

DATA STREET ACADEMY

Generative AI & Agentic AI Course Curriculum

The image shows a promotional graphic for the 'Generative AI & Agentic AI' course. On the left, a blue background features the Data Street Academy logo (a white square with a blue circle containing three white chevrons) and the text 'DATA STREET ACADEMY' in white. Below this, it says '50-Hour Professional Certification Course'. Further down, a green circle contains '50 HRS', and three smaller circles contain '10 Projects', 'Capstone', and 'Job-Ready'. On the right, a light blue background with a dotted pattern features the title 'Generative AI & Agentic AI' in large blue font. Below the title, it says 'From LLM Foundations to Production-Grade Autonomous Agents'. Three light blue boxes list the course components: 'Hands-On Engineering' (Every concept built in code — no passive watching), 'Agentic AI Mastery' (LangGraph · CrewAI · MCP · A2A protocols), and 'Production Deployment' (Docker · FastAPI · Cloud AgentOps · Monitoring).

DATA STREET ACADEMY
50-Hour Professional Certification Course

50 HRS

10 Projects Capstone Job-Ready

Generative AI & Agentic AI

From LLM Foundations to Production-Grade Autonomous Agents

- Hands-On Engineering**
Every concept built in code — no passive watching
- Agentic AI Mastery**
LangGraph · CrewAI · MCP · A2A protocols
- Production Deployment**
Docker · FastAPI · Cloud AgentOps · Monitoring

01 Why This Course, Why Now

The technology job market has split software professionals into two groups: those who can build with AI, and those who use it casually. This course is designed to move you decisively into the first group. It compresses what the industry has learned over the last three years of building LLM applications into a 50-hour, hands-on engineering program.

Market Data & Opportunity



Most courses teach impressive demos. Hiring managers repeatedly reject candidates whose portfolios contain only notebook demos with no error handling, no evaluation, and no deployment story. This course is deliberately engineered so you finish with production-quality, portfolio-ready systems and the architectural judgment to defend

your design decisions in interviews.

02 Who This Course Is For

Ideal Participants

- Software engineers, full-stack and backend developers who want to move into AI engineering roles.
- Solution architects and technical leads who must design and review GenAI and Agentic systems.
- Data engineers, QA/automation engineers, and DevOps/SRE professionals expanding into the AI stack.
- Engineering managers and product-minded technologists who need hands-on credibility with AI systems.

Prerequisites: Working knowledge of Python (reading and writing scripts, functions, classes) and familiarity with APIs and JSON. No prior machine learning, data science, or mathematics background is required. This course is GenAI-native by design.

03 What You Will Be Able to Do

- Design, build, and ship complete Generative AI applications using OpenAI, Anthropic Claude, and Google Gemini APIs, plus open-weight models.
- Build production-quality RAG systems with hybrid retrieval, reranking, citations, and measurable accuracy.
- Architect autonomous and multi-agent systems using the four core agentic design patterns: reflection, tool use, planning, and multi-agent collaboration.
- Work hands-on with frameworks employers ask for: LangChain/LangGraph, CrewAI, OpenAI Agents SDK, Agno, and Google ADK.
- Build and consume MCP (Model Context Protocol) servers and understand A2A for agent-to-agent interoperability.
- Engineer agent memory and context, evaluate agents with LLM-as-a-judge, trace with LangSmith/Langfuse, and secure against prompt injection.
- Use Agentic coding tools such as Claude Code — the new baseline workflow for software delivery.
- Deploy agents to the cloud with Docker and FastAPI, control costs, and operate with monitoring and human-in-the-loop controls.
- Walk into interviews with 10 in-course projects plus a deployed capstone, and a clear narrative for your transition into AI engineering.

04 Course Structure at a Glance — 50 Hours

The course is organized into four sections that mirror how AI engineering work progresses in industry: first you learn to harness models, then to give them knowledge, then to give them autonomy, and finally to make them reliable enough for production.

