Technical Assessment: AI Intern (Python)

Role: AI Intern

Topic: Local RAG (Retrieval-Augmented Generation) Pipeline

Est. Time: ~4-8 Hours (Deadline: 48 Hours)

1. Objective

We want to evaluate your ability to build practical GenAI applications using **local LLMs** and **Python**. Your goal is to build a "Document Q&A Assistant" that allows a user to upload a PDF (e.g., an employee handbook) and ask questions about it.

The system must run locally using **Ollama** to ensure data privacy and demonstrate your ability to work with open-source models.

2. The Challenge

Build a simple web app (Streamlit/Gradio) that performs the following steps:

1. **Ingest:** Load a PDF file.
2. **Process:** Split the text into manageable chunks.
3. **Embed:** Convert chunks into vector embeddings and store them in a local vector database.
4. **Retrieve:** When a user asks a question, find the top 3-5 most relevant chunks.
5. **Generate:** Pass the retrieved context + the question to a local LLM (via Ollama) to generate a concise answer.

Critical Requirement

The bot must answer based **ONLY** on the provided document.

- **Scenario A:** The answer is in the document -> Provide the answer.
- **Scenario B:** The answer is *not* in the document -> Explicitly state: *"I cannot find the answer to that question in the provided document."* (Do not hallucinate or use outside knowledge).

3. Tech Stack Requirements

You must use the following technologies. Please do not use paid APIs (like OpenAI GPT-4) for this task.

- **Language:** Python 3.9+
- **LLM Inference**: Ollama
- **Orchestration:** LangChain, LlamaIndex, or pure Python.
- **Vector Database:** ChromaDB, FAISS, or Qdrant (must run locally).
- **Embedding Model:** Any lightweight open-source model.

4. Deliverables

To complete this assessment, submit the following:

A. GitHub Repository

- **Source Code:** Clean, modular Python code. Avoid putting everything in a single main.py file or a Jupyter Notebook unless it is for exploration only.
- **`requirements.txt`:** A list of all dependencies.
- **README.md:** This is crucial. It must include:
    - **Prerequisites:** Which Ollama model to pull (e.g., ollama pull mistral).
    - **Installation:** How to set up the environment.
    - **Usage:** Exact command to run the bot.
    - **Architecture:** A brief sentence on why you chose your specific chunk size or overlap strategy.

B. Demo Video

- Record a video of your working application.
- Briefly show your code structure.

5. How to Submit

Please reply to the hiring email with:

1. The link to your GitHub repository.
2. The link to your Demo Video.

Good luck! We look forward to seeing your implementation.