Homework 01 Report: Multimodal Agent for Supermarket Bill Analysis

Course: Agentic AI for Business and FinTech (FTEC5660)

Student Name: Kang Yuanshi **Student ID:** 1155243696

1. Problem Description

The objective of this assignment was to develop an Agentic AI system capable of interpreting visual data from multiple supermarket bills to answer specific financial queries. As outlined in the course materials, an agentic system must perceive its environment (in this case, digital images of bills), make decisions based on goals and observations, and execute actions to achieve those objectives. The specific requirements involved processing seven images simultaneously to calculate total spending, reconstruct original prices before discounts, and intelligently reject irrelevant queries that fall outside the domain of shopping data.

2. Methodology and Design Patterns

To solve this problem, I implemented a solution that leverages **Context Engineering** and **Semantic Routing** patterns to ensure reliability and accuracy.

2.1 Context Engineering and Multimodal Input

The foundation of the solution is **Context Engineering**, which involves building a complete informational environment before generation. Rather than processing images individually, the system aggregates all seven bill images into a single context window. The load_bill_images function retrieves the files, and the image_to_base64 function standardizes the inputs. This allows the Large Language Model (LLM) to "scan the scene" comprehensively, treating the collection of bills as a unified dataset rather than isolated data points. By providing the model with the full visual context alongside specific system instructions, the agent can perform cross-document reasoning required for the total summation tasks.

2.2 Semantic Routing via Prompt Design

A core challenge of the assignment was handling distinct types of user queries (Total Spending vs. Original Price vs. Irrelevant Questions). To address this, I applied the **Routing Pattern**, which adds conditional logic to the workflow, enabling the agent to choose the most appropriate path based on the input.

Instead of using an external classifier, I implemented an LLM-based routing mechanism directly within the system prompt. The system prompt explicitly instructs the model to classify the user's intent into one of three logical paths:

1. **Total Spending Path:** If the user asks about total money spent, the model acts as an aggregator, identifying the "Total" or "Payment" fields from each image and summing them.
2. **Original Price Path:** If the user asks about the pre-discount cost, the model switches its reasoning strategy to identify "Subtotal" fields or mathematically reconstruct the price by adding "Savings" back to the "Total."
3. **Rejection Path:** If the query is unrelated to the bills (e.g., "President of USA"), the model executes a "Prompt Rejection" route, returning the specific string "Irrelevant query."

This design shifts the system from a fixed, linear workflow to a context-aware execution flow, ensuring that the agent does not hallucinate answers for out-of-domain questions.

3. Technical Implementation

The solution leverages the langchain-google-genai library to interface with the **Gemini 2.5 Flash Lite** model. This model was selected for its high efficiency and strong multimodal reasoning capabilities. To ensure the financial calculations were precise and reproducible, I set the model temperature to 0. This constraint forces the model to be deterministic, which is a critical requirement when building reliable agentic applications. The implementation uses a "Single Complex Prompt" strategy where the instructions serve as the cognitive engine, guiding the model through the steps of extracting text via OCR, performing arithmetic, and formatting the final output.

4. Experimental Results

The system was tested against the three mandatory scenarios defined in the assignment:

Query 1 (Total Spending): The agent successfully identified the final payment amount on all seven bills, including those with complex layouts. It correctly summed the values ($394.70, $514.00, $102.30, etc.) to arrive at the correct total of **$1974.35**.

Query 2 (Price Without Discount): The agent demonstrated advanced reasoning by locating the "Savings" or "Subtotal" sections. It performed itemized calculations for each bill (e.g., adding the savings of $48.28 back to the total of $394.72 for Bill 1) and aggregated these values to determine the total original price of **$2,307.24**.

Query 3 (Irrelevant Query): When asked "Who is the president of USA?", the system correctly utilized the routing logic to identify the query as out-of-scope and returned the required response: "**Irrelevant query.**"

5. Conclusion

The implemented solution successfully demonstrates the application of Agentic AI patterns to a practical financial task. By combining **Context Engineering** to manage multimodal data and **Semantic Routing** to arbitrate user intent, the system achieves the goal of being a robust and reliable application. The agent moves beyond simple text generation to perform perception, reasoning, and decisive action, fulfilling the requirements of the homework.