

Generative AI and Agentic AI training

Pre-requisites:

- Basics of Python language.
- Basics of Machine Learning

Table of Content

Module 01: Introduction of GenAI

- Introduction to Generative AI
- Architecture of Generative AI
- Applications of Generative AI using Transformer Library
- Labs

Module 02: Introduction of Large language Model

- Architecture of Large Language Models
- Types of Large Language Models (LLMs)
- Task based Text AI LLMs – Translation, Summarization, Sentence Similarity, Automatic Speech Recognition, Text to Speech, etc.
- Major Text AI LLMs - LLaMA, Qwen, Cohere, Falcon LLM
- Image AI Models & Services – Object Detection, Image Segmentation, Image Retrieval, Image, Image Captioning, Visual QnA, Zero-shot Image Classification, etc.
- Labs

Module 03: Learning Prompt Engineering using Open Source Models

- Introduction to Prompt Engineering
- Prompt Engineering Techniques: Zero-shot prompting, few shot prompting, Chain-of Thought, persona based prompting etc
- Best practices in prompt engineering

- Use cases in IT and operations
- Labs

Module 04: Multi-Modal Models

- Prompting with multi-modal models
- OpenAI Whisper: Speech-to-text model (open-source)
- DALL-E Mini: Image generation and understanding (open-source alternative)
- Multi-modal task processing

Module 05: LLM Systems (RAG) using Open Source Models

- Introduction to Retrieval Augmented Generation (RAG)
- Embedding generation and storage

Concept of Vector search and similarity scoring

- Introduction to LangChain
- Concept of Embedding, Retrieval, Chain and Agents using LangChain
- Lab: Build a Simple LLM Application using LangChain
- Lab: Build a Chatbot LangChain
- Difference between RAG & Question Answering system
- Build a Question Answering system over Tabular Data using LangChain
- Lab: Build vector stores and retriever using LangChain
- Lab: Build an Agent LangChain
- Lab: Build a Retrieval Augmented Generation (RAG) Application using LangChain
- Lab: Build a Conversational RAG Application using LangChain
- Introduction to Weaviate and vectorbase
- Multi-modal RAG using weaviate

Module 06: Deployment of Generative AI Systems

- LangChain: Building complex agent workflows

- Gradio: Creating interactive user interfaces
- Streamlit: Lightweight UIs for Agentic AI applications

Module 07: Introduction to Agentic AI

- What is Agentic AI?
- Characteristics of Agentic AI Systems
- Generative AI vs Agentic AI
- Use Cases and Applications
- Architecture of Agentic AI (Reflex, ReAct, Planning Agent, Multi-agent systems etc.)
- Agent Frameworks (LangChain, AutoGen etc)
- Ethical Considerations and Risks

Module 08: Building Chatbots with Agentic AI

- Using Llama 2 and Open-Assistant for Conversational Agents
- LangChain for Multi-Turn Conversations and Workflow Automation
- Integrating Gradio or Streamlit for Chatbot UIs
- Hands-On: Building a Task-Specific Chatbot

Module 09: Working with AutoGen

- Introduction to AutoGen
- AutoGen Architecture, components and ReAct pattern
- Agent-agent interaction, agent groups etc
- RAG and Function calling
- Labs

Module 10: Ethical and Responsible Use of Agentic AI

- Bias Detection and Mitigation in Open-Source Models
- Transparency in Decision-Making Processes

- Safeguards Against Unintended Outcomes

Module 11 Introduction to Agentic Frameworks

- Overview of Agentic AI: concept of autonomous agents
- Introduction to LangGraph
 - Graph-based execution for LLM workflows
 - Use cases and architecture
- Introduction to LangChain
 - Composable, modular LLM pipelines
 - Chains, agents, tools, and memory modules
- Introduction to multi-agent systems
 - Collaboration and task delegation between agents
 - Real-world business use cases

Module 12 : Agentic Retrieval-Augmented Generation (RAG)

- Refresh on traditional RAG pipelines
- What is Agentic RAG?
 - Combining retrieval with agent-based reasoning
 - Dynamic decision making in information retrieval
- Implementing advanced Q&A and research assistants
- Integrating vector databases (e.g., Chroma, Pinecone) with agents

Module 13: n8n and Model Context Protocol (MCP)

- Introduction to n8n
 - Visual workflow automation platform
 - Connecting LLM agents with business systems
- MCP (Model Context Protocol)

- Maintaining conversation context across steps
 - State management and memory persistence
- Example use cases: multi-turn workflows, context-aware virtual assistants, automated reporting