

JUELITES

Adnan Ibrahim Sawalha
Ayham Nedat Al-Hindi
Abdelhamid Ghannam
Moayad Khaldoun Abdo

May 10, 2025

Executive Summary

This project implements a Smart Financial Advisor using OpenAI and n8n, capable of answering questions based on transactional data from Supabase, with memory for long-term context.

Introduction

Project Objective

This project aims to develop a Smart Financial Advisor powered by OpenAI's language models and orchestrated through n8n. It allows users to ask financial and operational questions about retail transaction data—such as “What is the most purchased item in store C?”—and receive immediate, data-backed answers. The system supports long-term memory, enabling context-aware conversations across multiple queries.

Solution Overview

The solution integrates:

- 1- Supabase as the backend database to store and retrieve transaction data and conversational memory.
- 2- n8n as the workflow automation engine to manage the flow of user inputs, data queries, formatting, and AI interaction.
- 3- OpenAI GPT API to interpret questions and generate intelligent responses using natural language processing.

By combining these tools, the system eliminates the need for technical knowledge to query structured data, making business intelligence accessible through conversational AI.

System Architecture

The Retailify Bot is a smart financial assistant designed to interact with retail transaction data and respond intelligently to user queries. It leverages a hybrid architecture combining automation workflows (via n8n), vector similarity search (using Supabase), and OpenAI's language models. Below is a breakdown of its components and their interactions:

1. Input Handling

The bot accepts input from Telegram, either as text or audio. Upon receiving a message, the bot immediately replies to the user with a brief confirmation (e.g., "Just a moment...") while it processes the request.

If the input is audio, it is transcribed to text using the AssemblyAI API.

2. Retrieval-Augmented Generation (RAG)

After receiving input from the user, we perform the following steps:

- a. Context Retrieval: We query the transaction dataset to retrieve all records and store them in a variable called context.
- b. Memory Retrieval: We fetch the most recent entry from the memory database and store it in a variable called memory.
- c. Query Resolution: Using the context, memory, and the user's query, we construct a prompt and send it to the OpenAI API to generate a response.
- d. Memory Update: Finally, we store the user query along with the generated response into the memory database. This allows us to maintain continuity and context for future interactions.

3. Output Handling

The final response is returned to the user in the same format they used:

- If the input was text, the reply is sent as a text message.
- If the input was audio, the text response is converted to speech and sent back as an audio message.

4. Scheduled Reporting

Every three hours, the bot automatically generates a summary report of the entire transaction dataset.

- This report is formatted as a concise message and sent to the Telegram channel or user, providing an up-to-date financial snapshot.