AI Engineer - Trial Task

Business context

In the evolving healthcare landscape, pharmaceutical companies and payers are increasingly adopting value-based contracts (VBCs) to align drug pricing with patient outcomes. However, negotiating, structuring, and closing these contracts remains a complex and time-consuming process, often hindered by data fragmentation, regulatory challenges, and lengthy approval cycles.

Our company has a couple of AI initiatives currently ongoing, but for this trial task, we are focusing on developing a document processing and analysis system that enhances how pharma companies and payers interact with VBCs. This system aims to:

- Automate document extraction and classification for faster contract assessment
- Enable intelligent search and query capabilities to retrieve key contractual insights instantly
- Provide real-time insights and compliance checks to streamline negotiations

By leveraging AI-driven automation, we aim to help pharma and payers close value-based contracts faster and more efficiently.

Overview

Create a document processing and analysis system that demonstrates your ability to build production-ready AI applications. The system should handle both structured and unstructured data, incorporating multiple AI capabilities.

Core Requirements

1. Document Processing Pipeline

- Build a system that can process PDF documents and extract key information
- Handle multiple document types (contracts, reports, invoices)
- Implement intelligent document classification
- Extract structured data from semi-structured documents

2. Custom ChatBot Interface

- Create a chat interface that allows users to query the processed documents
- Implement semantic search capabilities
- Build a context-aware response system
- Handle follow-up questions

Technical Requirements

Backend (Python)

- Use LangChain for the document processing pipeline
- Implement vector storage for semantic search

- Create a FastAPI backend service
- Include proper error handling and logging

Frontend (React/JavaScript - minimal)

- Build a responsive web interface
- Implement real-time chat functionality
- Create a document upload interface
- Show processing status and results
- Write frontend tests

Integration

- Implement proper API documentation (OpenAPI/Swagger)
- Include authentication- Optional
- Handle document storage securely

Evaluation Criteria

- Code quality and organization
- Performance optimization
- System architecture decisions
- Error handling and edge cases
- Documentation quality
- Working prototype (code repository).
- Explanation of AI tool choices.

Time Limit: 8 hours

Recommended Tech Stack:

1. LangChain
2. FastAPI
3. React
4. PostgreSQL (for structured data)
5. Vector DB
6. OpenAI API

Note: Use the pdfs attached to the email.