**💼 Smart Financial Agent for Transaction Insights**

**AI Innovators Hackathon 2025 — Official Project Documentation**

**Executive Summary**

This project introduces a smart financial agent designed to transform raw retail transactions data into actionable insights using AI. Built for fast, intelligent querying, the system helps businesses and individuals detect anomalies, summarize transaction behavior, and explore patterns using natural language — without requiring any manual filtering or data science expertise.

By leveraging **semantic search**, **transformer-based LLMs**, and **vector similarity with FAISS**, our system delivers relevant results and natural language explanations based on transactional context.

**Problem Statement**

Retail chains and financial institutions in Jordan often deal with large volumes of transaction records. These records are:

* It is difficult to search or summarize without technical knowledge.
* Filled with repetitive failures, missing values, or suspicious activity.
* Not easy to query using natural questions (e.g., “What happened last weekend at C Mall?”).

**Proposed Solution**

We built an **AI-powered assistant** that allows anyone to:

* Ask natural-language questions about transactions (e.g., “What failed yesterday in City Mall?”).
* Receive clear, context-aware responses.
* Detect trends like repeated failures, high-spend patterns, or unusual activity.

Our approach:

* Embeds all transaction records semantically using MiniLM.
* Uses FAISS for fast vector search of top-matching entries.
* Applies Flan-T5 LLM to summarize and respond to queries.

**System Architecture**

CSV Data 🡪 Preprocessing and Corpus 🡪 Sentence Embedding (MiniLM) 🡪 FAISS Indexing 🡪 User Query Embedding 🡪 LLM (Flan-T5) Prompt + Response 🡪 Result and Summary

**Technologies Used**

|  |  |
| --- | --- |
| Purpose | Tool / Library |
| Language & Logic | Python 3.10+ |
| Data Processing | pandas, numpy |
| Semantic Embedding | sentence-transformers (MiniLM) |
| Similarity Search | faiss |
| Natural Language Output | transformers (Flan-T5) |
| Visualization / Logs | Streamlit (optional), CLI |
| Query Memory | Python Dicts + Timestamps |

**Features Implemented**

|  |  |
| --- | --- |
| Feature | Description |
| Natural Language Queries | **Ask any question about the data** |
| Semantic Search (Top-K) | **FAISS + MiniLM search for similar entries** |
| LLM-Powered Reasoning | **Flan-T5 interprets the results** |
| Automatic Summaries | **Optional pattern summaries** |
| Query History | **Logs query/response with timestamps** |
| Clean Output Formatting | **Readable and structured replies** |
|  |  |

**Team Role Breakdown**

|  |  |
| --- | --- |
| Member | Responsibilities |
| Nahla Bader | AI pipeline, embeddings, LLM integration |
| Rehab Amarah | Telegram bot (optional), UI design |
| Mohammad Hezan | Data cleaning, dataset management |
| Abdullah Mustafa | testing |

**References**

* MiniLM (SBERT): <https://www.sbert.net>
* FAISS: <https://github.com/facebookresearch/faiss>
* Flan-T5: https://huggingface.co/google/flan-t5-large
* Pandas: https://pandas.pydata.org/
* Hugging Face Transformers: https://huggingface.co/transformers