Practice Task: Mini AI Document Query System

Objective:

Build a small system where users upload documents, store and index them, and use multiple AI agents to answer questions based on the document content.

Requirements:

1. FastAPI

- Create endpoints:

• POST /upload: Upload PDF or text files

• POST /ask: Submit a question

• GET /documents: List uploaded documents

• GET /health: Health check endpoint

2. Document Upload

- Validate PDF/TXT files.

- Save files to a local uploads directory.

- Extract text from documents.

- Handle errors and invalid inputs.

3. Postgres Integration

- Create a table named ‘documents’ with fields:

id, filename, filepath, uploaded_at, status

- Store metadata when a file is uploaded.

- Provide basic CRUD operations if required.

4. Pinecone Integration

- Chunk extracted text.

- Generate embeddings.

- Store chunks in Pinecone using namespace = document_id.

- Implement simple similarity search.

5. LangGraph Multi-Agent Flow

Create a graph with these nodes:

- Router Agent: Determines if a user question requires document retrieval or a simple response.

- Retriever Agent: Queries Pinecone and returns relevant chunks.

- Answer Agent: Generates the final answer using retrieved context.

6. End-to-End Workflow

- User uploads a document.

- Document metadata is stored in Postgres.

- Content is embedded and indexed in Pinecone.

- User asks a question.

- Router Agent directs the flow.

- Retrieval and answer generation occur.

- Final answer is returned via the API.

7. Deliverables

- Functional API service.

- Database schema and migrations.

- LangGraph implementation with defined agents.

- Pinecone vector index integration.

- README with instructions to run the project.