

**.NET 8: Modern Full-Stack Development**

**Course Number:** NET-326  
**Duration:** 2 days

**Overview**

This .NET 8 training course teaches attendees how to use Microsoft's .NET 8 platform as the centerpiece of a tech stack that leverages modern C# features and integrates with other open-source languages and tools. The second day of this .NET course is dedicated to building a "green field" full-stack application.

**Prerequisites**

Students must have experience with the C# programming language and object-oriented programming concepts and some knowledge of HTML, CSS, and JavaScript concepts.

**Materials**

All .NET 8 training students receive comprehensive courseware.

**Software Needed on Each Student PC**

* .NET 8.0 SDK
* Visual Studio, VS Code, or Rider
* Lab file bundle provided with the course

**Objectives**

* Master the fundamentals of .NET 8, including its evolution, SDKs, runtimes, editors, IDEs, and project types
* Develop web applications using ASP.NET MVC and Razor Pages, Minimal APIs, Blazor Server and WebAssembly, and class libraries
* Design and develop modern C# applications
* Implement unit testing and source code repositories in .NET 8 projects
* Build a complete RESTful API using Blazor and Single Pages Applications (SPAs)
* Apply containerization and Docker to deploy .NET applications

**Outline**

* Introduction
  + Evolution of .NET
  + .NET SDKs and Runtimes
  + Editors and IDEs
* Overview of Project Types
  + ASP.NET MVC and Razor Pages
  + Minimal APIs
  + Blazor Server and WebAssembly
  + Class Libraries and .NET Standard
  + .NET MAUI
  + Disadvantages of project templates
* Modern C#
  + C#'s evolution as a multi-paradigm language
  + Pillars of Functional Programming
  + Nullable Reference Types
  + Immutability
  + Record Types
  + Exceptions vs. Result
  + Designing for Concurrency
  + Deferred Execution
* Multi-Project Solutions
  + Source Code Repositories
  + Project References
  + Unit Testing
* Hands-On Case-Study Project (Day 2)
  + Overview of Requirements
  + Architectural Choices
  + Entities and Business Logic
  + Data Persistence, EF, and LINQ
  + Server Generated Content
  + JavaScript, TypeScript, npm, htmx, and Tailwind CSS
  + Building a RESTful API
  + Authentication and Bearer Tokens
  + Blazor and Single Pages Applications (SPAs)
  + C# on the Client
  + Blazor Render Modes in .NET 8
  + Containerization and Docker
* Conclusion