Agentic AI Training Course – 32 hrs.

Course Objectives:

By the end of this 8-session course, learners will:

1. **Understand the core principles of Agentic AI** and how agents differ from traditional LLM-based systems or chatbots.
2. **Explore the anatomy of autonomous agents**, including memory, planning, reasoning, and action.
3. **Design and implement single-agent and multi-agent systems** using LangChain, OpenAI tools, and other libraries.
4. **Integrate memory, tool use, and real-world APIs** into intelligent agents.
5. **Learn to plan, reason, and execute tasks** using advanced agentic workflows and architectures.
6. **Implement monitoring, evaluation, and safety guardrails** in agentic systems.
7. **Apply agentic AI to real-world tasks and domain-specific use cases**, including collaborative and persistent agents.

Learning Outcomes:

After completing this course, participants will be able to:

1. **Explain the theoretical foundations** of agentic AI and its evolution from rule-based and LLM systems.
2. **Construct autonomous agents** that can perceive context, plan tasks, execute actions, and use tools effectively.
3. **Build and run agents using LangChain, OpenAI function-calling, and Python**, leveraging memory and planning features.
4. **Design multi-agent workflows** where agents collaborate, delegate, and communicate to achieve a common goal.
5. **Employ vector stores (FAISS, Pinecone, Chroma)** to implement persistent memory and context-aware reasoning.
6. **Create agents that interface with external systems** (e.g., web services, databases, enterprise APIs).
7. **Monitor and evaluate agent performance** using tools like LangSmith, Trulens, or PromptLayer.
8. **Address safety, ethical, and reliability concerns** in autonomous agent systems.

Prerequisites:

Participants should ideally have:

1. **Programming knowledge** – Intermediate-level Python skills, including working with APIs and basic object-oriented programming.
2. **Foundational understanding of AI/ML** – Awareness of what LLMs are, how they work, and general familiarity with NLP concepts.
3. **Basic familiarity with LLMs or Generative AI tools** – Prior usage of tools like ChatGPT, Claude, or similar models will help.

4. **Basic familiarity with REST APIs and JSON** – Since agents interact with external tools/services.
5. **(Optional but helpful)**: Exposure to LangChain, vector stores, or frameworks like OpenAI tools, AutoGen, or similar.

Session 1: Foundations of Agentic AI and Autonomy:

- What is Agentic AI? Key differences from chatbots, copilots, and traditional LLMs
- Core principles: autonomy, goal orientation, tool use, feedback
- Agent architectures: Reactive, Deliberative, Hybrid
- Evolution of AI to agents: from chat completion to autonomous agents
- Use cases across industries (healthcare, finance, logistics, retail)
- Interactive Demo: LLM chatbot vs agent

Session 2: Agent Anatomy – Components and Architectures

- Core components: memory, planning, reasoning, action execution
- Agent lifecycle and control flow
- Goal decomposition and task management
- Understanding feedback loops and agent learning
- Hands-on: Code walkthrough of a basic agent (Python + OpenAI functions)
- Mini Lab: Create a simple agent to fetch news and summarize

Session 3: Building Agents with LangChain and OpenAI

- LangChain introduction: tools, chains, and agents
- Agent types: ReAct, Tool-Using, Conversational agents
- Prompt engineering for agent decision-making
- Adding memory and tool use via LangChain
- Hands-on Lab: Build an agent to query APIs (weather, Wikipedia, calculator)

Session 4: Multi-Agent Systems and Collaboration

- What are Multi-Agent Systems (MAS)?
- Agent communication and coordination
- Role delegation and orchestration among agents
- Architectures: decentralized vs centralized control
- Multi-agent simulations: LangGraph, AutoGen, CrewAI (overview)
- Hands-on Lab: Simulate 2 agents working together (planner + executor)
- Case Study: Multi-agent systems in customer support automation

Session 5: Agent Memory, Context, and Learning

- Context windows vs memory systems
- Short-term vs long-term memory
- Retrieval-Augmented Generation (RAG) architecture
- Using vector databases: FAISS, Chroma, Pinecone
- Hands-on Lab: Build an agent with memory using FAISS and LangChain

Session 6: Reasoning, Planning, and Tool Use

- Planning frameworks: ReAct, Tree-of-Thoughts (ToT), Reflexion
- Role of self-reflection and meta-reasoning in agent intelligence
- Tool use: OpenAI function calling, LangChain Tools, APIs
- Designing custom tools and error handling
- Hands-on Lab: Create an agent that plans and executes a multi-step task
- Demo: Reflexion-style self-improving agent

Session 7: Monitoring, Evaluation, and Guardrails

- Evaluating agent performance: success, coherence, correctness
- Human-in-the-loop vs autonomous agents
- Setting guardrails to prevent hallucinations or unsafe outputs
- Monitoring with LangSmith, Trulens, PromptLayer
- Logging, debugging, and improvement loops
- Lab: Monitor and evaluate agent performance across two tasks

Session 8: Real-World Use Cases and System Integration

- Real-world case studies: travel agent, AI recruiter, legal assistant
- Integrating agents with CRMs, databases, and enterprise APIs
- Persistent agents and asynchronous task handling
- Deployment options: serverless, cloud, embedded
- UI/UX design considerations for agentic applications
- Hands-on Lab: Build an integrated agent with external API + storage