# ASP.NET Core 7.0 Programming

*Start learning how to program ASP.NET MVC Core applications using the C# programming language and become a professionalfrom beginner/novice user.*

## About the Course & Importance of ASP.NET Core

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.

***Computer programming is really fun in general, and programming with C# Language using Visual Studio is even better & simpler!***

## Learning ASP.NET Core :

**There are 3 phases to learn ASP.NET Core 7.0.**

1. Introduction to ASP.NET Core Fundamentals
2. ASP.NET Core Middleware
3. ASP.NET Core 7.0 Advanced Topics

| | |
|---|---|
| Pre-Requisites:<br>C# basic Knowledge | <br>Duration : 45 days |
| Course Materials & Textbooks : 1. Running Notes in class, 2. Pdfs, 3. HandOuts, 4. Certification Material in Pdfs will be provided for this course. | |
| Reference Websites:www.dotnetgurukul.com | |
| Instructor: Praveen Kumar M<br>Learning from Praveen Sir, is always different and you will experience it within few sessions – you attend and work regularly as per given guidance then you will be best in any team that you work in any company – assured. | |

# ASP.NET Core Course Syllabus by Praveen

## Phase I – Introduction to ASP.NET Core

### ASP.NET Core Introduction

- Introduction & Evolution of ASP.NET Core (Brief history of ASP , ASP.NET , ASP.NET MVC)
- 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

(Introducing Markup langs .html , .xhtml ,.xml , aspx , .cshtml )

### ASP.NET Core First Application

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

### Controllers

- Controllers and Action Methods Overview
- Action Methods and IActionResult object
- Passing data from Controller to View
- Understanding Action Selectors
- Action Filters
- Building Custom Action Filters
- Middleware
- Asynchronous Action Methods

### Views

- Introducing Razor View
- Advantages of Razor View
- Razor Syntax
- Types of Views
- Partial Views
- Layout Pages
- Special Views
- View Categorization based on Model

### Model Binding

- Html Form behavior
- Model Binder Overview
- DefaultModelBinder
- Binding to Complex Classes
- IFormCollection Model Binding
- IFormFile Model Binder
- Bind Attribute
- TryUpdateModelAsync

# Phase 2 – ASP.NET Core Middleware

- **Helpers**
  - Html Helpers
  - Built-In Html Helpers
  - URL helpers
  - Tag Helpers
  - Custom Tag Helpers
- **Validations & Data Annotations**
  - Data Annotations and Validations Overview
  - Validations with Data Annotation
  - Server Side and Client-Side Validation
  - Custom Server-side validation
  - Model level validation using IValidatableObject
  - Custom unobstrive Client-side Validation
  - Remote Validation
- **State management Techniques**
  - Cookies
  - Sessions
- **Routing**
  - Url Routing Overview
  - Custom Routes
  - Attribute Routing
  - Routing Constraints
- **Security**
  - Authentication and Authorization
  - Implementing Security using ASP.NET Core Identity
- **MVC and Entity Framework Core**
  - Basic CRUD Operations using Entity Framework
  - Separation of work using BO Classes
  - Writing Generic Class / Repository
  - Caching in Repository
- **ASP.NET Core - Web Caching**
  - Cache Tag Helpers
  - Memory Caching Introduction
  - In-Memory Caching
  - Response Cache
  - Distributed Cache
- **Module Development**
  - Understanding Areas
  - Adding Areas
  - Defining Area Routes
  - Linking between Areas
- **Web API and JQuery Ajax**
  - Introduction to Web API
  - AJAX implementation using JQuery
  - Calling the Web API with JQuery Ajax
  - Creating a Web API that Supports CRUD Operations using EF
- **Bundling & Minification**
  - What is Bundling and Minification in ASP.net Core?
  - Bundler and Minifier Extension
  - How to Bundle your files
  - How to minify your Bundles
  - Convert to Gulp

# Phase 3 – Upgrading concepts of ASP.NET Core 7.0

## ASP.NET Core 7.0 Changes

- Merger of the Program and Startup classes
- Program.cs file changes

## ASP.NET Core MVC and Razor improvements

- Minimal APIs . Create a minimal web API with ASP.NET Core
- Minimal APIs quick reference
- Differences between minimal APIs and APIs with controllers
- Code samples migrated to the new minimal hosting model in 6.0

## ASP.NET Core performance and API improvements

- Non-allocating app.Use extension method.
- Reduced memory allocations when accessing HttpRequest.Cookies.
- Use LoggerMessage.Define for the windows only HTTP.sys web server
- Reduce the per connection overhead in SocketConnection by ~30%.
- Reduce allocations by removing logging delegates in generic types.

## HTTP Logging middleware

- Support for the HTTP Logging middleware has been introduced in ASP.NET Core 6.
- You can take advantage of this middleware in ASP.NET Core 6 to log information about HTTP requests and responses that include one or more of the following.
- Request information
- Response information
- Request and response headers
- Body of the request
- Properties