Building Your First .NET Core MVC Application

One of the most popular methods for creating web application these days is Microsoft's .NET Core and the Model-View-Controller (MVC) design pattern. Using the MVC pattern makes your code reusable, testable and flexible. In this course you use .NET Core, MVC and Razor markup to build a web application. This course assumes you have little to no experience using MVC and .NET Core and want to learn to build web applications using this technology. This course walks you step-by-step through building a business application using the MVC Razor markup in .NET Core.

Throughout this course you learn the basics of the MVC design pattern and navigating from page to page. Tag helpers are an easy way to add great functionality to your web pages. Passing data from one page to the next, or from a controller to a view can be accomplished using a few different mechanisms and each is explored. Exception handling and logging are essential in today's applications and MVC has an easy way to deal with both.

After the basics of MVC have been explored, you start to build a list of product data and learn how to build forms for user input. Data validation can be accomplished using Data Annotations, and you learn how to display validation messages to the user. Finally, you build a full CRUD page so a user can retrieve, insert, update, and delete data in a SQL Server table.

Learning Objectives

Getting started with .NET Core MVC and Razor markup

Learn to manage state, catch exceptions, and perform logging

Work with tag helpers and the HTML helper class

Displaying, validating, and modifying data in a database

Student Prerequisites

No prior knowledge of MVC is required, however we assume you understand the basics web programming such as HTML and CSS. You should possess a working knowledge of C# and the .NET Framework.

Hardware Requirements

A computer must be supplied by the student with the following technologies installed.

- Visual Studio 2022 or later
- or Visual Studio Code V1.9 or later
- .NET Core
- SQL Server (Developer Edition or higher) or SQL Server Express
- AdventureWorksLT database
    - Available at https://github.com/PaulDSheriff/AdventureWorksLT

Course Level

Introduction

Course Length

8-16 hours

Module 1: Basics of MVC

Create a new .NET Core MVC project

Overall program flow

Folders and files

Using @Html.ActionLink helper

Using @Url.Action helper

Anchor Tag Helpers

Partial pages

CSS style sheets

Adding page head styles

Adding JavaScript

Retrieving configuration settings from appsettings.json

Modify the overall layout of your website

Module 2: Passing Data

Using ViewData

Using a ViewBag

Using TempData

Passing data as a query string

Module 3: State Management

Working with the Session object

Create your own session class wrapper

Send and retrieve Cookies

Working with the Cache object

Create your own state class and use dependency injection

Module 4: Exception Handling

Built-in global exception handling

Add a custom development page for retrieving exception information

Handle 404 status codes

Handle status code errors with redirect

Handle status code errors with re-execute

Module 5: Logging

Built-in logging

Logging levels

Configuring logging levels

Logging a product object

Using replaceable parameters in Log() methods

Logging to file

Module 6: Displaying Lists of Data

Add a data layer class library

Add a view model class library

Display an unordered list

Display an ordered list

Build a <select> list

Use the @Html.DropDownList HTML helper

Use the Select Tag Helper

Module 7 Working with HTML Forms

Create a product input form using @Html Helpers

Create a product input form using Input Tag Helpers

Add data annotations for displaying labels

Cancel out of a form

Work with related data in a <select> list

Working with radio buttons

Working with check boxes

Module 8: Validating Data

Add data annotations for validation (required, string length, datatype, range, etc.)

Bind/display validation messages using asp-validation-for

Use the validation summary tag helper

Create your own server-side custom validation attribute

Module 9: Build a CRUD Page

Using the MVVM design pattern

Select and display a product

Add and edit a product

Delete a product

Module 10: Working with HTML Tables

Build a table of product data

Searching product data

Searching using the MVVM design pattern

Sorting columns in a table

Module 11: Paging Tables

Creating hard-coded pager

Create a data-driven pager

Paging product data