

ASP.NET Core 10.0 Programming

Start learning how to program ASP.NET MVC Core applications using the C# programming language and become a professional from 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 4 phases to learn ASP.NET Core 10.0.

1. Introduction to ASP.NET Core Fundamentals
2. Developing ASP.NET Core Applications
3. Middleware and Security for ASP.NET Core Applications
4. Advanced Features and Deployment

Pre-Requisites:
C# basic Knowledge
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
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.NETCore Course Syllabus by Praveen

Phase I – Introduction to ASP.NET Core - Available Free on YouTube

Topic	Key Concepts / Coverage	Hands-on
ASP.NET Core Introduction – Client–Server Model	Client-side: HTML, CSS, JavaScript; Server-side; IIS Express/IIS/Kestrel; HTTP.sys; launchSettings.json; Dev vs Prod Ports; HTTP Request/Response; GET/POST/PUT/DELETE; Browser Lifecycle; DevTools; Rendering Model; Controller→Model→View; Razor; HTML/JSON/XML Output	HTML, Razor & Request/Response Demo
Client-Side Technologies: HTML & JavaScript Basics	Web evolution; HTML & JavaScript; Client-side limitations; Static UI; Security; Debugging; No Strong Typing; Dynamic UI; Type Inconsistency; Need for Server-side , Transition to ASP.NET Core	JavaScript Demos
Server-Side Technologies Comparison	Classic ASP→ASP.NET→MVC→ASP.NET Core; Web Forms; Code-behind; Lifecycle; DAL/BAL; MVC; SoC; Web API; Cross-platform; Razor Pages; Minimal APIs	Hands-on Lab
ASP.NET Core Fundamentals	Framework; MVC+Razor+APIs; Project Structure; Program.cs; appsettings.json; Controllers; Models; Views; wwwroot; Pipeline; Dependency Injection , Service registration & usage	Project Structure & DI Demo
Routing, Controllers & IActionResult	Routing; MVC vs API; Conventional & Attribute Routing; REST; IActionResult; Ok/NotFound/BadRequest; Json/File/Redirect; ActionResult<T>	Hands-on Lab
Razor Pages Fundamentals	Razor Pages; MVC vs Page Model; Pages Folder; PageModel; Pipeline; DI	Hands-on Demo
Razor Pages Routing & Return Types	Routing; @page; Route Parameters; Query Strings; IActionResult; Page; Redirect; Json; File; Content	Hands-on Lab
ASP.NET Core Execution Pipeline	Middleware; app.Use/app.Run; Custom Middleware; UseRouting; UseEndpoints; MVC & Razor Execution Flow	Pipeline Demo

Phase II – Developing ASP.NET Core Applications Premium Phase

Topic	Key Concepts / Coverage	Hands-on
Designing Models & Relationships in ASP.NET Core MVC	EF Core overview, Navigation Properties, Relationship Types (1:1, 1:M, M:M), Designing Models from SQL Server Tables, Customer–Address, Customer–Orders & Order–Products Relationships, DbContext & DbSet Configuration, Controllers for Relational Data, Displaying Related Data in Views	Build relational models, configure EF Core relationships, create controllers & views, navigation flow demo
ViewModels, DTOs & AutoMapper	Purpose of DTOs & ViewModels, Domain Models vs DTOs vs ViewModels, CustomerRequestDto & CustomerResponseDto, UI-specific ViewModels, Data Shaping, AutoMapper Benefits, Mapping Profiles Configuration	Create DTOs, configure AutoMapper, perform end-to-end object mapping
Model Binding in ASP.NET Core	Model Binding Fundamentals, Binding Sources (Route, Query, Form, Body), [FromForm], [FromQuery], [FromBody], [FromRoute], Request Processing Flow	Implement model binding using multiple binding sources, real-world form & API examples
Filters in ASP.NET Core – Part 1	Filter Pipeline Overview, Authorization Filters, Global/Controller/Action Filters, Role-based Authorization, Resource Filters, Custom Resource Filters	Implement authorization & resource filters with practical demonstrations
Filters in ASP.NET Core – Part 2	Action Filters, Logging & Validation, Exception Filters, Centralized Error Handling, Result Filters, Response Modification	Build custom filters and integrate multiple filters into a complete application

