CodeMug.in

CAREER-FOCUSED TECHNOLOGY PROGRAMS

AI Full Stack Developer Roadmap 2027

.NET Specialization | Full Stack • Cloud • Generative AI • Agentic AI

15 LEARNING STAGES	MODERN .NET STACK	AI ENGINEERING
C# to production AI applications	.NET 10 • Angular/React • Azure	LLMs • RAG • Agents • MCP

Program Overview

A structured learning roadmap across the modern .NET ecosystem. Progress from C# and frontend development to enterprise APIs, databases, architecture, microservices, Azure cloud, Generative AI, RAG, AI agents and MCP. The curriculum connects software engineering foundations with practical AI application development.

Who Should Learn

Aspiring software developers, .NET developers, backend/full-stack developers and working professionals interested in cloud-connected and AI-powered applications. Basic programming knowledge is helpful; consistent coding practice and project work are expected.

Complete Curriculum

Stage 01: Advanced C# Programming

- C# fundamentals, data types, operators, control flow, methods and exception handling
- OOP: classes, inheritance, polymorphism, abstraction and interfaces
- Collections, generics, delegates, events, LINQ and extension methods
- Async/await, Task Parallel Library, multithreading and concurrency
- SOLID principles, design patterns and Git

Stage 02: Modern Frontend Development

- HTML5, CSS3, responsive design, JavaScript and TypeScript
- Angular or React fundamentals, components and application structure
- Routing, forms, validation and state-management fundamentals
- RxJS and observables (Angular) or hooks (React)
- REST API integration with .NET applications

Stage 03: ASP.NET Core Web API

- .NET 10 and ASP.NET Core fundamentals
- Middleware pipeline, dependency injection and configuration
- REST API design, controllers, routing, DTOs and validation
- JWT authentication and role/policy-based authorization
- Swagger/OpenAPI, API versioning, rate limiting and idempotency

Stage 04: Database & Data Access

- SQL Server: database design, joins, indexes, views, stored procedures and transactions
- Entity Framework Core: DbContext, migrations, relationships and LINQ
- Dapper for lightweight data access
- Redis fundamentals and caching strategies

Stage 05: Software Architecture & Design

- Layered, Clean and Onion Architecture
- SOLID principles and dependency inversion
- Repository and Unit of Work patterns
- CQRS and MediatR fundamentals
- DDD concepts and Modular Monolith architecture

Stage 06: Microservices & API Gateway

- Microservices principles, service boundaries and communication
- REST and gRPC
- API Gateway with YARP and Ocelot
- Service discovery, load balancing and health checks

Stage 07: Distributed Systems & Messaging

- Event-driven architecture and asynchronous communication
- RabbitMQ and Apache Kafka
- Eventual consistency and Saga pattern
- Outbox pattern, Dead Letter Queues and message reliability
- Polly resilience: retry, circuit breaker and idempotency

Stage 08: DevOps, Containers & CI/CD

- Docker images, containers, volumes and networking
- Kubernetes: pods, deployments, services and configuration
- CI/CD using GitHub Actions and Jenkins
- Infrastructure as Code fundamentals with Terraform

Stage 09: Microsoft Azure Cloud

- Azure App Service, Azure SQL and Azure Functions
- Azure Storage, Key Vault and Service Bus
- Azure Container Apps and Azure Kubernetes Service
- Azure AI Services fundamentals
- Cloud deployment, monitoring and logging

Stage 10: Python & Machine Learning Foundations

- Python fundamentals for AI and data workflows
- AI/ML concepts and data preparation
- NumPy and Pandas fundamentals
- Supervised and unsupervised learning
- Neural networks and Transformer fundamentals
- ML.NET and ONNX model integration

Stage 11: Generative AI & Large Language Models

- LLM ecosystem: OpenAI, Azure OpenAI, Gemini, Claude and Hugging Face
- Running local models with Ollama
- Prompt engineering and structured outputs
- Function calling and tool-calling concepts

Stage 12: AI Integration with .NET

- OpenAI SDK and Microsoft.Extensions.AI
- IChatClient abstractions and provider integration
- Semantic Kernel fundamentals
- Streaming responses and prompt templates
- Function invocation, token usage and cost management

Stage 13: Embeddings, Vector Databases & RAG

• Embeddings and semantic representation
• ChromaDB, Qdrant, Pinecone, Azure AI Search and pgvector
• Document ingestion, chunking and indexing
• Semantic search, hybrid search and reranking
• RAG evaluation and Chat with PDF application

Stage 14: Agentic AI, AI Agents & MCP

• Agent architecture, planning, tool calling and memory
• Multi-agent systems and orchestration
• Microsoft Agent Framework fundamentals
• MCP host, client and server concepts
• MCP tools, resources, prompts and building a C# MCP server
• Agent and tool security fundamentals

Stage 15: Production AI, Security, Testing & Capstone Projects

• Production readiness, security and responsible AI
• Testing, observability, logging and monitoring for AI applications
• Deployment and operations for AI-powered systems
• Projects: AI Chatbot, RAG PDF Assistant and AI Recruitment Platform
• SQL Assistant, Customer Support Agent and Multi-Agent Business Automation
• AI SaaS application planning and implementation

Hands-on Project Direction

Paragraph('caseSensitive': 1 'encoding': 'utf8' 'text': '**Project**' 'frags': [ParaFrag(__tag__='b', bold=1, fontName='Helvetica-Bold', fontSize=8.2, greek=0, italic=0, link=[], rise=0, text='Project', textColor=Color(.090196,.12549,.2,1), us_lines=[])] 'style': 'bulletText': None 'debug': 0) #Paragraph	Paragraph('caseSensitive': 1 'encoding': 'utf8' 'text': '**Learning focus**' 'frags': [ParaFrag(__tag__='b', bold=1, fontName='Helvetica-Bold', fontSize=8.2, greek=0, italic=0, link=[], rise=0, text='Learning focus', textColor=Color(.090196,.12549,.2,1), us_lines=[])] 'style': 'bulletText': None 'debug': 0) #Paragraph
AI Chatbot	Conversational application integrated with an LLM.
RAG PDF Assistant	Document upload and context-aware question answering.
AI Recruitment Platform	AI-assisted recruitment workflows.
SQL Assistant	Natural-language interface for database questions.
Customer Support Agent	Agent that uses tools to support workflows.
Multi-Agent Business Automation	Coordinate multiple agents for a business process.

Tools & Technologies

C# • .NET 10 • ASP.NET Core Web API • Angular / React • TypeScript • SQL Server • EF Core • Dapper • Redis • Clean Architecture • CQRS • MediatR • YARP • Ocelot • gRPC • RabbitMQ • Kafka • Docker • Kubernetes • GitHub Actions • Jenkins • Terraform • Microsoft Azure • Python • ML.NET • ONNX • OpenAI • Azure OpenAI • Semantic Kernel • Vector Databases • RAG • AI Agents • MCP

Learning Outcomes

• Develop full-stack applications using modern .NET and a frontend framework.
• Design secure APIs and data-driven applications using enterprise architecture practices.
• Understand microservices, messaging, containers, CI/CD and Azure deployment.
• Integrate LLMs into .NET applications and build RAG-based solutions.
• Develop AI agents and connect tools through MCP concepts.
• Build portfolio-ready projects demonstrating practical software engineering skills.

Important Note
This syllabus is a learning roadmap. Sequence, depth, tools and project scope may be adapted to batch requirements, learner progress and technology updates. Course completion does not automatically award third-party certifications or guarantee employment or a specific salary.