

AI Agent Development

Course Duration ~90 Hours · 12 Weeks · Live Online · Project-First · Capstone Reviewed 1:1

[LangGraph](#)[MCP](#)[Multi-Agent](#)[Voice & Browser Agents](#)[Evals](#)[Deployment](#)

In short: This is a live, project-first deep-dive into building autonomous AI agents that plan, use tools, remember, collaborate, and run unattended in production — not demos that break on the second turn. Across 10 modules you ship tool-calling agents with validated outputs, custom MCP servers, durable LangGraph systems with human-in-the-loop gates, supervisor-led multi-agent pipelines, text-to-SQL, browser and realtime voice agents, and a full eval-and-observability harness — closing with a production-grade multi-agent capstone you design, build, evaluate, deploy and defend in a 1:1 review. Built for fresh IT graduates targeting an AI-first first role and for working professionals upgrading engineering or automation skills into agentic AI. Pairs with our Generative AI & LLM Engineering course: that one is the broad foundation (models, prompting, RAG, fine-tuning, LLMOps); this one is the specialist deep-dive into autonomous systems.

Duration	12 weeks · ~90 hours · live + hands-on	Format	Live online · project-first · capstone reviewed 1:1
Prerequisite	Working Python and basic familiarity with any LLM API. No ML/maths background required.	Curriculum	10 modules, from agent design patterns to deployment and capstone
Portfolio	7 production-style agent projects plus a defended capstone	Ideal for	Fresh IT graduates and working professionals moving into agentic AI

Course Highlights

- Every module leads with what you build, not what you "learn about"
- Graph-based agents with LangGraph: durable execution, interrupts, human-in-the-loop
- Model Context Protocol (MCP) servers and clients as the standard tool interface
- Multi-agent patterns — and the failure modes nobody warns you about
- Specialized agents: retrieval, text-to-SQL, browser/computer-use, voice, background
- Reliability engineering: trajectory evals, tracing, guardrails, OWASP LLM Top 10
- Framework fluency without lock-in: OpenAI Agents SDK, CrewAI, AutoGen and more

Hands-on Projects & Portfolio

- **Customer Support Agent** — RAG-backed knowledge, issue classification and automatic escalation to a human
- **Custom MCP Server** — internal tools and databases exposed to a customer-facing agent through one standard interface
- **Multi-Agent Research System** — parallel specialist agents coordinated by a supervisor into one research-and-report pipeline
- **Text-to-SQL Database Agent** — answers business questions conversationally against a real database
- **Browser-Operating Agent** — completes a real multi-step task end-to-end by operating a live browser
- **Voice Agent** — realtime speech with tool access and interruption handling
- **Agent Eval Harness** — trajectory scoring, nightly regression runs and a cost dashboard

Tools & Stack

Python · LangChain · LangGraph · LangSmith · OpenAI Agents SDK · CrewAI · AutoGen · LlamaIndex · MCP · A2A · Anthropic Claude · OpenAI · Gemini · Pydantic · pgvector · Qdrant · Mem0 · Langfuse · E2B · Playwright · FastAPI · Docker · Redis · Temporal

Career Outcomes

Agentic systems are where hiring demand has moved fastest, and agent orchestration plus eval design are now the strongest differentiators on an AI engineering resume. This course targets the roles at the centre of that shift:

- AI Agent Engineer · Agentic AI Engineer · Applied AI Engineer · AI Automation Engineer
- AI Solutions Architect · LLM Engineer · AI Platform Engineer

Who This Course Is For

- Final-year students and fresh IT graduates targeting an AI-first first role
- Python, backend and full-stack developers who can call an LLM API but can't yet ship a system that handles errors, scales, and survives production
- Data scientists and ML engineers moving from models into autonomous systems
- Automation, RPA and integration engineers upgrading rule-based workflows to agentic ones
- Tech leads and architects who must make build-vs-buy calls on agent platforms

Module 1: Agent Foundations & Design Patterns

8 Hours

YOU BUILD

- The same task three ways — prompt chain, workflow, and autonomous agent — with a decision memo on which one production actually needs

