



**STELLAR  
TECH PRO**



# ASP.NET MVC

**Course Duration  
35-40 hours**

**Sessions: 20**

**Level: Basic | Intermediate | Advanced**



**Pre-requisite:** Before you start learning this course, you should have a basic knowledge of

- HTML/CSS
- JavaScript

## ASP.Net Overview

- Introduction of available Web technologies
- Difference between ASP and ASP.Net
- Page Life Cycle
- Processing Directives
- Anatomy of ASP.Net Page
- Two way of build Web Application
- Web Site Template
- Web Application Template
- Difference between Web application and site.

## ASP.Net Core Contro

- Html Controls
- Html Container Controls
- Html Input Controls
- Web Controls
- Core Web Controls
- Miscellaneous Controls
- Controls Validation
- Client vs Server side
- Create custom control

## Data Access

- ADO.Net
- Difference between ADO and ADO.Net
- Perform CRUD using Stored Procedure
- Data-Bound and Data Source Control
- Prevent Sql Injection Attack
- Difference between
- DataReader vs DataTable
- Data Adapter vs DataSet

## Page Composition and Usability

- Master and Content Page
- Site Navigation
  - Menu
  - Breadcrumb
- Styling Asp.Net Page
  - Themes
  - Skin

## Configuration File

- Configure file hierarchy
- Web.Config
- Machine.Config
- Difference between above two

## Client-Side Programming

- Asp.Net Ajax
- Dynamically Modifying WebPages
- Asp.Net AjaxControlToolkit

## Troubleshooting

- Debugging
- Tracing
- Error handling

## State Management

- What is state management?
- Why it is required?
- Two types (Client vs Server)
- Difference between them



**Description:** In this course, we provide a practical hands-on introduction to developing and consuming ASP.NET Web API services using C# with Visual Studio. This is the latest framework from the Microsoft technologies stack which makes it easy to create RESTful Http services that can reach a broad range of clients (web, desktop, mobile and more). It is an excellent platform for creating RESTful Web applications.

## Pre-requisite: Experience in the following areas is necessary:

- Good working knowledge of C# and the .NET Framework.
- Basic knowledge of ASP.NET, HTML, and JavaScript

## What you will learn

- Gain an understanding of what ASP.NET Web API is and why it is useful.
- Perform validation of request data
- Consume HTTP services using both jQuery and the ASP.NET Web API client library
- Handle different media formats and content variations in Web API services
- Understand routing and binding in Web API

## Introduction

- Pragmatic REST
- What and Why
- Use Fiddler with HTTP method
- Web API Versus the WCF Summary

## Web API and HTTP

- HTTP Response Codes
- HttpResponseMessage
- Implementing POST
- Implementing PUT
- Implementing DELETE
- Summary

## NET Clients

- Web API Client Libraries
- HttpClient
- Issuing GET Requests
- Issuing POST Requests

## Binding, Validation and Routing

- Reading Raw HTTP Requests
- Route Data, Query String and Request Body
- Binding to Simple Types
- Validation Using Data Annotations
- Routing in ASP.NET Web API

## Media Formatters and Content Negotiation

- Internet Media Types
- Media Formatters
- JSON and XML Formatters
- Content Negotiation
- Accept and Content-Type Headers
- Using the Query String
- Custom Request Headers
- Summary

## JavaScript Clients

- Document Object Model
- jQuery
- Sending AJAX Requests
- summary



**Pre-requisite:** Before you start learning this course, you should have a basic knowledge of

- Html
- Css
- Jquery
- Ajax
- Javascript
- C#

## Overview of the ASP.NET

- Introduction of different Web Technology
- What is ASP.NET MVC
- Role of Model, View, and Controller
- How ASP.NET MVC Works
- Benefits of using ASP.NET MVC
- Summary

## Setting up and Installing ASP.NET MVC

- Installing Internet Information Server
- Installation of ASP.NET MVC
- Application Setting in IIS.
- Summary

## Microsoft SQL Server 2012

- Overview of SQL Server 2012
- Installation of SQL Server 2012
- Features of SQL Server Express
- SQL Server 2008 Express management tools
- Summary

## Getting Started with MVC

- ASP.NET MVC project templates
- Understanding the structure of ASP.NET MVC project
- Naming conventions
- Creating views
- Defining controllers
- Defining a data model
- Overview of coding standards follows during programming
- Summary

## Creating an application in MVC

- Creating strongly-typed views
- Understanding URLs and action methods
- Using HTML helpers
- Handling form post-backs
- Data validation

## Razor View Engine

- Razor Basics
- Razor design goals
- Implementation of Razor view
- Razor syntax
- Accessing Model Data in Razor views

## Strengthen ASP.NET MVC Applications

- ASP.NET application architecture best practices
- Implementing a Repository and Entity Framework Data Model
- Using Dependency Injection
- Implementing a custom controller factory

## View Techniques

- Defining and using custom HTML Helpers
- Defining a layout / MVC Master Page
- Using Styles
- Defining and using partial views
- Razor Helper Method syntax

## Implementing Navigation in MVC web apps

- Defining view-model classes
- Implementing Data Filtering in a Controller
- Understanding the Routing mechanism
- Adding custom entries to a route table
- Defining defaults, parameters, and validation
- Generating URLs and Hyperlinks
- Custom Route constraints



## MVC State Management

- Using hidden fields
- Session and Application State
- Custom model bindings

## Using AJAX and jQuery with ASP.NET MVC

- Overview of AJAX and ASP.NET MVC
- Unobtrusive AJAX
- Using AJAX Action Links
- Overview of jQuery
- jQuery Techniques
- Using jQuery UI

## ASP.NET Web API with MVC

- Overview of the ASP.NET Web API
- Building servers and clients
- Content negotiation
- Validation
- Query able Interfaces
- Dependency Injection

## FTP Management

- Understanding FTP
- Setting up FTP Server (Live)
- Uploading and downloading FTP contents
- Summary

## Sending Emails

- Designing email panel
- How to send an email to various users
- Sending auto emails
- Summary

## Deployment

- Deploying application on Web Server
- Summary

## Live Project

- Getting customer's requirements
- Preparing database and business logics
- Developing application
- Testing and implementing the project
- Troubleshooting the project application after implementation
- Summary

MVC

ASP.NET

