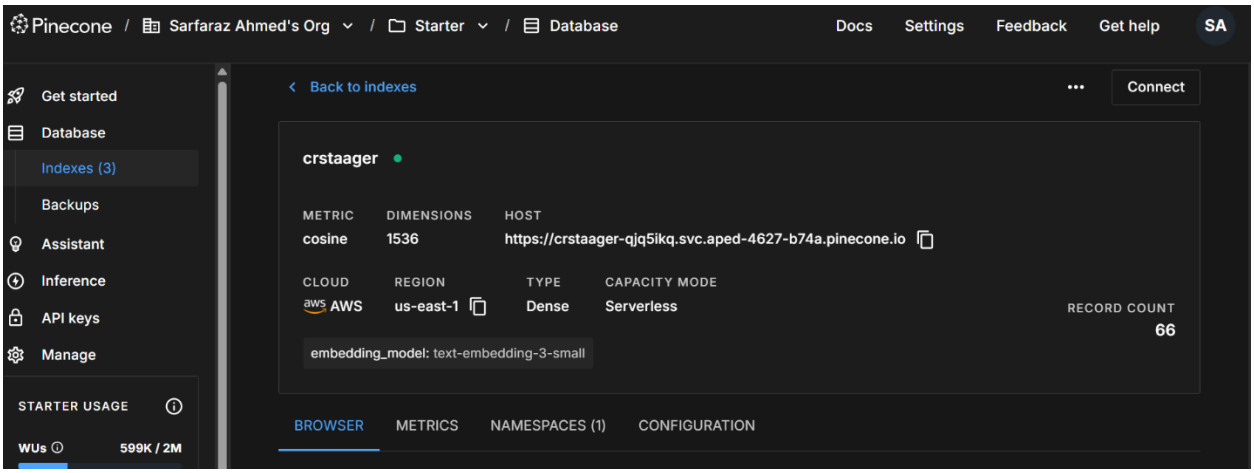
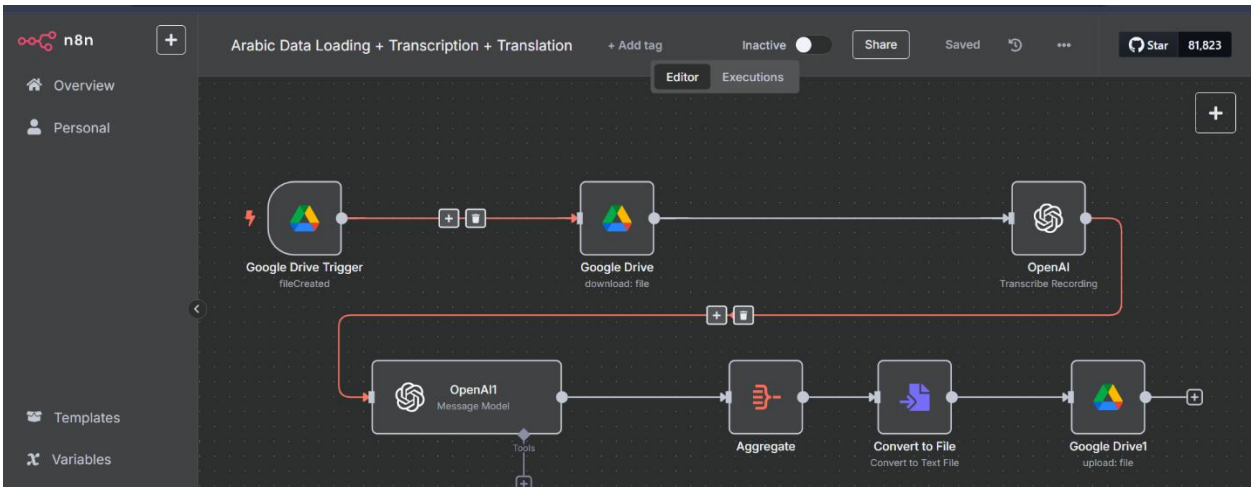
AI Position Task:

