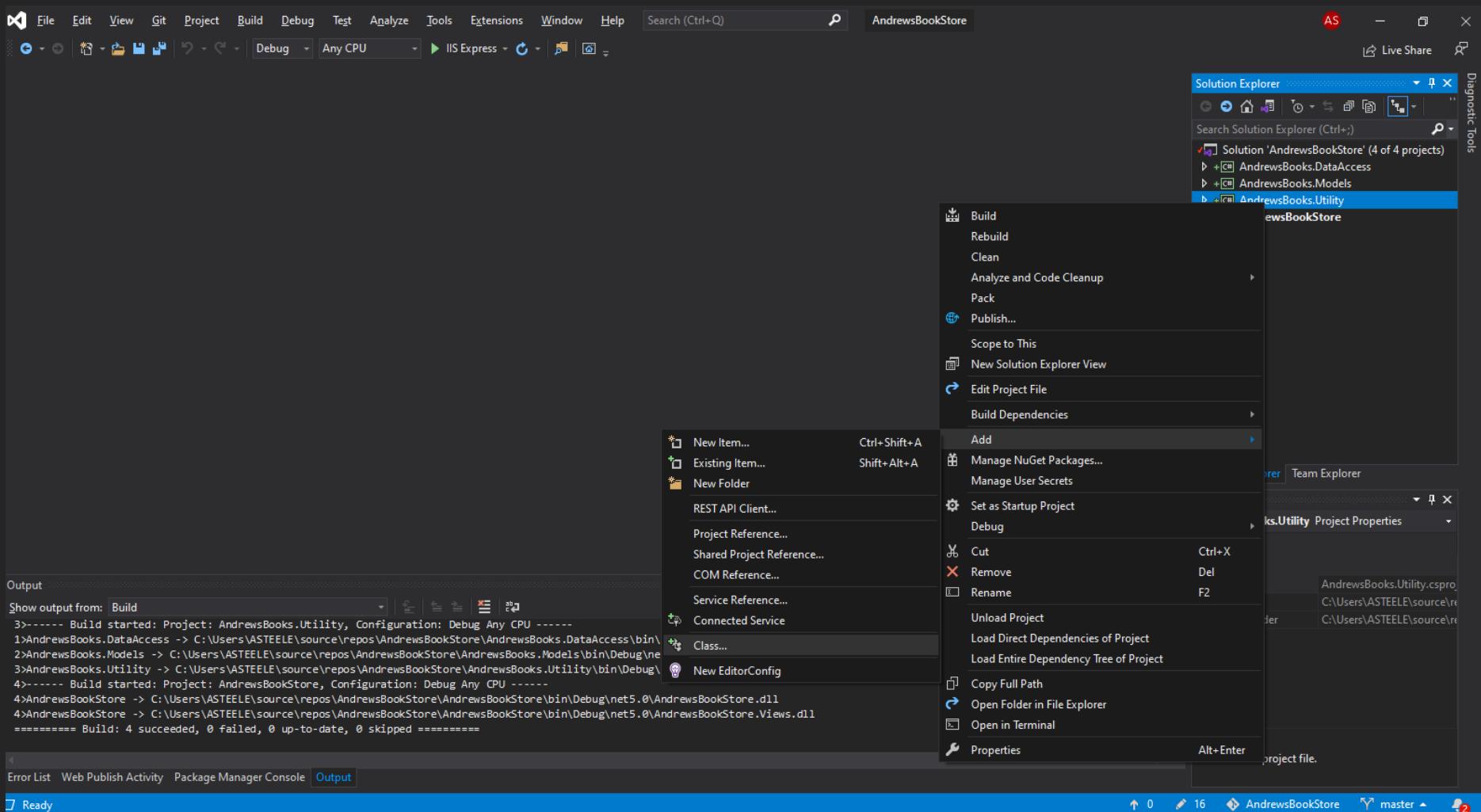




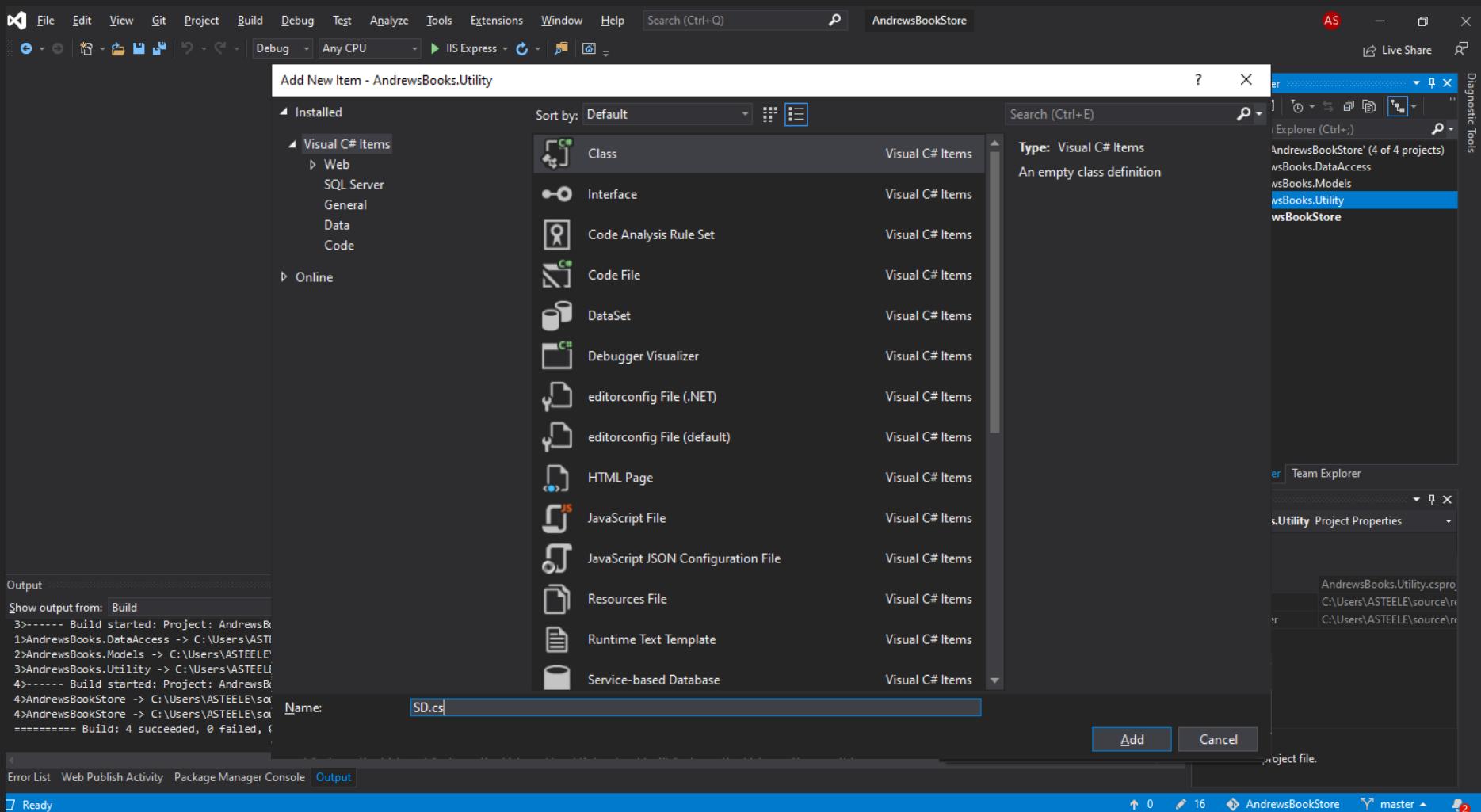
- In the Utility project, create a static details class called SD.cs



- In the Utility project, create a static details class called SD.cs



- In the Utility project, create a static details class called SD.cs



- In the Utility project, create a static details class called SD.cs

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows the solution 'AndrewsBookStore' containing four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore. The AndrewsBooks.Utility project is selected, and its folder 'Dependencies' contains the file 'SD.cs', which is highlighted in blue.
- SD.cs File Content:** The code defines a static class named SD within the namespace AndrewsBooks.Utility. The code is as follows:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace AndrewsBooks.Utility
6 {
7     class SD
8     {
9     }
10}
11
```

- Output Window:** Displays the build log for the 'Utility' project:

```
3>----- Build started: Project: AndrewsBooks.Utility, Configuration: Debug Any CPU -----
1>AndrewsBooks.DataAccess -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.DataAccess\bin\Debug\netcoreapp3.1\AndrewsBooks.DataAccess.dll
2>AndrewsBooks.Models -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Models\bin\Debug\netcoreapp3.1\AndrewsBooks.Models.dll
3>AndrewsBooks.Utility -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Utility\bin\Debug\netcoreapp3.1\AndrewsBooks.Utility.dll
4>----- Build started: Project: AndrewsBookStore, Configuration: Debug Any CPU -----
4>AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.dll
4>AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.Views.dll
===== Build: 4 succeeded, 0 failed, 0 up-to-date, 0 skipped ======
```

- Properties Window:** Shows the properties for the 'SD.cs' file, including the 'Build Action' set to 'C# compiler'.

- In the Utility project, create a static details class called SD.cs
- Modify the properties of the class

SD.cs

```

1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace AndrewsBooks.Utility
6 {
7     class SD
8     {
9     }
10}
11

```

Output

```

3>----- Build started: Project: AndrewsBooks.Utility, Configuration: Debug Any CPU -----
1>AndrewsBooks.DataAccess -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.DataAccess\bin\Debug\netcoreapp3.1\AndrewsBooks.DataAccess.dll
2>AndrewsBooks.Models -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Models\bin\Debug\netcoreapp3.1\AndrewsBooks.Models.dll
3>AndrewsBooks.Utility -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Utility\bin\Debug\netcoreapp3.1\AndrewsBooks.Utility.dll
4>----- Build started: Project: AndrewsBookStore, Configuration: Debug Any CPU -----
4>AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.dll
4>AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.Views.dll
===== Build: 4 succeeded, 0 failed, 0 up-to-date, 0 skipped =====

```

Properties

File Name	SD.cs
Full Path	C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Utility\SD.cs

Build Action

How the file relates to the build and deployment processes.

- In the Utility project, create a static details class called SD.cs
- Modify the properties of the class

The screenshot shows the Microsoft Visual Studio interface with the following components visible:

- Solution Explorer:** Shows the solution structure with four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore. The AndrewsBooks.Utility project is selected.
- Properties Window:** Visible on the right side of the interface.
- Output Window:** Shows the build log output.
- Code Editor:** Displays the SD.cs file from the AndrewsBooks.Utility project. The code defines a static class SD with no members.

```

1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace AndrewsBooks.Utility
6  {
7      public static class SD
8      {
9      }
10 }
11

```

```

146% No issues found
Output
Show output from: Build
3>----- Build started: Project: AndrewsBooks.Utility, Configuration: Debug Any CPU -----
1>AndrewsBooks.DataAccess -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.DataAccess\bin\Debug\netcoreapp3.1\AndrewsBooks.DataAccess.dll
2>AndrewsBooks.Models -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Models\bin\Debug\netcoreapp3.1\AndrewsBooks.Models.dll
3>AndrewsBooks.Utility -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Utility\bin\Debug\netcoreapp3.1\AndrewsBooks.Utility.dll
4>----- Build started: Project: AndrewsBookStore, Configuration: Debug Any CPU -----
4>AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.dll
4>AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.Views.dll
===== Build: 4 succeeded, 0 failed, 0 up-to-date, 0 skipped =====

```

- In the Utility project, create a static details class called SD.cs
- Modify the properties of the class
- Add project reference to the main project

The screenshot shows the Microsoft Visual Studio interface with the following components visible:

- Solution Explorer:** Shows the solution structure with four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore. The AndrewsBooks.Utility project is selected.
- Properties Window:** Visible on the right side of the interface.
- Output Window:** Shows the build log output.
- Code Editor:** Displays the SD.cs file from the AndrewsBooks.Utility project.

