RAG (Retrieval-Augmented Generation) Assignment

Objective

To design a comprehensive architecture for a Retrieval-Augmented Generation (RAG) system that can handle multiple data sources such as text documents, video transcripts, logs, FAQs, and user guides. The architecture should address the complete pipeline from data ingestion to query handling and response generation.

Assignment Overview

You are tasked with designing a RAG system that builds a knowledge base for a manufacturing execution system (MES) used in the chemical industry. The system should enable users to ask questions and receive precise, context-aware answers using Gen-AI techniques.

Task Details

Part 1: Architecture Design

1. **Data Sources**: Incorporate the following:
   - Text documents (e.g., FAQs, error logs, user guides).
   - Video and audio transcripts (e.g., training sessions, client process recordings).
   - Structured logs (e.g., bug reports, release notes).
   - Images (e.g., annotated diagrams, screenshots).
2. **Key Components**:
   - List out all the key components with usage
3. **Architecture Diagram**: Draw a clear, high-level architecture diagram showing all components and their interactions.

Part 2: Implementation Steps

Write detailed steps for implementing the RAG system:

1. **Data Preprocessing**: How would you process raw data? Discuss tools and libraries.
2. **Embedding Generation**: Specify the model and outline the embedding process.
3. **Vector Indexing**: Discuss how embeddings are stored and queried in the vector database.
4. **RAG Workflow**:
   - Input: Query from the user.
   - Process: Retrieval of relevant documents and context-aware generation by the LLM.

   - Output: Final user response.

5. **Evaluation**: Propose metrics to evaluate the system's performance.

Part 3: Extensions and Challenges

1. **Handling Multimedia**: Explain how you would integrate image data and video/audio data.
2. **Scaling**: Discuss how to scale the system to handle large datasets and multiple simultaneous user queries.

Part 4: Implementation with Sample Data and Deployment

Use example files (text, video, audio, and Microsoft Word & PDF) to implement the RAG system pipeline and deploy the results in a Streamlit application. Provide the code and documentation in a GitHub repository.

Submission Requirements

1. **Architecture Document**:
   - Description of each component in your architecture.
   - Diagram of the architecture.
2. **Implementation Plan**:
   - Step-by-step guide for building the RAG pipeline.
3. **Challenges and Extensions**: Solutions for handling multimedia and scaling.
4. **Application:** A working Streamlit application showcasing the complete flow from ingestion to response generation.

   GitHub repository with:
   - Complete codebase.
   - Sample data files.
   - README file with step-by-step setup and usage instructions.
   - Screenshots/short video of the working application.

Tips for Candidates

- Be specific about the tools, libraries, and frameworks you would use.
- Highlight potential challenges and how to overcome them.
- Prioritize simplicity in your design while ensuring it meets the project's requirements.
- Git Repos with similar solutions can be shared if implemented along with the assignment solution.
- A Short Video can be captured of the working solution.