Section	Name	Focus	Modules	Hours
A	GenAI Foundations	Models, prompting, RAG, fine-tuning, first applications	Modules 1–5	15
B	Agentic AI Core	Agent design patterns, frameworks, MCP/A2A, memory	Modules 6–9	16
C	Enterprise Ops	Agentic RAG, evaluation, security, deployment, frontier	Modules 10–14	15
D	Capstone Studio	Guided capstone build, architecture reviews, demo day	Capstone	4

Course Rhythm: Every module follows the same rhythm: concept (the minimum theory you need) → live build (instructor codes the pattern end to end) → lab (you practice it immediately) → debrief (where this pattern is used)

in industry, what breaks in production, and what interviewers ask about it).

05 Project Roadmap

This curriculum uses a three-layer project model refined from analysis of the strongest programs in the market and what hiring managers respond to.

- Layer 1 — In-module labs: Every module includes a short guided lab so no concept stays theoretical.
- Layer 2 — Real-world projects: 10 real-world projects placed directly after the module that unlocks them.
- Layer 3 — Capstone: A final production-grade, multi-agent platform built in the Capstone Studio, integrating everything in the course.

Placed After	#	Project Title	Theme
After Mod 3	1	Insurance Product Knowledge Assistant (RAG)	Retrieval & grounding
After Mod 4	2	Intelligent Document Processing for Insurance Claims	Document AI
After Mod 5	3	Meeting Minutes & Action Items Generator	Full-stack GenAI app
After Mod 6	4	Code & Test Case Generator with Reflection	Agent loops
After Mod 7	5	Stock Market Research & Analysis Agent	Tool-using agents
After Mod 7	6	AI Travel Planner Multi-Agent System with Agno	Multi-agent collaboration
After Mod 8	7	MCP Integration Hub with CrewAI	Interoperability
After Mod 9	8	Customer Support Desk with Persistent Memory	Memory & HITL
After Mod 10	9	Conversational Data Analyst (Text-to-SQL)	Enterprise data agents
After Mod 14	10	Agentic Coding Sprint with Claude Code	AI-driven SDLC

Section D	★	Capstone: Production-Grade Multi-Agent Platform	Everything combined

SECTION A: Generative AI Foundations (15 Hours)

Module 1 · The Generative AI Landscape and How LLMs Actually Work · 2 hours

A fast, engineer-friendly grounding in what large language models are, how they behave, and how to choose between them without unnecessary mathematics.

- From traditional software to AI-native software: what changed, and why 2023–2026 rewrote the technology job market.
- How LLMs work under the hood: tokens, embeddings, attention, and next-token prediction.
- The 2026 model landscape: GPT, Claude, Gemini, Llama, Mistral, DeepSeek, Qwen — proprietary vs open-weight models.
- Reasoning models and test-time compute; context windows, token pricing, latency, and rate limits.
- Multimodal models: text, vision, audio, and document understanding; hallucinations and non-determinism.

 **Hands-on Lab:** Run the same set of business tasks across three frontier models, compare quality, cost, and latency, and produce a model-selection matrix.

Module 2 · Prompt Engineering, LLM APIs, and Structured Outputs · 3 hours

The core craft of working with models programmatically.

- Working with OpenAI, Anthropic, and Gemini APIs: chat messages, roles, system prompts, streaming, and SDK patterns.
- Sampling controls: temperature, top-p, max tokens, stop sequences; structured outputs with Pydantic.
- Function calling / tool calling: the single most important API capability for agentic workflows.
- Context engineering, prompt versioning, prompt caching, and batching for cost control.

 **Hands-on Lab:** Build a document-extraction service that converts unstructured contracts into validated JSON with retry and fallback logic.

Module 3 · Embeddings, Vector Databases, and RAG · 4 hours

Building retrieval pipelines that are measurable and tunable.

