

## ***TƏDRİS PLANI (AZ)***

### **Programlaşdırmaya giriş**

Bu plan backend inkişafına **.NET** vasitəsilə başlamaq istəyən tələbələr üçün hazırlanmışdır. Programın sonunda tələbələr .NET memarlığını dərinlən anlayacaq, C# ilə praktiki şəkildə işləməyi öyrənəcək, ASP.NET Core ilə API-lərin hazırlanmasını mənimsəyəcək.

**Bu ixtisaslaşma aşağıdakıları birləşdirir:**

- **Nəzəriyyə + praktiki məşğələlər**
- **Təlimçi ilə laboratoriya (kod tapşırıqlarının izahı)**
- **Aylıq qiymətləndirmələr**
- **Yekun backend layihəsi**

Programın sonunda tələbələr sadə Web API yaradacaq, Swagger inteqrasiya edəcək və Microsoft Copilot vasitəsilə kodu təkmilləşdirəcəklər.

### **Qiymətləndirilən Praktiki Suallar**

- 30 test əsaslı suallar
- C# kodlaşdırma tapşırıqları (praktiki tətbiq)
- Hər bir tələbənin balı ayrılıqda qeyd olunur və aylıq performans hesabatına əlavə edilir.

### **Praktiki məşğələ**

**Tapşırıqların yoxlanması:**

- Ev tapşırıqları və suallar detallı şəkildə izah edilir
- Müəllim çətin və anlaşılmaz mövzuları ayrıca izah edir
- Tələbələr mövzunu tam başa düşənə qədər praktikanı davam etdirirlər

## ***MODULE 1 — .NET ilə Backend İnkişafı***

Bu modulda tələbələr .NET ekosistemini, onun strukturu və müasir .NET-in klassik .NET Framework-dən necə fərqləndiyini öyrənəcəklər. Tələbələr C#, .NET kitabxanaları, NuGet paketləri və backend inkişafının ümumi iş prinsipi ilə tanış olacaqlar.

*Qazanılan bacarıqlar:* .NET memarlığı (Framework vs .NET Core vs Modern .NET), Cross-platform inkişaf, C# ilə işləmək, NuGet paket idarəetməsi, DI & SOLID və arxitektura pattern-ləri, .NET layihələrində praktiki kodlaşdırma, .NET kitabxanaları ilə işləmə təcrübəsi.

## **MÖVZULAR:**

- Backend development-ə giriş
- Backend texnologiyaları
- AI texnologiya: Microsoft Copilot
- .NET strukturu və inkişaf mərhələləri
- .NET-in cross-platform xüsusiyyətləri
- Runtime və layihə strukturu
- .NET-də C# ilə kodlaşdırmanın əsasları
- NuGet paket istifadəsi
- .NET kitabxanalarının tətbiq olunması
- .NET Core istifadəsinin üstünlükləri
- DI (Dependency Injection) anlayışı
- SOLID prinsiplərinə giriş

- Backend arxitektura pattern-lərinə giriş

## **MODULE 2 — Error Handling, Logging & Serialization**

Bu modul .NET-də məlumatın emalı üçün vacib olan konsepsiyaları öyrədir. Tələbələr düzgün error handling tətbiqi, strukturlaşdırılmış loglama və JSON/XML serialization ilə işləməyi öyrənirlər.

**Qazanılan bacarıqlar:** ASP.NET Core-da error handling, Logging framework-lərin istifadəsi, JSON/XML serialization & deserialization, Serialization performans və təhlükəsizliyi, Structured logs ilə debugging

### **MÖVZULAR:**

- ASP.NET Core və .NET-də Error Handling
- Logging əsasları
- Logging provider-lər (Console, Debug, File, Serilog və s.)
- Serialization & Deserialization
- Serialization texnikaları
- JSON və XML tətbiqi
- Serialization riskləri və təhlükəsizlik
- Serialized məlumatların yoxlanması və qorunması
- Serialization performans

## **MODULE 3 — ASP.NET Core-də Middleware**

Bu modul ASP.NET Core tətbiqlərinin əsasını təşkil edən middleware pipeline-ını öyrədir. Tələbələr middleware-in necə işlədiyini, hazır komponentləri və öz custom middleware-lərini yaratmağı öyrənəcəklər.