Step 1: I created a vector database at https://www.pinecone.io/

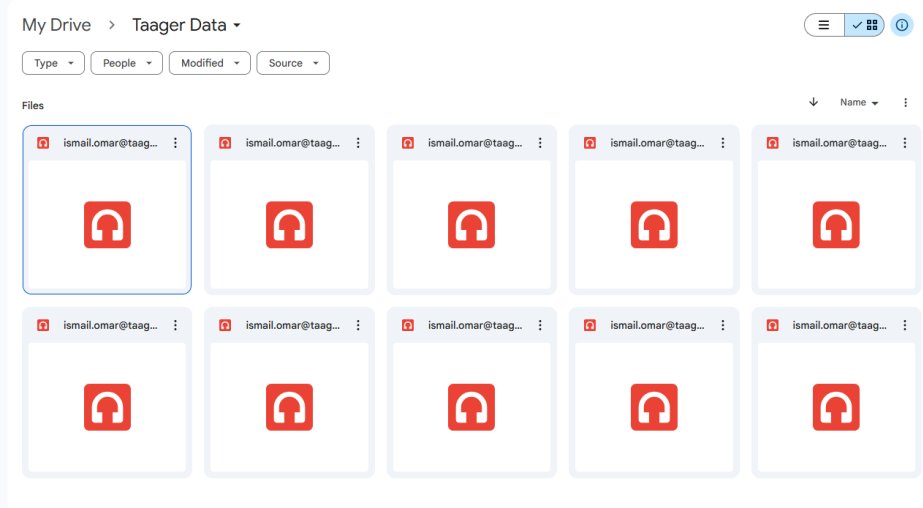


Step 2: I created an **n8n workflow using Google Drive and other nodes**, as shown in the picture (including OpenAI). I connected it to a Google Drive folder containing audio data. When I ran the workflow, it successfully **transcribed the data, translated it into English, and saved it** in another Google Drive folder.



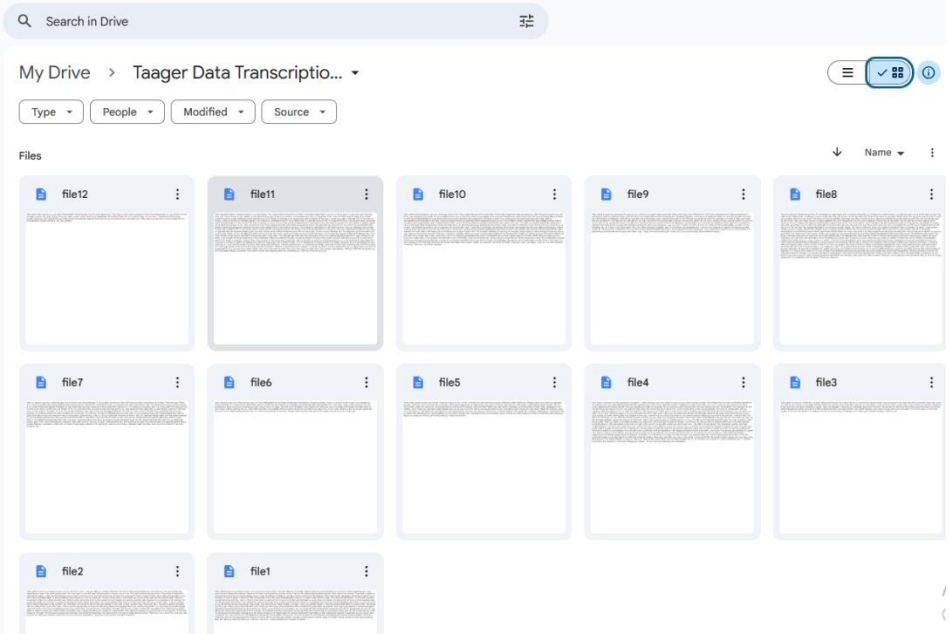
Input to n8n workflow:

Arabic Audio Input Data: https://drive.google.com/drive/folders/12mS9c2ZA03-FFVvECg1KYd8kNQ7HhlZx?usp=sharing

