Technical Interview Task

AI-Enhanced Document QA System

Objective

Create a document ingestion and question-answering system that leverages advanced AI models and vector databases. This system should demonstrate your ability to work with state-of-the-art AI tools and APIs, as well as your understanding of natural language processing and information retrieval concepts.

Components

1. Backend (Node.js)

- Set up an Express server with the following endpoints:
    - Document ingestion (accept PDF or plain text files)
    - Question answering
- Implement a document processing pipeline:
    - Text extraction (for PDFs) and chunking
    - Named entity recognition
    - Vector embedding generation
- Integrate with Pinecone for vector storage and search
- Implement error handling and input validation

2. Frontend (React)

- Create a user interface for:
    - Document upload and processing status
    - Asking questions and displaying answers
    - Visualizing relevant document chunks and confidence scores

3. AI Integration

- Use the Claude 3.5 API or GPT-4 API for:
    - Advanced text analysis during document processing
    - Question answering
- Implement Retrieval Augmented Generation (RAG):
    - Retrieve relevant document chunks from Pinecone
    - Enhance prompts with retrieved context

4. Python Script

- Create a Python script that:
    - Performs basic topic modeling on ingested documents
    - Updates document metadata in Pinecone with extracted topics

Requirements

1. Use Node.js (Express) for the backend and React for the frontend
2. Integrate with Pinecone for vector storage and retrieval
3. Use either the Claude 3.5 API or GPT-4 API for text analysis and question answering
4. Implement a basic RAG system
5. Create a Python script for topic modeling
6. Provide clear setup instructions, including environment variables for API keys
7. Include basic error handling and input validation
8. Optimize for performance, considering rate limits of external APIs

Evaluation Criteria

- Successful integration of Pinecone and chosen AI API
- Effective implementation of the RAG system
- Quality of document processing and information retrieval
- Frontend design and user experience
- Code quality, organization, and documentation
- Error handling and edge case management
- Creative problem-solving in system design and implementation

Time Limit

Candidates should aim to complete a working prototype within 2 hours. Quality of implementation is valued over completeness.

Submission

Provide a GitHub repository with your solution, including:

- Source code for backend, frontend, and Python script
- README with setup instructions and any assumptions made
- Brief explanation of your approach and any challenges faced