Course Outline

ASP.NET Core MVC, Rev. 6.0

Course Summary

Description

This course is for anyone who knows the basics of HTML/CSS and C# and wants to learn how to use the ASP.NET Core platform to develop MVC web apps. That's true whether you're new to web development or whether you have experience with another web development platform.

Objectives

At the end of this course, students will have the skills to:

- Develop database-driven web apps
- Use endpoint routing
- Use Razor views
- Work with model binding, data validation, EF (Entity Framework) Core for database handling and dependency injection
- Use Bootstrap for responsive design
- Work with xUnit and Moq for unit testing
- Use Identity for authentication

Topics

- An introduction to web programming with ASP.NET Core MVC
- How to develop a single-page MVC web app
- How to make a web app responsive with Bootstrap
- How to develop a data-driven MVC web app
- How to manually test and debug an ASP.NET Core web app
- How to work with controllers and routing
- How to work with Razor views
- How to transfer data from controllers
- How to work with session state and cookies
- How to work with model binding
- How to validate data
- How to use EF Core
- The Bookstore website
- How to use dependency injection and unit testing
- How to work with tag helpers, partial views, and view components
- How to authenticate and authorize users
- How to deploy ASP.NET Core MVC apps
- How to use Visual Studio Code

Audience

This course is for developers that are familiar with .NET and C#.

Prerequisites

C# and HTML/CSS

Duration

Five days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

ASP.NET Core MVC, Rev. 6.0

Course Outline

I. An introduction to web programming and ASP.NET Core MVC

A. An introduction to web apps
B. The components of a web app
C. How static web pages are processed
D. How dynamic web pages are processed
E. An introduction to the MVC pattern
F. An introduction to ASP.NET Core MVC
G. Four ASP.NET programming models for web apps
H. Some web components of .NET Framework and .NET Core
I. An introduction to ASP.NET Core middleware
J. How state works in a web app
K. Tools for working with ASP.NET Core MVC apps
L. An introduction to Visual Studio
M. An introduction to Visual Studio Code
N. How an ASP.NET Core MVC app works
O. How coding by convention works
P. How a controller passes a model to a view
Q. How a view uses Razor code, tag helpers, and Bootstrap CSS classes
R. How the Program.cs file configures the middleware for an app

II. How to develop a one-page MVC web app

A. How to create a Core MVC web app
B. How to start a new web app
C. How to configure a web app
D. How to set up the MVC folders
E. How to add a controller
F. How to add a Razor view
G. How to configure the middleware for an HTTP request pipeline
H. How to run a web app and fix errors
I. How to run a web app
J. How to find and fix errors
K. How to work with a model
L. How to add a model
M. How to add a Razor view imports page
N. How to code a strongly-typed view
O. How to handle GET and POST requests
P. The Future Value app after handling GET and POST requests
Q. How to organize the files for a view
R. How to add a CSS style sheet
S. How to add a Razor layout and view start
T. The code for a Razor layout and view start
U. How to add a Razor view when using a layout with a view start
V. How to validate user input
W. How to set data validation rules in the model
X. The model class with data validation
Y. How to check the data validation
Z. How to display validation error messages
AA. The Future Value app after validating data

III. How to make a web app responsive with Bootstrap

A. An introduction to responsive web design
B. A responsive user interface
C. How to add client-side libraries such as Bootstrap and jQuery
D. How to manage client-side libraries with LibMan
E. How to enable client-side libraries
F. How to get started with Bootstrap
G. The classes of the Bootstrap grid system
H. How the Bootstrap grid system works
I. How to work with forms
J. How to work with buttons and images
K. How to work with margins and padding
L. The code for the view of the Future Value app
M. More skills for Bootstrap CSS classes
N. How to format HTML tables
O. How to align and capitalize text
P. How to provide context
Q. More skills for Bootstrap components
R. How to work with button groups
S. How to work with icons and badges
T. How to work with button dropdowns
U. How to work with list groups
V. How to work with alerts and breadcrumbs
W. How to work with navigation bars
X. How to create navs
Y. How to create navbars
Z. How to position navbars

