AZ-2002: Develop an ASP.NET Core web app that Consumes an API Course

Course Duration: 4 Hours **Course code: AZ-2002**

1. Course Overview

In AZ-2002, you'll build a real-world ASP.NET Core application that consumes one or more RESTful APIs. The course takes you step-by-step—designing the web app, integrating external data sources, handling JSON, managing authentication, and deploying to production. By the end, you'll understand how to build maintainable, scalable web applications that communicate with APIs.

2. What you'll learn?

By the end of this course, you will be able to:

- Design and scaffold ASP.NET Core web applications from scratch
- Consume RESTful APIs using HttpClient and modern typed clients
- Parse and map JSON data efficiently using System.Text.Json or Newtonsoft.Json
- Authenticate and authorize API calls using OAuth 2.0, JWT, or API keys
- Implement best practices: dependency injection, logging, retry policies, and error handling
- Secure sensitive data (e.g., secrets, tokens) with Azure Key Vault or User Secrets
- Deploy your web app to Azure App Service or Azure Container Instances

3. Target Audience

This course is ideal for:

- Intermediate .NET developers familiar with MVC/Razor pages
- Developers moving into full-stack roles who need to interact with APIs
- Engineers working with microservices or distributed systems
- Anyone preparing for ASP.NET Core certifications or modern web development

4. Pre-Requisites

Learners should have:

- Basic experience with C# and ASP.NET Core MVC or Razor Pages
- Familiarity with RESTful APIs and HTTP fundamentals (methods, status codes, headers)
- A working development machine with Visual Studio 2022+ or Visual Studio Code
- .NET 7 SDK (or higher) installed

5. Course content

Module 1: Introduction & Setup

- Course intro, learning objectives, and project scaffold
- Project structure overview and tooling

Module 2: API Fundamentals & HttpClient

- What is a REST API? Reviewing endpoints and JSON payloads
- Configuring HttpClient and consuming a public API
- Handling errors, timeouts, and retries

Module 3: JSON Handling & Data Mapping

- Parsing JSON using System.Text.Json
- Creating model classes and DTOs, using automapper

- Data validation and transformation

Module 4: Authentication & Authorization

- Integrating API keys and secure tokens
- Using JWT or OAuth 2.0 with external APIs
- Storing secrets securely

Module 5: Dependency Injection & Typed HttpClient

- Registering named and typed HttpClients
- Best practices for sustainability and testability
- Mocking HTTP responses for testing

Module 6: Caching & Performance Optimization

- Implementing in-memory or distributed caching
- Reducing API calls and improving response times

Module 7: Logging & Resilience

- Configuring logging middleware and capturing API errors
- Using Polly to implement retry and circuit breaker policies

Module 8: Front-End Views & UX

- Displaying API data in Razor views
- Building forms and filters
- Handling loading states and error messages

Module 9: Unit Testing & Integration Testing

- Writing unit tests for API services
- Testing controllers and integration scenarios
- Tools: xUnit, Moq, TestServer

Module 10: Deployment & CI/CD

- Deploying to Azure App Service or Containers
- Setting up GitHub Actions, Azure Pipelines, or Azure DevOps
- App settings, environment variables, and secrets in production