Building Scalable Multi-Agent AI Systems

Aditya Mehta

This document outlines the workflow followed during the Seasons of Code 2025 project titled *Building Scalable Multi-Agent AI Systems*, under the mentorship of Shubham Ingale and Atharv Kurde. The primary objective was to build a Retrieval-Augmented Generation (RAG) agent using modern tools such as Hugging Face, LangChain, and local inference libraries like Ollama. The project was executed over nine weeks, with the final deliverable being a functional AI agent capable of retrieving and generating context-aware responses from custom documents.

The initial part of the project (Weeks 1–3) focused on building a foundational understanding of deep learning through hands-on implementation. In Week 1, I trained a simple multi-layer perceptron (MLP) to classify data, gaining familiarity with PyTorch basics. Week 2 involved building a convolutional neural network (CNN) for image classification tasks. In Week 3, I implemented a recurrent neural network (RNN) using LSTM units for sequential data, and supplemented this by studying the intuition behind LSTM networks. These initial exercises helped strengthen core concepts and prepared the foundation for deploying more advanced models in later stages of the project. In Week 4, I implemented a basic self-attention layer and studied the transformer encoder-decoder architecture.

In the second phase (Weeks 5–9), I worked with large language models and modern frameworks. I started with the Hugging Face `transformers` library to explore text generation, summarization, and classification. For the main project, I first built a ReAct agent, then extended it into a RAG agent using text chunking with `CharacterTextSplitter`, Hugging Face embeddings, FAISS for vector storage, and LangChain's `RetrievalQA` to connect retrieval with generation.

To demonstrate practical usage, I implemented examples using Hugging Face's summarization and classification pipelines and integrated them with LangChain. Though I did not run Ollama locally due to environment limitations, I focused on cloud-based and Hugging Face-hosted models. The project culminated in a working RAG system, tested on user-provided text inputs and capable of producing informative, context-aware answers.

As practical demonstrations, I built two applications using the RAG agent. The first was a text summarizer that generated concise summaries from long-form text inputs. The second was a question-answering bot built around a PDF containing stock market performance data from 2024, which could respond accurately to user queries by retrieving relevant content from the document.