```

SD.cs -> X
AndrewsBooks.Utility
AndrewsBooks.Utility.SD

1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace AndrewsBooks.Utility
6  {
7      public static class SD
8      {
9      }
10 }
11

```

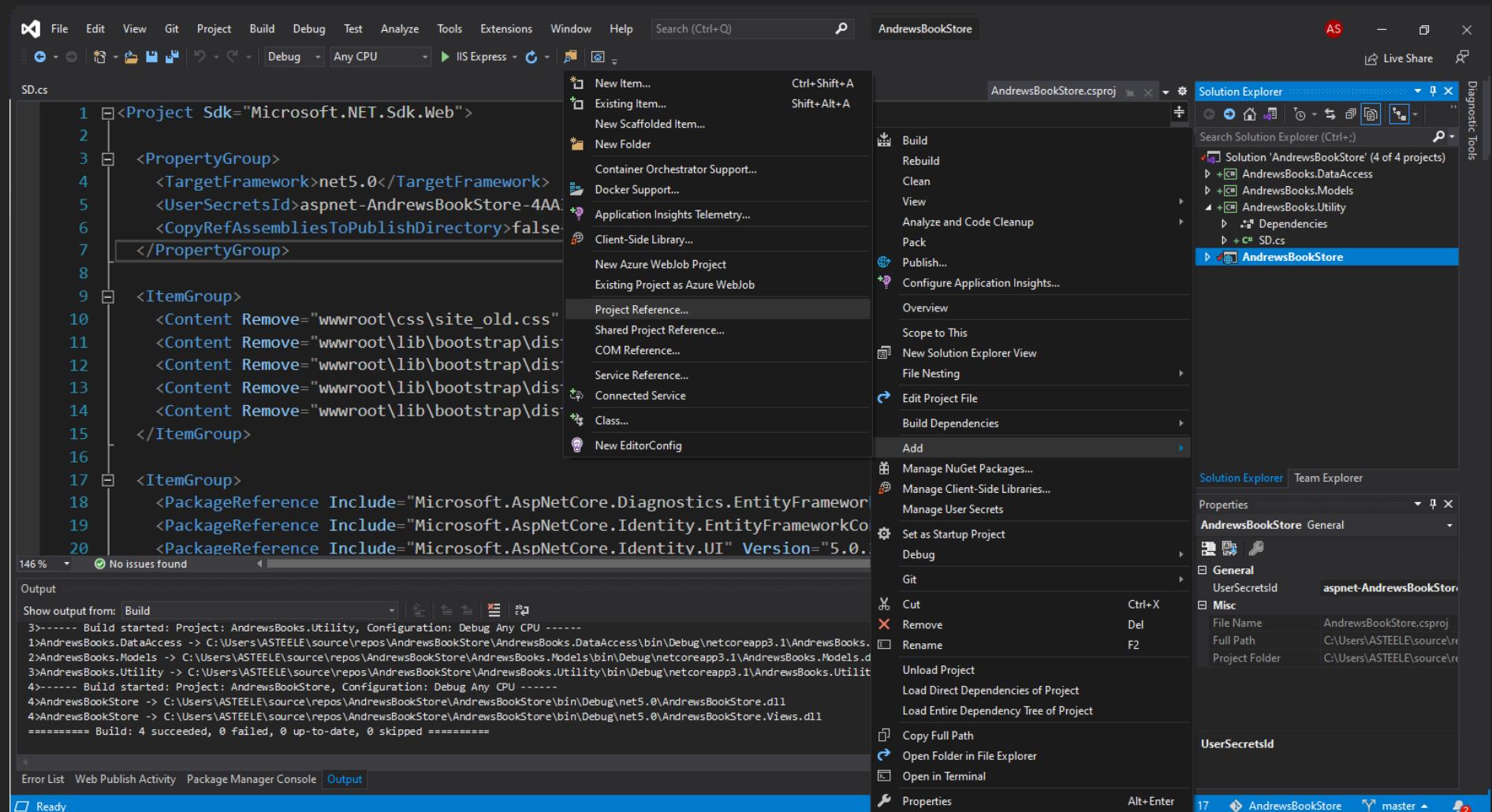
Output Window Content:

