

## ASP.NET Core Syllabus

### Overview:

All .NET Beginner(s)/Professional(s) who are keen to develop modern, light weight and cloud-based web applications should go for this course. ASP.NET Web form and ASP.NET MVC available because of its age is considered to be very matured for web application development, it is because of the popularity of ASP.NET Core Framework and many exclusive features of ASP.NET Core that today it is dominating over Web Forms and MVC are first choice for large sized enterprise web application development.

### Course Objectives:

- ❖ Understand the benefits of Asp.net MVC Core 3 design over traditional ASP.NET Web Forms.
- ❖ Acquiring sufficient knowledge on role of Model, View and Controller in integrating them to develop complete web application
- ❖ Understand how Routing API maps requests to action methods in controller.
- ❖ Learn how to reuse code rendering HTML using custom HTML Helper methods and Tag Helpers.
- ❖ Building Custom Model Binders for typical conditions in which built-in default binders are not usable.
- ❖ Understanding and applying validation framework for both client and server validations.
- ❖ Access databases and performing CRUD operations using LINQ and Entity Framework ❖ Implement security in ASP.Net Core applications.
- ❖ Use Ajax and partial page updates and rendering and executing JavaScript code from ASP.NET Core.
- ❖ Modularize using Areas.
- ❖ Compress the output using bundling and minification features.
- ❖ Develop Service Oriented RESTful services using Web API feature of ASP.NET Core.
- ❖ Build and deploy ASP.NET Core application to the production server.

### Tools/SDK/IDE:

- ❖ Visual Studio 2017 or higher, SQL Server 2012 or higher.
- ❖ LINQ, Entity Framework and SQL Server for database related operations.

**Module 1:- Introduction to ASP.NET Core**

In this module you learn what is ASP.NET Core and its features. Also you find here the advantages of ASP.NET Core framework over the existing frameworks for developing a web application.

- ❖ Introduction
- ❖ What is ASP.NET Core?
- ❖ ASP.NET Core Features
- ❖ Advantages of ASP.NET Core
- ❖ MVC Pattern
- ❖ Understanding ASP.NET Core MVC
- ❖ ASP.NET Core vs. ASP.NET MVC vs. ASP.NET Web Forms

**Module 2: - ASP.NET Core First Application**

In this chapter you get knowledge about how to create an ASP.NET Core application and setting up project initials with basic web page example.

- ❖ ASP.NET Core Environment Setup
- ❖ ASP .NET Core First Application
- ❖ Project Layout
- ❖ Understanding Life Cycle of ASP.Net Core Request

**Module 3:- Controllers & Action Methods**

In this chapter you will know what is a controller and action method and the responsibilities of a controller, the basic functionality of filters with few examples.

- ❖ Controllers Overview
- ❖ Action Methods
- ❖ Passing data from Controller to View
- ❖ Middleware

**Module 4:- Views**

This deals with what is a view, razor syntax and different types of views with examples.

- ❖ Introducing Razor View
- ❖ Advantages of Razor View
- ❖ Razor Syntax
- ❖ Partial Views
- ❖ Layout Pages

**Module 5:- Testing**

In this chapter we have different types Helper used to reduce number of key strokes when writing the html content in views and writing reusable components.

- ❖ Introduction to Unit Testing
- ❖ Project Arch
- ❖ Unite test

**Module 6:- Model Binding**

This module contains the model binding examples that describes how a form bind to its corresponding action method with when the form is submitted.

- ❖ Html Form behavior
- ❖ Model Binder Overview
- ❖ DefaultModelBinder

**Module 7:- Validations & Data Annotations**

In this chapter you will learn Build application with proper error messages in minimum code.

- ❖ Data Annotations and Validations Overview
- ❖ Validations with Data Annotation
- ❖ Server Side and Client Side Validation

**Module 8:- State management Techniques**

In this chapter you will find how data can be maintained temporarily in different locations and used that for multiple requests.

- ❖ Sessions

**Module 9:- Security**

In this chapter you will learn how to implement security for a web application using ASP.NET Identities which is recommended by Microsoft for all web applications.

- ❖ Authentication and Authorization
- ❖ Implementing Security using ASP.NET Core Identity

**Module 10:- MVC and Entity Framework Core**

In this you will learn how perform simple database operations like add, update, delete and retrieve the data from SQL database using entity framework.

- ❖ Basic CRUD Operations using Entity Framework
- ❖ Separation of work using BO Classes
- ❖ Writing Generic Class / Repository

**Module 11:- ASP.NET Core - Web Caching**

In this chapter you will learn how to improve the performance of a web application by storing the frequently used data using different caching techniques.

- ❖ Cache Tag Helpers
- ❖ Memory Caching Introduction

**Module 12:- Routing**

In this chapter you learn how the URL is mapping to controller action methods for rendering the particular web page and the examples for customizing the URL.

- ❖ Url Routing Overview
- ❖ Custom Routes
- ❖ Attribute Routing

**Module 13:- Module Development**

In this chapter you will learn the concept of splitting a large web app into multiple modules so that maintains will be easy by using Areas.

- ❖ Understanding Areas
- ❖ Adding Areas
- ❖ Defining Area Routes
- ❖ Linking between Areas

**Module 14:- Web API and JQuery Ajax**

In this chapter you will learn how Web API works as web service which delivers json/xml data that can be used with JQuery Ajax client side calls for applying asynchronous behavior in any type of application.

- ❖ Introduction to Web API
- ❖ AJAX implementation using JQuery
- ❖ Creating a Web API that Supports CRUD Operations using EF

**Module 15:- Bundling & Minification**

In this module you will learn how Bundling and Minification is used for reducing the amount data transferred from the web app while rendering to improve the performance of your web app.

- ❖ What is Bundling and Minification in ASP.net Core?
- ❖ Bundler and Minifier Extension