Trainer Preparation Guide for Course 20486D: Developing ASP.NET Core MVC Web Applications

What You Need to Know

The course materials are updated to reflect the use of Microsoft Visual Studio 2017 with Microsoft .NET Core 2.1. This course now ships without a virtual machine environment. The required software to run the labs and demos is specified in the Classroom Setup Guide. For better and easier maintenance of course content, the source code that accompanies this course (namely, the Allfiles directory and the step-by-step instructions) are hosted on GitHub. Make sure all the student machines include these files prior to the training. The GitHub repository is large, downloading it during the training may take too long.

Design of the Course

In this course, students will learn to develop advanced Microsoft ASP.NET Core Model-View-Controller (MVC) web applications by using .NET Core 2.1 tools and technologies. The focus will be on coding activities that enhance the performance and scalability of the website application. ASP.NET Core MVC will be introduced and compared with other ASP.NET programming models so that students know when each model can be used, or should be used.

The course is structured around a collection of labs that take the student through the process of designing and developing ASP.NET Core MVC web applications. In each lab, the students learn a different aspect of developing an MVC web application.

It is imperative that you explain the purpose of each lab to the students so that they understand what they will be working on throughout the course. There are demonstrations for you to show students the key points of each lab. Use these to show both the end-user experience of the lab results and the tasks that the student will complete in Visual Studio. These demonstrations will help students to be more confident when completing the lab and to be aware of the results of the tasks that they undertake.

Encourage students to read the scenarios before each lab and each exercise so that they understand what they are doing in the context of developing a real-world application.

Module 1: “Exploring ASP.NET Core MVC” describes the Microsoft Web Technologies stack and compares the various ASP.NET programming models. It explores the structure of an MVC web application and identifies features such as models, views, controllers, configuration files, style sheets, and script files. The lab at the end of this module is used to explore the structure of the three programming models available in ASP.NET Core, which are Razor Pages, Web API, and MVC.

Module 2: “Designing ASP.NET Core MVC Web Applications” explains how to design an MVC application. This module emphasizes agile and extreme programming, in which only large-scale design is done before starting coding. This emphasis continues in later modules. In the lab at the end of this module, the students design an ASP.NET Core MVC web application that meets a set of functional requirements. They will record the design in an accurate, precise, and informative manner.

Module 3: “Configure Middleware and Services in ASP.NET Core” explains how to configure the pipeline of ASP.NET Core web applications by using middleware. It also describes how to configure services and inject services to controllers. In the lab at the end of this module, the students create a new empty ASP.NET Core project, configure its pipeline using middleware, and use dependency injection to inject services.

Module 4: "Developing Controllers" explains how to develop controllers in an ASP.NET Core MVC web application. The first lesson describes how to add a controller to a web application that responds to user actions. The second lesson describes how to add routes to the ASP.NET routing engine and ensure that URLs are user-friendly throughout an MVC web application. The third lesson describes how to write code in action filters that executes before or after a controller action. In the lab at the end of this module, the students add controllers and actions, configure routes, and add an action filter.

Module 5: "Developing Views" explains how to develop views in an ASP.NET Core MVC web application. The first lesson describes how to create an MVC view and add Razor markup to it, to display data to users. The second lesson describes how to create HTML helpers and tag helpers, and embed them in views. The third lesson describes how to reuse Razor markup in multiple locations throughout an application. In the lab at the end of this module, the students add views to construct the user interface of a cities web application.

Module 6: "Developing Models" explains how to develop models in an ASP.NET Core MVC application. The first lesson describes how to add a model to an MVC application. The second lesson describes how to display the model data in a view by using helpers. The third lesson describes how to use forms with data annotations and validation. In the lab at the end of this module, the students create and code an MVC model that will be used in a butterflies' shop application. The model includes properties that describe a butterfly.

Module 7: "Using Entity Framework Core in ASP.NET Core" explains how to use Entity Framework Core in an ASP.NET Core application. The first lesson introduces Entity Framework. The second lesson describes how to use Entity Framework Core in a web application. The third lesson describes how to use Entity Framework to connect to Microsoft SQL Server from MVC applications. In the lab at the end of this module, the students use Entity Framework Core to connect a cupcakes and bakeries web application to a database.

Module 8: "Using Layouts, CSS, and JavaScript in ASP.NET Core MVC" explains how to use layouts, CSS, and JavaScript in an ASP.NET Core application. It demonstrates how to implement a consistent look and feel across an entire MVC web application. The first lesson demonstrates how to create a layout and link views to a layout. The second lesson describes how to add styles and JavaScript code to an application and execute the JavaScript code in response to user actions. The third lesson describes how to use jQuery to write compelling pages rapidly. In the lab at the end of this module, the students use layout, CSS, and JavaScript in a zoo web application.

Module 9: "Client-Side Development" explains how to use add client-side packages and configure task runners. The first lesson demonstrates how to use Bootstrap, Less, and Sass in an ASP.NET Core application. The second lesson describes how to use task runners in an ASP.NET Core application. The third lesson describes how to ensure that a web application displays correctly on devices with different screen sizes. In the lab at the end of this module, the students use Bootstrap and Sass to style a web-based ice cream application. They will use gulp to compile, minify, and bundle files.