```

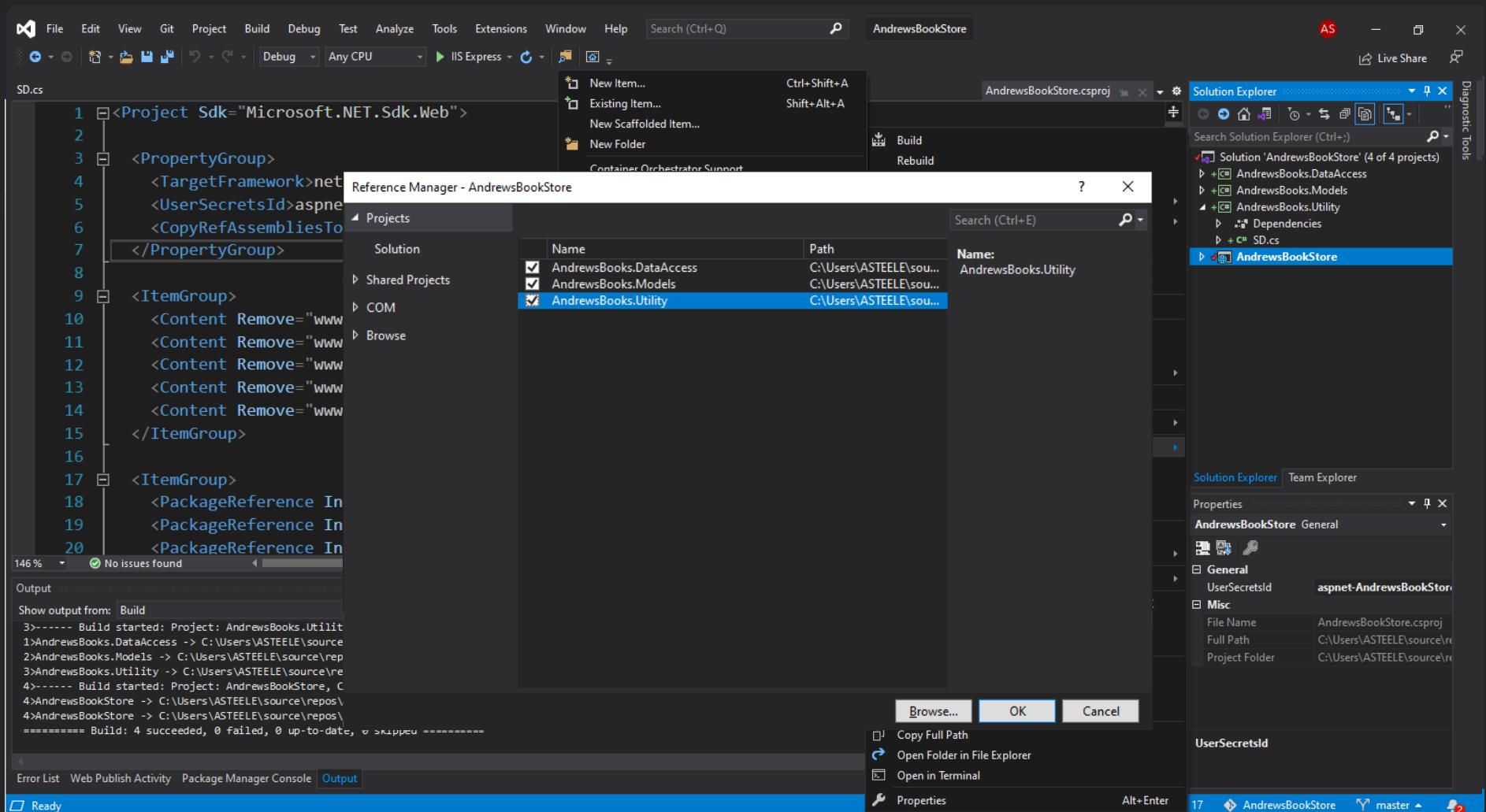
1>----- Build started: Project: AndrewsBooks.Utility, Configuration: Debug Any CPU -----
1> AndrewsBooks.DataAccess -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.DataAccess\bin\Debug\netcoreapp3.1\AndrewsBooks.DataAccess.dll
2> AndrewsBooks.Models -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Models\bin\Debug\netcoreapp3.1\AndrewsBooks.Models.dll
3> AndrewsBooks.Utility -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Utility\bin\Debug\netcoreapp3.1\AndrewsBooks.Utility.dll
4>----- Build started: Project: AndrewsBookStore, Configuration: Debug Any CPU -----
4> AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.dll
4> AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.Views.dll
===== Build: 4 succeeded, 0 failed, 0 up-to-date, 0 skipped =====

```

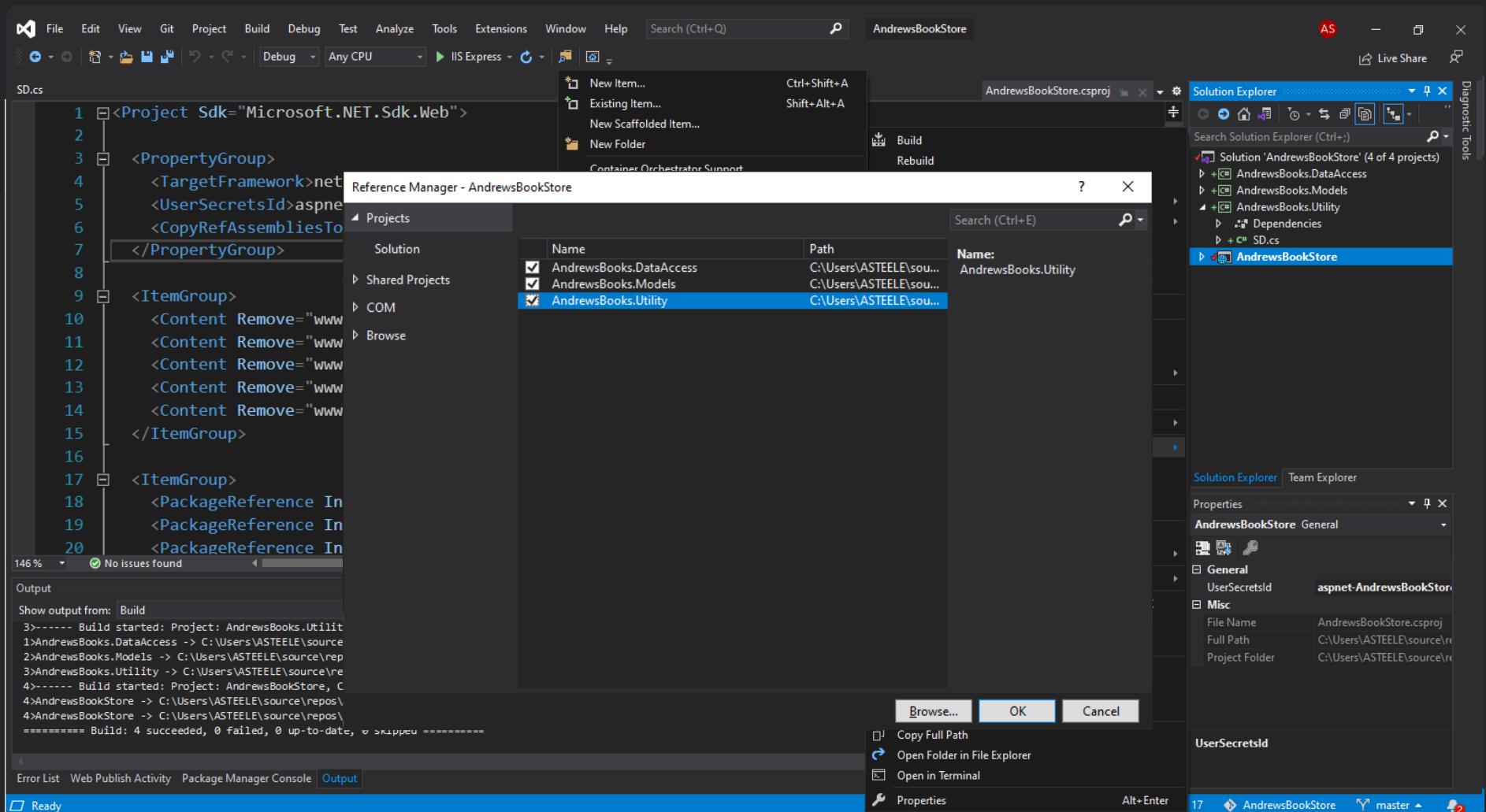
- In the Utility project, create a static details class called SD.cs
- Modify the properties of the class
- Add project reference to the main project



- In the Utility project, create a static details class called SD.cs
- Modify the properties of the class
- Add project reference to the main project



- In the Utility project, create a static details class called SD.cs
- Modify the properties of the class
- Add project reference to the main project
- In the DataAccess project add project references to Models and Utility

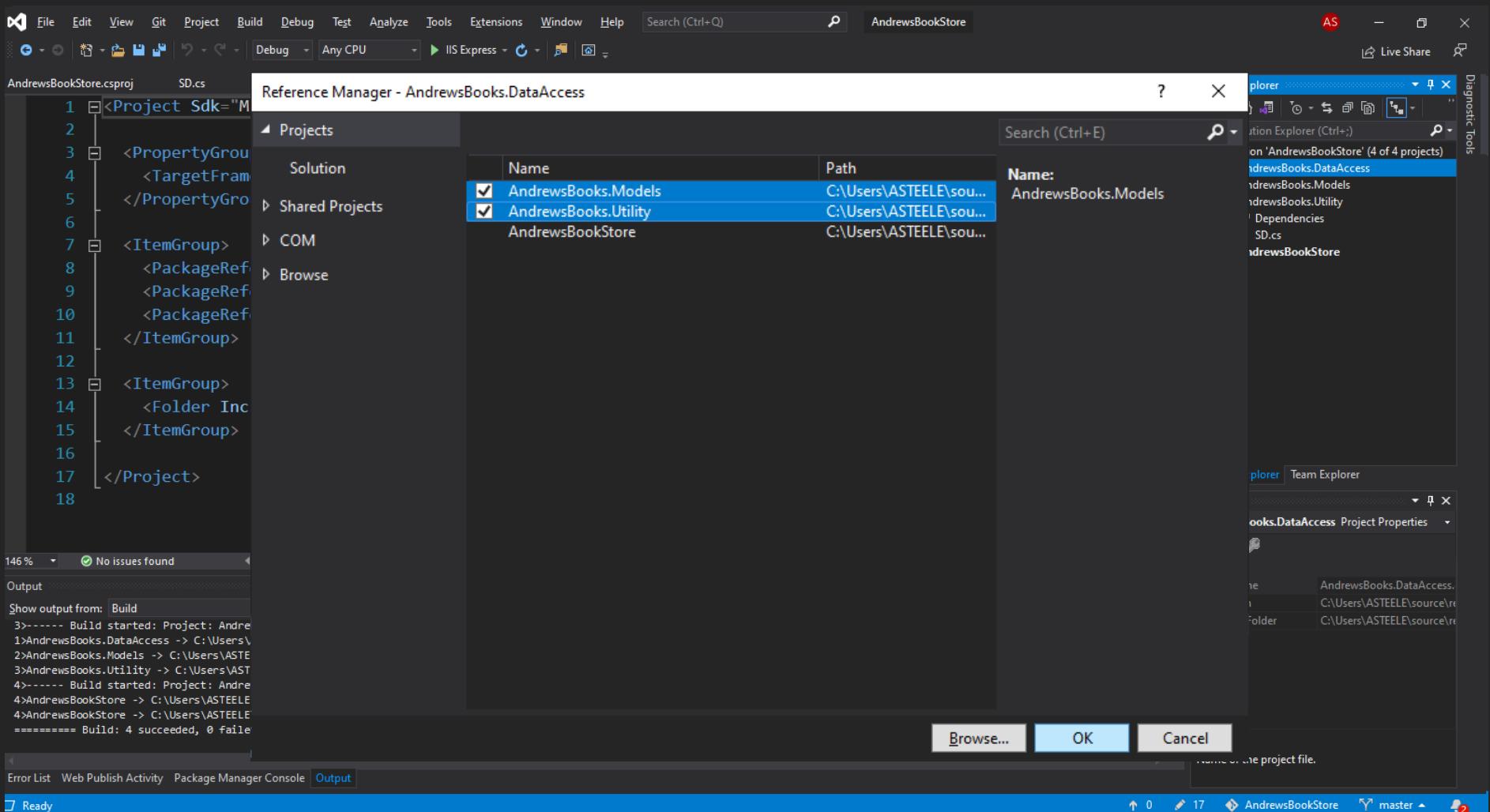


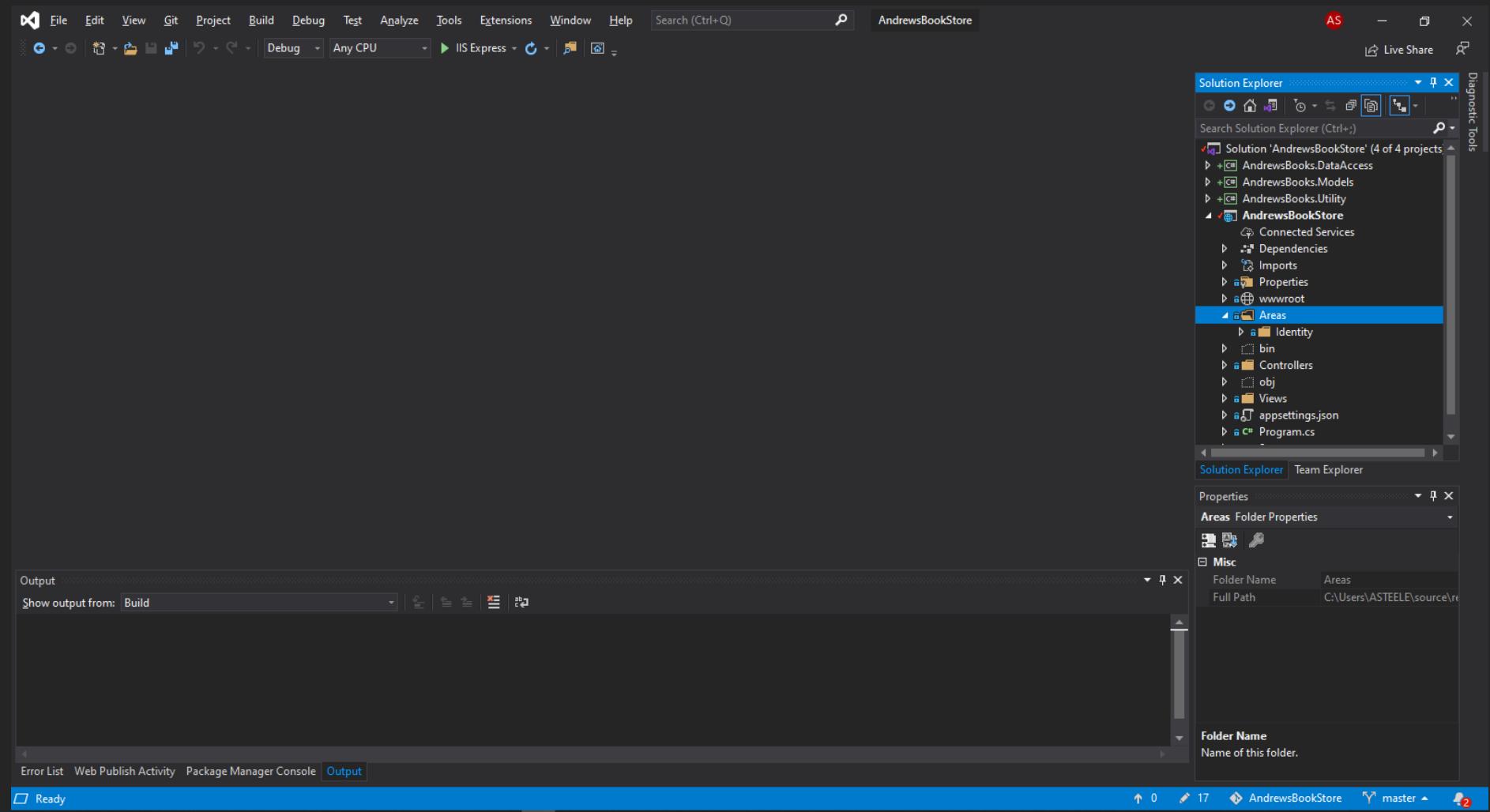
- In the Utility project, create a static details class called SD.cs
- Modify the properties of the class
- Add project reference to the main project
- In the DataAccess project add project references to Models and Utility

The screenshot shows the Microsoft Visual Studio interface with the following details:

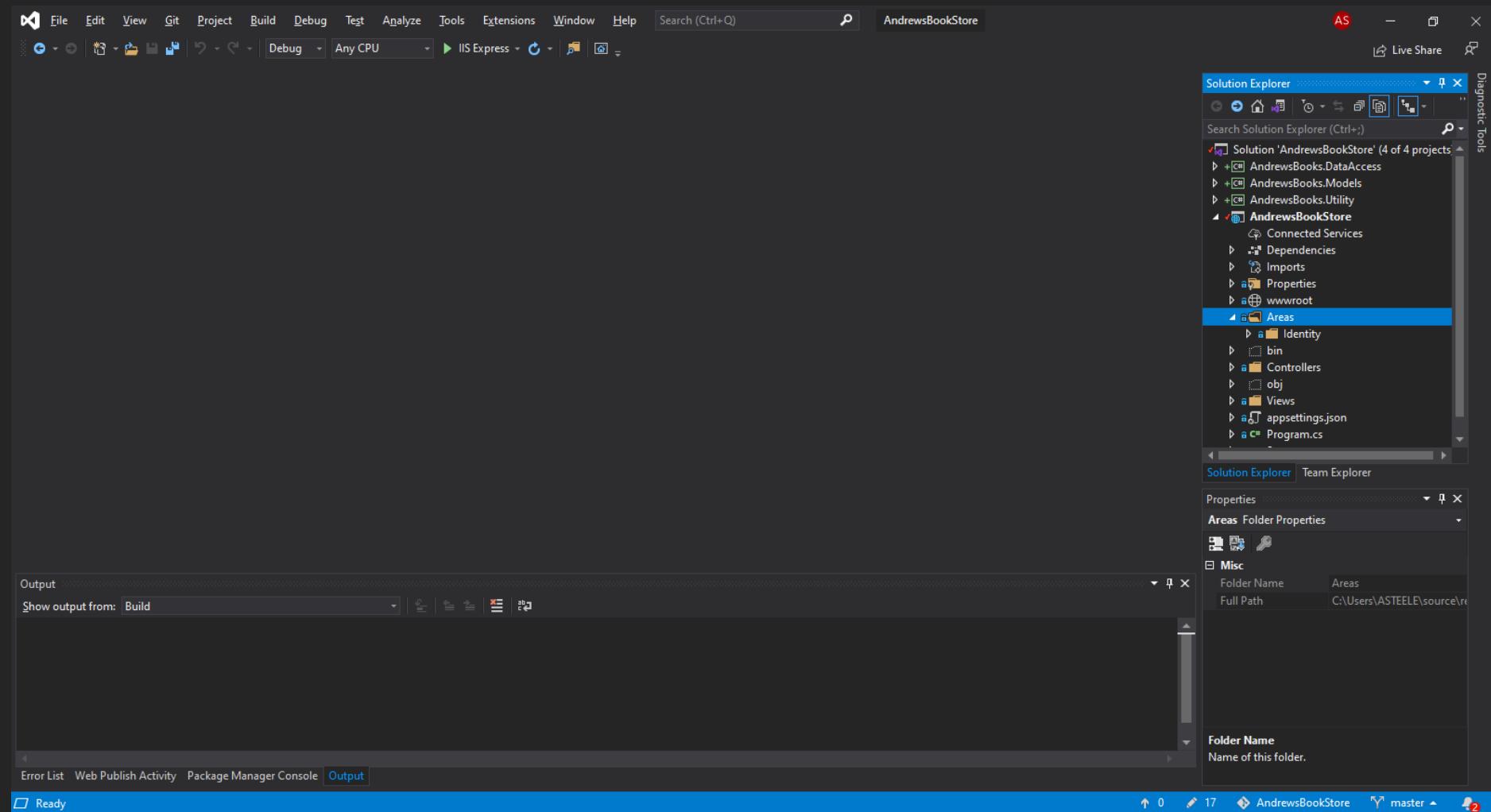
- Solution Explorer:** Shows the solution 'AndrewsBookStore' containing four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore.
- Code Editor:** Displays the contents of the `AndrewsBooks.DataAccess.csproj` file, specifically the project configuration XML. Key parts include:
 - `<Project Sdk="Microsoft.NET.Sdk">`
 - `<PropertyGroup>` block setting `TargetFramework` to `netcoreapp3.1`.
 - `<ItemGroup>` blocks for package references to Entity Framework Core and a folder named `Data\`.
 - `</Project>`
- Output Window:** Shows the build log output for four projects:
 - Build started: Project: AndrewsBooks.Utility, Configuration: Debug Any CPU -----
 - 1>AndrewsBooks.DataAccess -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.DataAccess\bin\Debug\netcoreapp3.1\AndrewsBooks.DataAccess.dll
 - 2>AndrewsBooks.Models -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Models\bin\Debug\netcoreapp3.1\AndrewsBooks.Models.dll
 - 3>AndrewsBooks.Utility -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBooks.Utility\bin\Debug\netcoreapp3.1\AndrewsBooks.Utility.dll
 - 4>----- Build started: Project: AndrewsBookStore, Configuration: Debug Any CPU -----
 - 4>AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.dll
 - 4>AndrewsBookStore -> C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\bin\Debug\net5.0\AndrewsBookStore.Views.dll
 - ===== Build: 4 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
- Properties Window:** Shows the properties for the `AndrewsBooks.DataAccess` project, including the file name, full path, and project folder.

- In the Utility project, create a static details class called SD.cs
- Modify the properties of the class
- Add project reference to the main project
- In the DataAccess project add project references to Models and Utility

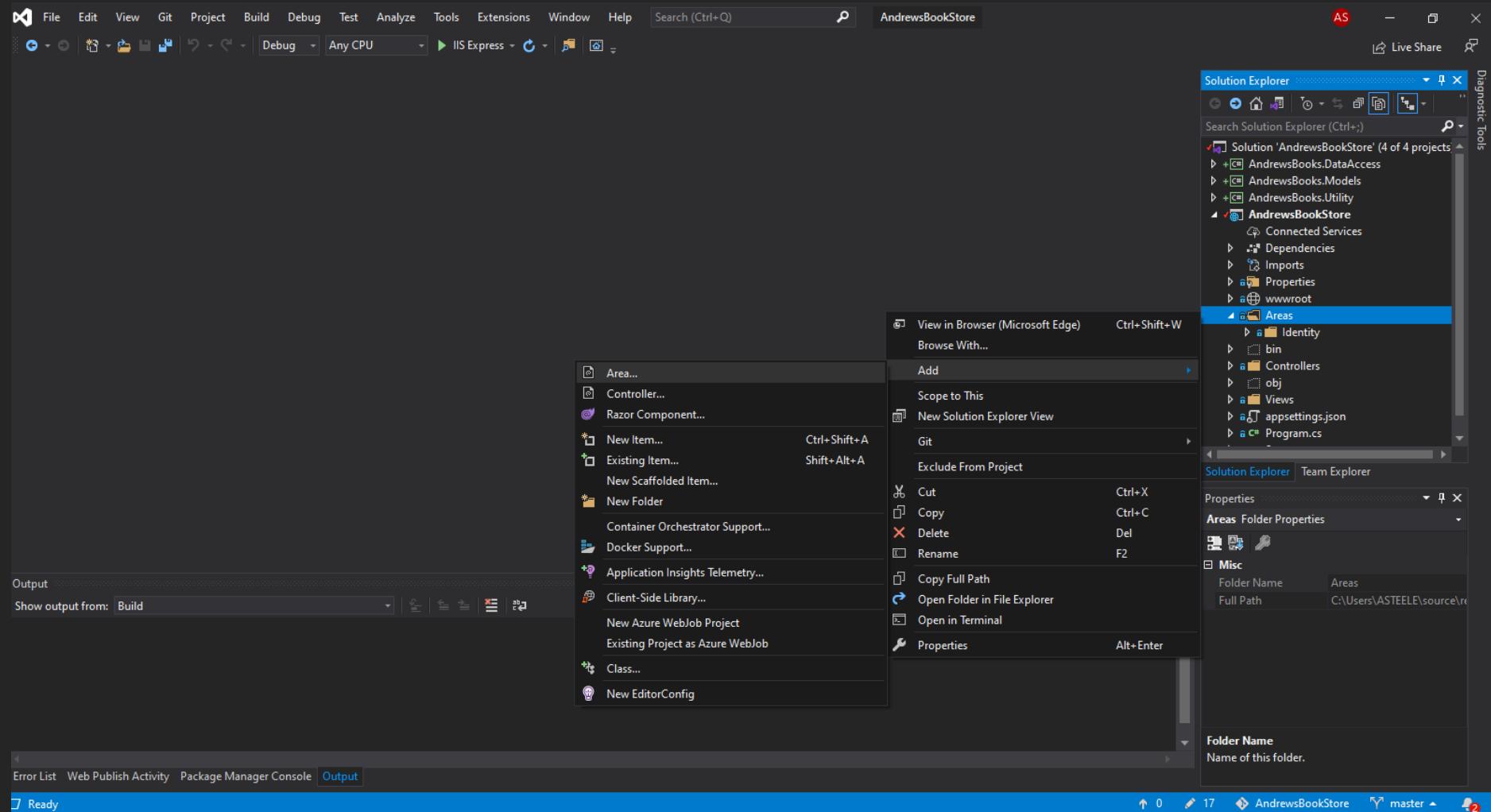




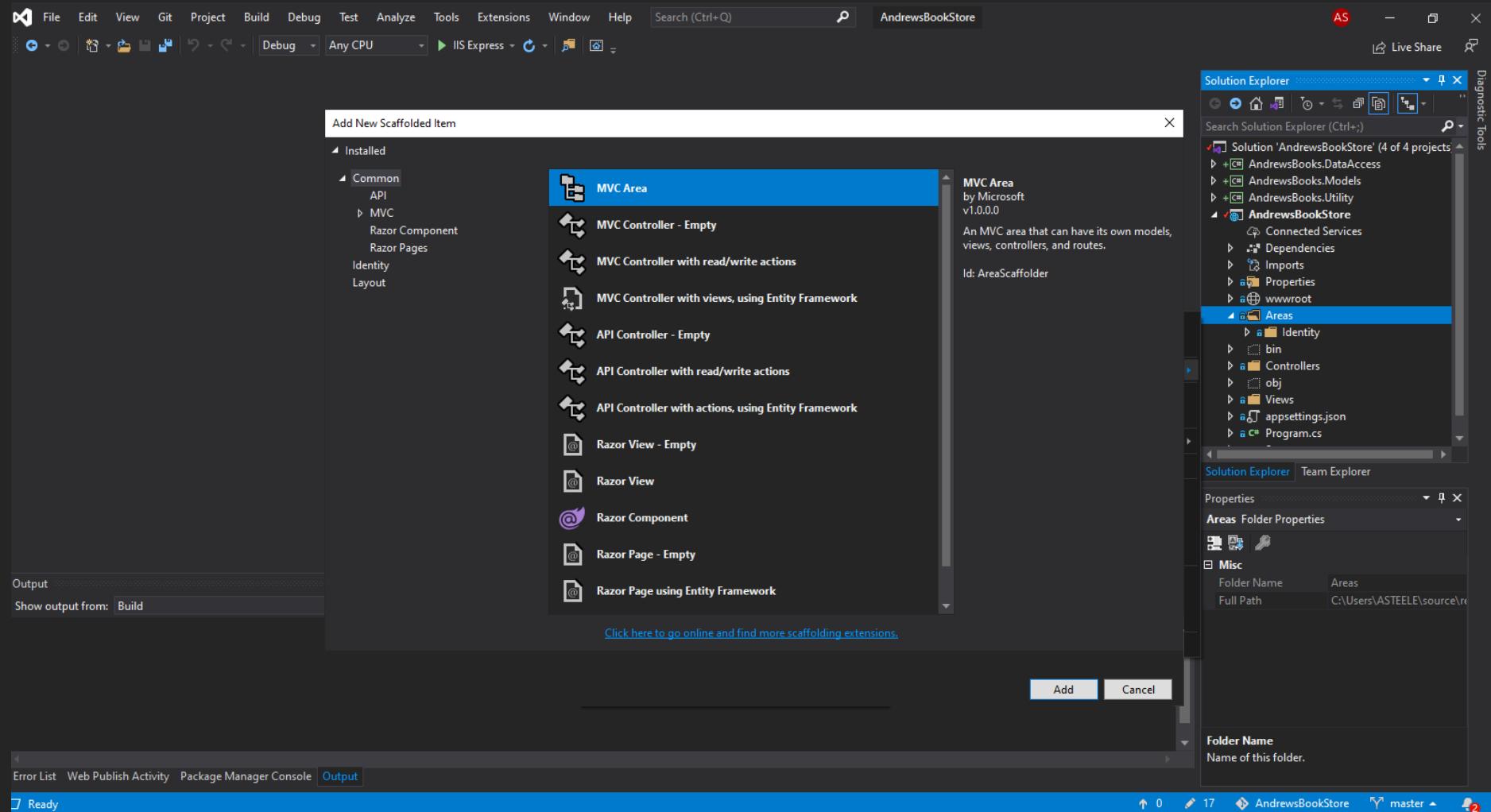
- Add a 'Customers' area to Areas



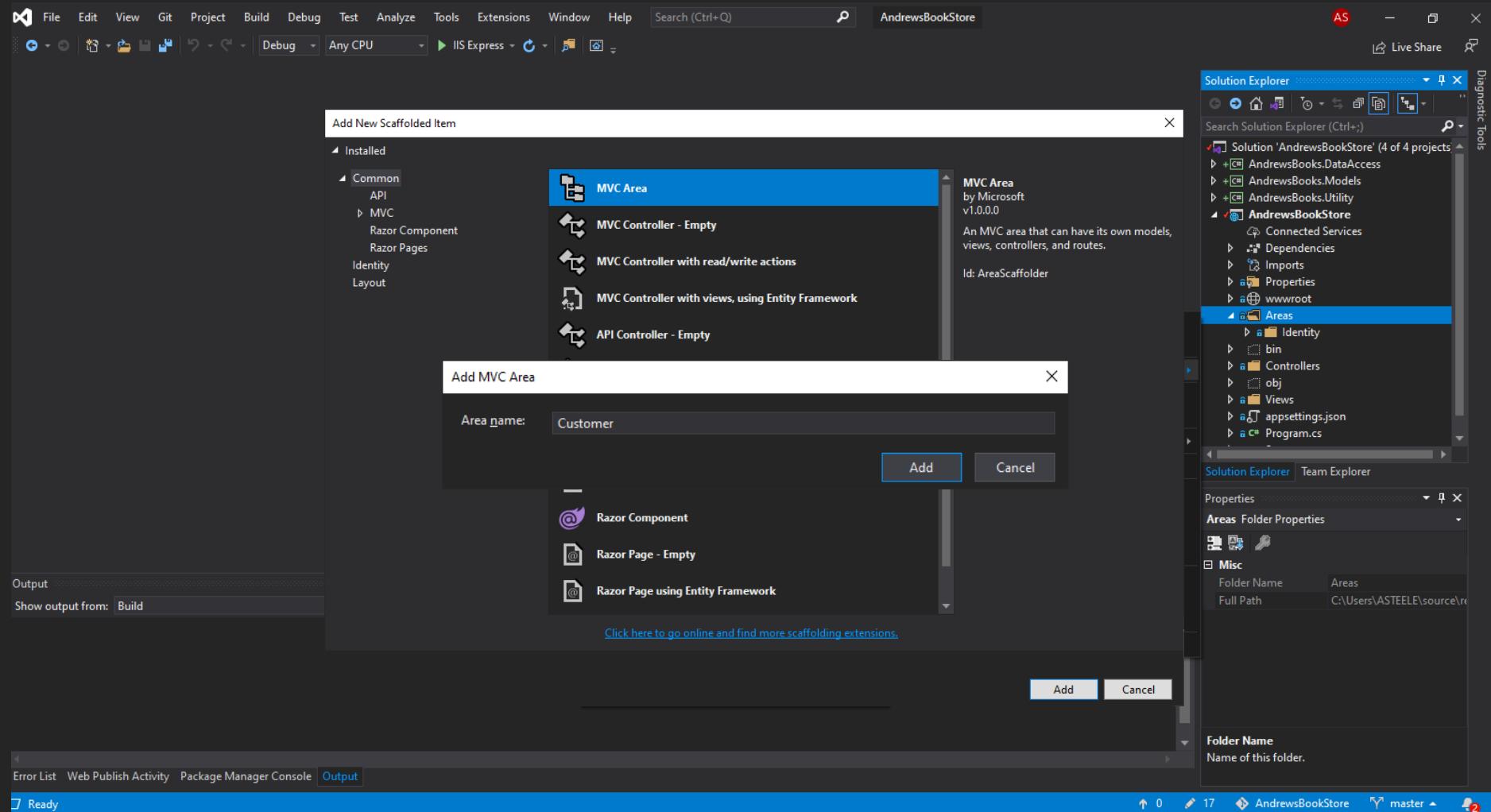
- Add a 'Customers' area to Areas



- Add a 'Customers' area to Areas



- Add a 'Customers' area to Areas



- Add a 'Customers' area to Areas

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows a solution named "AndrewsBookStore" containing four projects: "AndrewsBooks.DataAccess", "AndrewsBooks.Models", "AndrewsBooks.Utility", and "AndrewsBookStore". The "AndrewsBookStore" project is expanded, showing its structure: Connected Services, Dependencies, Imports, Properties, wwwroot, Areas, Customer, Identity, bin, Controllers, Views, and appsettings.json.
- Code Editor:** An open file is titled "ScaffoldingReadMe.txt" which contains instructions for adding an area. It includes code for configuring endpoints, specifically mapping a route for the "areas" controller.
- Output Window:** Displays log messages from the code generator:
 - Finding the generator 'area'...
 - Running the generator 'area'...
- Status Bar:** Shows "7 Item(s) Saved" and other status indicators.

- Add a 'Customers' area to Areas
- Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt

The screenshot shows a Visual Studio interface with the following details:

- Solution Explorer:** Shows the solution 'AndrewsBookStore' containing four projects: AndrewsBooks.DataAccess, AndrewsBooks.Models, AndrewsBooks.Utility, and AndrewsBookStore. The AndrewsBookStore project is expanded, showing its structure including Areas, Customer, Identity, bin, Controllers, Views, and appsettings.json.
- Code Editor:** The file 'ScaffoldingReadMe.txt' is open, displaying the following text and code:

