

# Developing ASP.NET Core 9 Web Applications

**Course Duration:** 48 Hours (6 Days)

## Course Overview

**Developing ASP.NET Core 9 Web Applications is an intensive, hands-on program designed to help developers master modern web application development using .NET 9, C# 12, and the ASP.NET Core 9 framework.**

This course equips participants with the knowledge and skills to **design, develop, test, secure, and deploy** scalable web applications and RESTful APIs using **MVC, Razor Pages, and Minimal APIs**.

## Pre-requisites

Participants should have:

- Prior experience with **C# programming** (preferably C# 10 or above).
- Basic understanding of **HTML, CSS, and JavaScript**.
- Familiarity with **Visual Studio 2022** or **VS Code**.
- (Optional) Exposure to **SQL Server** and **Git/GitHub** is recommended.

## Module 1: Exploring ASP.NET Core 9

- ASP.NET Core 9 architecture and request pipeline
- .NET 9 minimal hosting model updates
- Razor Pages vs MVC vs Minimal APIs
- Modern development tools and templates **Lab:** Build your first ASP.NET Core 9 web app

## Module 2: Application Design and Architecture

- MVC, Razor Pages, and Minimal APIs: choosing the right model
- Folder structure, environment configuration, and DI setup
- Environment variables, secrets, and configuration providers **Lab:** Design a multi-layered ASP.NET Core 9 project

## Module 3: Middleware and Services

- Creating custom middleware
- Dependency Injection and Service Lifetimes
- Logging and telemetry (OpenTelemetry integration in .NET 9) **Lab:** Build and register custom middleware for request tracking

## Module 4: Controllers and Routing

- Attribute routing and route groups
- Action filters and custom filters
- Minimal API routing vs MVC routing **Lab:** Implement controllers and routes for student operations

## Module 5: Razor Views and Tag Helpers

- Razor view syntax and model binding
- Partial views and layout views
- Tag Helpers and View Components **Lab:** Create a student dashboard using Razor views

## Module 6: Models, Validation, and Data Binding

- Model binding improvements in .NET 9
- Data annotations and Fluent Validation
- Handling forms and JSON payloads **Lab:** Add validation and input forms to your app

## Module 7: Data Access with EF Core 9

- EF Core 9 features (bulk operations, interceptors)
- Repository and Unit of Work patterns
- Database migrations and seeding **Lab:** Integrate EF Core 9 with MVC and perform CRUD operations

## Module 8: Testing, Logging, and Troubleshooting

- Unit testing with xUnit and Moq
- Integration testing in .NET 9
- Centralized logging and exception handling **Lab:** Write unit tests and implement logging middleware

## Module 9: Security in ASP.NET Core 9

- Authentication (Identity, OAuth2, and OpenID Connect)
- Authorization policies and role-based access
- Security headers, CSRF, and input sanitization **Lab:** Secure your application with ASP.NET Core Identity

## Module 10: Client-Side Integration and Blazor

- Blazor Server vs Blazor WebAssembly
- Razor Components in .NET 9
- Using Bootstrap 5 and modern front-end libraries **Lab:** Create a responsive UI using Razor components and Bootstrap

## Module 11: API Development with Minimal APIs

- Minimal API syntax and performance benefits
- Route groups and filters in .NET 9
- Versioning and Swagger documentation **Lab:** Create and document a RESTful API with Swagger UI

## Module 12: Real-Time Communication and Performance

- SignalR for real-time apps
- Caching with distributed memory and Redis
- Background services and IHostedService **Lab:** Implement live notifications using SignalR

## Module 13: Deployment and Cloud Integration

- Publishing apps to Azure App Service
- Using Docker with ASP.NET Core 9
- CI/CD pipelines with GitHub Actions / Azure DevOps **Lab:** Containerize and deploy your app to Azure

### Advance Topics(Quick examples(Overview))1 Day (No labs(only trainer examples))

- Overview of C# advanced features in .NET 9
- async and await in-depth
- Tasks, threads, and cancellation tokens
- Improved Threading Model
- Parallel Test Execution
- Native AOT (Ahead-of-Time Compilation):
- Meta programming in .NET Core 9
- Persisted Assembly Builder
- Enhanced Reflection APIs
- Optimized Garbage Collector