Module 10: "Testing and Troubleshooting" explains how to run unit tests and debugging tools against a web application in Visual Studio and configure an application for troubleshooting. The first lesson demonstrates how to run tests against MVC components, such as models, and locate potential bugs. The second lesson describes how to build an MVC application that handles exceptions smoothly and robustly. The third lesson describes how to log MVC applications. In the lab at the end of this module, the students add testing and troubleshooting to a store web application to improve its quality.

Module 11: "Managing Security" explains how to manage security in ASP.NET Core applications. The first lesson demonstrates how to use authentication in ASP.NET Core applications. The second lesson describes how to use authorization in ASP.NET Core applications. The third lesson describes several common hacking techniques and explains how to make sure an MVC application is not vulnerable to each technique. In the lab at the end of this module, the students manage security in a web-based library application.

Module 12: “Performance and Communication” explains how to improve the performance and reliability of ASP.NET Core applications. The first lesson demonstrates how to configure ASP.NET caches to accelerate the delivery of content and ensure that out-of-date information is not displayed. The second lesson describes how to persist application, session, and user state information in the most appropriate location. The third lesson describes how to use Web Sockets to create a live data feed in an MVC application. In the lab at the end of this module, the students manage the performance and the communication of a web-based electric store application.

Module 13: “Implementing Web APIs“ explains how to implement Web APIs. The first lesson introduces Web APIs. The second lesson describes how to develop a Web API. The third lesson describes how to call a Web API. In the lab at the end of this module, the students create a server-side Web API application and a client-side ASP.NET Core MVC application. In the client-side application, they call the Web API actions.

Module 14: “Hosting and Deployment” explains how to deploy an ASP.NET Core MVC web application from a development computer to a web server. The first lesson describes common deployment scenarios for web applications and how to choose the most appropriate deployment destination for a given scenario. The second lesson describes how to deploy a completed MVC web application to Microsoft Azure. The third lesson describes Azure fundementals. In the lab at the end of this module, the students deploy an ASP.NET Core MVC application to Azure.

Required Materials to Teach This Course

To teach this course, you need the following materials:

- Course Handbook
- Microsoft PowerPoint files
- Microsoft OneNote Trainer Pack on the Microsoft Learning Download Center https://learningdownloadcenter.microsoft.com
- Classroom Setup Guide
- The latest lab files from GitHub. Microsoft Learning is using GitHub to publish the lab steps and lab scripts for many courses that cover ASP.NET Core and Azure. Using GitHub allows for collaboration between a course’s authors and Microsoft Certified Trainers (MCTs) to ensure the content is current with the ASP.NET Core and Azure platform changes. Additionally, using GitHub allows the MCTs to provide feedback and suggestions for lab changes, and enables a course’s authors to update lab steps and scripts quickly and relatively easily.
- An Azure user account and Azure user accounts for each student.

Important We recommend that you use PowerPoint 2007 or newer to display the slides for this course. If you use PowerPoint Viewer or an older version of PowerPoint, the slide features might not display correctly.

Prerequisite Knowledge to Teach This Course

To present this course, you must have the following knowledge and skills:

- Experience in developing applications by using Visual Studio 2017.

- Experience in developing ASP.NET Core MVC web applications.
- Experience in developing applications by using the Visual C# language.
- Experience in developing websites that use the JavaScript language.
- Experience in using the jQuery JavaScript library.
- Experience in using Microsoft Azure to host websites, services, and databases.

Preparation Tasks

Complete the following tasks to prepare for this course.

Courses or Workshops

We highly recommend that you audit the latest versions of the following courses:

- Course 20483: Programming in C#
- Course 20480: Programming in HTML5 with JavaScript and CSS3
- Course 20532: Developing Microsoft Azure Solutions

Exams

To identify your technical proficiency with the content of this course, we highly recommend that you pass the following exam:

- Exam 70-486: Developing ASP.NET MVC Web Applications

Technical Preparation Activities

We highly recommend that you complete the following technical preparation activities:

- Use the OneNote Trainer Pack (OTP) on the Microsoft Learning Download Center https://learningdownloadcenter.microsoft.com to prepare for delivering the course.
    - The OTP includes the following content on each page (and in this order):
        - Slides
        - Instructor Notes
        - Student Handbook Content
- If you are using Digital Microsoft Official Courses (MOC) via the Arvato Skillpipe reader, become familiar with how the digital content is accessed and configured to ensure smooth setup and access for students. You should also ensure that everything is in place for students to have a smooth experience when signing in for the first time and accessing their content.

  Digital MOC can be purchased for use during course delivery at the same time you order the course materials.
- Review the learning product error log, which is available on the Microsoft Learning Download Center: https://learningdownloadcenter.microsoft.com.
- Practice using the Microsoft products and tools that are associated with this learning product:
    - Microsoft Visual Studio 2017 Community Edition
    - Microsoft Edge, including F12 Developer Tools

        - The Microsoft Azure administration portal
    - Review the following websites for updated information on the relevant Microsoft products:
        - The ASP.NET website at http://www.asp.net
        - The Microsoft Azure homepage at http://azure.microsoft.com
- Review the course change log, also available on the MCT Download Center, if you have taught a previous version of this course.

