Gerwin D. Dogui-is

AI-Powered Inventory Automation with n8n

Overview

This project demonstrates an AI-driven inventory automation system built using n8n, which is hosted locally with Docker and exposed securely with Ngrok. The goal was to automate the extraction of details regarding purchases from receipts (images or PDFs) and log them into a structured Google Sheet for inventory management. This was achieved by integrating the APIs of Telegram, OpenAI, and Google Sheets, creating a seamless end-to-end pipeline that processes receipts, extracts product data, and automatically logs inventory.

Tools Used

- n8n – Orchestration engine for building no-code/low-code automations.
- Docker – Used for self-hosting n8n in a containerized environment.
- ngrok – Secure tunneling to expose local n8n server for Telegram API webhook.
- Telegram Bot API – For input (receipts) and feedback (confirmation).
- OpenAI GPT API – AI agent for OCR, data parsing, and JSON structuring.
- Google Sheets API – For structured inventory storage and record-keeping.

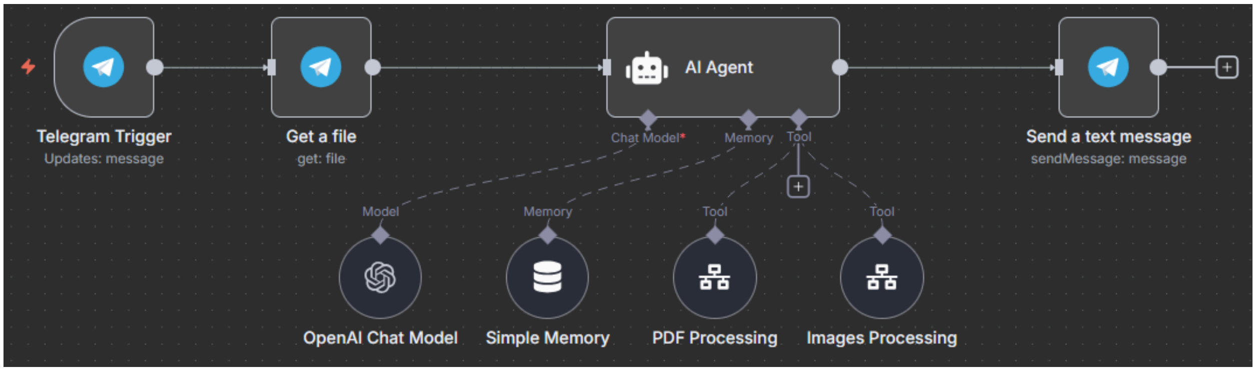


Fig 1. Main Workflow

The workflow begins when the Telegram bot receives a message from the user. The Telegram Trigger node picks it up and extracts the file_id, which is then used to download the file. Once the file is available, an AI Agent makes a request to the OpenAI API. At this point, the system decides which path to take depending on the file type, PDF or image. When the data has been processed, the structured details are written into Google Sheets, and the Telegram bot sends a confirmation back to the user, letting them know that the receipt has been recorded.

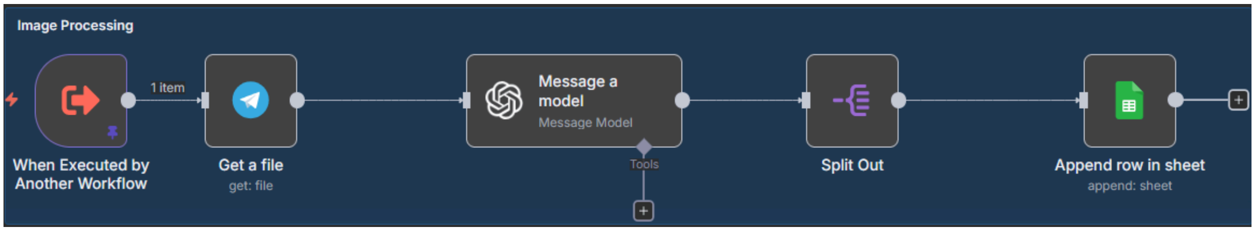


Fig 2. Image Processing Workflow

If the file is an image, the Image Processing Workflow takes over. The file path is passed in, and the system downloads the image. From there, the image is sent to the OpenAI API, which extracts the text and returns it as structured JSON data. The JSON is then split into individual items, and each item is appended to Google Sheets, ensuring that every product from the receipt is logged.

