ASP.NET MVC Using C#

This course provides a practical hands-on introduction to developing Web applications using ASP.NET MVC 5 and C#. This Web development framework from Microsoft emphasizes separation of concerns in the architecture and testability of applications. The course includes coverage of the Razor View Engine, Entity Framework 6 and ASP.NET Web API 2. It also introduces Windows Azure and the deployment of ASP.NET MVC applications to the Azure cloud. The course covers the fundamentals of the Model-View-Controller design pattern and its implementation in ASP.NET MVC. This technology is compared with classical ASP.NET Web Forms. The two technologies share a common ASP.NET infrastructure. Visual Studio 2013 with .NET 4.5.1 is used as a productive platform for creating ASP.NET MVC applications. After presenting the fundamentals of the technology with several examples, the main components of Model, Controller and View are covered in detail. The discussion of the Model incorporates modern Microsoft data access technologies, including LINQ and ADO.NET Entity Framework 6. The routing mechanism of ASP.NET MVC is covered. The course introduces automated unit testing of Web applications, one of the chief advantages of the new technology. Security is covered, with some hands-on illustrations of attacks and defenses against them. The course includes a discussion of how ASP.NET MVC and Web Forms can be used together in the same application. The course includes with an introduction to ASP.NET Web API and concludes with deployment to Windows Azure. An appendix covers deployment on IIS 7.5.

Course Objectives:

- Gain a thorough understanding of the philosophy and architecture of Web applications using ASP.NET MVC.
- Acquire a working knowledge of Web application development using ASP.NET MVC 5 and Visual Studio 2013.
- Access databases using ADO.NET, LINQ, and Entity Framework 6.
- Perform automated testing of Web applications created using ASP.NET MVC and the built-in Visual Studio Unit Test Framework.
- Implement security in ASP.NET MVC applications.
- Combine ASP.NET MVC and Web Forms when appropriate.
- Create HTTP services using ASP.NET Web API2.
- Deploy ASP.NET MVC applications to the Windows Azure cloud.

Audience: Experienced application developers or architects responsible for Web applications in a Microsoft environment.

Prerequisites: Good working knowledge of C# and the .NET Framework. Basic knowledge of ASP.NET and HTML is also required.

Number of Days: 3 days

1 Introduction to ASP.NET MVC

Review of ASP.NET Web Forms
Advantages and Disadvantages of Web Forms
Model-View-Controller Pattern
ASP.NET MVC
Advantages and Disadvantages of ASP.NET MVC
Goals of ASP.NET MVC
Unit Testing

- 2 **Getting Started with ASP.NET MVC**
- ASP.NET MVC 5 Testbed
- Starter Application
- Simple App with Controller Only
- Action Methods and Routing
- Action Method Return Type
- Rendering a View
- Creating a View in Visual Studio
- The View Web Page
- Dynamic Output
- Razor View Engine
- Embedded Scripts
- Using a Model with a ViewBag
- Controller Using the Model & ViewBag
- View Using the Model and ViewBag
- Using Model Directly
- A View Using Model in Visual Studio
- View Created by Visual Studio
- Passing Parameters in Query String
- 3 **ASP.NET MVC Architecture**
- The Controller in ASP.NET MVC
- The View in ASP.NET MVC
- The Model in ASP.Net MVC
- How MVC Works
- Using Forms
- HTML Helper Functions
- Displaying Sign In Form
- Handling Form Submission
- Model Binding
- Greet View
- Input Validation
- Nullable Type
- Checking Model Validity
- Validation Summary
- 4 **The Model**
- Microsoft Technologies for the Model
- SmallPub Database
- ADO.NET
- Language Integrated Query (LINQ)
- ADO.NET Entity Framework
- SmallPub Tables
- SmallPub Entity Data Model
- Entity Data Model Concepts
- Class Diagram
- Context Class
- List of Categories
- List of Books
- A LINQ Query
- IntelliSense
- Controller
- Web.config
- Using a Parameter
- Modifying a Data Source
- Entity Framework in a Class Library
- Data Access Class Library
- Client Code
- ASP.NET MVC Database Clients
- NuGet Package Manager
- Visual Studio Extensions
- Online Gallery
- Entity Framework 6 via NuGet
- 5 **The Controller**
- IController Interface
- A Low Level Controller
- Displaying Context
- Add and Subtract
- Controller Base Class
- Action Methods
- HomeController
- Math Controller
- Invoking MathController
- Receiving Input
- Non-Nullable Parameters
- Nullable Parameters
- Using a Model
- HTML Output
- Filters
- Asynchronous Controllers
- 6 **The View**
- Action Results
- JavaScript Object Notation
- Action Method Attributes
- View Responsibility
- A Program without a View
- A Program with a View
- View Page
- Passing Data to the View
- Passing Lists to the View
- HTML Helper Methods
- Link-Building Helpers
- Form Helpers
- Validation Helpers