Razor View Engine: Standard vs Strongly Typed Helpers	Razor Syntax Fundamentals, Razor Directives, HTML Helpers Overview, Standard HTML Helpers, Strongly Typed HTML Helpers, IntelliSense & Compile-time Safety	Create Razor views using standard and strongly typed HTML helpers
Digging into HTML Helpers (Forms & Controls)	Form Helpers, Input Controls (TextBoxFor, CheckBoxFor, etc.), Model Binding with Helpers, ViewBag & ViewData, Display Helpers, Editor Helpers	Build data entry forms and implement complete form submission workflow
Passing Data from Controller to View	Data Passing Techniques, Model-based Approach, ViewData vs ViewBag, TempData, Cross-request Data Flow	Implement different controller-to-view communication techniques
Tag Helpers in ASP.NET Razor Forms	Introduction to Tag Helpers, HTML Helpers vs Tag Helpers, Advantages, asp-for, asp-action, asp-controller, Built-in Tag Helpers	Develop forms using Razor Tag Helpers with model binding
Advanced Tag Helpers & Validation	Form Tag Helpers, Input Controls, TextArea, CheckBox, Radio Buttons, DropDown Lists, Data Annotations, Validation Rules, Validation Tag Helpers, Client-side Validation	Build a validated registration form with client-side validation
Navigation & Rendering Tag Helpers	Anchor Tag Helper, Routing, Query Strings, Page Handlers, Partial Tag Helper, Environment Tag Helper, Cache Tag Helper, Image Tag Helper	Create reusable UI components and implement navigation & caching
Layouts, ViewStart & ViewImports	Layout Pages, RenderBody, RenderSection, _ViewStart.cshtml, _ViewImports.cshtml, Namespace Management, Tag Helper Registration	Create consistent application layouts and shared Razor configurations

Phase 3: Middleware and Security for ASP.NET Core Applications Premium

Topic	Key Concepts / Coverage	Hands-on
Securing ASP.NET Core Applications using Microsoft Identity (Individual Accounts)	Authentication vs Authorization, Login Workflow, Authentication Options, Individual Accounts, ASP.NET Core Identity Packages, ApplicationDbContext, Identity Database Creation	Create MVC project with Individual Accounts, implement User Registration & Login
Securing Applications using Microsoft Identity Platform & Windows Authentication	Microsoft Identity Platform, Azure Active Directory (AAD), Tenant ID, Client ID, Client Secret, App Registration, Redirect URI, Callback Paths, Authentication Flow, Windows Authentication	Configure Azure AD authentication and Windows Authentication
Role & Policy Authorization with Middleware Introduction	Role-Based Authorization, Policy-Based Authorization, UserManager, RoleManager, Authorize Attribute, Middleware Fundamentals, Request Pipeline, HttpContext.Items, Authorization Middleware	Implement Admin vs Non-Admin access using Roles, Policies & Middleware
ASP.NET Core Identity – Default Authentication, Roles & Claims Caching	Identity Authentication Flow, SignInManager, UserManager, Claims-Based Authentication, Authentication Cookies, Session Behavior, Claims & Roles Caching, Default vs Manual Role Checks	Configure Identity authentication and inspect cached claims & roles
Securing Web APIs using JWT Authentication	Cookie vs Token Authentication, JWT Fundamentals, JWT Structure (Header, Payload, Signature), Token Generation, JWT Validation Middleware, Securing APIs with Authorize Attribute	Build Login API, generate JWT tokens, secure API endpoints using Postman
Implementing OAuth 2.0 Authentication in ASP.NET Core Web API	OAuth 2.0, OpenID Connect, Authorization Code Flow, Access Token vs ID Token, Google OAuth, Authorization & Token Endpoints, OAuth Scopes, ID Token Decoding, Google Cloud Console	Integrate Google OAuth 2.0 authentication into Web API
ASP.NET Core Middleware Pipeline – HTTPS & HSTS	Middleware Pipeline, Middleware Execution Order, HTTP vs HTTPS, HTTPS Redirection, HSTS Concepts, Strict-Transport-Security Headers	Configure HTTPS, HSTS and understand middleware execution flow
ASP.NET Core Middleware	Same-Origin Policy, CORS Fundamentals, CORS Policies,	Configure CORS policies and