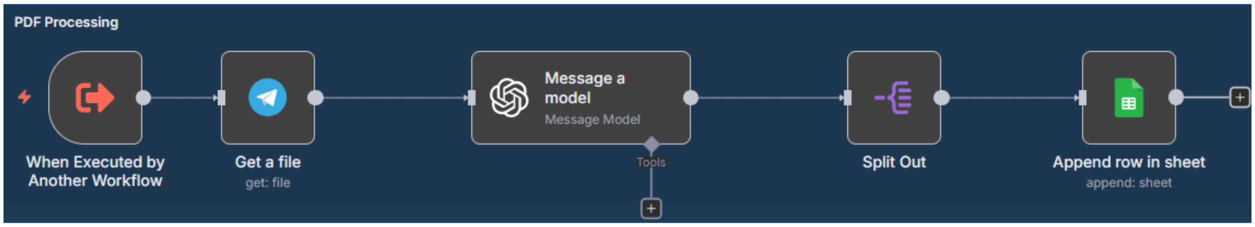
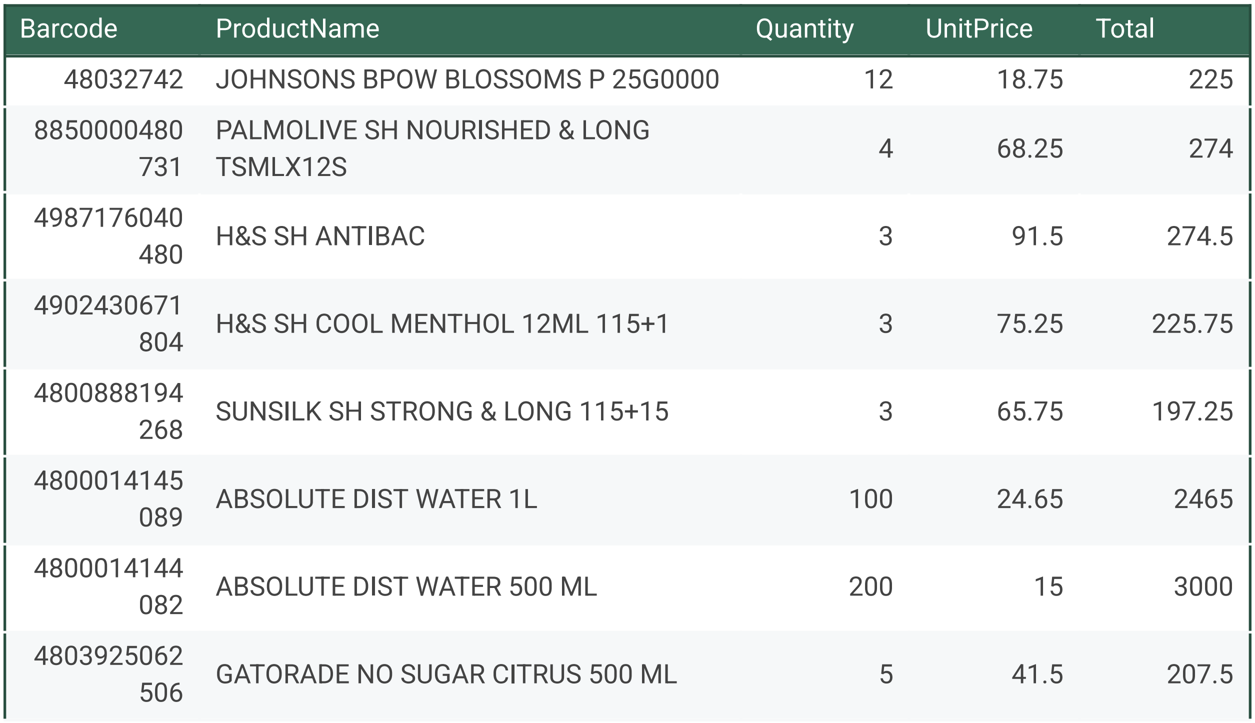


Fig 3. PDF Processing Workflow

The PDF Processing Workflow follows a similar approach, but it's tailored for handling PDFs. The workflow downloads the file, then calls the OpenAI API with a prompt designed specifically for text-rich PDF receipts.

The API response is once again returned as structured JSON data, which is split into items and added to Google Sheets. This keeps the process consistent across both PDFs and images.



Barcode	ProductName	Quantity	UnitPrice	Total
48032742	JOHNSONS BPOW BLOSSOMS P 25G0000	12	18.75	225
8850000480731	PALMOLIVE SH NOURISHED & LONG TSMLX12S	4	68.25	274
4987176040480	H&S SH ANTIBAC	3	91.5	274.5
4902430671804	H&S SH COOL MENTHOL 12ML 115+1	3	75.25	225.75
4800888194268	SUNSILK SH STRONG & LONG 115+15	3	65.75	197.25
4800014145089	ABSOLUTE DIST WATER 1L	100	24.65	2465
4800014144082	ABSOLUTE DIST WATER 500 ML	200	15	3000
4803925062506	GATORADE NO SUGAR CITRUS 500 ML	5	41.5	207.5

Table 1. Sample Data

Conclusion

This project shows how automation can make inventory tracking easier. Using n8n, Docker, ngrok, and APIs from Telegram, OpenAI, and Google Sheets, the system moves data from receipts into a structured sheet with little manual work. The result is a reliable way to record inventory without typing data by hand. It is a clear example of how automation and AI can handle routine tasks in a practical setting.

Limitation

This project has a limitation related to cost and usage limits. The system depends on the OpenAI API, which enforces token limits. When many images are processed, the token allowance can run out. Once this happens, the user must wait for the token quota to reset, which usually occurs monthly. But it can be easily overcome by purchasing more tokens.