



.NET Development and Visual Studio

20486: Developing ASP.NET Core MVC Web Applications

\$2,995.00

- 5 Days

Upcoming Dates

Course Description

In this 5-day course, the professional web developers will learn to develop advanced ASP.NET Core MVC applications using .NET Core tools and technologies. The focus will be on coding activities that enhance the performance and scalability of the Web site application. This course will also prepare the student for exam 70-486.

Course Outline

Module 1: Exploring ASP.NET Core MVC Microsoft ASP.NET Core MVC and the other web technologies of the ASP.NET Core can help you create and host dynamic, powerful, and extensible web applications. ASP.NET Core, of which ASP.NET Core MVC is part, is an open-source, cross-platform framework that allows you to build web applications. You can develop and run ASP.NET Core web applications on Windows, macOS, Linux, or any other platform that supports it. ASP.NET Core MVC supports agile, test-driven development cycle. It also allows you to use the latest HTML standard and Front-End frameworks such as Angular, React, and more.

Lessons

- 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

After completing this course, students will be able to:

- Understand the variety of technologies available in the Microsoft web stack.
- Describe the different programming models available for developers in ASP.NET.
- Choose between ASP.NET Core and ASP.NET 4.x.
- 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.
- Distinguish between MVC models, MVC controllers, and MVC views.

Module 2: Designing ASP.NET Core MVC Web Applications Microsoft ASP.NET Core MVC is a programming model that you can use to create powerful and complex web applications. However, all complex development projects, and large projects in particular, can be challenging and intricate to fully understand. Without a complete understanding of the purposes of a project, you cannot develop an effective solution to the customer's problem. You need to know how to identify a set of business needs and plan the Model-View-Controller (MVC) web application to meet those needs. The project plan that you create assures stakeholders that you understand their requirements and communicates the functionality of the web application, its user interface, structure, and data storage to the developers. By writing a detailed and accurate project plan, you can ensure that the powerful features of MVC are used effectively to solve the customer's business problems. Lessons

- 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

After completing this module, students will be able to:

- Plan the overall architecture of an ASP.NET Core MVC web application and consider aspects such as state management.
- Plan the models, controllers, and views that are required to implement a given set of functional requirements.

Module 3: Configure Middlewares and Services in ASP.NET Core ASP.NET Core is a framework that allows us to build many different kinds of applications. In this module, you will learn how to leverage the ASP.NET Core framework to handle requests and responses via existing, and custom middleware, and how to configure services for use in middleware and throughout other parts of the application, such as controllers. A middleware is a segment of code that can be used as part of the request and response pipeline that allows us to handle them according to any relevant parameter. This potentially allows multiple separate requests to be handled in a completely different fashion and receive separate responses. Services are classes that expose functionality which you can later use throughout different parts of the application, without having to keep track of scope manually in each individual location and instantiate any dependencies. This is done by using Dependency Injection. Dependency Injection is a technique used by ASP.NET Core that allows us to add dependencies into the code without having to worry about instantiating objects, keeping them in memory, or passing along required dependencies. This allows the application to become more flexible and to reduce potential points of failure whenever you change a service. Lessons