TASK 1

Title

RAG-Based Question Answering System

Objective

Assess your ability to build an applied AI system using embeddings, retrieval, background jobs, and APIs.

Problem Statement

Create an API that allows users to upload documents and ask questions based on those documents using a **Retrieval-Augmented Generation (RAG)** approach.

Functional Requirements

Your system should:

1. Accept documents (minimum two formats, e.g., PDF and TXT)
2. Chunk and embed documents
3. Store embeddings in a local vector store or cloud based vector store (e.g., FAISS/ PINECONE)
4. Retrieve relevant chunks based on user queries
5. Generate answers using an LLM

Technical Requirements

- FastAPI or Flask
- Embedding generation
- Similarity search
- Background job for document ingestion
- Request validation (Pydantic)
- Basic rate limiting

Mandatory Explanations

You must include a document explaining:

- Why you chose a specific chunk size
- One retrieval failure case you observed
- One metric you tracked (latency, similarity score, etc.)

Deliverables

- GitHub repository link
- Architecture diagram (draw.io or hand-drawn)
- README.md with setup and usage instructions

Constraints

- Do not use default RAG templates without explanation
- Avoid heavy frameworks unless justified

Evaluation Criteria

- Chunking strategy
- Retrieval quality
- API design
- Metrics awareness
- System explanation clarity