

## **Agentic AI Course Syllabus**

### **Section 1: Introduction to Agentic AI & LLM Ecosystem**

- What is Agentic AI?
- Role of LangGraph, AutoGen, CrewAI in the ecosystem
- OpenAI vs Azure OpenAI vs AWS Bedrock
- Introduction to foundational concepts: Agents, Tasks, Graphs

### **Section 2: Basics of LangChain and LangGraph**

- LangChain recap: Chains, Tools, Memory
- LangGraph architecture and why it matters
- Installation, environment setup, and first LangGraph DAG

### **Section 3: Exploring LangGraph Core Concepts**

- Nodes, Edges, State Machines
- Understanding transitions and handlers
- Building a simple agentic task flow

### **Section 4: Python SDK and Node Configuration**

- Deep dive into LangGraph Python SDK
- Defining nodes and reactive transitions
- Testing individual components with unit test strategy

### **Section 5: Multi-Agent Setup with LangGraph**

- Multi-agent interaction via graph state
- Introducing dynamic task allocation
- Conditional logic and loops in graphs

### **Section 6: Context Handling in Graphs**

- Memory, buffers, and shared state
- Prompt engineering for modular agents
- Using LangChain tools inside LangGraph

## **Section 7: Introduction to AutoGen**

- AutoGen vs LangGraph
- AutoGen architecture and agent design
- Basic use-cases and sample projects

## **Section 8: Building Custom Agents with AutoGen**

- Defining roles and communication protocols
- Tool integration in AutoGen
- Building helper agents and supervisor agents

## **Section 9: Combining AutoGen and LangGraph**

- Orchestrating AutoGen inside LangGraph
- Handling multi-turn conversations
- Error handling and edge case design

## **Section 10: Invoice Parsing with LangGraph**

- Designing agents for invoice interpretation
- Simulating document variations
- Defining success metrics for extraction

## **Section 11: Image Processing Pipeline (OCR)**

- Tools: Azure Cognitive Vision, AWS Textract, Tesseract
- Building OCR extractor modules
- Integration with LangGraph pipeline

## **Section 12: Intermediate Graph Building**

- Data validation agents
- Retry and fallback agents
- Visualizing graphs with tools like Graphviz

### **Section 13: Intro to Containerization with Docker**

- Docker basics, images, volumes, networks
- Writing Dockerfiles for Python/Node.js agents
- LangGraph inside Docker

### **Section 14: Docker Compose and Multi-Service Setup**

- Docker Compose YAML structure
- Orchestrating multiple agents/services
- Environment variables and secrets

### **Section 15: Testing Dockerized LangGraph Solutions**

- Local dev + container testing
- Bind mounts, logging, and debug modes
- Container-to-container communication

### **Section 16: Introduction to Kubernetes (K8s)**

- Pods, Deployments, Services, ConfigMaps
- K8s vs Docker Compose
- Setup Minikube for local testing

### **Section 17: Deploying LangGraph in K8s**

- Writing Kubernetes manifests
- Helm vs Kubectl
- Deploying a sample LangGraph pipeline

### **Section 18: AutoGen on Kubernetes**

- Scaling agents
- Managing state in distributed environment
- Logging and monitoring via Prometheus & Grafana

### **Section 19: Azure Cloud Deployment**

- Resource Group, App Service, Container Registry
- Deploying Docker container to Azure Web App

- Azure OpenAI API authentication & quota handling

## **Section 20: AWS Cloud Deployment**

- ECS + Fargate for LangGraph
- Using Bedrock for model inference
- Integration with CloudWatch, IAM, Textract

## **Section 21: CI/CD for Agentic AI Pipelines**

- GitHub Actions basics
- Docker build & push workflow
- Kubernetes auto-deploy pipeline

## **Section 22: Production-Ready Agentic Systems**

- Rate limiting & retries
- API Gateway/Reverse Proxy integration
- Secure key management

## **Sections 23: Logging and Observability**

- LangGraph/AutoGen internal logs
- Using OpenTelemetry
- Tracing long-running agent flows

## **Sections 24: Performance Benchmarking**

- Token usage analysis
- Latency optimization
- Cost-performance balance in cloud

## **Section 25: Advanced Prompt Engineering for Agents**

- Structured outputs with ReAct and CoT
- Use of external toolkits (LlamaIndex, Vector DBs)
- Model adaptation and few-shot strategies

## **Section 26: User Feedback Loops in Agentic Systems**

- Capturing feedback on agent outputs
- Self-healing agents with AutoGen feedback loops
- Dynamic policy adjustment

## **Section 27: Simulation & Testing Frameworks**

- End-to-end pipeline testing
- A/B test experiments
- Integration with synthetic data generation

## **Section 28: Case Study – Enterprise Invoice Agent**

- Simulating multilingual invoices
- Table extraction logic
- Structured JSON/Excel output via agents

## **Section 29: Agent Behavior Tuning**

- Prompt templating with LangChain
- Personality config for agents
- Context vs history vs memory tradeoffs

## **Section 30: Capstone Design Review (Part 1)**

- Each participant/team presents their initial design
- Review and feedback from mentors

## **Section 31: Capstone Development Support**

- Debugging session
- Review state transitions
- Test cases writing session

## **Section 32: Deployment of Capstone to Cloud**

- Pick Azure or AWS for deployment
- Secure deployment best practices

### **Section 33: Monitoring and Final Test Runs**

- Review CI/CD
- Final testing of K8s deployment

### **Section 34: Capstone Presentations – Round 1**

- Showcase final projects
- Peer review

### **Section 35: Capstone Presentations – Round 2**

- Continuation of presentations
- Instructor evaluation

### **Section 36: Graduation and Wrap-up**

- Program summary and learning's
- Certification distribution
- Discussion on next-level topics (RAG, LLMOps, etc.)

CREDO SYSTEMZ  
CREDO SYSTEMZ