RAG System Homework Project

Overview

Your task is to build a **Retrieval-Augmented Generation (RAG)** system that answers questions using external documents. This demonstrates how large language models (LLMs) can be grounded in up-to-date, domain-specific knowledge.

The chosen model has a knowledge cutoff in **August 2024**. Therefore, your system must rely on retrieved documents to answer questions about events **after this date** — not on the model's internal knowledge.

Project Requirements

Document Preparation

You must choose a document that describes an event that occurred after August 2024. Store and index the document using **ChromaDB** with persistence enabled. Apply a text-splitting strategy to divide the content into at least **50 chunks** to enable meaningful retrieval.

System Implementation

The system must use the `gemini-2.0-flash` model. You are required to build the RAG pipeline using either **LangChain** or **LlamaIndex**, and integrate with either **LangSmith** or **LangFuse** for tracing and observability.

Pre-built agents are not allowed. You must implement core components manually.

The system should support:

- **Dialog flow** – handle multi-turn interactions.
- **Memory** – retain context across turns.

Effectiveness Testing

Design a set of at least **five questions** that can only be answered with the help of the retrieved document. The model should **fail or hallucinate** when answering these questions without retrieval, proving the necessity of RAG.

You should also experiment with **different system prompts** and describe their impact on behavior and results.

Code Quality

Your repository must follow best practices:

- No large files in Git history.
- No secret tokens in commit history.
- Code must be documented and reproducible.

Submission Guidelines

Deadline: 11.05 at 23:59

Each student has a dedicated branch in the repository. You must open a **Pull Request (PR)** from your working branch to your assigned branch before the deadline.

Your PR must include:

- Full implementation of the RAG system.
- A Jupyter notebook or script demonstrating:
  - Indexing
  - Retrieval
  - Answering
  - Prompt experimentation
- A link to your project in **LangSmith** or **LangFuse**.

Bonus Requirements (Extra Credit)

To receive extra credit, your system must implement both of the following:

1. **Metadata filtering** – restrict retrieval to relevant document subsets.
2. **Multi-query retrieval** – use rephrased or alternative queries to improve retrieval depth.

Final Note

Your RAG system must prove its value by answering questions that the model **cannot answer on its own**. Focus on demonstrating that correct answers emerge only when retrieval is active and relevant.