AI Engineer (GenAI Focus) Interview Challenge

Background

The dataset consists of a collection of internal company documents including technical specifications, product manuals, meeting notes, and customer support tickets. The company wants to implement a generative AI system that can make this knowledge accessible to employees via natural language queries, providing accurate and contextual answers.

This challenge simulates a real-world business need for information retrieval and synthesis using large language models. You'll need to design a system that not only retrieves relevant information but also generates coherent, contextual responses while maintaining factual accuracy.

Objective

Build a Retrieval-Augmented Generation (RAG) system that effectively answers questions about company information using the provided document corpus. Your solution should demonstrate an understanding of LLM integration, vector search, prompt engineering, and system design principles.

Dataset Overview

- **Documents**: 500 text files in various formats (PDF, DOCX, TXT, MD)
- **Sample queries**: 50 example questions with expected answers for evaluation
- **Document metadata**: JSON file with creation dates, departments, and document types

Scope of Work

1. Document Processing & Embedding

- Design a pipeline to extract and process text from different document formats
- Implement chunking strategies appropriate for the content
- Create and store vector embeddings for efficient retrieval

2. Retrieval System

- Build a vector database for storing and querying document chunks
- Implement semantic search functionality
- Design a ranking mechanism to prioritize the most relevant context

3. LLM Integration

- Select and justify your choice of foundation model
- Create effective prompts for the generation component

- Implement techniques to improve response accuracy and reduce hallucinations

4. System Evaluation

- Evaluate using the provided sample queries
- Measure retrieval accuracy, response quality, and factual correctness
- Compare different approaches and explain trade-offs

5. Production Considerations

- Address scalability as the document corpus grows
- Consider strategies for keeping information up-to-date
- Design an approach for monitoring and improving system performance

Technical Requirements

- Modular, clean, and documented code
- Use of appropriate libraries for text processing, embeddings, and LLM integration
- Thoughtful prompt design with clear reasoning for your choices
- Analysis of system performance and limitations

Evaluation Criteria

Your solution will be evaluated based on:

- Quality of document processing and retrieval implementation
- Effectiveness of your prompt engineering approach
- System performance on test queries
- Reasoning behind design decisions and trade-offs
- Code quality and documentation

Discussion Questions

- How would you handle document updates in a production environment?
- What techniques did you implement to reduce hallucinations?
- How would your approach scale to 100,000+ documents?
- What limitations does your current solution have?
- How would you implement user feedback mechanisms to improve system performance?

Deliverables

- Complete code repository with setup instructions

- Technical write-up explaining your approach, findings, and limitations
- Brief presentation (5-10 slides) summarizing key aspects of your solution
- Performance analysis with metrics and discussion of results

Final Notes

The focus of this challenge is not just on technical implementation but on demonstrating thoughtful design decisions in working with large language models and retrieval systems. We're interested in your reasoning, your understanding of LLM limitations, and your approach to solving real-world information retrieval problems.