Course duration

- 5 days

Course Benefits

- Describe the Microsoft Web Technologies stack and select an appropriate technology to use to develop any given application.
- Design the architecture and implementation of a web application that will meet a set of functional requirements, user interface requirements, and address business models.
- Configure the pipeline of ASP.NET Core web applications using middleware, and leverage dependency injection across MVC application.
- Add Controllers to an MVC Application to manage user interaction, update models, and select and return Views.
- Develop a web application that uses the ASP.NET Core routing engine to present friendly URLs and a logical navigation hierarchy to users.
- Create Views in an MVC application that display and edit data and interact with Models and Controllers.
- Create MVC Models and write code that implements business logic within Model methods, properties, and events.
- Connect an ASP.NET Core application to a database using Entity Framework Core.
- Implement a consistent look and feel across an entire MVC web application.
- Write JavaScript code that runs on the client-side and utilizes the jQuery script library to optimize the responsiveness of an MVC web application.
- Add client side packages and configure Task Runners.
- Run unit tests and debugging tools against a web application in Visual Studio 2017.
- Write an MVC application that authenticates and authorizes users to access content securely using Identity.
- Build an MVC application that resists malicious attacks.
- Use caching to accelerate responses to user requests.
- Use SignalR to enable two-way communication between client and server.
- Describe what a Web API is and why developers might add a Web API to an application.
- Describe how to package and deploy an ASP.NET Core MVC web application from a development computer to a web server.

Microsoft Certified Partner

Webucator is a Microsoft Certified Partner for Learning Solutions (CPLS). This class uses official Microsoft courseware and will be delivered by a Microsoft Certified Trainer (MCT).

Course Outline

1. Exploring ASP.NET Core MVC
    1. Overview of Microsoft Web Technologies
    2. Overview of ASP.NET 4.x
    3. Introduction to ASP.NET Core MVC
    4. Lab: Exploring ASP.NET Core MVC
        1. Exploring a Razor Pages Application
        2. Exploring a Web API Application
        3. Exploring an MVC Application
        4. After completing this course, students will be able to:
        5. Understand the variety of technologies available in the Microsoft web stack.
        6. Describe the different programming models available for developers in ASP.NET.
        7. Choose between ASP.NET Core and ASP.NET 4.x.
        8. Describe the role of ASP.NET Core MVC in the web technologies stack, and how to use ASP.NET Core MVC to build web applications.
        9. Distinguish between MVC models, MVC controllers, and MVC views.
2. Designing ASP.NET Core MVC Web Applications
    1. Planning in the Project Design Phase
    2. Designing Models, Controllers and Views
    3. Lab: Designing ASP.NET Core MVC Web Applications
        1. Planning Model Classes
        2. Planning Controllers
        3. Planning Views
        4. Architecting and MVC Web Application
3. Configure Middlewares and Services in ASP.NET Core
    1. Configuring Middlewares
    2. Configuring Services
    3. Lab: Configuring Middleware and Services in ASP.NET Core
        1. Working with Static Files
        2. Creating custom middleware
        3. Using dependency injection
        4. Injecting a service to a controller
4. Developing Controllers
    1. Writing Controllers and Actions
    2. Configuring Routes
    3. Writing Action Filters
    4. Lab: Developing Controllers
        1. Adding controllers and actions to an MVC application
        2. Configuring routes by using the routing table
        3. Configuring routes using attributes
        4. Adding an action filer
5. Developing Views
    1. Creating Views with Razor Syntax
    2. Using HTML Helpers and Tag Helpers

3. Reusing Code in Views
4. Lab: Developing Views
    1. Adding Views to an MVC Application
    2. Adding a partial view
    3. Adding a view component

6. Developing Models
    1. Creating MVC Models
    2. Working with Forms
    3. Validate MVC Application
    4. Lab: Developing Models
        1. Adding a model
        2. Working with Forms
        3. Add Validation
7. Using Entity Framework Core in ASP.NET Core
    1. Introduction to Entity Framework Core
    2. Working with Entity Framework Core
    3. Use Entity Framework Core to connect to Microsoft SQL Server
    4. Lab: Using Entity Framework Core in ASP.NET Core
        1. Adding Entity Framework Core
        2. Use Entity Framework Core to retrieve and store data
        3. Use Entity Framework Core to connect to Microsoft SQL Server
8. Using Layouts, CSS and JavaScript in ASP.NET Core MVC
    1. Using Layouts
    2. Using CSS and JavaScript
    3. Using jQuery
    4. Lab: Using Layouts, CSS and JavaScript in ASP.NET Core
        1. Applying a layout and link views to it
        2. Using CSS
        3. Using JavaScript
        4. Using jQuery
9. Client-Side Development
    1. Applying Styles
    2. Using Task Runners
    3. Responsive design
    4. Lab: Client-Side Development
        1. Use gulp to run tasks
        2. Styling using Sass
        3. Using Bootstrap
10. Testing and Troubleshooting
    1. Testing MVC Applications
    2. Implementing an Exception Handling Strategy
    3. Logging MVC Applications
    4. Lab: Testing and troubleshooting
        1. Testing a Model
        2. Testing a controller using a fake repository
        3. Implementing a repository in MVC project
        4. Add exception handling

5. Add logging
6. After completing this course, students will be able to:
7. Run unit tests against the Model–View–Controller (MVC) components, such as model classes and controllers, and locate potential bugs.
8. Build a Microsoft ASP.NET Core MVC application that handles exceptions smoothly and robustly.
9. Run logging providers that benefit your applications and run them by using a common logging API.

11. Managing Security
    1. Authentication in ASP.NET Core
    2. Authorization in ASP.NET Core
    3. Defending from Attacks
    4. Lab: Managing Security
        1. Use Identity
        2. Add Authorization
        3. Avoid the Cross-Site Request Forgery Attack
12. Performance and Communication
    1. Implementing a Caching Strategy
    2. Managing State
    3. Two-way communication
    4. Lab: Performance and Communication
        1. Implementing a Caching Strategy
        2. Managing state
        3. Two-Way communication
13. Implementing Web APIs
    1. Introducing Web APIs
    2. Developing a Web API
    3. Calling a Web API
    4. Lab: Implementing Web APIs
        1. Adding Actions and Call Them Using Microsoft Edge
        2. Calling a Web API using server-side code
        3. Calling a Web API using jQuery
14. Hosting and Deployment
    1. On-premise hosting and deployment
    2. Deployment to Microsoft Azure
    3. Microsoft Azure Fundamentals
    4. Lab: Hosting and Deployment
        1. Deploying a Web Application to Microsoft Azure
        2. Upload an Image to Azure Blob Storage

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this ASP.NET class:

- A minimum of two to three years of experience developing web-based applications by using Microsoft Visual Studio and Microsoft ASP.NET, proficiency in using the .NET Framework, and some familiarity with the C# language.