

ASP.NET Razor and Blazor Development

Day 1: Introduction to ASP.NET Razor

Module 1: Exploring ASP.NET Core MVC and Razor

- Lesson 1: Overview of Microsoft Web Technologies
- Lesson 2: Overview of ASP.NET
- Lesson 3: Understanding Core form of .NET
- Lesson 4: What is ASP.NET Razor?
- Lesson 5: Benefits of Using Razor

Module 2: Setting Up Your Development Environment

- Lesson 1: Installing Visual Studio 2022
- Lesson 2: Configuring Your Development Environment
- Lesson 3: Creating a New Razor Project
- Lesson 4: Planning in the Project Design Phase

Module 3: Basic Razor Syntax and Structure

- Lesson 1: Understanding Razor Syntax
- Lesson 2: Writing Your First Razor Script
- Lesson 3: Razor Syntax vs. HTML

Day 2: Building Dynamic Web Pages

Module 4: Developing Controllers and Razor Components

- Lesson 1: Writing Controllers and Actions
- Lesson 2: Configuring Routes
- Lesson 3: Writing Action Filters

- **Lesson 4: Razor Syntax & Directives**
- **Lesson 5: Pages vs Components**
- **Lesson 6: Routing in Razor**
- **Lesson 7: Rendering Components**
- **Lesson 8: Data Binding**
- **Lesson 9: Communication Between Components**
- **Lesson 10: Passing Parameters**

Module 5: Developing Views

- **Lesson 1: Creating Views with Razor Syntax**
- **Lesson 2: Using HTML Helpers and Tag Helpers**
- **Lesson 3: Reusing Code in Views**

Module 6: Validation

- **Lesson 1: Introduction to Validation**
- **Lesson 2: Using EditForm Component**
- **Lesson 3: Input Components**
- **Lesson 4: Validation Components**

Module 7: Using Bootstrap and Client-Side Development in Blazor Applications

- **Lesson 1: Providing a Responsive User Interface**
- **Lesson 2: Layout Using Bootstrap**
- **Lesson 3: Using CSS and JavaScript**
- **Lesson 4: Using jQuery**
- **Lesson 5: Applying Styles**
- **Lesson 6: Using Task Runners**

- Lesson 7: Responsive Design

Day 3: Advanced Features and Techniques

Module 8: Using REST Services (Web API) and Entity Framework Core

- Lesson 1: Creating Web API Services
- Lesson 2: Web API Controllers
- Lesson 3: Implementing Controller Actions
- Lesson 4: Configuring and Injecting HttpClient
- Lesson 5: Implementing GET, PUT, POST, and DELETE
- Lesson 6: Introduction to Entity Framework Core
- Lesson 7: Migrations and Database Creation
- Lesson 8: Adding Migrations

Module 9: Optimizing Razor Applications for Performance

- Lesson 1: Performance Tuning Techniques
- Lesson 2: Caching Strategies
- Lesson 3: Minimizing Page Load Time

Module 10: Razor Class Libraries

- Lesson 1: Creating a Razor Class Library
- Lesson 2: Using Components from a Library
- Lesson 3: Routing to Pages in a Library

Day 4: Performance Optimization and Security, Real-World Applications, and Deployment

Module 11: Testing and Troubleshooting

- Lesson 1: Testing MVC Applications
- Lesson 2: Implementing an Exception Handling Strategy
- Lesson 3: Logging MVC Applications
- Lesson 4: Debugging Razor Code
- Lesson 5: Logging and Monitoring

Module 12: Securing MVC-Based Apps Using Razor

- Lesson 1: Dealing with Common Security Threats in Web Applications
- Lesson 2: Authentication and Authorization
- Lesson 3: Configuring Authorization
- Lesson 4: Securing Pages and Components

Module 13: Hosting and Deployment

- Lesson 1: Deployment to Microsoft Azure
- Lesson 3: Microsoft Azure Fundamentals
- Lesson 4: Preparing for Deployment
- Lesson 5: Configuring the Hosting Environment
- Lesson 6: Monitoring and Maintaining Applications

Day 5: Implementing Microservices

Module 14: The Emergence of Microservices Architecture

1. Explore the ideal software development practice
2. Learn how a fine-grained Service-Oriented Architecture (SOA) can help to achieve the ideal

3. Learn how Microservices attempts to achieve the ideal .
4. Understanding Microservices vs. Monolithic Architecture

Module 15: Microservice Design Principles

1. Designing small microservices
2. Designing independent microservices
3. Designing resilient microservices

Module 16: Developing and Securing Microservices

1. Setting Up a Microservices Solution in Visual Studio
2. Creating [ASP.NET](#) Core Microservices
3. Security
 - Authentication and Authorization

Module 17: Monitoring, Logging, and Testing

1. Testing Microservices
 - Unit Testing,
 - Integration Testing, and
 - Contract Testing.