```
1 Scaffolding has generated all the files and added the required dependencies.
2
3 However the Application's Startup code may require additional changes for things to work end to end.
4 Add the following code to the Configure method in your Application's Startup class if not already done:
5
6     app.UseEndpoints(endpoints =>
7     {
8         endpoints.MapControllerRoute(
9             name : "areas",
10            pattern : "{area:exists}/{controller=Home}/{action=Index}/{id?}")
11    );
12 });
13
```
- Output Window:** Shows the generator log:

```
Finding the generator 'area'...
Running the generator 'area'...
```
- Status Bar:** Shows the current branch is 'master'.

- Add a 'Customers' area to Areas
- Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt

The screenshot shows the Visual Studio IDE interface with the following details:

- Startup.cs:** The main code editor window displays the `Configure(IApplicationBuilder app, IWebHostEnvironment env)` method. The route configuration for the 'Customer' area is highlighted, showing the pattern `"{area=Customer}/{controller=Home}/{action=Index}/{id?}"`.
- Solution Explorer:** Shows the project structure for "AndrewsBookStore". It includes the `Program.cs` file, `Startup.cs`, and a newly created `Areas` folder containing `Customer` and `Identity` subfolders.
- Output Window:** The bottom-left window shows the generator output:


```
Finding the generator 'area'...
Running the generator 'area'...
```
- Status Bar:** The bottom right shows the status bar with "18" items saved, the branch name "master", and a red notification badge with the number "2".

- Add a 'Customers' area to Areas

- Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt

- Move the HomeController.cs to the Area > Customer > Controller folder and delete Data and Models.

The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Project Name:** AndrewsBookStore.
- Startup.cs Content:**

```
57
58     app.UseRouting();
59
60     app.UseAuthentication();
61     app.UseAuthorization();
62
63     app.UseEndpoints(endpoints =>
64         {
65             endpoints.MapControllerRoute(
66                 name: "default",
67                 pattern: "[area=Customer]/[controller=Home]/[action=Index]/[id?]");
68             endpoints.MapRazorPages();
69         });
70     });
71 }
72 }
73 }
```
- Solution Explorer:** Shows the project structure:
 - AndrewsBooks.Utility
 - AndrewsBookStore
 - Connected Services
 - Dependencies
 - Imports
 - Properties
 - wwwroot
 - Areas
 - Customer
 - Identity
 - bin
 - Controllers
 - obj
 - Views
 - appsettings.json
 - Program.cs
 - ScaffoldingReadMe.txt
 - Startup.cs
- Output Window:** Shows log messages:

```
Finding the generator 'area'...
Running the generator 'area'...
```
- Bottom Status Bar:** Item(s) Saved, 18, AndrewsBookStore, master.

- Add a 'Customers' area to Areas
- Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt
- Move the HomeController.cs to the Area > Customer > Controller folder and delete Data and Models.

The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q)
- Toolbars:** Standard toolbar with icons for Open, Save, Print, etc.
- Startup.cs and ScaffoldingReadMe.txt:** Tabs in the top left.
- Code Editor:** Shows the HomeController.cs code:

```

1 using AndrewsBookStore.Models;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.Extensions.Logging;
4 using System;
5 using System.Collections.Generic;
6 using System.Diagnostics;
7 using System.Linq;
8 using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Controllers
11 {
12     public class HomeController : Controller
13     {
14         private readonly ILogger<HomeController> _logger;
15
16         public HomeController(ILogger<HomeController> logger)
17         {
18
19         }
20     }
21 }
```
- Solution Explorer:** Shows the project structure:
 - AndrewsBookStore
 - AndrewsBookStore.Controllers
 - AndrewsBookStore.Controllers.HomeController
 - logger
 - HomeController.cs
 - Properties
 - wwwroot
 - Areas
 - Customer
 - Controllers
 - Views
 - Data
 - Models
 - Identity
 - bin
 - obj
 - Views
 - appsettings.json
 - Program.cs
 - ScaffoldingReadMe.txt
- Properties Panel:** Shows the file properties for HomeController.cs:
 - Build Action: C# compiler
 - Copy to Output Direct: Do not copy
 - Custom Tool
 - Custom Tool Namespace
 - Misc:
 - File Name: HomeController.cs
 - Full Path: C:\Users\ASTEEL\source\repos\AndrewsBookStore\AndrewsBookStore\Controllers\HomeController.cs
 - Build Action: How the file relates to the build and deployment processes.
- Output Window:** Shows log output from the generator:

```

Finding the generator 'area'...
Running the generator 'area'...

```
- Bottom Bar:** Error List, Web Publish Activity, Package Manager Console, Output, and a status bar showing 7 Ready.

- Add a 'Customers' area to Areas
 - Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt
 - Move the HomeController.cs to the Area > Customer > Controller folder and delete Data and Models.

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a Search bar. The title bar displays "AndrewsBookStore".

The code editor window contains the following C# code for `HomeController.cs`:

```
1 using AndrewsBookStore.Models;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.Extensions.Logging;
4 using System;
5 using System.Collections.Generic;
6 using System.Diagnostics;
7 using System.Linq;
8 using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Controllers
11 {
12     public class HomeController : Controller
13     {
14         private readonly ILogger<HomeController> _logger;
15
16         public HomeController(ILogger<HomeController> logger)
17         {
18             _logger = logger;
19         }
20     }
21 }
```

The Solution Explorer window on the right shows the project structure:

- Connected Services
- Dependencies
- Imports
- Properties
- wwwroot
- Areas
 - Customer
 - Controllers
 - HomeController.cs
 - Views
- bin
- Controllers
- obj
- Views
- appsettings.json
- Program.cs
- ScaffoldingReadMe.txt
- Startup.cs

The `HomeController.cs` file is selected in the Solution Explorer.

The Properties window shows the following settings for `HomeController.cs`:

- Build Action: C# compiler
- Copy to Output Directory: Do not copy
- Custom Tool
- Custom Tool Namespace
- Misc
 - File Name: HomeController.cs
 - Full Path: C:\Users\ASTEELE\source\repos\AndrewsBookStore\AndrewsBookStore\Controllers\HomeController.cs

The Output window at the bottom shows the following messages:

- Show output from: Code Generation Log
- Finding the generator 'area'...
- Running the generator 'area'...

The status bar at the bottom indicates "Ready".

- Add a 'Customers' area to Areas
 - Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt
 - Move the HomeController.cs to the Area > Customer > Controller folder and delete Data and Models.
 - Edit the HomeController.cs to explicitly define that the controller is in the Customer Area

- Add a 'Customers' area to Areas
- Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt
- Move the HomeController.cs to the Area > Customer > Controller folder and delete Data and Models.
- Edit the HomeController.cs to explicitly define that the controller is in the Customer Area

```

1  using AndrewsBookStore.Models;
2  using Microsoft.AspNetCore.Mvc;
3  using Microsoft.Extensions.Logging;
4  using System;
5  using System.Collections.Generic;
6  using System.Diagnostics;
7  using System.Linq;
8  using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Controllers
11 {
12     [Area("Customer")]
13     public class HomeController : Controller
14     {
15         private readonly ILogger<HomeController> _logger;
16
17         public HomeController(ILogger<HomeController> logger)
18         {
19             _logger = logger;
20         }
21
22         public IActionResult Index()
23         {
24             return View();
25         }
26
27         public IActionResult Privacy()
28         {
29             return View();
30         }
31
32         [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, Skip��lter = true)]
33         public IActionResult Error()
34         {
35             return View();
36         }
37     }
38 }

```

No issues found

Output

Show output from: Code Generation Log

Finding the generator 'area'...

Running the generator 'area'...

Ready

- Add a 'Customers' area to Areas
- Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt
- Move the HomeController.cs to the Area > Customer > Controller folder and delete Data and Models.
- Edit the HomeController.cs to explicitly define that the controller is in the Customer Area
- Move Views > Home and modify the HomeController namespace

The screenshot shows the Visual Studio IDE interface with the following details:

- Solution Explorer:** Shows the project structure for "AndrewsBookStore". An "Areas" folder under the root contains a "Customer" folder, which in turn contains a "Controllers" folder with an "HomeController.cs" file.
- HomeController.cs:** The code editor displays the following C# code:

```

1  using AndrewsBookStore.Models;
2  using Microsoft.AspNetCore.Mvc;
3  using Microsoft.Extensions.Logging;
4  using System;
5  using System.Collections.Generic;
6  using System.Diagnostics;
7  using System.Linq;
8  using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Controllers
11 {
12     [Area("Customer")]
13     public class HomeController : Controller
14     {
15         private readonly ILogger<HomeController> _logger;
16
17         public HomeController(ILogger<HomeController> logger)
18         {
19             _logger = logger;
20         }
21
22         public IActionResult Index()
23         {
24             return View();
25         }
26
27         public IActionResult Privacy()
28         {
29             return View();
30         }
31
32         [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, SkipStatusCodeCheck = true)]
33         public IActionResult Error()
34         {
35             return View();
36         }
37     }
38 }
```
- Output Window:** Shows the output of the code generation process:

```

Finding the generator 'area'...
Running the generator 'area'...

```
- Status Bar:** Displays "Ln: 12 Ch: 5 SPC: CRLF".

- Add a 'Customers' area to Areas
- Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt
- Move the HomeController.cs to the Area > Customer > Controller folder and delete Data and Models.
- Edit the HomeController.cs to explicitly define that the controller is in the Customer Area
- Move Views > Home and modify the HomeController namespace

```

1  using AndrewsBookStore.Models;
2  using Microsoft.AspNetCore.Mvc;
3  using Microsoft.Extensions.Logging;
4  using System;
5  using System.Collections.Generic;
6  using System.Diagnostics;
7  using System.Linq;
8  using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Area.Customer.Controllers
11 {
12     [Area("Customer")]
13     public class HomeController : Controller
14     {
15         private readonly ILogger<HomeController> _logger;
16
17         public HomeController(ILogger<HomeController> logger)
18         {
19             _logger = logger;
20         }
21
22         public IActionResult Index()
23         {
24             return View();
25         }
26
27         public IActionResult Privacy()
28         {
29             return View();
30         }
31
32         [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, Skip��lter = true)]
33         public IActionResult Error()
34         {
35             return View();
36         }
37     }
38 }

```

The Solution Explorer shows the project structure:

- Connected Services
- Dependencies
- Imports
- Properties
- wwwroot
- Areas
 - Customer
 - Controllers
 - Views
- Identity
- bin
- Controllers
- obj
- Views
 - Shared
 - _ViewImports.cshtml
 - _ViewStart.cshtml

The Properties tab shows the following settings for the 'Home' folder:

Folder Name	Home
Full Path	C:\Users\ASTEEL\source\repos\AndrewsBookStore\Areas\Customer\Views\Home

The Output window shows the generator log:

```

Finding the generator 'area'...
Running the generator 'area'...

```

- Add a 'Customers' area to Areas
- Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt
- Move the HomeController.cs to the Area > Customer > Controller folder and delete Data and Models.
- Edit the HomeController.cs to explicitly define that the controller is in the Customer Area
- Move Views > Home and modify the HomeController namespace
- Run the application...why?

```

1  using AndrewsBookStore.Models;
2  using Microsoft.AspNetCore.Mvc;
3  using Microsoft.Extensions.Logging;
4  using System;
5  using System.Collections.Generic;
6  using System.Diagnostics;
7  using System.Linq;
8  using System.Threading.Tasks;
9
10 namespace AndrewsBookStore.Area.Customer.Controllers
11 {
12     [Area("Customer")]
13     public class HomeController : Controller
14     {
15         private readonly ILogger<HomeController> _logger;
16
17         public HomeController(ILogger<HomeController> logger)
18         {
19             _logger = logger;
20         }
21
22         public IActionResult Index()
23         {
24             return View();
25         }
26
27         public IActionResult Privacy()
28         {
29             return View();
30         }
31
32         [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, Skip��lter = true)]
33         public IActionResult Error()
34         {
35             return View();
36         }
37     }
38 }