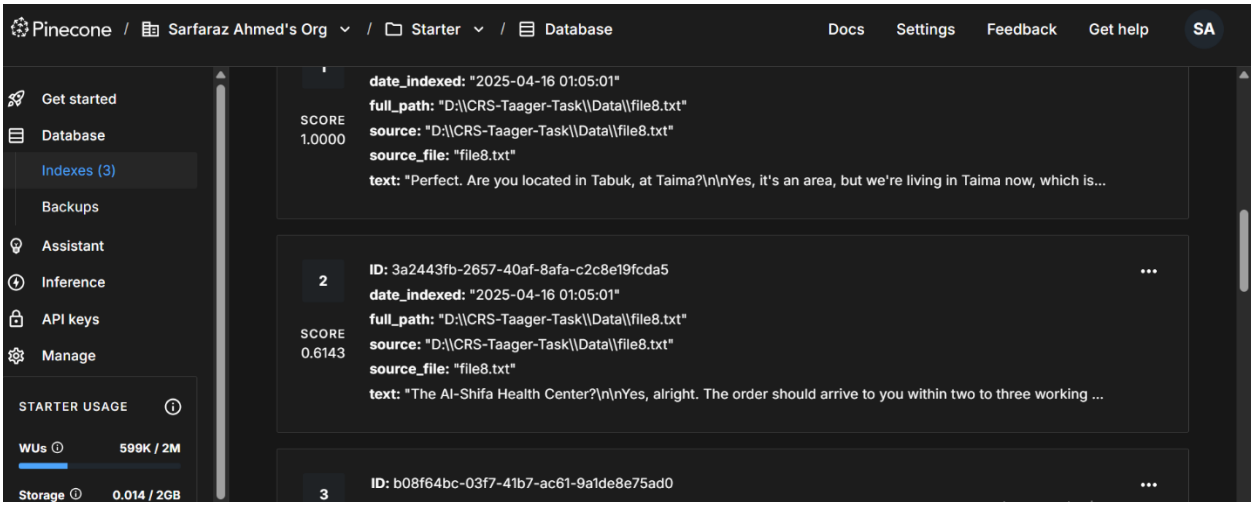


Output From n8n workflow:

Arabic Transcribed + Translated Dataset Link: https://drive.google.com/drive/folders/17N8Iac2oEIQR3m3FDPdi9uLGZBPcMhgH?usp=sharing



Step 3: I then indexed the entire translated text data into a Pinecone vector database using the **text-embedding-3-small model** and **LangChain**, with a chunk size of **500** and an overlap of **50**. All the code is available in **indexing.py.**

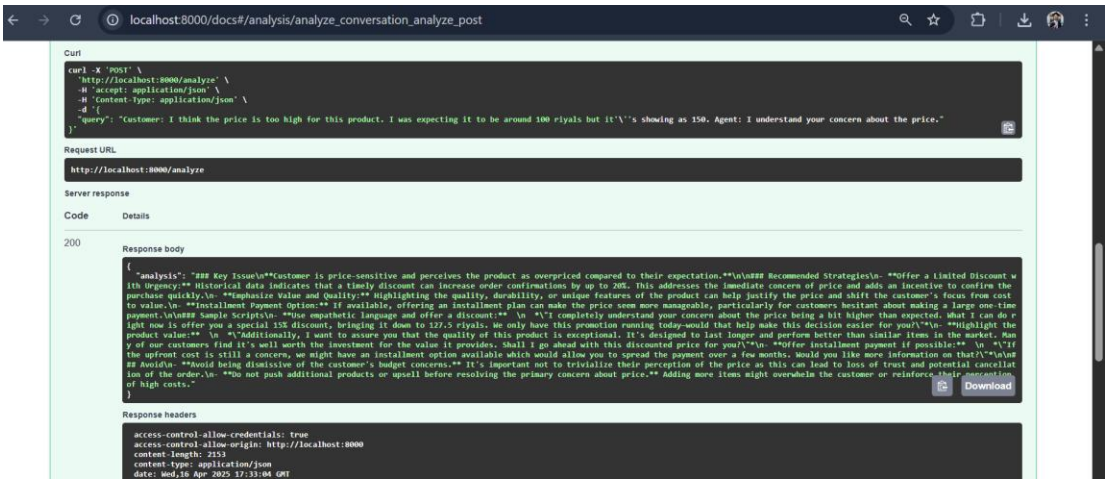


Step 4: Finally, I developed an agentic RAG system using LangChain and LangGraph. I integrated the Pinecone vector database and OpenAI's GPT-4o LLM with custom prompt instructions and utilized FastAPI to create an endpoint.

For reference, please see the following files:

- app.py
- prompt.py
- api.py

Output:



Key Technologies:

- **Framework:** Langchain + Langgraph
- **LLM:** GPT-4 o
- **RAG Pipeline**: Pinecone vector database for retrieving historical interactions and openai's embedding model.
- **Automation**: n8n workflow for audio-to-text processing and translation.
- **API**: FastAPI for serving recommendations.

Core Features:

- Automated Arabic audio transcription → English translation → structured dataset.
- Real-time analysis of customer objections (e.g., price sensitivity, confusion).
- Actionable strategies (discounts, empathy scripts, urgency tactics) grounded in historical data.