## **AI Hiring Assignment:**

Hi,

Thanks for your time. Below are 2 assignments that you have to execute. Please choose 1 of the 2 assignments and develop a full solution. Once you have built a solution to your satisfaction, please reach out to us and an interview will be scheduled.

## **Project Assignment 1 (LLM-powered): Chatbot to Chat with a MySQL database**

### **Project Title:**

**"LLM-powered Chatbot to interact with a MySQL Knowledge Base"**

### **Objective:**

Build a chatbot that answers user queries by intelligently retrieving and generating responses from structured data stored in a MySQL database — using **LLM-based techniques** (RAG, prompt engineering, embeddings, etc.).

### **Requirements:**

#### **1. Database Layer**

* Store the attached data in a MySQL database



#### **2. Data Retrieval & Processing**

* Use **SQL-based retrieval** to pull relevant rows based on user intent.
* Alternatively, or additionally, extract data from MySQL and create **embeddings** (e.g., using OpenAI or HuggingFace models), then store them in a vector store (like FAISS or ChromaDB).

#### **3. LLM Integration**

* Use a Retrieval-Augmented Generation (RAG) setup:
  + Retrieve top-k results from MySQL or vector DB based on user query.
  + Feed the context into an LLM (e.g., GPT-4, LLaMA) via prompt engineering to generate natural responses.
* Implement fallback logic when no relevant data is found.

#### **4. Chat Interface**

* Build a simple CLI or web UI (Streamlit/Gradio is fine).
* Maintain conversation context for better follow-up queries.

#### **5. Questions to be answered**

* What are the monthly sales across platform1 since Jan 2025
* What is the share of units sold across various platforms since Jan 2025
* Can you tell me the top 5 days with the highest daily units sold

### **Deliverables:**

* ✅ Source code (Python preferred) with clear folder structure.
* ✅ Sample SQL dump of your database.
* ✅ Preprocessing scripts for embedding creation.
* ✅ README with:
  + Project overview
  + Tech stack
  + How to run the bot
  + Design decisions

### **Optional (but impressive):**

* Deploy your chatbot on Hugging Face Spaces or Streamlit Cloud.
* Submit a demo video (2–3 min) walking through the flow.

## **Project Assignment 2: Chatbot using Documents (LLM-based)**

### **Project Title:**

**"Ask-the-Docs: AI Assistant for Company Policies"**

### **Objective:**

Develop a chatbot that can answer user queries based on a collection of PDF and Word documents

### **Requirements:**

* Use tools like LangChain or Haystack to:
  + Parse and chunk documents.
  + Generate embeddings using OpenAI, Cohere, or HuggingFace models.
  + Store them in a vector database like FAISS or ChromaDB.
* Implement a simple frontend or CLI where users can ask questions.
* Use RAG (retrieval-augmented generation) to generate context-rich responses.

### **Bonus Tasks:**

* Add support for real-time document uploads.
* Implement feedback collection for answers to improve the system.

### **Sample Data:**



### **Deliverables:**

* Source code with requirements.txt .
* Preprocessed document samples.
* Demo video (optional but highly recommended).
* README explaining architecture, tools used, and how to run the bot.