**Qazanılan bacarıqlar:** Middleware strukturu, Built-in middleware-lərlə işləmə, Custom middleware dizaynı və tətbiqi, Middleware pipeline-ının təhlükəsizliyi,

### **MÖVZULAR:**

- Middleware nədir?
- Middleware pipeline
- Built-in middleware-lər
- Custom middleware yaradılması
- Middleware komponentlərinin tətbiqi

- Dizayn strukturu
- Middleware təhlükəsizlik qatları

## **MODULE 4 — ASP.NET Core ilə Web API-lərin Qurulması**

Bu modulda tələbələr RESTful Web API-lərin qurulmasını öyrənəcəklər. CRUD əməliyyatları, routing, controller-lər və middleware integrasiyası tətbiq olunacaq.

**Qazanılan bacarıqlar:** ASP.NET Core Web API yaratmaq, RESTful endpoint-lər, Attribute və Conventional routing, CRUD API-lər, Middleware ilə API pipeline konfigurasiyası

### **MÖVZULAR:**

- Web API-lərin xüsusiyyətləri
- Web API layihəsinin yaradılması və qurulması
- Controller-lər, Endpoint-lər və HTTP verb-lər
- Web API-də CRUD əməliyyatları
- Routing texnikaları (Attribute, Conventional, Map)
- Web API-də DI istifadəsi
- API pipeline (Middleware konfigurasiyası)

## **MODULE 5 — OpenAPI/Swagger və Microsoft Copilot ilə API İnkişafı**

Bu modulda API sənədləşdirilməsi və AI-assisted development öyrədilir. Tələbələr Swagger integrasiyası, API client yaradılması və Copilot vasitəsilə kod generasiyası ilə tanış olurlar.

**Qazanılan bacarıqlar:** OpenAPI/Swagger integrasiyası, Avtomatik API sənədləşdirilməsi, Swagger ilə API client yaradılması, Copilot ilə kod generasiyası və debugging, AI ilə API kodunun təkmilləşdirilməsi

### **MÖVZULAR:**

- OpenAPI və Swagger
- Swagger-in API-yə integrasiyası
- Sənədləşdirmə konfigurasiyası
- API Client yaradılması
- Microsoft Copilot istifadəsi
- Copilot ilə kod generasiyası

- Copilot ile API debugging
- Copilot vasiteleri kod keyfiyetinin artırılması

## ***SYLLABUS (EN)***

### **Introduction to the Specialization**

This specialization is designed for students 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 (coding tasks explanation)**
- **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
- Coding Exercises (Hands-On Implementation)
- 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
- .NET Architecture & Evolution
- Cross-platform features of .NET
- Runtime and project structure
- Basics of coding with C# in .NET
- NuGet package management
- Implementation of .NET Libraries
- Benefits of using .NET Core
- DI concepts
- Introduction to SOLID Principles
- Introduction to backend architectures patterns

## ***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:** Error 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

- Logging Providers (Console, Debug, File, Serilog, Etc.)
- Serialization & deserialization
- Serialization techniques
- Implementing JSON & XML
- Securing serialized data (risks)
- Securing & validating serialized data
- Performance considerations

## **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:**

- What is middleware?
- Middleware pipeline
- Middleware pipeline
- Creating custom middleware
- Implementing Middleware Components
- Designing flows
- 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:**

- Feature overview of Web APIs
- Creating & setting Web API project
- Controllers & HTTP verbs
- CRUD operations in Web API

- Routing Techniques (Attribute, Conventional, Map)
- DI in Web API
- API pipeline setup (Configuring Middleware)

## ***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:***

- OpenAPI & Swagger
- Integrating Swagger into an API
- Customizing documentation
- Generating API Clients
- Using Copilot
- Code generation with Copilot
- Debugging API logic using Copilot
- Enhancing API Code Quality With Copilot