TOPICS

- What actually makes a system "agentic"; the spectrum from prompt chain to workflow to autonomous agent
- ReAct, plan-and-execute, reflection and self-critique loops, router patterns
- Critically: when *not* to use an agent

Assessment: Quiz + Lab

Module 2: Tool Use & Structured Outputs

8 Hours

YOU BUILD

- A tool-calling agent with Pydantic-validated outputs, parallel tool calls, retries, and sandboxed code execution via E2B/Modal

TOPICS

- Function/tool calling; JSON schemas and Pydantic validation
- Tool design that models can actually use; error handling and retries; parallel tool calls
- Sandboxed code execution (E2B / Modal)

Assessment: Quiz + Lab

Module 3: Model Context Protocol (MCP)

8 Hours

YOU BUILD

- A custom MCP server exposing internal systems and a database as agent tools, with auth, plus a client that consumes third-party servers

TOPICS

- Building MCP servers and clients; transport options; auth
- Exposing internal systems and databases as agent tools
- Replacing bespoke integration layers with one standard interface; publishing and consuming third-party servers

Assessment: Quiz + Lab

Module 4: Agent Memory & State

8 Hours

YOU BUILD

- An agent with checkpointed state and long-term semantic memory that survives restarts and remembers users across sessions

TOPICS

- Short-term vs long-term memory; conversation state; TypedDict and Pydantic state schemas; reducers
- Checkpointing and persistence; semantic memory with vector stores; Mem0
- Context-window management and compaction

Assessment: Quiz + Lab

Module 5: Building with LangGraph

10 Hours

YOU BUILD

- A durable graph-based agent with conditional routing, streaming, human-in-the-loop approval gates, and time-travel debugging

TOPICS

- Graph-based agents with nodes, edges and conditional routing; subgraphs
- Durable execution; streaming; interrupts
- Human-in-the-loop approval gates; time-travel debugging

Assessment: Quiz + Lab

Module 6: Multi-Agent Systems

10 Hours

YOU BUILD

- A supervisor-led research-and-report system with parallel specialist agents, shared state, and quality loops that actually converge

TOPICS

- Supervisor and swarm/decentralized patterns; task delegation; message passing; shared state
- Agent-to-agent (A2A) communication; convergence and quality loops
- The failure modes nobody warns you about: infinite loops, cost blowouts, responsibility diffusion

Assessment: Quiz + Lab

Module 7: Framework Landscape

6 Hours

YOU BUILD

- One agent ported across two frameworks, plus a build-vs-buy evaluation matrix your team can reuse

TOPICS

- OpenAI Agents SDK, CrewAI, AutoGen/AG2, LlamaIndex Workflows, Google ADK, AWS Strands
- What each is good at; how to choose
- How to avoid framework lock-in

Assessment: Quiz + Lab

Module 8: Specialized Agents

12 Hours

YOU BUILD

- Four working agents: a retrieval agent over enterprise knowledge, a text-to-SQL agent, a browser-operating agent, and a realtime voice agent with barge-in

TOPICS

- Retrieval agents over enterprise knowledge; text-to-SQL database agents
- Browser and computer-use agents
- Voice agents: STT/TTS, realtime APIs, barge-in and latency budgets
- Scheduled and event-driven background agents

Assessment: Quiz + Lab

Module 9: Reliability — Evaluation, Observability & Security

10 Hours

YOU BUILD

- An agent eval harness with trajectory scoring, a calibrated LLM-as-judge, CI regression runs, full tracing, and a hardened guardrail layer

TOPICS

- Trajectory and task-completion evals; LLM-as-judge with calibration; regression suites in CI

- Tracing with LangSmith / Langfuse / Phoenix; cost and latency dashboards
- Guardrails; prompt injection and indirect injection defense; OWASP LLM Top 10; least-privilege tool permissions; audit trails

Assessment: Quiz + Lab

Module 10: Deployment & Capstone

10 Hours

YOU BUILD

- One production-grade multi-agent system, end-to-end: designed, built, evaluated, deployed — and defended in a 1:1 review

TOPICS

- FastAPI serving; containerization; durable long-running orchestration
- Concurrency and rate limits; cost control at scale; versioning agents safely
- Capstone design review, deployment, and 1:1 defense

Assessment: Capstone Project + 1:1 Review