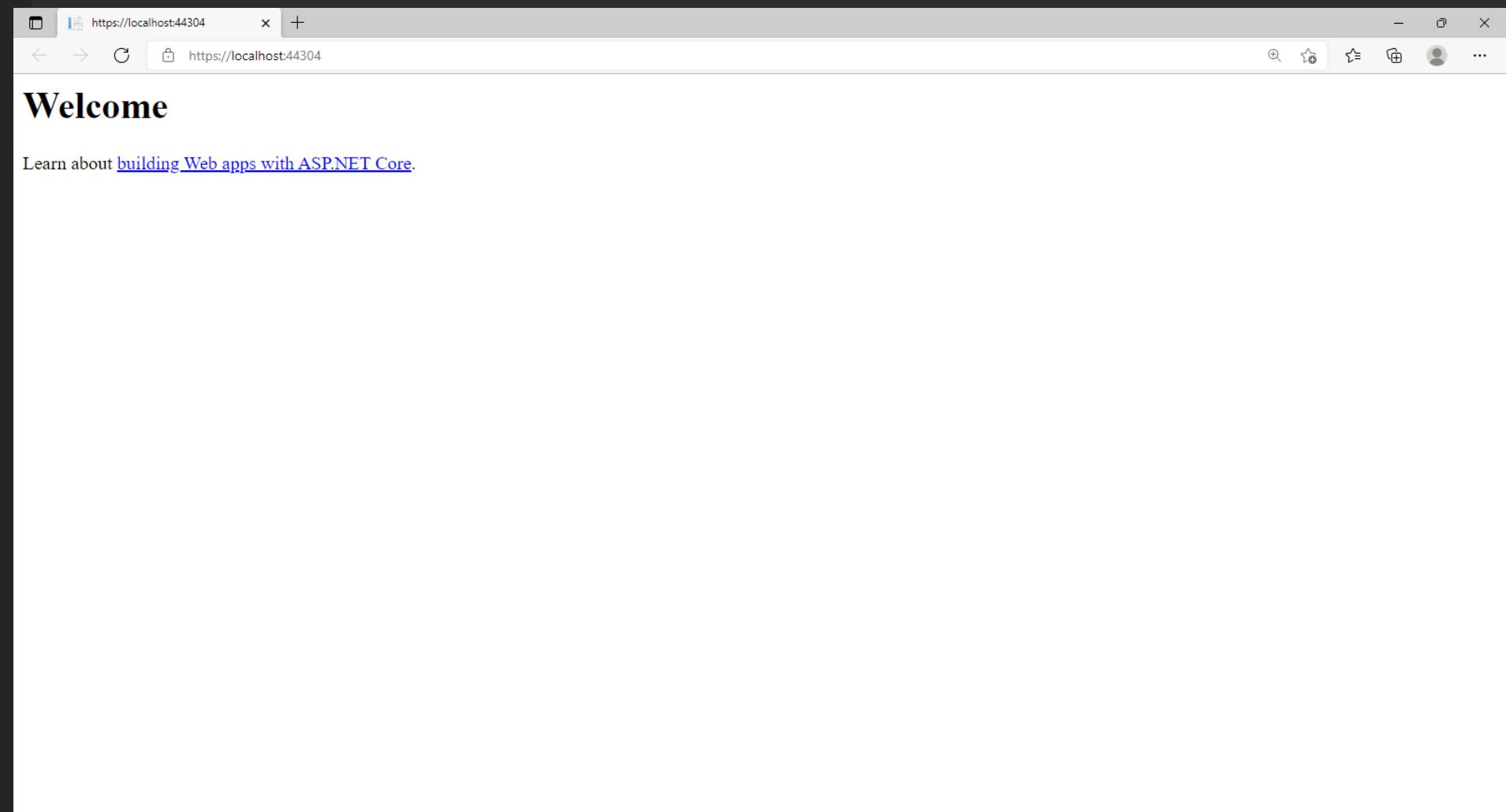
```

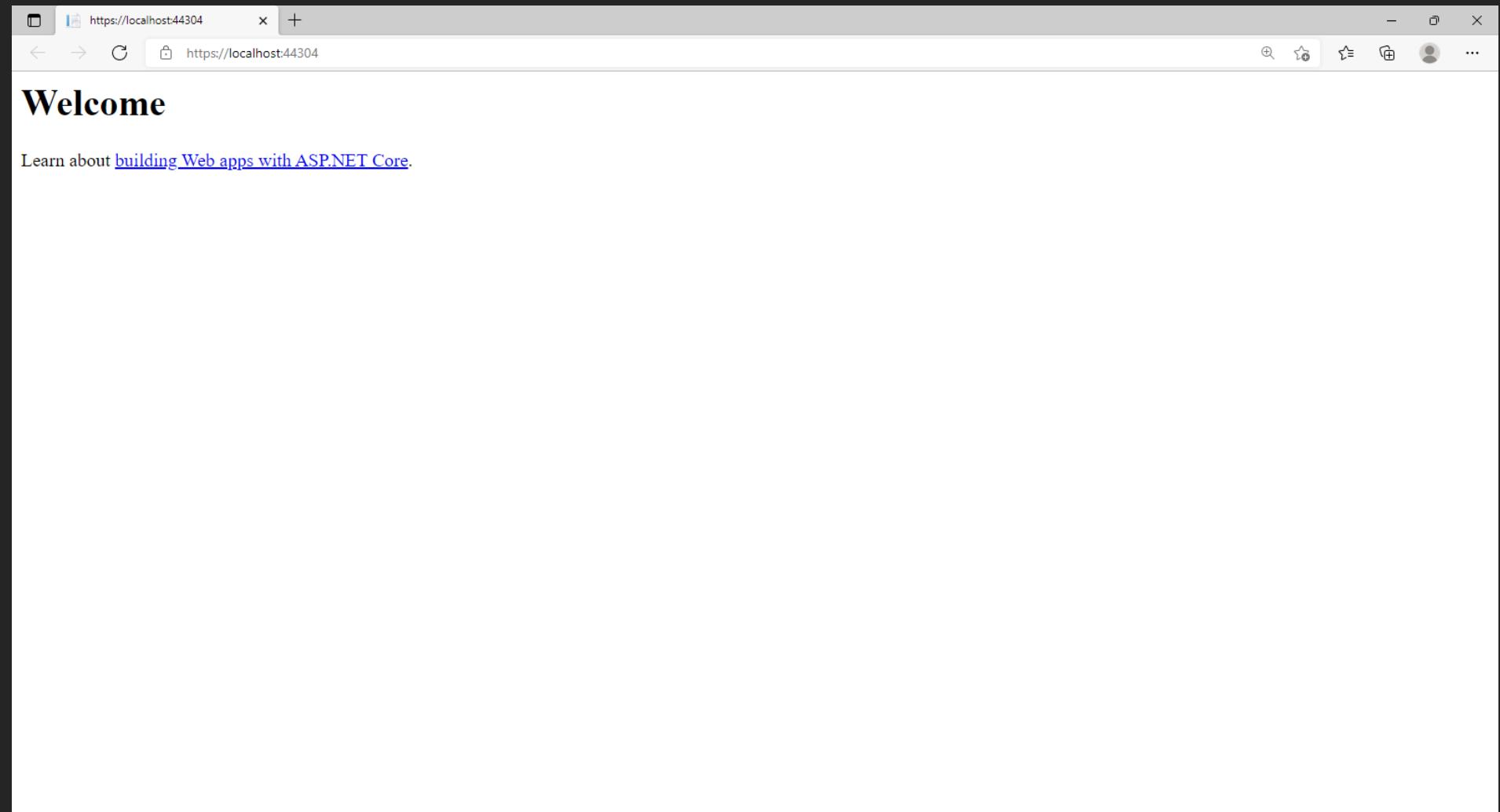
No issues found

Finding the generator 'area'...
Running the generator 'area'...

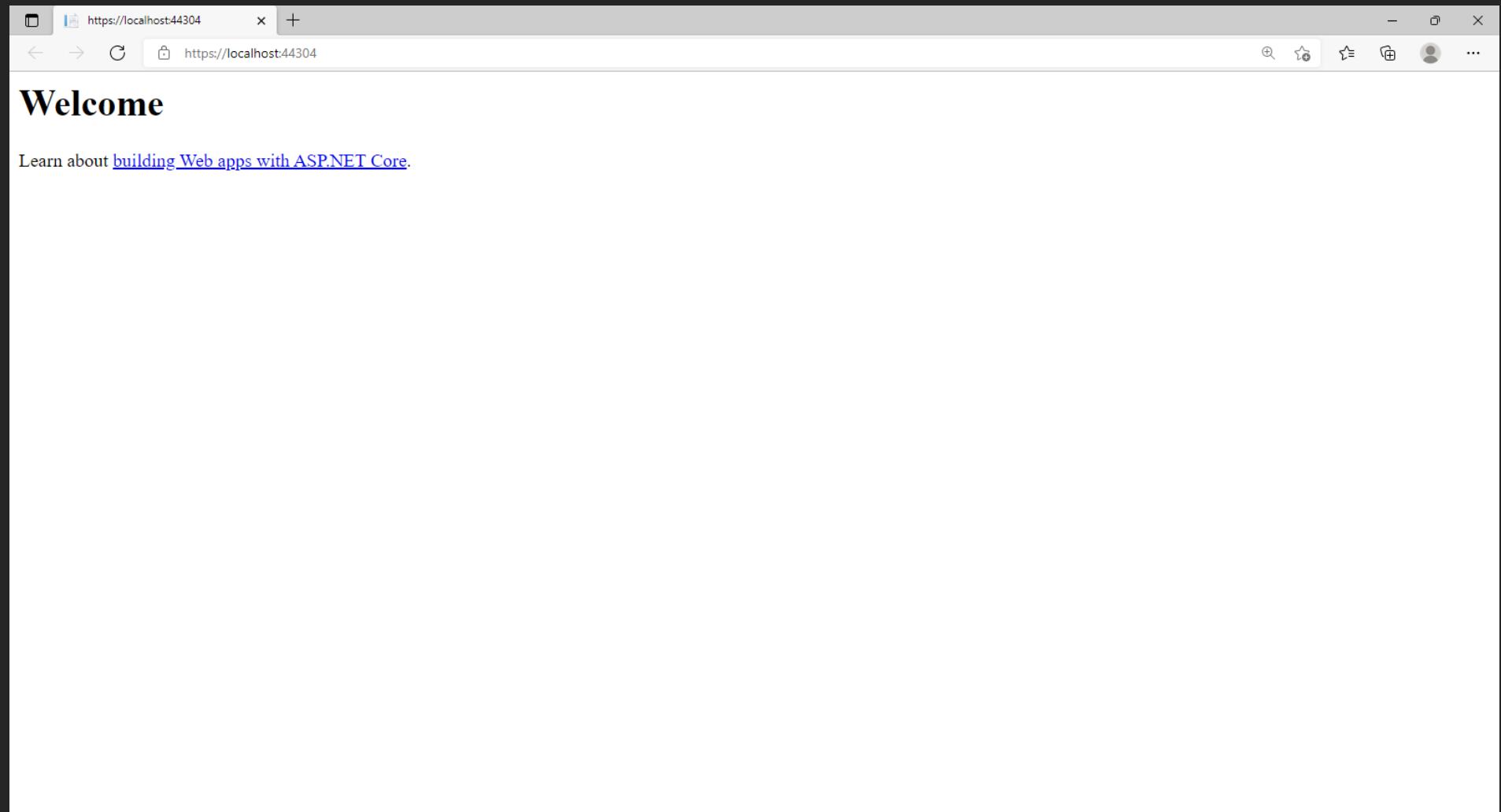
Ready

- Add a 'Customers' area to Areas
- Change the routes in Startup.cs like the one outlined in the ScaffoldingReadMe.txt
- Move the HomeController.cs to the Area > Customer > Controller folder and delete Data and Models.
- Edit the HomeController.cs to explicitly define that the controller is in the Customer Area
- Move Views > Home and modify the HomeController namespace
- Run the application...why?





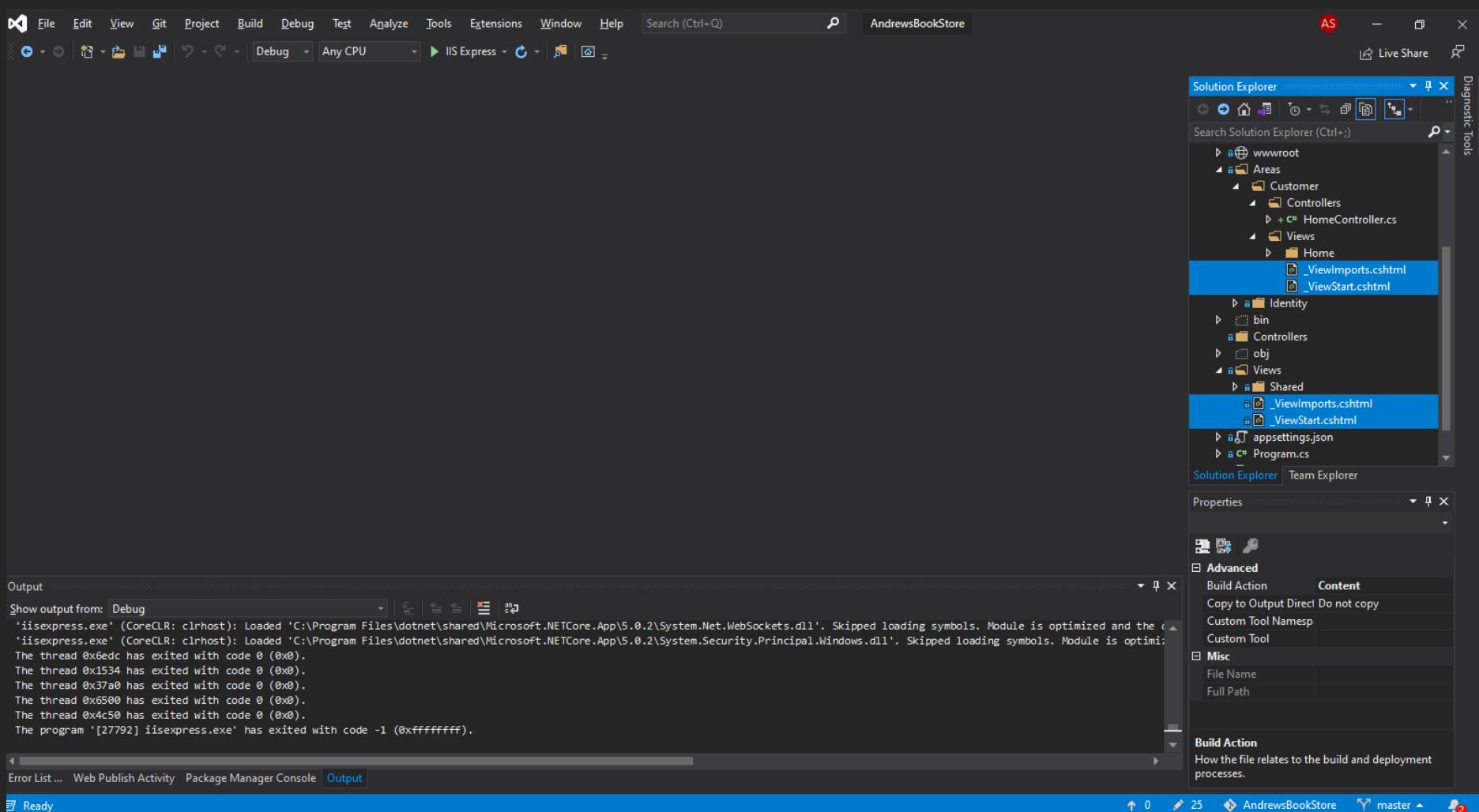
- Views are now in Areas, but master page is defined in _ViewStart



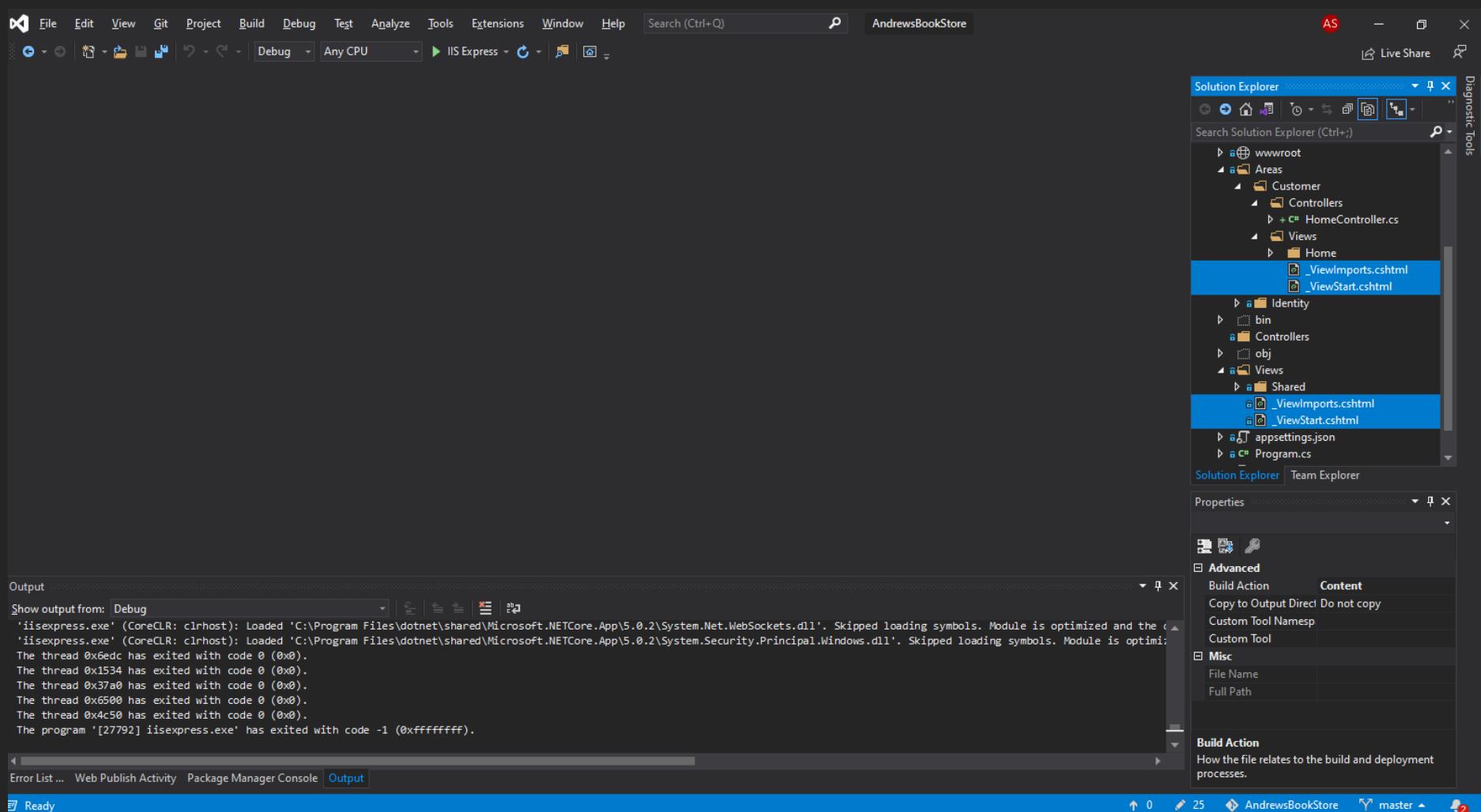
- Views are now in Areas, but master page is defined in _ViewStart

- Views are now in Areas, but master page is defined in _ViewStart
- Copy _ViewImport and _ViewStart to Customer Area

- Views are now in Areas, but master page is defined in _ViewStart
- Copy _ViewImports and _ViewStart to Customer Area



- Views are now in Areas, but master page is defined in _ViewStart
- Copy _ViewImport and _ViewStart to Customer Area
- Modify the _ViewStart.cshtml to reflect the new path



- Views are now in Areas, but master page is defined in _ViewStart
- Copy _ViewImport and _ViewStart to Customer Area
- Modify the _ViewStart.cshtml to reflect the new path

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** The main window displays the file `_ViewStart.cshtml` containing the following code:

```

1  @{
2     Layout = "~/Views/Shared/_Layout.cshtml";
3 }
4

```
- Solution Explorer:** Shows the project structure for "AndrewsBookStore". The `Customer` area contains a `Views` folder which includes `_ViewImports.cshtml` and `_ViewStart.cshtml`.
- Output Window:** The bottom window shows the output of the application's run. It includes messages from 'iisexpress.exe' about loading shared DLLs and exiting with code -1.

```

'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.2\System.WebSockets.dll'. Skipped loading symbols. Module is optimized and the code is不做脱机调试.
'iisexpress.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\5.0.2\System.Security.Principal.Windows.dll'. Skipped loading symbols. Module is optimized and the code is不做脱机调试.
The thread 0x6edc has exited with code 0 (0x0).
The thread 0x1534 has exited with code 0 (0x0).
The thread 0x37a0 has exited with code 0 (0x0).
The thread 0x6500 has exited with code 0 (0x0).
The thread 0x4c50 has exited with code 0 (0x0).
The program '[27792] iisexpress.exe' has exited with code -1 (0xffffffff).

```

- Views are now in Areas, but master page is defined in _ViewStart
- Copy _ViewImport and _ViewStart to Customer Area
- Modify the _ViewStart.cshtml to reflect the new path
- Run the application now

The screenshot shows the Microsoft Visual Studio interface with the following details:

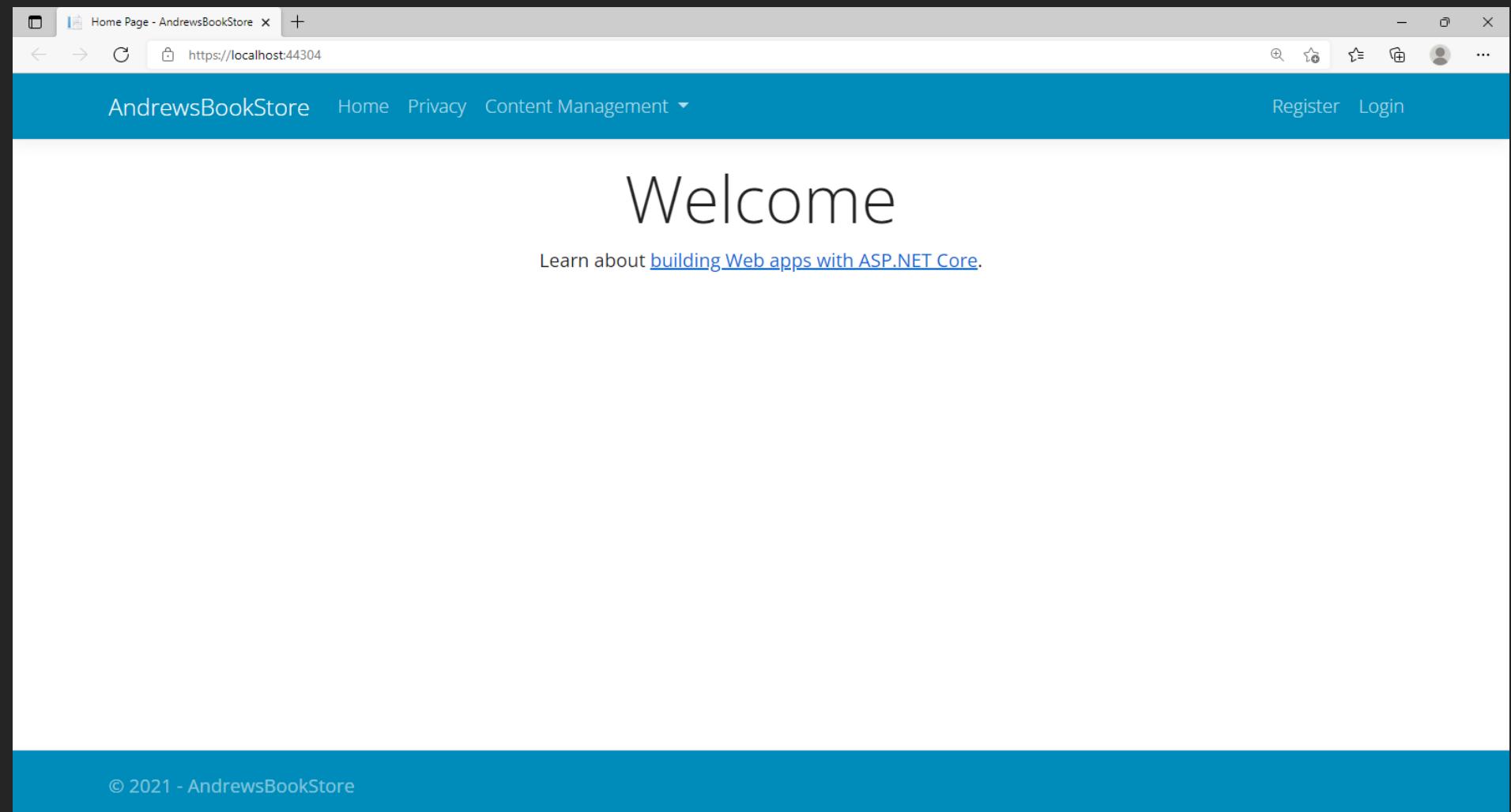
- Code Editor:** The main window displays the file `_ViewStart.cshtml` containing the following code:

```

1  @{
2     Layout = "~/Views/Shared/_Layout.cshtml";
3 }
4

```
- Solution Explorer:** On the right, the Solution Explorer shows the project structure for "AndrewsBookStore". It includes a Customer area with a HomeController.cs and a Views folder containing _ViewImports.cshtml and _ViewStart.cshtml.
- Output Window:** The bottom window shows the output of the application run. It includes messages from iisexpress.exe indicating module loading and thread exits, followed by the message: "The program '[27792] iisexpress.exe' has exited with code -1 (0xffffffff)".

- Views are now in Areas, but master page is defined in _ViewStart
- Copy _ViewImport and _ViewStart to Customer Area
- Modify the _ViewStart.cshtml to reflect the new path
- Run the application now

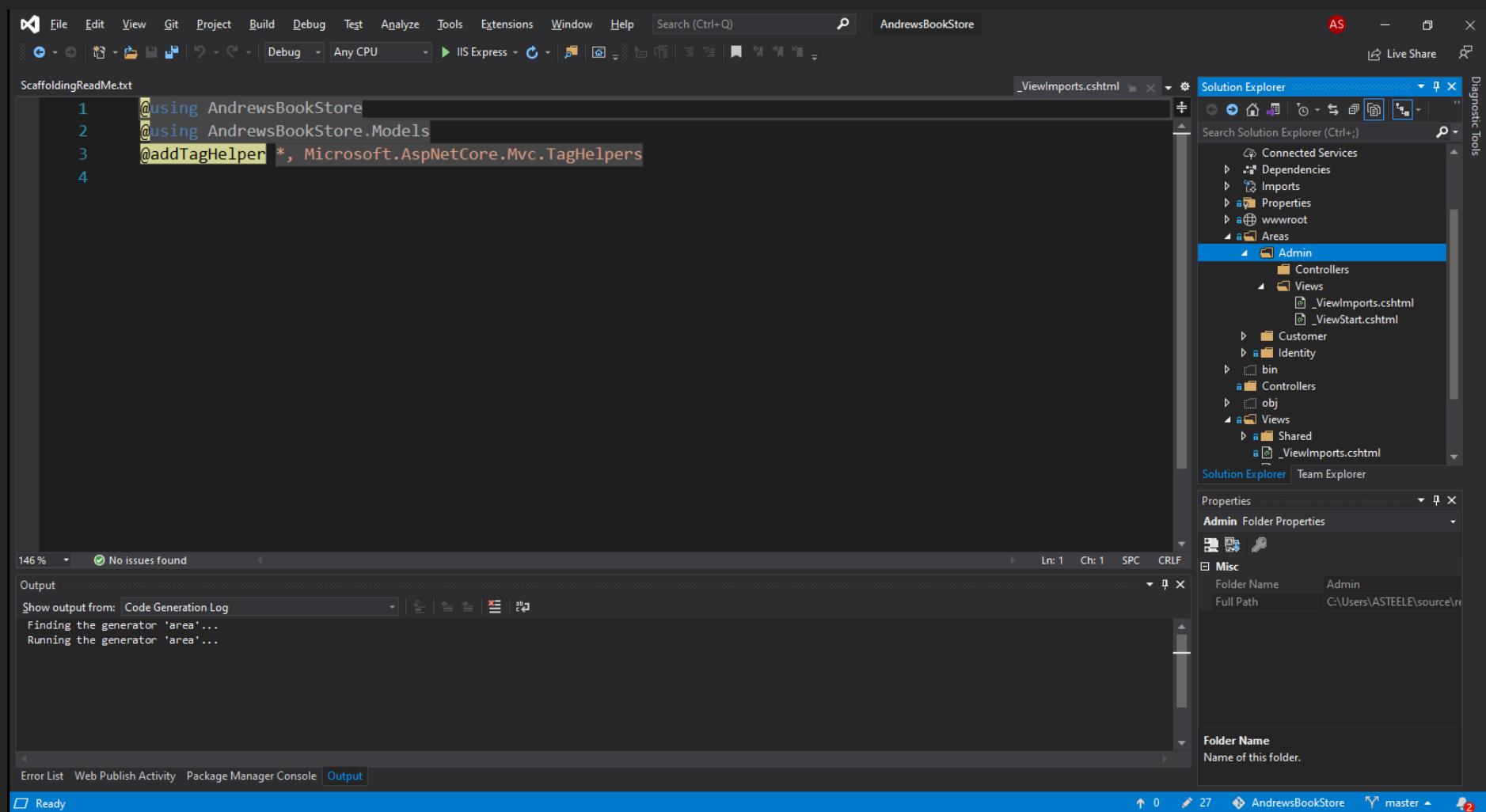


The screenshot shows the Visual Studio 2019 IDE interface with the following details:

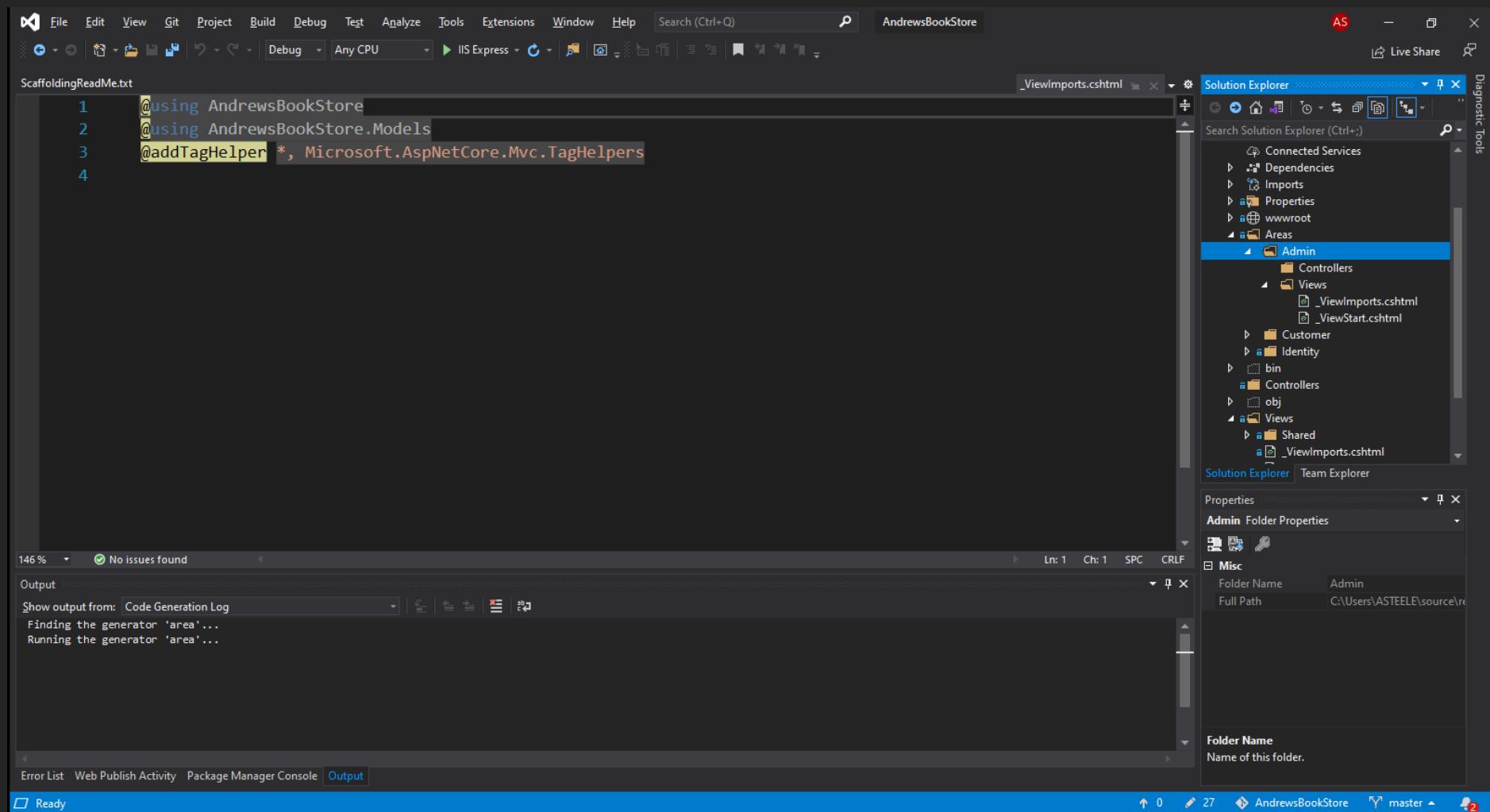
- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q) and a magnifying glass icon.
- Project Name:** AndrewsBookStore
- Toolbars:** Standard toolbar with icons for New, Open, Save, Print, etc.
- Solution Explorer:** Shows the project structure:
 - Connected Services
 - Dependencies
 - Imports
 - Properties
 - wwwroot
 - Areas
 - Admin
 - Controllers
 - Views
 - _ViewImports.cshtml
 - _ViewStart.cshtml
 - Customer
 - Identity
 - bin
 - obj
 - Views
 - Shared
 - _ViewImports.cshtml
- Properties Explorer:** Shows properties for the Admin folder.
- Output Window:** Displays log messages:

