

Full Stack .NET 10 + AI Engineering

Build full-stack applications with C#, ASP.NET Core and React, then add AI with Microsoft.Extensions.AI, Semantic Kernel, RAG, tool calling and agents.

DURATION	SESSIONS	TRAINING HOURS
16 weeks	80 one-hour sessions	80 hours
MODE	LEVEL	INCLUDES
Live online, instructor-led	Intermediate	Weekly labs, capstone project

What you will learn

- C# and .NET 10: object-oriented programming, LINQ and async programming
- SQL, Entity Framework Core and data modelling
- ASP.NET Core Web APIs: validation, caching, logging and testing with xUnit
- Security with JWT, ASP.NET Core Identity and policies
- React with TypeScript, including a streaming chat interface
- How large language models work: tokens, context and cost
- Prompt engineering and Microsoft.Extensions.AI
- Semantic Kernel: plugins, prompt templates and memory
- Embeddings, vector search and Retrieval-Augmented Generation (RAG)
- Tool calling, AI agents and the Model Context Protocol (MCP)
- Prompt-injection defence, evaluation and observability for AI features
- Docker, Azure deployment and CI/CD with GitHub Actions

Curriculum

Sixteen weekly modules of five one-hour sessions, each ending with a hands-on lab. The first nine modules build the .NET full stack: C#, SQL and EF Core, ASP.NET Core APIs, security and React. The next six cover AI engineering with .NET: Microsoft.Extensions.AI, Semantic Kernel, RAG, tool calling and agents, production AI and cloud deployment. The course ends with a capstone project.

Module 1 C# Foundations

The language and tools you will use every day.

- 1 Course orientation, the .NET 10 SDK, CLI and IDE setup
- 2 Types, variables, operators and control flow
- 3 Methods, parameters and return values
- 4 Arrays, strings and collections
- 5 Lab: C# fundamentals exercises

Module 2 Object-Oriented C#

Designing programs with classes and interfaces.

- 6 Classes, records and structs
- 7 Inheritance, interfaces and abstract classes
- 8 Generics
- 9 Exceptions and nullable reference types
- 10 Lab: build a small object-oriented application

Module 3 Modern C# and Async Programming

The features modern .NET code relies on.

- 11 LINQ for querying data
- 12 Delegates, lambdas and events
- 13 async/await and Tasks
- 14 Pattern matching and newer C# language features
- 15 Lab: LINQ and async exercises

Module 4 SQL and Data Access

Relational data and Entity Framework Core.

- 16 SQL fundamentals: tables, keys, CRUD and filtering
- 17 Joins, indexes and transactions
- 18 EF Core models and DbContext
- 19 Migrations and relationships
- 20 Lab: data model with EF Core

Module 5 ASP.NET Core Web API – Part 1

Building HTTP APIs the .NET way.

- 21 Hosting, dependency injection and configuration
- 22 Minimal APIs and controllers
- 23 Validation and error handling
- 24 API documentation with OpenAPI
- 25 Lab: build a CRUD Web API

Module 6 ASP.NET Core Web API – Part 2

Production-quality API design and testing.

- 26 EF Core in the API with service layers
- 27 Paging, filtering and caching
- 28 Logging and health checks
- 29 Unit and integration tests with xUnit
- 30 Lab: harden and test the API

Module 7 Security

Protecting APIs and users.

- 31 Authentication and authorization concepts
- 32 JWT and refresh tokens
- 33 ASP.NET Core Identity
- 34 Roles, policies, CORS and OWASP basics
- 35 Lab: secure the API

Module 8 React – Part 1

A modern front end in React with TypeScript.

- 36 JavaScript and TypeScript essentials
- 37 Components, props and state
- 38 Hooks and effects
- 39 Forms and validation
- 40 Lab: React screens for the application

Module 9 React – Part 2 and Full Stack

Connecting the React front end to the .NET API.

- 41 Routing
- 42 Calling the .NET API and the sign-in flow
- 43 State management
- 44 Building a streaming chat interface
- 45 Lab: full-stack integration

Module 10 AI Fundamentals for .NET

How large language models work and how to use them from C#.

- 46 How LLMs work: tokens, context windows and cost
- 47 Prompt engineering patterns
- 48 Microsoft.Extensions.AI and the IChatClient abstraction
- 49 Streaming and structured output
- 50 Lab: first AI features in the API

Module 11 Semantic Kernel

Orchestrating AI features with Semantic Kernel.

- 51 Kernel, plugins and functions
- 52 Prompt templates
- 53 Chat history and memory
- 54 Azure OpenAI and other model providers
- 55 Lab: AI assistant with Semantic Kernel

Module 12 Embeddings and RAG

Answers grounded in your own documents and data.

- 56 Embeddings and vector search
- 57 Vector stores: pgvector and Azure AI Search
- 58 Document ingestion and chunking
- 59 Building a RAG API with citations
- 60 Lab: document question-answering

Module 13 Tool Calling and AI Agents

Models that act through your code and data.

- 61 Function and tool calling
- 62 Building agents in .NET
- 63 Model Context Protocol (MCP) servers and clients in C#
- 64 Multi-step workflows and guardrails
- 65 Lab: an agent that uses your APIs

Module 14 Production AI

Running AI features reliably, safely and affordably.

- 66 Cost, latency and caching
- 67 Prompt injection and protecting personal data
- 68 Testing and evaluating AI features
- 69 Observability with OpenTelemetry
- 70 Lab: production-ready AI features

Module 15 Cloud and DevOps

Shipping the application.

- 71 Docker for .NET and React
- 72 Deploying to Azure App Service and Container Apps
- 73 CI/CD with GitHub Actions
- 74 Configuration and secrets management
- 75 Lab: deploy the full-stack application

Module 16 Capstone Project

Design, build and deploy your own full-stack AI application.

- 76 Capstone design: requirements and architecture
- 77 Capstone build: backend and data
- 78 Capstone build: AI features
- 79 Capstone build: front end and deployment
- 80 Capstone demo and review

How you learn

- Live coding in every session
- A weekly lab at the end of each module
- One application built up step by step through the course
- Code reviews
- A capstone project you deploy and present

Projects

- **AI-Powered Web API:** An ASP.NET Core API with chat and structured-output features built on Microsoft.Extensions.AI.
- **Document Question-Answering (RAG):** Answers questions over your own documents with citations, using embeddings and vector search.
- **Capstone: Full-Stack AI Application:** A deployed .NET + React application with AI features of your choice, presented in a final demo.

Who should enrol

- BE / B.Tech, BCA, MCA and BSc Computer Science / IT students
- Final-year students and recent graduates
- .NET developers who want to add AI skills
- Anyone aiming for a backend, full-stack or AI engineering role on .NET

Prerequisites

Basic programming knowledge in any language.
C# is taught from the fundamentals.

Course completion

Learners who complete the training, the weekly labs and the capstone project receive a course completion certificate through GeeksArray, and finish with a deployed full-stack AI application for their portfolio.

Questions about batch dates, timings, fees or anything else?

Contact the course coordinator, **Kaumodini Kedar** · WhatsApp **+91 91726 23305**

Or see the course page on geeksarray.com.