Agentic AI Master Syllabus

WEEK 1 — Introduction to Agentic AI & Building AI Agents

Understand how agents think and act, then build your first agent.

Day 01 What is Agentic AI? Generative AI vs Agentic AI; AI assistants, workflows and autonomous agents.
Day 02 Agent architecture and components: goals, planning, reasoning and execution.
Day 03 Reactive vs planning-based agents (ReAct pattern); single-agent vs multi-agent systems; human-in-the-loop AI.
Day 04 The LLM as the reasoning engine; writing agent instructions and objectives.
Day 05 Tool and function calling; connecting agents to external APIs.
Day 06 Planning and task decomposition; observation, action and feedback loops.
Day 07 Memory and state management; stopping conditions, iteration limits and handling failed tool calls; weekly assessment.

WEEK 2 — Agent Frameworks & Agentic RAG

Work with the leading agent frameworks and make retrieval smarter.

Day 08 LangChain: LLM app development, tools, chains, integrations and agents.
Day 09 LangGraph in depth: stateful agents and graph workflows.
Day 10 LangGraph checkpoints and human approval; CrewAI role-based agent teams.
Day 11 Microsoft Agent Framework (with AutoGen background); OpenAI Agents SDK, Claude Agent SDK and Google ADK awareness.
Day 12 Traditional RAG vs agentic RAG; query rewriting, query decomposition and agent-driven retrieval.
Day 13 Multiple knowledge sources, SQL and document retrieval (text-to-SQL); GraphRAG and knowledge graphs (awareness).
Day 14 Retrieval validation, corrective retrieval, self-reflection and answer verification; Project 1 (Intermediate): AI Research Agent that searches, collects evidence and produces a sourced report.

WEEK 3 — Multi-Agent Systems & External Tools

Make agents work in teams and connect them to real systems.

Day 15 Multi-agent architecture; agent roles and responsibilities.
Day 16 Supervisor and worker patterns; sequential and parallel execution.
Day 17 Agent delegation, coordination, shared state and communication; conflict resolution, result aggregation, evaluation and debugging.
Day 18 REST APIs and authentication (API keys, OAuth); web search and information extraction.
Day 19 SQL databases and structured data; file systems and document processing.
Day 20 Python execution in controlled environments; email, calendar and business-system integrations.
Day 21 Browser automation fundamentals, tool permissions and approval workflows; Project 2 (Advanced): Multi-Agent Content System with researcher, writer and reviewer agents.

WEEK 4 — Memory, Advanced Design & Agent Security

Build agents that remember, run reliably and stay safe.

Day 22 Short-term and long-term memory; conversation history management; persistent state and checkpoints.
Day 23 User preferences and session context; summarization and context compression; vector-based memory retrieval; preventing stale or incorrect memory; privacy-aware memory.
Day 24 Planning and replanning, workflow orchestration, reflection and verification loops.
Day 25 Model Context Protocol (MCP): tools, resources and prompts; agent-to-agent communication.
Day 26 Event-driven and asynchronous agents, background tasks and scheduled workflows; reliable execution, idempotency and human approval for sensitive actions.
Day 27 Agent security: prompt injection and indirect prompt injection, data leakage and privacy, tool abuse and least-privilege permissions, sandboxing and access control.
Day 28 Authentication and secrets management, output validation, audit logs and traceability, human oversight and responsible AI; OWASP Top 10 for LLM Applications.

Greens Technologies

WEEK 5 — Evaluation, Deployment & Final Project

Test, ship and monitor your agents like a production team.

Day 29 Testing and evaluation: LLM output quality, factual accuracy, LLM-as-a-judge, human evaluation and hallucination detection.
Day 30 RAG retrieval and answer evaluation; agent task-completion rates and tool-call success and failure analysis; regression testing and benchmark datasets.
Day 31 Latency and token-cost tracking; tracing and debugging agent workflows (RAGAS, LangSmith, Langfuse).
Day 32 Deployment: APIs with FastAPI, frontends with Streamlit, Docker fundamentals and Kubernetes orchestration.
Day 33 Cloud deployment on AWS, Azure and GCP; database and vector database integration; logging, monitoring and observability (OpenTelemetry); caching, scaling, rate limiting and cost management; CI/CD with GitHub Actions.
Day 34 Emerging topics: building MCP servers and clients, A2A protocol, computer-use, coding and voice agents, long-running workflows, enterprise AI automation and AI governance. Project 3 (Advanced): Enterprise Support Agent that retrieves company knowledge, classifies tickets, drafts responses and asks for human approval.
Day 35 Project presentation, final assessment, portfolio review and interview preparation.

Greens Technologies