Agentic AI Developer Certification Projects Overview

The Agentic AI Developer Certification Program is built around three progressively challenging projects that demonstrate participants' mastery of key concepts in agentic system design. Each project corresponds to a distinct learning module and culminates in a functional system, portfolio-ready submission, and peer/community review.

Project 1: Build a RAG-Powered AI Assistant

- Timing: Week 4 (end of Module 1)
- Focus: Retrieval-Augmented Generation, Single-Agent Systems
- Theme: Knowledge retrieval and question answering

Objective

- Participants will build a retrieval-augmented AI application that can intelligently respond to user queries by grounding its responses in external data sources (e.g., documents, datasets).

Requirements

- A LangChain-based pipeline that includes: Prompt formulation, Vector store retrieval (e.g., FAISS, Chroma), Response generation from an LLM
- Optional enhancements: Basic memory components (e.g., session memory), Intermediate reasoning using ReAct or CoT-style steps
- Evaluation loop or basic logging for QA
- Basic UX (CLI, notebook, or minimal UI)

Deliverables

- A working RAG system with: Clear chain structure, Custom documents or knowledge base
- Source code with inline documentation
- README explaining how to run and evaluate the agent

Project 2: Design a Multi-Agent System

- Timing: Week 8 (end of Module 2)
- Focus: Multi-Agent Workflows, Communication, LangGraph
- Theme: Collaborative Problem Solving

Objective

- Participants will design and implement a system composed of multiple specialized agents that coordinate to accomplish a complex task, showcasing multi-agent, role-based behavior, and inter-agent collaboration.

Requirements

- Use of LangGraph or similar orchestration framework
- At least two distinct agents with: Different roles (e.g., planner vs executor), Defined communication channels or memory sharing
- Use of Model Context Protocol (MCP): implement agent interfaces and communication patterns compatible with MCP to ensure interoperability, modularity, and persistence support
- A goal-driven flow: agents must work together to solve a user-defined problem or task (e.g., travel planner, multi-document summarizer, multi-turn form filler)

Deliverables

- A multi-agent system with a clearly orchestrated workflow
- Demonstration script or UI that showcases the collaboration
- README explaining agent roles, task flow, and evaluation logic
- Optional logs or performance summaries

Final Project: Production-Aware Agentic AI System

- Timing: Week 12 (end of Module 3)
- Focus: Safety, Observability, Deployment
- Theme: Real-World Readiness

Objective

- Participants will finalize and productionize the multi-agent system created in Project 2 by adding robustness, safety, and deployment features. The final project reflects the full lifecycle of agentic AI development—from ideation to deployment.

Requirements

- Build on Project 2 with the following additions: Guardrails, Observability, Deployment, Documentation
- Guardrails: Input/output validation, prompt protection, or structured response constraints
- Observability: Basic logging, user feedback capture, or instrumentation (e.g., using LangSmith or custom logs)
- Deployment: Lightweight deployment via FastAPI, Gradio, or Streamlit; Hosted locally or on a cloud service (Render, Hugging Face, etc.)
- Documentation: Clearly stated limitations and assumptions; Usage guide or demo walkthrough; Safety and monitoring considerations

Deliverables

- A fully runnable and hosted demo of the productionized system
- Source code and deployment files (FastAPI, Gradio/Streamlit app, etc.)
- README with: System diagram or flow explanation, Safety practices and logging, Deployment steps and limitations

Submission and Certification

Each project will be submitted via the Ready Tensor platform in the form of a publication. Participants will receive final certification after completion and approval of all three projects.