ASP.NET MVC Programming Using C# and .NET6

Duration: 35 hours

Prerequisites: Knowledge of fundamental HTML syntax is helpful, but not a requirement. Prior experience with any scripting or programming language is required.

Course Description: This course provides students with hands on experience using Visual Studio 2022 to create dynamic web applications using ASP.NET MVC, C#, and .NET6. The course provides a thorough introduction to the C# programming language in the context of .NET6, and includes coverage of the essentials of the C# programming language, built in data types, operators, control structures, classes and methods, collections and exception handling.

This course teaches students how to leverage the power of the Model-View-Controller design pattern with the ASP.NET MVC Framework to separate the layers of a web application. Students will use the Razor view engines to design a user interface. Students will learn how to build models to manage an application's data layer using both the Entity Framework and LINQ to SQL. Students will also learn how to build controllers containing action methods to manage communication between views and models.

Other topics include data scaffolding; URL routing; implementing security; unit testing; and deploying ASP.NET MVC applications. Comprehensive labs provide the students with experience creating, debugging, testing and deploying dynamic ASP.NET MVC applications.

Students Will Learn

- Using Visual Studio to create C# applications
- Introduction to ASP.NET MVC and .NET 6
- Building ASP.NET MVC Applications
- Building Controllers
- Working with Views
- Working with Models
- Understanding ASP.NET MVC Routing
- Creating controllers containing action methods to process HTTP requests
- Securing and Deploying ASP.NET MVC Applications
- Using Visual Studio's tools to create and run tests for ASP.NET MVC applications

Overview

Introduction to .NET

- Overview of the .NET Framework
- How .NET is Different from Traditional Programming
- Common Language Runtime (CLR)
- Common Language Specification (CLS)
- Common Type System (CTS)
- .NET Assemblies
- Microsoft Intermediate Language (CIL)

Introduction to Visual Studio

- Creating a Project
- Using the Code Editor
- Correcting Syntax Errors
- Setting Project Properties
- Adding References
- Compiling a Program
- Running a Program

- .NET Namespaces
- .NET Framework Class Library

Language Fundamentals

- C# Program Structure
- Defining Namespaces
- Understanding C# Data Types
- Defining Variables and Constants
- Comparing Value Types vs. Reference Types
- Working with Operators and Expressions
- Performing Type Conversions
- Using Console I/O
- Formatting Numbers, Date and Times

Methods and Parameters

- Defining Static and Instance Methods
- Passing Parameters by value and by reference
- Overloading Methods
- Using Variable Length Parameter Lists

Collections

- Defining and Using Arrays
- Understanding `System.Array`
- .NET Collections vs Generic Collections
- Working with Lists
- Working with Dictionaries
- Using LINQ to Objects

Overview of ASP.NET MVC

- Overview of Model-View-Controller Design Pattern
- ASP.NET MVC Application Architecture
- Understanding the MVC Execution Process
- Building an ASP.NET MVC Application Using Visual Studio
- Visual Studio MVC Project Templates
- Using a `web.config` File

Developing Views

- Creating Views
- Using the Razor View Engine
- Using ViewBag to Pass Controller Data
- Using HTML Helpers

Developing Controllers

- Debugging a Program
- Using the MSDN (Help)

Conditionals and Looping

- `if/else`
- `switch`
- `while` and `do/while`
- `for`
- `foreach`

Exception Handling

- What are Exceptions?
- .NET Exception Hierarchy
- Catching Exceptions
- Throwing Exceptions
- Managing Resources with Finally

Object-Oriented Programming

- Overview of Object-Oriented Programming
- Building Classes
- Defining Properties
- Using Auto-Implemented Properties
- Defining Methods
- Understanding Constructors
- Extending .NET Classes via Inheritance
- Defining and Implementing Interfaces
- Understanding the Role of Interfaces in .NET

Startup and Services

- Fundamentals of Dependency Injection
- Using the default DI container
- Using the Startup file
- Creating Services and Controlling Service Lifetime
- Building an ASP.NET MVC Application Using Visual Studio Project Templates
- Using Bootstrap CSS for the UI

Developing Models

- Creating Model Classes
- Working with Strongly-Typed Views
- Validating User Input
- Using Data Annotations for Display
- Working with the Entity Framework
- Working with LINQ to SQL
- Using Scaffolding to Generate Views

Routing Control

- Creating Controllers
- Defining Action Methods
- Working with HTML Form Data
- Mapping URLs to Action Methods
- Understanding `ActionResult` Types
- Using Model Binding
- Understanding Routing in ASP.NET MVC
- Defining URL Routes
- Registering Routes
- Adding Constraints to Routes
- Using Attribute Routing

Related Bootcamp

Track	Duration	Price
Microsoft .NET Developer: C#	161 hours	$4,475

Contact Us

Address: 6 Hemlock Drive Chelmsford, MA 01824

Phone: 978.250.4983

Mon - Thur: 9 am - 5 pm EST
Fri: 9 am - 4 pm EST

E-mail: info@developer-bootcamp.com

Copyright© 2025 Developer Bootcamp