

# MVS3-14 - MOC 20486D - DEVELOPING ASP.NET CORE MVC WEB APPLICATIONS

Categoria: **Visual Studio**

## INFORMAZIONI SUL CORSO



**Durata:**  
5 Giorni



**Categoria:**  
Visual Studio



**Qualifica Istruttore:**  
Microsoft Certified  
Trainer



**Dedicato a:**  
Sviluppatore



**Produttore:**  
Microsoft

## OBIETTIVI

After completing this course, students will be able to:

- 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.

## PREREQUISITI

Before attending this course, students must have:

- Experience with Visual Studio 2017.

- Experience with C# programming, and concepts such as Lambda expressions, LINQ, and anonymous types.
- Experience in using the .NET Framework.
- Experience with HTML, CSS and JavaScript.
- Experience with querying and manipulating data with ADO.NET.
- Knowledge of XML and JSON data structures.

## CONTENUTI

### **Module 1: Exploring ASP.NET Core MVC**

- Overview of Microsoft Web Technologies
- Overview of ASP.NET 4.x
- Introduction to ASP.NET Core MVC

#### **Lab : Exploring ASP.NET Core MVC**

- Exploring a Razor Pages Application
- Exploring a Web API Application
- Exploring an MVC Application

### **Module 2: Designing ASP.NET Core MVC Web Applications**

- Planning in the Project Design Phase
- Designing Models, Controllers and Views

#### **Lab : Designing ASP.NET Core MVC Web Applications**

- Planning Model Classes
- Planning Controllers
- Planning Views
- Architecting and MVC Web Application

### **Module 3: Configure Middlewares and Services in ASP.NET Core**

- Configuring Middlewares
- Configuring Services

#### **Lab : Configuring Middleware and Services in ASP.NET Core**

- Working with Static Files
- Creating custom middleware
- Using dependency injection
- Injecting a service to a controller

### **Module 4: Developing Controllers**

- Writing Controllers and Actions
- Configuring Routes
- Writing Action Filters

#### **Lab : Developing Controllers**

- Adding controllers and actions to an MVC application
- Configuring routes by using the routing table
- Configuring routes using attributes
- Adding an action filter

### **Module 5: Developing Views**

- Creating Views with Razor Syntax
- Using HTML Helpers and Tag Helpers

- Reusing Code in Views

#### **Lab : Developing Views**

- Adding Views to an MVC Application
- Adding a partial view
- Adding a view component

#### **Module 6: Developing Models**

- Creating MVC Models
- Working with Forms
- Validate MVC Application

#### **Lab : Developing Models**

- Adding a model
- Working with Forms
- Add Validation

#### **Module 7: Using Entity Framework Core in ASP.NET Core**

- Introduction to Entity Framework Core
- Working with Entity Framework Core
- Use Entity Framework Core to connect to Microsoft SQL Server

#### **Lab : Using Entity Framework Core in ASP.NET Core**

- Adding Entity Framework Core
- Use Entity Framework Core to retrieve and store data
- Use Entity Framework Core to connect to Microsoft SQL Server

#### **Module 8: Using Layouts, CSS and JavaScript in ASP.NET Core MVC**

- Using Layouts
- Using CSS and JavaScript
- Using jQuery

#### **Lab : Using Layouts, CSS and JavaScript in ASP.NET Core**

- Applying a layout and link views to it
- Using CSS
- Using JavaScript
- Using jQuery

#### **Module 9: Client-Side Development**

- Applying Styles
- Using Task Runners
- Responsive design

#### **Lab : Client-Side Development**

- Use gulp to run tasks
- Styling using Sass
- Using Bootstrap

#### **Module 10: Testing and Troubleshooting**

- Testing MVC Applications
- Implementing an Exception Handling Strategy
- Logging MVC Applications

#### **Lab : Testing and troubleshooting**

- Testing a Model

- Testing a controller using a fake repository
- Implementing a repository in MVC project
- Add exception handling
- Add logging

### **Module 11: Managing Security**

- Authentication in ASP.NET Core
- Authorization in ASP.NET Core
- Defending from Attacks

#### **Lab : Managing Security**

- Use Identity
- Add Authorization
- Avoid the Cross-Site Request Forgery Attack

### **Module 12: Performance and Communication**

- Implementing a Caching Strategy
- Managing State
- Two-way communication

#### **Lab : Performance and Communication**

- Implementing a Caching Strategy
- Managing state
- Two-Way communication

After completing this module, students will be able to:

- Implement caching in a Microsoft ASP.NET Core application.
- Use state management technologies to improve the client experience, by providing a consistent experience for the user.
- Implement two-way communication by using SignalR, allowing the server to notify the client when important events occur.

### **Module 13: Implementing Web APIs**

- Introducing Web APIs
- Developing a Web API
- Calling a Web API

#### **Lab : Implementing Web APIs**

- Adding Actions and Call Them Using Microsoft Edge
- Calling a Web API using server-side code
- Calling a Web API using jQuery

### **Module 14: Hosting and Deployment**

- On-premise hosting and deployment
- Deployment to Microsoft Azure
- Microsoft Azure Fundamentals

#### **Lab : Hosting and Deployment**

- Deploying a Web Application to Microsoft Azure
- Upload an Image to Azure Blob Storage

**Materiale didattico:** Materiale didattico in formato digitale

**Costo materiale didattico:** incluso nel prezzo del corso a Calendario

**Natura del corso:** Operativo (previsti lab su PC)