Instructional Preparation Activities

We highly recommend that you complete the following instructional preparation activities:

- Read the About This Course at the beginning of the course handbook for the learning product.
- Walk through the Introduction slide deck for the learning product.
- Walk through each module presentation slide deck and read the corresponding Instructor Notes (located in the notes view of the presentation slide deck) for the module. Note that additional hidden slides are used in each slide deck to accommodate the amount of Instructor Notes information for a given topic.
- Practice presenting each module:
    - Identify the key points and must-know information for each topic.
    - Perform each demonstration and hands-on lab.
    - Anticipate the questions that students might have.
    - Identify examples, analogies, impromptu demonstrations, and additional delivery tips that will help to clarify module content and provide a more meaningful learning experience for your specific audience.
    - Note any problems that you might encounter during a demonstration or a lab exercise, and determine a course of action for how you will resolve the problems in the classroom. To access the lab answer keys, refer to GitHub: https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/tree/master/Instructions.
    - Work through the Module Review and Takeaways section at the end of each module and determine how you will use this section to reinforce student learning and promote knowledge transfer to on-the-job performance.
    - Customize and enhance your instructor notes.
- Consult the Born to Learn forums for additional tips and strategies—posted by your fellow Microsoft Certified Trainers (MCTs)—for teaching the learning product.
- Review the updated information about the Microsoft Certification Program on the Microsoft Learning Certifications website.

Instructor Computer Setup

Set up the instructor computer by following the setup instructions in the Classroom Setup Guide. This document provides hardware requirements for the instructor computer in addition to detailed setup instructions.

Course Timing

The following schedule is an estimate of the course timing. Your timing might vary. Every student might not finish every lab. Use your judgment to set a reasonable time to move on to the next module.

This schedule has been developed to provide about six hours of training per day. Each day starts at 9:00 A.M. and ends between 4:30 and 5:00 P.M. and includes two 15 minute breaks and one hour for lunch. Each day also includes time to review the previous day's topics and to answer questions from the students. Be realistic about your timings.

Day 1

Start	End	Module
9:00	9:30	Introduction
9:30	10:30	Module 1: Exploring ASP.NET Core MVC
10:30	10:45	Break
10:45	12:05	Lab: Exploring ASP.NET Core MVC
12:05	13:05	Lunch
13:05	13:50	Module 2: Designing ASP.NET Core MVC Web Applications
13:50	14:35	Lab: Designing ASP.NET Core MVC Web Applications
14:35	14:50	Break
14:50	15:45	Module 3: Configure Middleware and Services in ASP.NET Core
15:45	16:45	Lab: Configure Middleware and Services in ASP.NET Core

Day 2

Start	End	Module
9:00	9:30	Day 1 review
9:30	10:45	Module 4: Developing Controllers
10:45	11:00	Break
11:00	12:10	Lab: Developing Controllers
12:10	12:50	Module 5: Developing Views
12:50	13:50	Lunch
13:50	14:15	Module 5: Developing Views (*continued*)
14:15	15:20	Lab: Developing Views
15:20	15:35	Break
15:35	16:45	Module 6: Developing Models

Day 3

Start	End	Module
9:00	9:30	Day 2 review
9:30	10:35	Lab: Developing Models
10:35	10:50	Break
10:50	11:45	Module 7: Using Entity Framework Core in ASP.NET Core
11:45	12:15	Lab: Using Entity Framework Core in ASP.NET Core
12:15	13:15	Lunch
13:15	14:00	Lab: Using Entity Framework Core in ASP.NET Core (*continued*)
14:00	15:00	Module 8: Using Layouts, CSS and JavaScript in ASP.NET Core MVC
15:00	15:15	Break
15:15	16:30	Lab: Using Layouts, CSS and JavaScript in ASP.NET Core MVC

Day 4

Start	End	Module
9:00	9:30	Day 3 review
9:30	10:45	Module 9: Client-Side Development
10:45	11:00	Break
11:00	12:00	Lab: Client-Side Development
12:00	12:35	Module 10: Testing and Troubleshooting
12:35	13:35	Lunch
13:35	14:10	Module 10: Testing and Troubleshooting (*continued*)
14:10	15:20	Module 11: Managing Security
15:20	15:35	Break
15:35	16:55	Lab: Managing Security

Day 5

Start	End	Module
9:00	9:30	Day 4 review
9:30	10:30	Module 12: Performance and Communication
10:30	10:45	Break
10:45	11:35	Lab: Performance and Communication
11:35	12:25	Module 13: Implementing Web APIs
12:25	13:25	Lunch
13:25	14:30	Lab: Implementing Web APIs
14:30	15:30	Module 14: Hosting and Deployment
15:30	15:45	Break
15:45	16:45	Lab: Hosting and Deployment