Code Challenge: Q&A Chatbot

Job Position: AI Engineer

Case Study

You are given:

- A **tabular dataset**:Fraud Dataset Kaggle , which contains information about simulated credit-card transactions containing both legitimate and fraudulent records.
- A **document**: choose between Understanding Credit Card Frauds or 2024 REPORT ON PAYMENT FRAUD

Tasks

As an AI engineer at our company, your task is to develop an internal Agent Chatbot tool capable of extracting meaningful information from both tabular dataset and file. The tool should provide insightful responses to a questions, such as:

Unset

```
Q: "“How does the daily or monthly fraud rate fluctuate over the two-year
period?”
---
Q:"Which merchants or merchant categories exhibit the highest incidence of
fraudulent transactions?"
---
Q: "What are the primary methods by which credit card fraud is committed?"
---
Q: What are the core components of an effective fraud detection system,
according to the authors?"
---
Q: How much higher are fraud rates when the transaction counterpart is
located outside the EEA?"
---
Q: What share of total card fraud value in H1 2023 was due to cross-border
transactions?"
```

a) Framework

- Preferably use Python.
- Store tabular data in your database.
- Integrate with existing online or offline Language Models (LLM) such as chatGPT, BERT, etc. (We allow you to use free or open source model)
- No limit on the use of RAG tools (e.g., Pinecone, Faiss).
- Utilizing AI agent libraries like PydanticAI or CrewAI is allowed.
- Prompt design is not limited to a single layer (multiple layers allowed).
- Provide quality scoring for your chatbot’s answers.

b) UI

- Develop a simple UI for the chatbot using Streamlit or Gradio.

What we Assess

1. Accuracy: The primary focus is on the accuracy and relevance of generated responses.
2. Coverage: Evaluate the adaptability to a variety of management questions. 3. Readability: Assess how well you organize code structure and name variables for clarity and modifiability.
4. Exception Handling: Examine how you handle invalid data, edge cases, and make assumptions regarding those cases.
5. Performance: Evaluate the overall system performance.
6. Data Processing: Review the method of embeddings (and Retrieval Augmented Generation), preprocessing, and postprocessing.

Deliverables

Your solution needs to be delivered as a link to a Git repository in your personal Github account, preferably a private repository (please create one if you don’t have any). It’s better to write a clear README to guide us, the reviewers, to understand your code easily. Please provide a video and screenshots of your application.