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Course Summary (cont'd)

IV. How to develop a data-driven MVC web app

A. An introduction to the Movie List app
B. The pages of the app
C. The folders and files of the app
D. How to use EF Core
E. How to add EF Core to your project
F. How to create a DbContext class
G. How to seed initial data
H. How to add a connection string
I.
J. How to enable dependency injection
K. How to use migrations to create the database
L. How to work with data
M. How to select data
N. How to insert, update, and delete data
O. How to view the generated SQL statements
P. The Movie List app
Q. The Home controller
R. The Home/Index view
S. The Movie controller
T. The Movie/Edit view
U. The Movie/Delete view
V. How to work with related data
W. How to relate one entity to another
X. How to update the DbContext class and the seed data
Y. How to use migrations to update the database
Z. How to select related data and display it on the Movie List page
AA. How to display related data on the Add and Edit Movie pages
BB. How to make user-friendly URLs
CC. How to make URLs lowercase with a trailing slash
DD. How to add a slug

V. How to manually test and debug an ASP.NET Core web app

A. How to test an ASP.NET Core web app
B. How to run a web app
C. How to use the Hot Reload feature
D. How to use the browser's developer tools
E. How to use the Internal Server Error page
F. How to use the Exception Helper
G. How to use the debugger
H. How to use breakpoints
I. How to work in break mode
J. How to monitor variables and expressions
K. How to use tracepoints

VI. How to work with controllers and routing

A. How to use the default route
B. How to configure the default route
C. How the default route works
D. How to code a simple controller and its actions
E. How to code a controller that uses the id segment
F. How to create custom routes
G. How to include static content in a route
H. How to work with multiple routing patterns
I. How to use attribute routing
J. How to change the routing for an action
K. More skills for changing the routing for an action
L. How to change the routing for a controller
M. Best practices for creating URLs
N. How to work with areas
O. How to set up areas
P. How to associate controllers with areas

VII. How to work with Razor views

A. How to use Razor syntax
B. How to work with code blocks and inline expressions
C. How to code inline loops
D. How to code inline conditional statements
E. Essential skills for Razor views
F. The starting folders and files for an app
G. How to code controllers that return views
H. How to create a default layout and enable tag helpers
I. How to use tag helpers to generate URLs for links
J. Three views that use the default layout and tag helpers
K. The three views displayed in a browser
L. More skills for Razor views
M. More tag helpers for generating URLs for links

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Course Summary (cont'd)

N. How to format numbers in a view
O. How to work with a model
P. How to pass a model to a view
Q. How to display model properties in a view
R. How to bind model properties to HTML elements
S. How to bind a list of items to a element
T. How to display a list of model objects
U. How to work with Razor layouts
V. How to create and apply a layout
W. How to nest layouts
X. How to use view context
Y. How to use sections
Z. The Guitar Shop website
AA. The user interface for customers
BB. The user interface for administrators

VIII. How to transfer data from controllers

A. How to use ActionResult objects
B. An introduction to ActionResult subtypes
C. How to return ActionResult objects
D. How to use the ViewData and ViewBag properties
E. How to use the ViewData property
F. How to use the ViewBag property
G. The NFL Teams 1.0 app
H. The user interface
I. The model layer
J. The Home controller
K. The layout
L. The Home/Index view
M. How to work with view models
N. How to create a view model
O. The updated Index() action method
P. The updated Home/Index view
Q. How to redirect a request
R. How to use the ActionResult classes for redirection
S. How to use the Post-Redirect-Get pattern
T. How to use the TempData property
U. How to get started with TempData
V. How to use methods of the TempData dictionary
W. The NFL Teams 2.0 app
X. The user interface
Y. The Home/Index view
Z. The Details() action method of the Home controller
AA. The Home/Details view
BB. The Add() action method of the Favorites controller
CC. The layout view

