Assignment 2 - Building Custom RAG pipelines with LangChain API

Assignment Overview

In this assignment, you will explore the capabilities of the LangChain API by building three custom Retrieval-Augmented Generations (RAGs). Each RAG should utilize a different document loader and vector database to demonstrate the versatility and power of LangChain in various scenarios.

Objectives

- Understand and implement the RAG model using the LangChain API.
- Experiment with different document loaders to handle various data sources.
- Explore different vector databases for efficient information retrieval.
- Develop skills in reading documentation effectively.

Tasks

RAG Development

Create three different RAGs using the LangChain API:

1. Each RAG should use a unique document loader. You cannot use the WebbaseLoader as this was already used in class. Possible document loaders include:
   - Local file system loader
   - Cloud storage loader (e.g., AWS S3)
   - Web crawler loader
2. Each RAG must utilize a different vector database for document retrieval. You may choose from the following vector databases. You should not use databases that were used in class such as FAISS.
   - Chromadb

- Pinecone
- Weaviate

Video Presentation

Prepare a video presentation or a live demo of your RAGs. The presentation should be maximum 10 minutes and explain the code and clearly show the RAGs working with a live demo. A demo *is not* you simply going over code or screenshots without actually running the code.

Submission Guidelines

Submit the following by the deadline:

- Separate python code for each RAG
- A maximum 10 minute video presentation demonstrating the working of your RAGs and explaining the code.

Marking Scheme

Criteria	**Marks**
Three different RAG created and demonstrated as working	15
Using three different document loaders and explaining their working clearly	5
Using three different vector databases and explaining their working clearly	5
Explanation of overall code in own words	5
Video presentation (clear explanations, concise, relevant)	10
Discretionary marks	5
Total	**45**