Pipeline – CORS & Anti-Forgery	AllowAnyOrigin, AllowAnyMethod, AllowCredentials, CSRF Attacks, Anti-Forgery Tokens, ValidateAntiForgeryToken	implement Anti-Forgery protection
Custom Middleware in ASP.NET Core	Request Processing Pipeline, Inline Middleware, Class-Based Middleware, Reusable Middleware, Real-World Middleware Scenarios	Build custom inline and reusable middleware components
Introduction to State Management in ASP.NET Core	Stateless HTTP, State Management Fundamentals, Client-Side vs Server-Side State Management	Demonstrate various state management techniques
Client-Side State Management in ASP.NET Core	Cookies, Cookie Expiration & Persistence, Query Strings, Hidden Fields, Client-Side Data Storage	Implement Cookies, Query Strings and Hidden Fields in MVC applications
Server-Side State Management in ASP.NET Core	ViewData, ViewBag, TempData, Session State, Session Lifecycle, Application State, MemoryCache	Implement Session Management, TempData and Server-Side State Handling
Caching in ASP.NET Core	Caching Fundamentals, IMemoryCache, Output Caching, Distributed Caching, Redis Concepts, IDistributedCache	Implement Memory Cache, Output Cache and Distributed Caching demonstrations

Phase 4: Advanced Features and Deployment 📦 Premium Phase

Topic	Key Concepts / Coverage	Hands-on
Creating RESTful APIs in ASP.NET Core	Web API Fundamentals, REST vs Non-REST, REST Principles (Resources, HTTP Verbs, Statelessness), HTTP Status Codes, JSON Responses, RESTful API Design, Routing, Controllers, CRUD Operations using [ApiController], Swagger & Postman	Build RESTful CRUD APIs, document APIs with Swagger, test endpoints using Postman
API Versioning & JSON Formatting	API Versioning Fundamentals, URL-based Versioning, Header-based Versioning, Configuring API Versioning, JSON Serialization Options, camelCase, Null Handling, Indented Formatting, Custom Date Formatting	Implement API Versioning and customize JSON serialization settings
Consuming APIs (HttpClient & Fetch API)	API Consumer Architecture, Client-Server Communication, HttpClient in .NET, JSON Parsing, JavaScript Fetch API, Frontend Integration	Consume Web APIs using HttpClient and Fetch API in an end-to-end application
Real-Time Communication with SignalR	Traditional HTTP Limitations, Real-Time Communication, SignalR Architecture, WebSockets, SignalR Hub, Client-Server Messaging, Browser Integration	Build a real-time messaging application using SignalR
Exception Handling & Logging with ILogger and Serilog	Exception Handling Review, Production Logging, ILogger Interface, Log Levels (Information, Warning, Error, Critical), Serilog Integration, Console Logging, File Logging, Database Logging	Configure Serilog and implement centralized application logging
Unit Testing in ASP.NET Core (xUnit & NUnit)	Unit Testing Fundamentals, Automated Testing Benefits, xUnit vs NUnit, Assertions, AAA (Arrange-Act-Assert) Pattern, Test Execution	Write and execute unit tests using xUnit and NUnit
Mocking with Moq Framework	Mocking Fundamentals, Moq Framework, Mock Objects, Mock Interfaces, Fake Responses, Method Verification, Success & Failure Scenario Testing	Create mock dependencies and test services using the Moq framework

Frequently Asked Questions (FAQ)

1. How do I purchase a course?

Purchasing a course is simple.

1. Contact us through ****WhatsApp:9000366774**** or ****Email : contactdotnetgurukul@gmail.com****
2. We'll share the payment details (**UPI / Bank Transfer**) based on your preferred payment method.
3. Once the payment is completed, please share the payment confirmation (screenshot or transaction ID).
4. After **verifying your payment, we'll create your account (if required) and provide access to your purchased course through our learning portal (Learnyst).**
5. You can then log in to Learnyst and start learning immediately.

Our team typically provides course access within a few hours after payment verification. In most cases, access is granted on the same day.

2. Will I get support if I have technical questions or doubts?

Yes. We provide support for course-related technical questions and concept clarifications.

If you have any doubts **while learning, you can reach out to us via ****WhatsApp****, ****Email****, or ****Google Meet / Microsoft Teams****. We'll do our best to help you understand the concepts, resolve technical issues, and guide you in the right direction.**

Our goal is to ensure you have a smooth learning experience and successfully complete the course.