Templated Helpers
Validation in Model
Validation in Controller
ValidationMessage Helper
7 Routing
ASP.NET Routing
Routing in ASP.NET MVC
RouteCollectionExtensions Class
Default Values for URL Parameters
Using a Default Route
Home Controller
Assigning Parameter Values
Controller Code
View Code
Properties of Routes
UrlParameter Optional
A Problem with the Route Map
Matching URLs to Route
Route Debugging Utility
Multiple Matches
Fixing the Bug
Debugging Routes
Areas
8 Unit Testing
Test-Drive Development
Functional Tests
Unit Tests
Test Automation
Refactoring
Regression Testing
Unit Testing in ASP.NET MVC
Installing UNnit via NuGet
A Test Class Library
The Model
Testing the Model
Structure of Unit Tests
Assertions
Visual Studio Unit Framework
Assert Class
Assert.AreEqual()
CollectionAssert Class
StringAssert Class
Test Case
Test Methods, Class, Runner
Ignoring Tests
Fixing the Bug
Testing Controllers
Classes with External Dependencies
Dependency Injection
Mocking Frameworks
Using Moq
Models
Controller
Index View
Monthly Payment View
Unit Tests
Inversion of Control (IoC) Contain
9 Security
Web Security
Input Forgery
Cross-Site Scripting
Entering JavaScript
The Attack
What Allowed the Attack
Blocking the Attack
Using Razor
Session Hijacking
Cross-Site Request Forgery
Controller and Model Code
View Code
Attacker Code
The Defense
SQL Injection
Using the MVC Framework Securely
Authorize Attribute
ChildActionOnly Attribute
RequireHttps Attribute
ValidateInput Attribute
10 Combining ASP.NET MVC and Web Forms
Using Web Forms in an MVC Application
Linking and Redirecting from Web Forms Pages to MVC Actions
Default.aspx
Sharing Data between ASP.Net MVC and ASP.NET Web Forms
Using Web Form Controls in MVC Views
Using MVC in a Web Forms Application

Steps to Create Hybrid MVC – Web Forms Application
Web Forms MVC Interop in VS 2013
11 ASP.NET Web API
REST
Representation, State and Transfer
Collections and Elements
Specifying a Start Page
Implementing PUT Verb
Using Fiddler
Composing a Request
ASP.NET MVC and We API
Route Registration
HTTP Response Codes
POST Response Code
HttpResponseException
Web API Clients
HttpClient
Initializing HttpClient
Issuing a GET Request
Issuing a POST Request
Performing CRUD Operations
Book Repository
BooksController
View
Exercise the Server Using Fiddler
Client Code: Book Class, Program Class
ShowBooks()
AddBooks()
Update and Delete
12 Deployment on Windows Azure
What is Windows Azure?
A Windows Azure Testbed
Publish to Azure
Import Publish Settings
Manage Azure Subscriptions
Sign into Windows Azure
Download Subscripion File
New Publish Settings
Create a Site on Windows Azure
Publish
Web Deployment Completed
Modifying a Web Application
Publish to a New Site
Deploy to Original Site
Publish Preview
13 Appendix A: Learning Resources
14 Appendix B: Deployment in IIS 7.5
Internet Information Services
Installing IIS 7.5
MVC with IIS 7.5
.NET Framework Version
Registering ASP.NET
Convert to an Application
XCOPY Deployment