```
Finding the generator 'area'...
Running the generator 'area'...
```
- Status Bar:** Shows the status "Ready" and other system information.

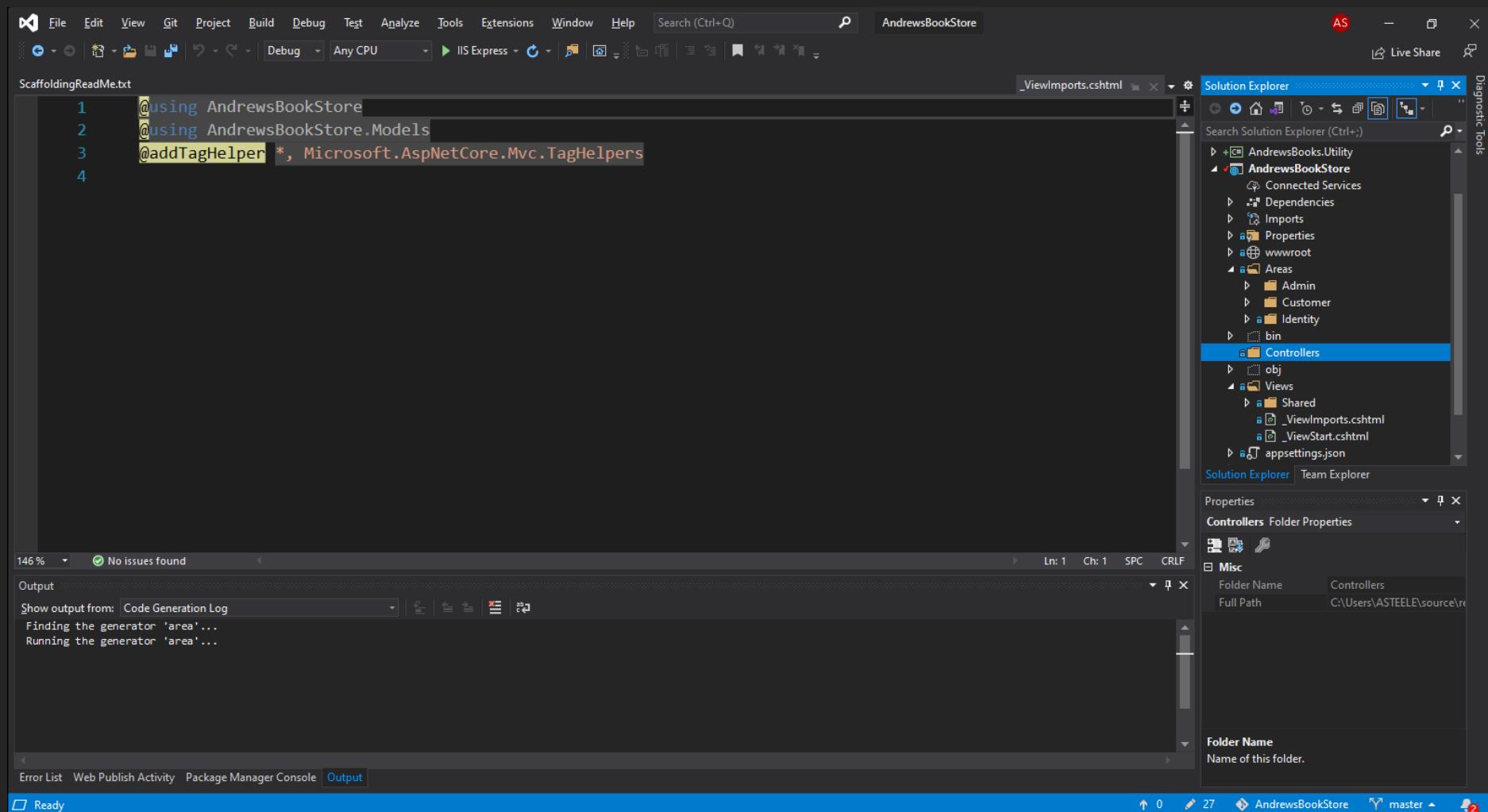
- Add a new Admin area in Areas



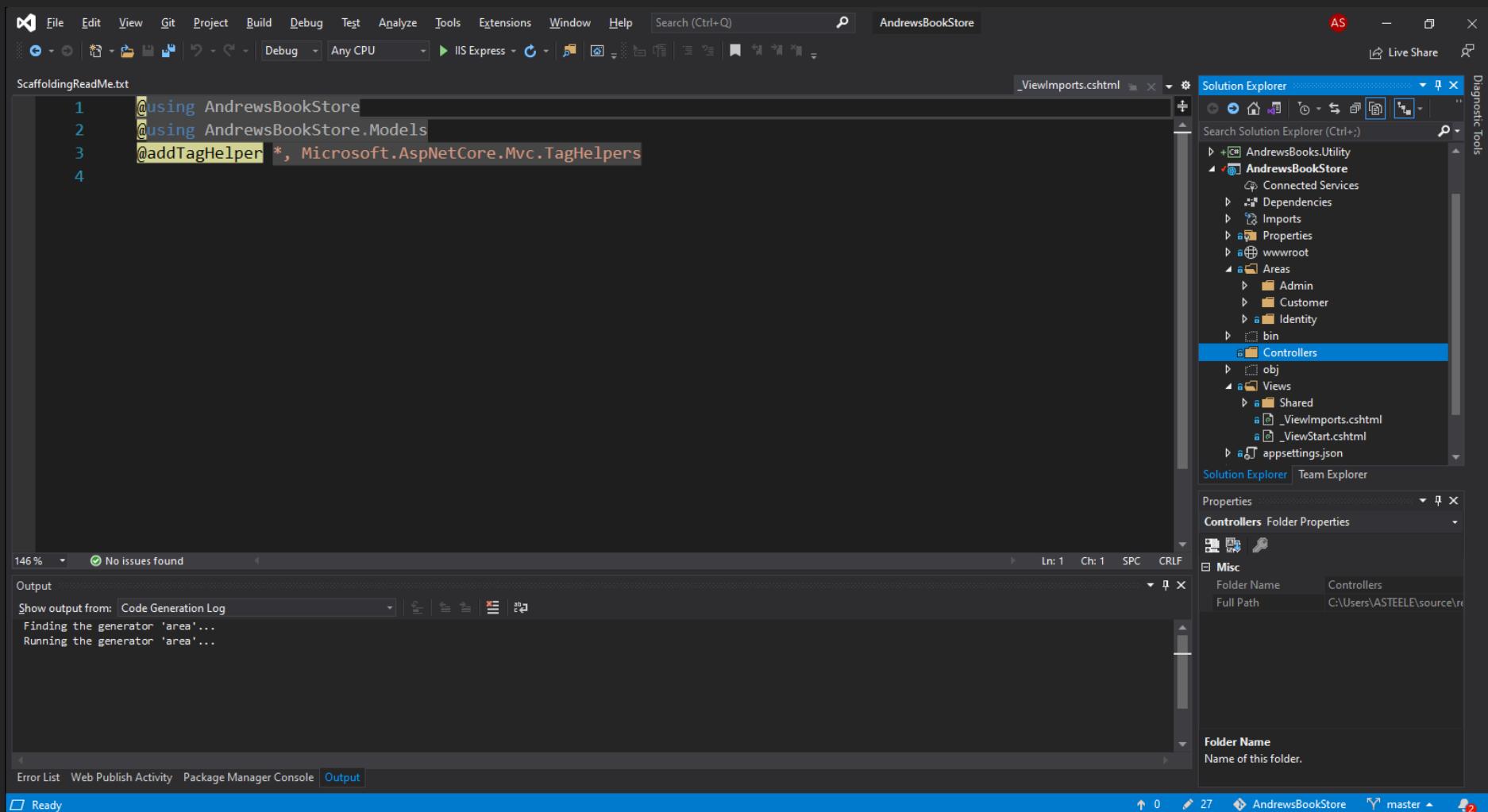
- Add a new Admin area in Areas
- Add the proper view files and delete the Data and Models folder



- Add a new Admin area in Areas
- Add the proper view files and delete the Data and Models folder



- Add a new Admin area in Areas
- Add the proper view files and delete the Data and Models folder
- Delete the Controllers folder



- Add a new Admin area in Areas
- Add the proper view files and delete the Data and Models folder
- Delete the Controllers folder
- Update the GitHub repo

The screenshot shows the Visual Studio IDE interface with the following details:

- Project Title:** AndrewsBookStore
- File Menu:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q)
- Solution Explorer:** Shows the project structure:
 - AndrewsBooks.Utility
 - AndrewsBookStore
 - Connected Services
 - Dependencies
 - Imports
 - Properties
 - wwwroot
 - Areas
 - Admin
 - Customer
 - Identity
 - bin
 - Controllers (selected)
 - obj
 - Views
 - Shared
 - _ViewImports.cshtml
 - _ViewStart.cshtml
 - appsettings.json
- Properties:** Shows Controller properties.
- Output Window:** Shows the message "No issues found". Below it, under "Output", it says "Show output from: Code Generation Log" and displays the log messages:


```
Finding the generator 'area'...
Running the generator 'area'...
```
- Bottom Status Bar:** Shows "Ready", "0", "27", "AndrewsBookStore", "master", and a refresh icon.

- Add a new Admin area in Areas
- Add the proper view files and delete the Data and Models folder
- Delete the Controllers folder
- Update the GitHub repo

