SYLLABUS

Introduction to the Specialization

This specialization is designed for teens who want to build a strong foundation in **backend development using .NET**.
By the end, students will understand .NET architecture also will gain hands-on experience with C# and API development using ASP.NET Core.

The specialization combines:

- **Theory + Hands-on practice**
- **Guided labs + “You Try It” coding tasks**
- **Monthly assessments**
- **One final backend project**

By the end, students will build a simple Web API, integrate Swagger, and use Microsoft Copilot to enhance code.

Graded Practice Quiz

Homework Tasks:

- 30 knowledge-based questions
- Assigned as at-home work
- Each student’s score is recorded and included in their monthly performance review

Guided Lab

Practice Session:

- Homework questions and tasks are reviewed in detail
- Instructor provides individual explanations to clarify difficult or unclear topics
- Students continue practicing until they fully understand the concepts

MODULE 1 — Backend development with .NET

In this module, you will explore the .NET ecosystem, understand its architecture, and learn how modern .NET differs from the legacy .NET Framework. You will gain hands-on experience with C#, .NET libraries, NuGet package management, and the overall workflow of back-end development using .NET. By the end of this module, you will understand the foundational building blocks required to develop scalable back-end.

Skills Gained: Understanding .NET architecture (Framework vs. .NET Core vs. Modern .NET), Cross-platform development with .NET, Using C# in the context of .NET, NuGet package management, Understanding DI, SOLID, and architectural patterns, Hands-on coding with .NET projects, Working with .NET libraries

TOPICS:

- **Introduction to Backend Development**
- **Tools for Backend Development**
- **AI Tool: Microsoft Copilot Overview**
- **Introduction to .NET and Its Architecture**
- **Evolution of the .NET Platform**
- **Cross-Platform Features of Modern .NET**
- **Modern .NET Runtime and Project Structure**
- **Hands-On Coding With C# in .NET**
- **NuGet and Package Management**
- **Implementation of .NET Libraries**
- **Benefits of Using .NET Core**
- **Concept of Dependency Injection (DI)**
- **Introduction to SOLID Principles**
- **Introduction to Backend Architectures**

MODULE 2 — Error Handling, Logging & Serialization

This module introduces essential reliability and data-processing concepts in .NET. You will learn how to implement proper error handling, create structured logs, and work with serialization/deserialization for data interchange. The module also covers security considerations and performance impacts of serialization.

Skills Gained: Eror handling in ASP.NET Core, Using logging frameworks, JSON/XML serialization & deserialization, Understanding serialization performance and security, Debugging using structured logs

TOPICS:

- **Error Handling in .NET & ASP.NET Core**
- **Logging Fundamentals in ASP.NET Core**
- **Logging Providers (Console, Debug, File, Serilog, Etc.)**
- **Introduction to Serialization & Deserialization**
- **Serialization Techniques in .NET**
- **Implementing JSON & XML Serialization**
- **Security Risks in Serialization**
- **Securing and Validating Serialized Data**

- **Performance Impact of Serialization**

MODULE 3 — Middleware in ASP.NET Core

This module focuses on the middleware pipeline—the foundation of every ASP.NET Core application. You will learn how middleware works, explore built-in components, and implement your own custom middleware for processing requests and responses.

Skills Gained: Understanding middleware architecture, Working with built-in middleware, Designing and implementing custom middleware, Securing middleware pipelines

TOPICS:

- **Introduction to Middleware in .NET Core**
- **Concept of Middleware Pipelines**
- **Built-In Middleware Components**
- **Creating Custom Middleware**
- **Implementing Middleware Components**
- **Designing Middleware Workflows**
- **Securing Middleware Layers**

MODULE 4 — Building Web APIs with ASP.NET Core

In this module, you will learn how to build RESTful Web APIs using ASP.NET Core. You will set up your API project, configure routing, implement CRUD operations, and integrate middleware to support a complete API workflow.

Skills Gained: Creating ASP.NET Core Web API Projects, Implementing RESTful API Endpoints, Attribute Routing & Conventional Routing, Building CRUD APIs, Middleware configuration for APIs

TOPICS:

- **Features of API Development in .NET**
- **Setting Up a Web API Project**
- **Controllers, Endpoints & HTTP Verbs**
- **CRUD Operations in Web API**
- **Routing Techniques (Attribute, Conventional, Map)**
- **Dependency Injection in API Projects**
- **Configuring Middleware for API Request Pipeline**

MODULE 5 — OpenAPI/Swagger & Microsoft Copilot for API Development

This module covers API documentation and AI-assisted development. You will integrate OpenAPI (Swagger) into your project and learn how to use Microsoft Copilot to generate code, enhance logic, and identify bugs.

Skills Gained: OpenAPI/Swagger integration, Auto-generating API documentation, Creating API clients from Swagger, Using Copilot for code generation & debugging, Enhancing API logic with AI

TOPICS:

- **Introduction to OpenAPI and Swagger**
- **Integrating Swagger Into ASP.NET Core**
- **Customizing Swagger Documentation**
- **Generating API Clients With Swagger**
- **Introduction to Microsoft Copilot**
- **Generating Code Snippets With Copilot**
- **Debugging API Code Using Copilot**
- **Enhancing API Code Quality With Copilot**