Generative AI and Agents Course Content

Topics Covered

- What is Generative AI?
- What are LLMs (Large Language Models)
- Introduction to OpenAI API and ChatGPT
- Using ChatGPT for various tasks using manual prompting
- What kind of applications can be developed using these LLMs
- Understanding ReAct prompt
- Understanding Agent and how it works conceptually

- Understanding about embeddings and RAG Conceptually

What You Will Learn

- Grasp the fundamentals of Generative AI and Large Language Models
- Understand the capabilities and limitations of LLMs
- Learn to interact with OpenAI API and ChatGPT effectively
- Master manual prompting techniques for various use cases
- Explore real-world applications built with LLMs
- Understand the ReAct (Reasoning + Acting) prompting paradigm
- Conceptually understand how AI agents work and their architecture
- Learn the fundamentals of embeddings and RAG (Retrieval-Augmented Generation)

Hands-on Labs

- **Lab 1.1:** Creating a chatbot using OpenAI API and Streamlit/Gradio

2

Using Langchain 1.x (Latest Version)

Topics Covered

- Why Langchain?
- Using Langchain to make API call to OpenAI API
- Understanding various message types
- Understanding and creating a first chain
- Understanding and using Prompt Templates
- Understanding Runnables in detail
- Chaining 2 chains
- LCEL (LangChain Expression Language)

- Using Server Side tools with model

What You Will Learn

- Understand why Langchain is essential for building LLM applications
- Master making API calls to OpenAI using Langchain
- Learn different message types (System, Human, AI, Tool messages)
- Create your first Langchain chain from scratch
- Design and implement effective prompt templates
- Deep dive into Runnables and their capabilities
- Combine multiple chains for complex workflows
- Master LCEL for declarative chain composition
- Integrate server-side tools with LLM models

Hands-on Labs

- **Lab 2.1:** Creating a chatbot using Langchain and Streamlit
- **Lab 2.2:** Using Runnables for chain composition

3

Understanding Tools and Agents

Topics Covered

- Understanding how agents work
- Creating tools and creating agent using tools
- Understanding Middleware

- Using @wrap_model_call, @wrap_tool_call and @dynamic_prompt
- Static and Dynamic Models
- Error Handling In Tools using middleware
- Static and dynamic prompts
- Structured output using Tool Strategy and Provider Strategy
- Understanding Memory
- Memory Management Techniques like Trimming Messages, Message Summarization
- Using DatabaseToolkit and Code execution tools
- Agent response streaming
- Using Tool Runtime to access State, context, Store Commands and Stream Writer
- Mail Agent and Human In the Loop

- Implementing Guard rails

What You Will Learn

- Deep understanding of agent architecture and decision-making process
- Create custom tools and integrate them with agents
- Master middleware for intercepting and modifying agent behavior
- Use decorators for model calls, tool calls, and dynamic prompts
- Implement both static and dynamic model selection
- Build robust error handling mechanisms using middleware
- Create static and dynamic prompt systems
- Generate structured outputs using advanced strategies
- Implement various memory management techniques
- Use database toolkits and code execution capabilities
- Stream agent responses for better user experience
- Leverage Tool Runtime for state management and context
- Build mail agents and human-in-the-loop systems
- Implement guardrails for safe AI agent deployment

Hands-on Labs

- **Lab 3.1:** Using Middlewares for custom functionality
- **Lab 3.2:** Dynamic models and Dynamic Prompts implementation
- **Lab 3.3:** Building a Database Agent

- **Lab 3.4:** Human In the Loop implementation
- **Lab 3.5:** Implementing Guard rails for agent safety

4

Understanding Embeddings, Vector Store and RAG

Topics Covered

- Understanding embeddings
- What is RAG?
- Using Document Loaders, Splitters
- Indexing
- Using Chroma DB as vector store
- Understanding Similarity Search
- Retrieving using retrievers
- RAG with Agents

- RAG with Chains

What You Will Learn

- Master the concept of embeddings and vector representations
- Understand RAG (Retrieval-Augmented Generation) architecture
- Load and process documents using various loaders
- Implement document splitting strategies
- Create and manage vector indexes
- Set up and use Chroma DB as a vector database
- Implement similarity search algorithms
- Build and use custom retrievers
- Integrate RAG with agents for enhanced capabilities
- Combine RAG with chains for complex workflows

Hands-on Labs

- **Lab 4.1:** Implementing a complete RAG system with document processing, vector store, and retrieval

5

Model Context Protocol (MCP)

Topics Covered

- Why MCP?
- Creating a custom MCP server
- Using MCP Servers with Claude desktop
- Stdio Client
- SSE client

- Using MCP with Langchain

What You Will Learn

- Understand the need and benefits of Model Context Protocol
- Build custom MCP servers from scratch
- Integrate MCP servers with Claude Desktop
- Implement Stdio-based MCP clients
- Set up Server-Sent Events (SSE) clients
- Seamlessly integrate MCP with Langchain applications

Hands-on Labs

- **Lab 5.1:** Creating a custom MCP server
- **Lab 5.2:** Integrating MCP with Claude Desktop

- **Lab 5.3:** Using MCP with Langchain applications

6

Langgraph

Topics Covered

- Why Langgraph?