- Embeddings and semantic search; document ingestion and chunking strategies (fixed, recursive, semantic, layout-aware).
- Vector databases compared: Pinecone, Chroma, Qdrant, Weaviate, and pgvector.
- Advanced retrieval: hybrid search, reranking, query rewriting, metadata filtering.
- Measuring RAG: faithfulness, answer relevance, context precision/recall with RAGAS-style evaluation.

 **Hands-on Lab:** Assemble a full RAG pipeline step by step, then tune chunking/retrieval settings and observe score changes.

PROJECT 1 · Insurance Product Knowledge Assistant (RAG)

Build a production-quality RAG assistant over a corpus of insurance policy documents. Customers and agents can ask questions and receive grounded answers with clickable citations to the exact clause and graceful refusal when the answer is not in the corpus.

Module 4 · Open-Source Models, Fine-Tuning, and Build-vs-Buy · 2 hours

When prompting and RAG are not enough — and how enterprises handle data privacy and compliance.

- Running models locally and privately: Ollama for development, vLLM for production serving.
- Decision framework: prompt engineering vs RAG vs fine-tuning (LoRA/QLoRA) vs distillation.
- Small language models and model routing; data privacy drivers (HIPAA, GDPR, SOC 2).

PROJECT 2 · Intelligent Document Processing for Insurance Claims

An automated pipeline that ingests claim documents (forms, photos, scanned PDFs), extracts and validates structured data, flags anomalies, and routes exceptions to a human queue.

Module 5 · Building and Shipping Your First GenAI Application · 2 hours

Turning model skills into a deployed, full-stack application.

- Reference architecture: FastAPI backends, Streamlit/Gradio UIs, React frontends.
- AI-assisted engineering workflows: Claude Code, Cursor, GitHub Copilot.
- Audio pipelines, streaming responses, and cloud container deployment.

PROJECT 3 · Meeting Minutes & Action Items Generator

Full-stack application that takes a meeting recording, transcribes it, identifies speakers, and produces structured minutes with owner/deadline action items.

SECTION B: Agentic AI Core (16 Hours)

Module 6 · From Generation to Action: Agents & Agentic Design Patterns · 3 hours

Building an agent from scratch in pure Python before touching frameworks.

- The perceive → plan → act → observe loop; workflows vs. agents.
- The 4 core patterns: Reflection, Tool Use, Planning, and Multi-Agent Collaboration.
- ReAct architecture, tool schema design, error handling, infinite loops, and safety budgets.

PROJECT 4 · Code & Test Case Generator with Reflection

An agent that produces working code plus tests, executes them in a sandbox, reads failures, and iterates until all tests pass.

Module 7 · Agent Frameworks: LangGraph, CrewAI, Agno, OpenAI Agents SDK · 4 hours

Hands-on mastery of the top orchestration frameworks.

- LangGraph deep dive: graphs, nodes, edges, state, checkpointers, human-in-the-loop interrupts.
- CrewAI role-based crews and Agno lightweight high-performance agents.
- OpenAI Agents SDK and multi-agent topologies (supervisor, hierarchical, pipeline).

PROJECT 5 · Stock Market Research & Analysis Agent

Pulls live market data and news, computes indicators, analyzes sentiment, and outputs structured briefs.

PROJECT 6 · AI Travel Planner Multi-Agent System with Agno

Destination research, flights, itinerary planner, and budget critic agents collaborate to build complete itineraries under constraints.

Module 8 · Agent Interoperability: MCP and A2A · 3 hours

The protocol layer that turns agents into enterprise infrastructure.

- MCP (Model Context Protocol) architecture: hosts, clients, servers, tools, resources, and transports.
- Building custom MCP servers in Python to safely expose internal databases and APIs.
- A2A (Agent-to-Agent protocol) and the 2-layer enterprise stack.

PROJECT 7 · MCP Integration Hub with CrewAI

Design an MCP server exposing CRM data, consumed by a CrewAI crew with human-approval gates for write operations.

Module 9 · Agent Memory and Context Engineering · 3 hours

Engineering what the model sees for long-running agents.