IX. How to work with session state and cookies

A. How ASP.NET MVC handles state
B. Six ways to maintain state
C. An introduction to session state
D. How to work with session state
E. How to configure an app to use session state
F. How to work with session state items in a controller
G. How to get session state values in a view
H. How to use JSON to store objects in session state
I. How to encapsulate session values
J. How to extend the ISession interface
K. How to use a wrapper class
L. The NFL Teams 3.0 app
M. The user interface
N. The session classes
O. The view model
P. The Home controller
Q. The layout
R. The Home/Details view
S. The Favorites controller
T. The Favorites/Index view
U. How to work with cookies
V. How to work with session cookies
W. How to work with persistent cookies
X. The NFL Teams 4.0 app
Y. The cookies class
Z. The Home controller
AA. The Favorites controller

X. How to work with model binding

A. An introduction to MVC model binding
B. How to use controller properties to retrieve GET and POST data
C. How to use model binding to retrieve GET and POST data
D. How to use model binding with complex types
E. How to use model binding with nested complex types

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Course Summary (cont'd)

F. More skills for binding data
G. How to use a submit button to post a value
H. How to post an array to an action method
I. How to control the source of bound values
J. How to control which values are bound
K. The ToDo List app
L. The user interface
M. The entity classes
N. The database context class
O. A utility class for filtering
P. The Home controller
Q. The layout
R. The Home/Index view
S. The Home/Add view

XI. How to validate data

A. How data validation works in MVC
B. The default data validation provided by model binding
C. How to use data attributes for validation
D. A Registration page with data validation
E. How to control the user experience
F. How to format validation messages with CSS
G. How to check validation state and use code to set validation messages
H. How to display model-level and property-level validation messages
I. How to enable client-side validation
J. How to customize server-side validation
K. How to create a custom data attribute
L. How to pass values to a custom data attribute
M. How to check multiple properties
N. How to customize client-side validation
O. How to add data attributes to the generated HTML
P. How to add a validation method to the jQuery validation library
Q. How to work with remote validation
R. The Registration app
S. The user interface and CSS
T. The Customer and RegistrationContext classes
U. The MinimumAgeAttribute class
V. The minimum-age JavaScript file
W. The Validation and Register controllers
X. The layout
Y. The Register/Index view

XII. How to use EF Core

A. How to create a database from code
B. How to code entity and DB context classes
C. How to configure the database
D. How to manage configuration files
E. EF commands for working with a database
F. How to use EF migration commands
G. How to work with relationships
H. How entities are related
I. How to configure a one-to-many relationship
J. How to configure a one-to-one relationship
K. How to configure a many-to-many relationship
L. How to seed data in the linking table of a many-to-many relationship
M. How to control delete behavior
N. The Bookstore database classes
O. The entity classes
P. The context and configuration classes
Q. How to create code from a database
R. How to generate DB context and entity classes
S. How to configure a generated DB context class
T. How to modify a generated entity class
U. How to work with data in a database
V. How to query data
W. How to work with projections and related entities
X. How to insert, update, and delete data
Y. How to handle concurrency conflicts
Z. How to check for concurrency conflicts
AA. How handle a concurrency exception
BB. How to encapsulate your EF code
CC. How to code a data access class
DD. How to use a generic query options class
EE. How to use the repository pattern

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Course Summary (cont'd)

XIII. The Bookstore website

A. The user interface and folder structure
B. The end user pages
C. The admin user pages
D. The folders and files
E. Some general-purpose code
F. Extension methods for strings
G. The generic QueryOptions class
H. The generic Repository class
I. The paging and sorting of the Author Catalog
J. The user interface and custom route
K. The abstract GridData class
L. The AuthorGridData subclass
M. The Author/List view model and the Author controller
N. The Author/List view
O. The paging and sorting of the Book Catalog
P. The user interface and custom route
Q. The BookGridData subclass
R. The Book/List view model and the Book controller
S. The Book/List view
T. The Cart page
U. Extension methods for session and cookies
V. The user interface
W. The model classes
X. The Cart controller
Y. The Cart/Index view
Z. The Admin area
AA. The BookRepository subclass
BB. The Related Books page
CC. The Delete() and Related() action methods