- Understanding Langgraph components
- Understanding State of Graph and creating StateGraph
- Tool Node
- Conditional Branching in graph
- Adding Memory to graph
- Graph with Multiple Schemas
- Implementing a chatbot using StateGraph for Summarization
- Parallel processing in langgraph
- Dynamic parallelization in Langgraph
- Multi Agentic Application using Supervisor Pattern
- SubGraphs

- Building SQL Agent

What You Will Learn

- Understand why Langgraph is essential for complex agent workflows
- Master all Langgraph components and their usage
- Create and manage StateGraph for agent state management
- Implement tool nodes in graph workflows
- Build conditional branching logic in graphs
- Add memory capabilities to graph-based agents
- Work with multiple schemas in a single graph
- Build a summarization chatbot using StateGraph
- Implement parallel processing in graphs
- Create dynamically parallelized workflows
- Build multi-agent systems using supervisor pattern
- Design and implement subgraphs for modularity
- Create a complete SQL agent using Langgraph

Hands-on Labs

- **Lab 6.1:** Creating a state graph with memory
- **Lab 6.2:** Creating a custom ReAct Agent using Lang Graph
- **Lab 6.3:** Understanding parallel processing in Langgraph
- **Lab 6.4:** Creating a multi agentic application using Langgraph

- **Lab 6.5:** Creating a customer support bot using subgraphs

7

Google ADK

Topics Covered

- What is ADK?
- Create an agent project
- Running adk web, adk run and adk api server
- ADK Streaming
- Context Management for Agents
- Tool definition and usage
- Invoking LLM directly
- Running agent manually
- Session Service
- Runner
- Using LiteLLM
- Agent Delegation to Sub Agents
- Understanding Session State
- Understanding ToolContext
- Understanding about Session Service
- Using Session variables in agent instructions
- Understanding how ADK runtime executes your agent in detail
- Understanding callbacks in ADK
- Input GuardRails
- Sequential Agents
- Parallel Agents
- Loop Agents

What You Will Learn

- Understand Google ADK architecture and purpose
- Set up and create agent projects
- Run ADK web interface, CLI, and API server
- Implement streaming responses in ADK
- Manage context for agents effectively

- Define and use tools in ADK
- Invoke LLMs directly through ADK
- Run agents without the ADK CLI
- Understand and use Session Service
- Work with ADK Runner for programmatic execution
- Integrate LiteLLM for multi-provider LLM access
- Implement agent-to-agent delegation
- Create hierarchical agent structures
- Manage communication between parent and sub-agents
- Master session state management in ADK
- Understand and use ToolContext effectively
- Deep dive into Session Service capabilities
- Use session variables in agent instructions
- Understand ADK runtime execution flow in detail
- Implement callback mechanisms in ADK
- Build input guardrails for agent safety
- Intercept and modify agent behavior using callbacks
- Design and implement sequential agent workflows
- Create parallel agent execution patterns
- Build loop-based agent systems

Hands-on Labs

- **Lab 7.1:** Understand ADK basics - Running Agent using ADK
- **Lab 7.2:** Running Agents manually and using LiteLLM
- **Lab 7.3:** Auto agent delegation implementation
- **Lab 7.4:** Using Session Service and State management
- **Lab 7.5:** Implementing Guardrails using callbacks
- **Lab 7.6:** Implementing Sequential, Parallel and Loop Agents

Advanced ADK Topics

- Understanding Simple Parent rule
- Creating Custom Agents
- Understanding state management using Invocation Context in detail
- Understanding EventActions
- Agent Escalation

- Understanding Agent Tool
- Controlling agent flow with tool actions
- Tool Sets in ADK
- Using ADK's Builtin code executor
- Integrating Langchain and Crew AI Tools

- Implementing LongTerm memory in ADK using memory Service

Advanced Learning Outcomes

- Understand Simple Parent rule for agent hierarchies
- Create custom agents with specialized behaviors
- Master Invocation Context for state management
- Implement EventActions for event-driven workflows
- Build agent escalation mechanisms
- Use Agent Tools for enhanced capabilities
- Control agent flow using tool actions
- Organize tools using Tool Sets
- Leverage ADK's built-in code executor
- Integrate Langchain and Crew AI tools with ADK
- Implement long-term memory using Memory Service

Advanced Labs

- **Lab 7.7:** Implementing all advanced topics using practical examples

ADK Design Patterns

- Implementing Coordinator/Dispatcher Pattern
- Hierarchical Task decomposition pattern
- Critic Pattern

- Human in the loop pattern

Design Pattern Learning Outcomes

- Implement Coordinator/Dispatcher pattern for agent orchestration
- Build hierarchical task decomposition systems
- Create Critic pattern for agent self-evaluation
- Implement Human-in-the-loop patterns for oversight

Design Pattern Labs

- **Lab 7.8:** Implementing all design patterns using ADK

Agent Config, MCP Integration, and Artifact Service

- Creating agentic application using agent config
- Using MCP with ADK
- Understanding and using Artifact Service

Integration Learning Outcomes

- Build agentic applications using agent configuration
- Integrate MCP (Model Context Protocol) with ADK
- Use Artifact Service for managing agent outputs

Integration Labs

- **Lab 7.9:** Implementing agentic application using agent config
- **Lab 7.10:** Using MCP Tools with ADK
- **Lab 7.11:** Using Artifact Service