- Context failure modes: poisoning, distraction, compaction strategies.
- Memory taxonomy: short-term, long-term, episodic, semantic, procedural (Mem0 / LangMem).
- LangGraph checkpoints for durable, resumable agent sessions.

PROJECT 8 · Multi-Agent Customer Support Desk with Persistent Memory

Supervisor support system with triage, knowledge, and action agents, persistent customer memory, and human escalation.

SECTION C: Production & Enterprise Engineering (15 Hours)

Module 10 · Agentic RAG and Enterprise Data Agents · 2 hours

Reasoning-driven retrieval and enterprise database agents.

- Query planning, multi-hop retrieval, GraphRAG concepts, and self-correcting retrieval loops.
- Text-to-SQL agents: schema awareness, semantic layers, read-only guardrails.

PROJECT 9 · Conversational Data Analyst (Text-to-SQL)

Interprets natural language queries, writes guarded read-only SQL, self-corrects on errors, and renders visual charts with insights.

Module 11 · Evaluation and Observability · 3 hours

Making agents reliable with systematic testing.

- Golden datasets, LLM-as-a-judge with calibrated rubrics, trajectory analysis, tool accuracy.
- Tracing with LangSmith and Langfuse; CI/CD regression testing gates.

Module 12 · Safety, Security, and Responsible AI · 2 hours

Securing agents against modern attack vectors.

- OWASP Top 10 for LLMs, direct and indirect prompt injection defense.
- Guardrails libraries, PII redaction, NIST AI RMF, EU AI Act, auditability logging.

Module 13 · Deployment, Scaling, and AgentOps · 2 hours

Production engineering for AI workloads.

- FastAPI agent services, Docker containers, session concurrency, prompt caching, cost engineering.

Module 14 · The Expanding Frontier · 1 hour

Agentic coding, computer use, real-time voice, and career positioning.

- Claude Code deep dive: CLAUDE.md conventions, plan mode, subagents, and test verification.

PROJECT 10 · Agentic Coding Sprint with Claude Code

Modernize legacy modules and build new features end-to-end using Claude Code as an autonomous engineering agent under human lead supervision.

SECTION D: Capstone Studio (4 Hours)

Participants design and build a production-grade multi-agent platform for a business domain (support, claims, finance, recruiting, or logistics).

Capstone Deliverables
● Supervisor architecture with specialized sub-agents ● MCP tools integration for enterprise data access ● Persistent agent memory across sessions ● Evaluation harness with LLM-as-a-judge rubrics ● Guardrails and security controls ● Cloud deployment with monitoring ● Architecture Decision Records (ADR) document ● Demo Day presentation

06 Tools & Technologies Covered

Category	Technologies
Models & APIs	OpenAI GPT, Claude (Anthropic), Gemini, Llama, Mistral, DeepSeek, Ollama, vLLM
Core Dev	Python, FastAPI, Pydantic, Streamlit, Gradio, Docker, Git
RAG & Vector	Pinecone, Chroma, Qdrant, Weaviate, pgvector, RAGAS, GraphRAG
Agent Frameworks	LangGraph, CrewAI, Agno, OpenAI Agents SDK, AutoGen/AG2
Interoperability	MCP (Model Context Protocol), A2A (Agent-to-Agent Protocol)
Memory & Evals	LangGraph Checkpointers, Mem0, Langfuse, LangSmith
Safety & Deployment	OWASP Top 10 for LLMs, Guardrails AI, Docker, AWS/GCP Cloud Deployment, Claude Code

07 What You Walk Away With

 GitHub Portfolio 10 real-world systems plus a deployed production-grade capstone.	 Full AI Stack Fluency RAG · LangGraph · CrewAI · MCP · Evals · Guardrails · Cloud AgentOps.	 Career Toolkit Resume positioning, interview question bank, and architectural design confidence.
--	---	--

DATA STREET ACADEMY · Generative AI & Agentic AI · 50 Hours · Production-Grade AI Engineering

Data Street Academy · Generative AI & Agentic AI · 50 Hours