

.NET AI Agents en Apps

h3>Doelgroep Cursus Building .NET Agents and Apps

De cursus is bedoeld voor .NET-softwareontwikkelaars die AI-functionaliteit willen integreren in hun applicaties. Zowel backend- als full-stack developers, AI-engineers en softwarearchitecten profiteren van deze training.

Voorkennis Cursus Building .NET Agents and Apps

Deelnemers dienen ervaring te hebben met C# en .NET-ontwikkeling. Basiskennis van AI, machine learning en REST API's is aanbevolen.

Uitvoering Training Building .NET Agents and Apps

De cursus wordt verzorgd door een trainer en combineert theorie met praktische oefeningen. Voorbeelden en use-cases uit de praktijk ondersteunen het leerproces.

Certificaat Building .NET Agents and Apps

Na succesvolle afronding ontvangen deelnemers een certificaat van deelname aan de cursus Building .NET Agents and Apps.

Duur: 3 dagen

Prijs: € 2499

[Open Rooster](#)



Inhoud Cursus .NET AI Agents en Apps

De cursus Building .NET Agents and Apps leert je hoe je AI-gedreven applicaties en agents ontwikkelt met moderne .NET-tools en -platformen. Je leert AI integreren in .NET 8, werkt met ML.NET, Azure AI en OpenAI, en bouwt slimme agents met Microsofts Semantic Kernel. Na afloop ben je in staat om intelligente full-stack .NET-apps te bouwen met machine learning en natural language features.

Introductie

De cursus start met een overzicht van AI in .NET, inclusief ML.NET, OpenAI, Azure AI en ONNX. Deelnemers leren de basisbegrippen en hoe de AI-stack in .NET is opgebouwd.

Slimme .NET-apps met ML.NET

Deze module laat je ML.NET gebruiken voor het bouwen van sentimentanalysemodellen, pipelines, en integratie in ASP.NET-apps. Je leert hoe je modellen traint, evalueert en inzet.

OpenAI en Azure AI in .NET

Leer hoe je GPT-modellen integreert in C#, inclusief authenticatie, streamingoutput, kostenbeheer en cognitieve services zoals vertaling, spraak en beeldherkenning.

AI-agenten en tooling in .NET

Ontdek hoe je AI-agenten bouwt met Semantic Kernel, planningsstrategieën definieert, geheugen beheert en agents koppelt aan externe APIs zoals kalenders of weerdiensten.

Natuurlijke taal en prompts

Leer effectieve prompts ontwerpen in Semantic Kernel, semantisch zoeken implementeren, vector stores integreren en een intelligente Q&A-app opzetten met RAG.

AI-applicaties bouwen en deployen

De slotmodule behandelt hoe je complete AI-apps bouwt met Blazor of ASP.NET, ze host op Azure, SignalR gebruikt, caching beheert en de AI logica verpakt als NuGet-pakket.

Modules Cursus .NET AI Agents en Apps

Module 1: Intro to AI and .NET	Module 2: Smart .NET Apps with ML.NET	Module 3: OpenAI & Azure AI
Overview of AI/ML in .NET Key concepts: Models, Inference, Agents ML.NET, Azure AI, ONNX LLMs and modern app development Cognitive services and APIs Cloud vs local AI models .NET 8 AI features Setting up environment Copilot in .NET productivity Use cases in .NET AI	Intro to ML.NET Building a sentiment model Data processing & features Using Model Builder Saving/loading models Evaluation & tuning ML in ASP.NET Deploying prediction APIs Production model usage Model integration patterns	OpenAI API in .NET Azure OpenAI differences API key setup & auth First GPT request in C# Creating chat assistant Token cost management Streaming responses Vision/Speech/Language APIs Azure Translator in apps Building Azure + OpenAI bots
Module 4: AI Agents & Tooling in .NET	Module 5: Prompt Engineering & NL Interfaces	Module 6: Deploying .NET AI Apps
What is an AI Agent? LangChain vs Semantic Kernel Semantic Kernel SDK intro First AI agent in .NET Plugins & skills in SK Planning strategies Memory and context External API integration Logging and debugging Use case: Calendar agent	Prompt engineering basics Templated prompts in SK Chaining prompts Managing conversation history Prompt tips for .NET devs Vector search with embeddings Using Pinecone, Redis, AI Search RAG pattern implementation Use case: Doc Q&A system Fallback and hallucination handling	Building full-stack AI apps Blazor vs ASP.NET AI UIs Secure HTTP APIs SignalR for live AI updates Caching/throttling responses Long-running workflow handling Azure App Service & Containers Testing strategies & logging NuGet packaging of agents Final project: AI assistant demo