XIV. How to use dependency injection and unit testing

A. How to use dependency injection (DI)
B. How to configure your app for DI
C. How to use DI with controllers
D. How to create an interface for an existing class
E. How to use DI with an HttpContextAccessor object
F. How to use DI with action methods
G. How to use DI with views
H. How to get started with unit testing
I. How unit tests work
J. How to add an xUnit project to a solution
K. How to write a unit test
L. How to run a unit test
M. How to test methods that have dependencies
N. How to use a fake repository object
O. How to use a fake TempData object
P. How to create fake objects with Moq
Q. How to work with mock objects
R. How to mock a repository object
S. How to mock a TempData object
T. How to mock an HttpContextAccessor object
U. The Bookstore.Tests project
V. The Test Explorer
W. The Usings.cs file
X. The BookControllerTests class
Y. The AdminBookControllerTests class
Z. The CartTests class

XV. How to work with tag helpers, partial views, and view components

A. An introduction to tag helpers
B. How to use tag helpers
C. How to register tag helpers
D. How to create custom tag helpers
E. How to create a custom tag helper
F. How to create a tag helper for a non-standard HTML element
G. How to use extension methods with tag helpers
H. How to control the scope of a tag helper
I. How to use a tag helper to add elements
J. More skills for custom tag helpers
K. How to use properties with a tag helper
L. How to work with the model property that an element is bound to
M. How to use a ViewContext object with a tag helper
N. How to use dependency injection with a tag helper
O. How to create a conditional tag helper
P. How to generate URLs in a tag helper
Q. How to work with partial views
R. How to create and use a partial view
S. How to pass data to a partial view

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Course Summary (cont'd)

T. How to work with view components
U. How to create and use a view component
V. How to pass data to a view component
W. The Bookstore app
X. The Book Catalog page
Y. The updated ActiveNavbar tag helper
Z. The layout
AA. The Book/List view

XVI. How to authenticate and authorize users

A. An introduction to authentication
B. Three types of authentication
C. How individual user account authentication works
D. An introduction to ASP.NET Identity
E. How to restrict access to controllers and actions
F. How to get started with Identity
G. How to add Identity classes to the DB context
H. How to add Identity tables to the database
I. How to configure the middleware for Identity
J. How to add Log In/Out buttons and links to the layout
K. How to start the Account controller
L. How to register a user
M. The Register view model
N. The Account/Register view
O. The Register() action method for POST requests
P. The LogOut() action method for POST requests
Q. How to log in a user
R. The Login view model
S. The Account/Login view
T. The LogIn() action method for POST requests
U. How to work with roles
V. Properties and methods for working with roles
W. The User entity and view model
X. The User controller and its Index() action method
Y. The User/Index view
Z. Other action methods of the User controller
AA. The code that restricts access
BB. How to seed roles and users
CC. More skills for working with Identity
DD. How to change a user's password
EE. How to add more user registration fields

XVII. How to deploy ASP.NET Core MVC apps

A. How to work with Visual Studio's publish feature
B. How work with publish profiles
C. How to work with the Publish wizard
D. How to publish to the file system
E. How to create a publish profile for the file system
F. How to deploy to a publish folder
G. How to change the publish folder
H. How to publish a simple app to Azure
I. How to get started with Azure
J. How to create a publish profile for an Azure App Service
K. How to deploy to the Azure App Service
L. How to publish a data-driven app to Azure
M. How to add an Azure SQL Database to a publish profile
N. How to configure a connection string and migrations in a publish profile
O. How to deploy a data-driven app to the Azure App Service
P. How to work with an Azure app
Q. An introduction to the Azure portal
R. How to delete an Azure App Service

XVIII. How to use Visual Studio Code

A. How to work with existing projects
B. How to install VS Code
C. How to open and close a project folder
D. How to view and edit files
E. How to run and stop a project
F. How to create the database for a project
G. A summary of .NET EF Core commands
H. How to start a new project
I. How to create a new project
J. How to add NuGet packages to a project
K. How to work with the folders and files
L. How to install and manage client-side libraries
M. How to debug a project
N. How to set a breakpoint
O. How to work in break mode

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically