Agentic AI Implementation for Resort Management

Assignment By: Revanth Prasad Yedla
Date: 27 December 2025

---

1 Introduction and Problem Statement

1.1 Overview

In the modern hospitality sector, managing a resort with extensive data requires automated coordination and management. This assignment utilizes an Agentic AI framework to bridge the gap between complex SQL backends and natural language queries from guests.

1.2 Objectives

The primary objective of this assignment was to develop a chat-based agentic AI that ensures:

1. **Unified Access:** A single natural language entry point for all services.
2. **Data Integrity:** Persistent storage of transactions in a centralized SQL database.
3. **Identity Security:** Mandatory verification protocols before service fulfillment.

2 Agentic AI Framework

2.1 Conceptual Framework

This system utilizes Agentic AI, enabling a supervisor model (Gemini 2.0 Flash) to reason through user intent and delegate tasks to specialized sub-agents, thereby forming a multi-agent system (MAS).

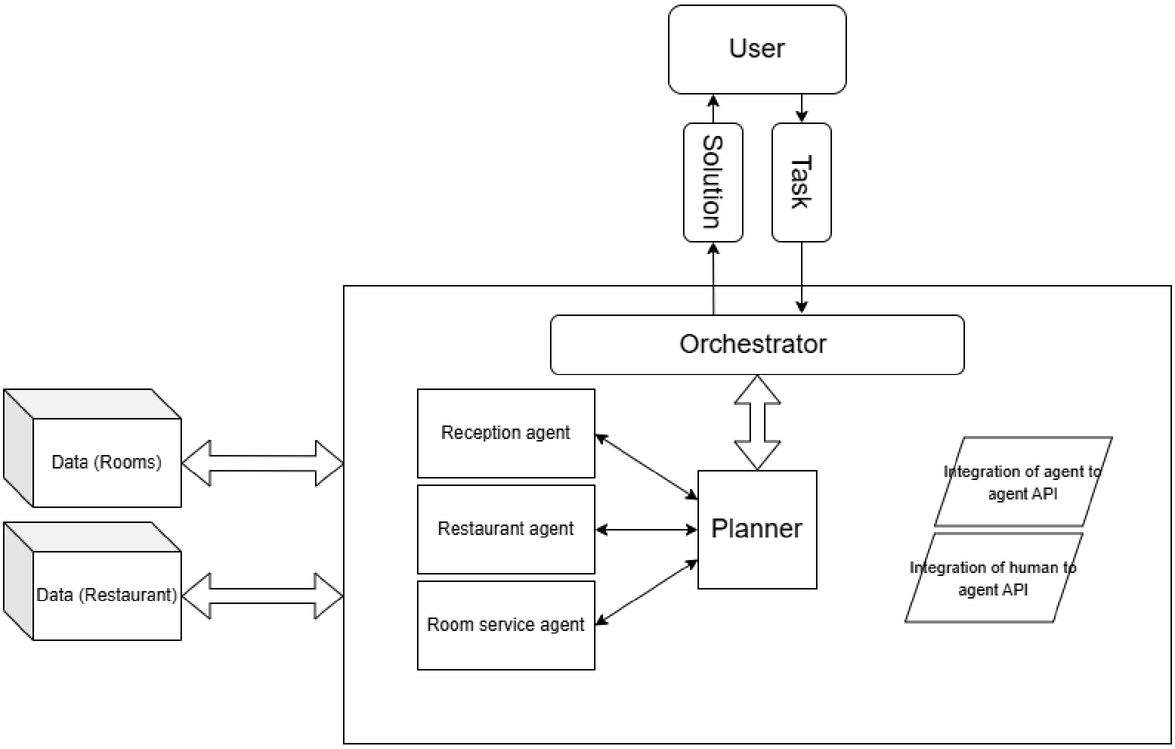


Figure 1: A task-specific Multi-Agent System

2.2 Multi-Agent Orchestration

The system architecture follows a **Manager-Worker pattern**:

- **Resort manager (Orchestrator):** Performs semantic analysis to route tasks.
- **Reception agent:** Manages room inventory and facility inquiries.
- **Restaurant agent:** Navigates menu categories and calculates orderly billing.
- **Room service agent:** Logs amenity requests (laundry, toiletries) to the database.

2.3 Tool Integration and Function Calling

The system's "Agentic" capability is derived from the integration of specialized Python functions, which the LLM utilizes as **Tools** to interact with the resort's physical data. These tools ensure that the AI's responses are deterministic and grounded in the SQL backend, rather than relying on generative hallucinations.

- `verify_guest_stay(room_no, guest_id)`: Acts as the primary security handshake, authenticating users against the dummy guest registry before permitting transactional actions.
- `get_available_rooms()`: Performs a live scan of the dummy room inventory to provide real-time availability and categorized pricing.
- `get_menu_by_category(category)`: Dynamically retrieves item names and prices from the 9-category Excel-synced menu database.
- `place_restaurant_order(room_no, item, qty, guest_id)`: Executes a multi-step transaction that logs orders into the `food_orders` table and calculates the immediate total for orderly payment.
- `request_room_service(room_no, type)`: Facilitates non-billing requests (e.g., laundry, amenities) by logging them into the `service_requests` tracking table.

3 Assignment Coverage and Technical Scope

3.1 Functional Implementation

The project covers the core pillars of automated resort management:

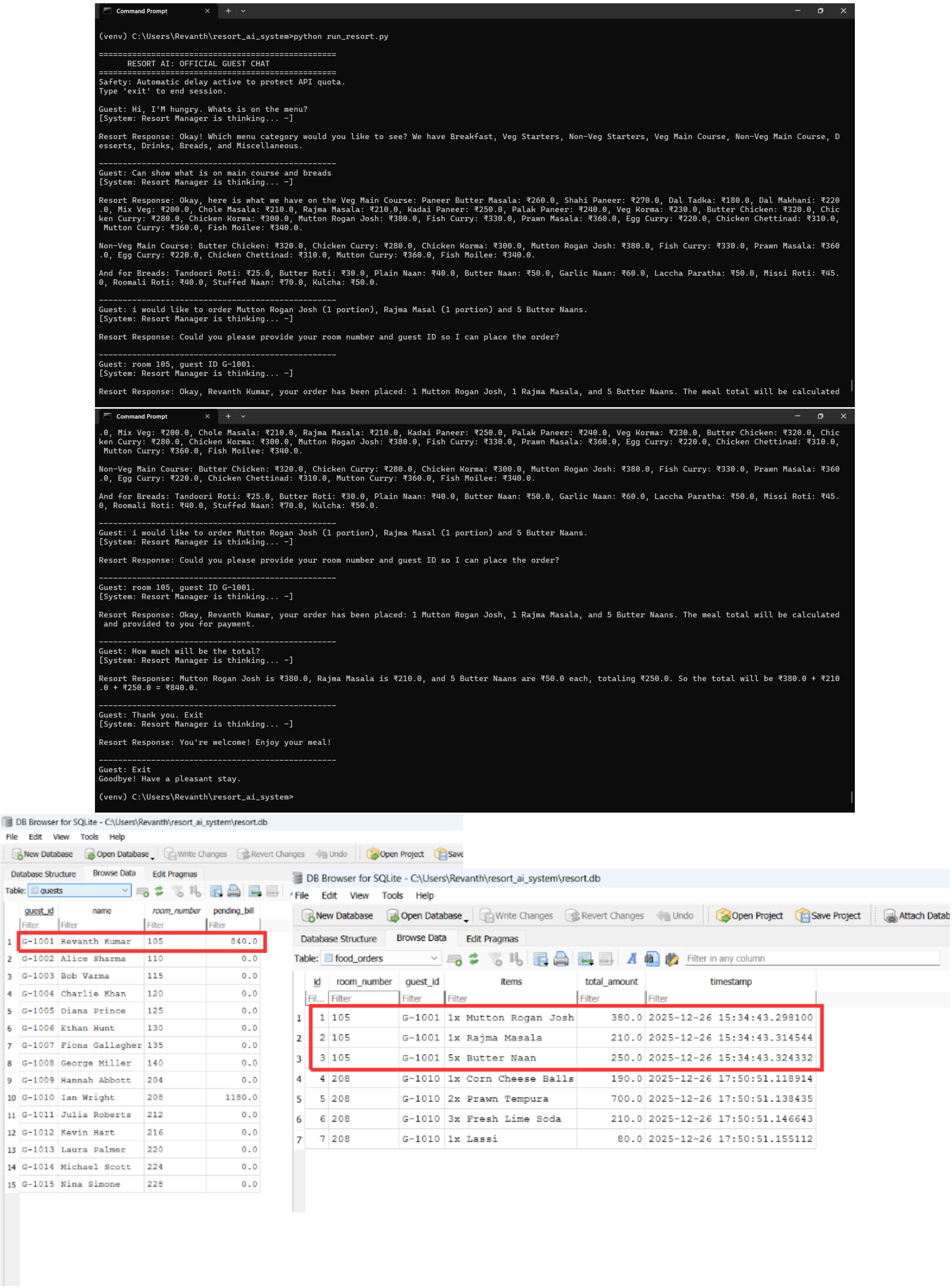
Feature	**Implementation Strategy**	**Status**
Inventory Sync	90 rooms categorized by type and price (dummy)	Complete
Guest Registry	15-guest database for verification (dummy)	Complete
Dining Logic	90-item menu across 9 categories	Complete
Orderly Billing	Real-time calculation of meal-wise totals	Complete
Service Logging	Room service and amenity request tracking	Complete

3.2 Technical Stack

The implementation utilizes a robust Python stack:

- **Intelligence:** Google Gemini 2.0 Flash (Generative AI Client).
- **Database:** SQLite with SQLAlchemy ORM for relational mapping.
- **Data Handling:** Pandas for importing menu items from Excel.
- **Security:** `python-dotenv` for protected API key management.

3.3 Traceability: From Natural Language Intent to Database Entry



```
(venv) C:\Users\Revanth\resort_ai_system>python run_resort.py

==================================================
      RESORT AI: OFFICIAL GUEST CHAT
==================================================
Safety: Automatic delay active to protect API quota.
Type 'exit' to end session.

Guest: Hi, I'M hungry. Whats is on the menu?
[System: Resort Manager is thinking... -]

Resort Response: Okay! Which menu category would you like to see? We have Breakfast, Veg Starters, Non-Veg Starters, Veg Main Course, Non-Veg Main Course, D
esserts, Drinks, Breads, and Miscellaneous.

--------------------------------------------------
Guest: Can show what is on main course and breads
[System: Resort Manager is thinking... -]

Resort Response: Okay, here is what we have on the Veg Main Course: Paneer Butter Masala: ₹260.0, Shahi Paneer: ₹270.0, Dal Tadka: ₹180.0, Dal Makhani: ₹220
.0, Mix Veg: ₹200.0, Chole Masala: ₹210.0, Rajma Masala: ₹210.0, Kadai Paneer: ₹250.0, Palak Paneer: ₹240.0, Veg Korma: ₹230.0, Butter Chicken: ₹320.0, Chic
ken Curry: ₹280.0, Chicken Korma: ₹300.0, Mutton Rogan Josh: ₹380.0, Fish Curry: ₹330.0, Prawn Masala: ₹360.0, Egg Curry: ₹220.0, Chicken Chettinad: ₹310.0,
 Mutton Curry: ₹360.0, Fish Moilee: ₹340.0.

Non-Veg Main Course: Butter Chicken: ₹320.0, Chicken Curry: ₹280.0, Chicken Korma: ₹300.0, Mutton Rogan Josh: ₹380.0, Fish Curry: ₹330.0, Prawn Masala: ₹360
.0, Egg Curry: ₹220.0, Chicken Chettinad: ₹310.0, Mutton Curry: ₹360.0, Fish Moilee: ₹340.0.

And for Breads: Tandoori Roti: ₹25.0, Butter Roti: ₹30.0, Plain Naan: ₹40.0, Butter Naan: ₹50.0, Garlic Naan: ₹60.0, Laccha Paratha: ₹50.0, Missi Roti: ₹45.
0, Roomali Roti: ₹40.0, Stuffed Naan: ₹70.0, Kulcha: ₹50.0.

--------------------------------------------------
Guest: i would like to order Mutton Rogan Josh (1 portion), Rajma Masal (1 portion) and 5 Butter Naans.
[System: Resort Manager is thinking... -]

Resort Response: Could you please provide your room number and guest ID so I can place the order?

--------------------------------------------------
Guest: room 105, guest ID G-1001.
[System: Resort Manager is thinking... -]

Resort Response: Okay, Revanth Kumar, your order has been placed: 1 Mutton Rogan Josh, 1 Rajma Masala, and 5 Butter Naans. The meal total will be calculated
```

```
.0, Mix Veg: ₹200.0, Chole Masala: ₹210.0, Rajma Masala: ₹210.0, Kadai Paneer: ₹250.0, Palak Paneer: ₹240.0, Veg Korma: ₹230.0, Butter Chicken: ₹320.0, Chic
ken Curry: ₹280.0, Chicken Korma: ₹300.0, Mutton Rogan Josh: ₹380.0, Fish Curry: ₹330.0, Prawn Masala: ₹360.0, Egg Curry: ₹220.0, Chicken Chettinad: ₹310.0,
 Mutton Curry: ₹360.0, Fish Moilee: ₹340.0.

Non-Veg Main Course: Butter Chicken: ₹320.0, Chicken Curry: ₹280.0, Chicken Korma: ₹300.0, Mutton Rogan Josh: ₹380.0, Fish Curry: ₹330.0, Prawn Masala: ₹360
.0, Egg Curry: ₹220.0, Chicken Chettinad: ₹310.0, Mutton Curry: ₹360.0, Fish Moilee: ₹340.0.

And for Breads: Tandoori Roti: ₹25.0, Butter Roti: ₹30.0, Plain Naan: ₹40.0, Butter Naan: ₹50.0, Garlic Naan: ₹60.0, Laccha Paratha: ₹50.0, Missi Roti: ₹45.
0, Roomali Roti: ₹40.0, Stuffed Naan: ₹70.0, Kulcha: ₹50.0.

--------------------------------------------------
Guest: i would like to order Mutton Rogan Josh (1 portion), Rajma Masal (1 portion) and 5 Butter Naans.
[System: Resort Manager is thinking... -]

Resort Response: Could you please provide your room number and guest ID so I can place the order?

--------------------------------------------------
Guest: room 105, guest ID G-1001.
[System: Resort Manager is thinking... -]

Resort Response: Okay, Revanth Kumar, your order has been placed: 1 Mutton Rogan Josh, 1 Rajma Masala, and 5 Butter Naans. The meal total will be calculated
 and provided to you for payment.

--------------------------------------------------
Guest: How much will be the total?
[System: Resort Manager is thinking... -]

Resort Response: Mutton Rogan Josh is ₹380.0, Rajma Masala is ₹210.0, and 5 Butter Naans are ₹50.0 each, totaling ₹250.0. So the total will be ₹380.0 + ₹210
.0 + ₹250.0 = ₹840.0.

--------------------------------------------------
Guest: Thank you. Exit
[System: Resort Manager is thinking... -]

Resort Response: You're welcome! Enjoy your meal!

--------------------------------------------------
Guest: Exit
Goodbye! Have a pleasant stay.

(venv) C:\Users\Revanth\resort_ai_system>
```

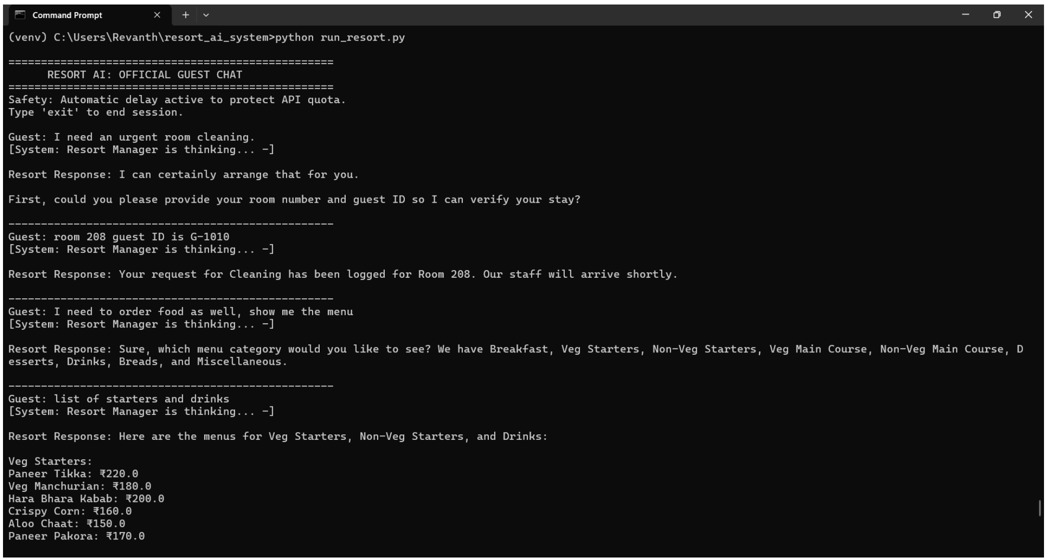
Table: guests

	guest_id	name	room_number	pending_bill
	Filter	Filter	Filter	Filter
1	G-1001	Revanth Kumar	105	840.0
2	G-1002	Alice Sharma	110	0.0
3	G-1003	Bob Varma	115	0.0
4	G-1004	Charlie Khan	120	0.0
5	G-1005	Diana Prince	125	0.0
6	G-1006	Ethan Hunt	130	0.0
7	G-1007	Fiona Gallagher	135	0.0
8	G-1008	George Miller	140	0.0
9	G-1009	Hannah Abbott	204	0.0
10	G-1010	Ian Wright	208	1180.0
11	G-1011	Julia Roberts	212	0.0
12	G-1012	Kevin Hart	216	0.0
13	G-1013	Laura Palmer	220	0.0
14	G-1014	Michael Scott	224	0.0
15	G-1015	Nina Simone	228	0.0

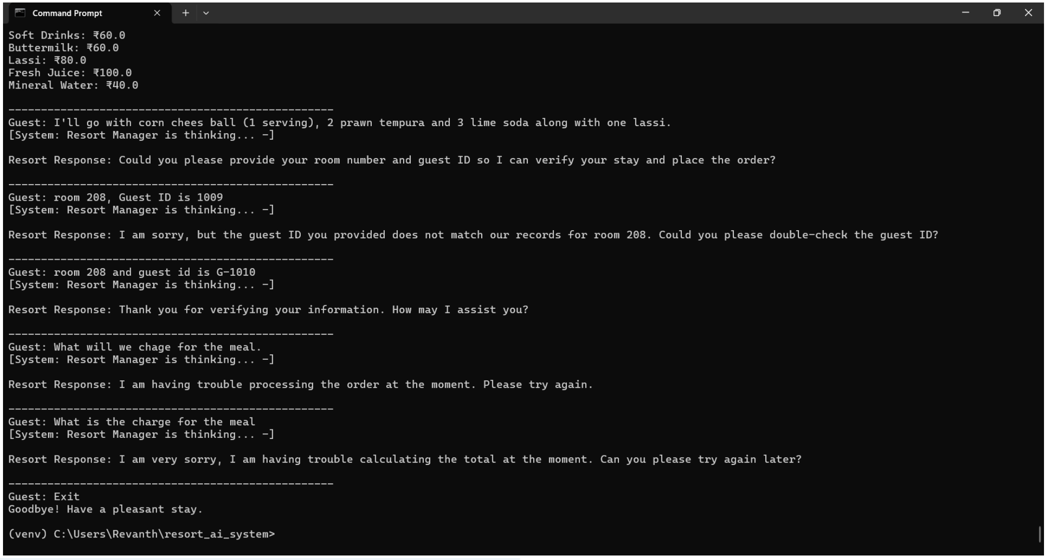
Table: food_orders

	id	room_number	guest_id	items	total_amount	timestamp
	Fil...	Filter	Filter	Filter	Filter	Filter
1	1	105	G-1001	1x Mutton Rogan Josh	380.0	2025-12-26 15:34:43.298100
2	2	105	G-1001	1x Rajma Masala	210.0	2025-12-26 15:34:43.314544
3	3	105	G-1001	5x Butter Naan	250.0	2025-12-26 15:34:43.324332
4	4	208	G-1010	1x Corn Cheese Balls	190.0	2025-12-26 17:50:51.118914
5	5	208	G-1010	2x Prawn Tempura	700.0	2025-12-26 17:50:51.138435
6	6	208	G-1010	3x Fresh Lime Soda	210.0	2025-12-26 17:50:51.146643
7	7	208	G-1010	1x Lassi	80.0	2025-12-26 17:50:51.155112

Figure 2: Instance where the guest is ordering food. The agent verifies the guest, then takes the order, which is then reflected in the database.



```
(venv) C:\Users\Revanth\resort_ai_system>python run_resort.py

==================================================
      RESORT AI: OFFICIAL GUEST CHAT
==================================================
Safety: Automatic delay active to protect API quota.
Type 'exit' to end session.

Guest: I need an urgent room cleaning.
[System: Resort Manager is thinking... -]

Resort Response: I can certainly arrange that for you.

First, could you please provide your room number and guest ID so I can verify your stay?

--------------------------------------------------
Guest: room 208 guest ID is G-1010
[System: Resort Manager is thinking... -]

Resort Response: Your request for Cleaning has been logged for Room 208. Our staff will arrive shortly.

--------------------------------------------------
Guest: I need to order food as well, show me the menu
[System: Resort Manager is thinking... -]

Resort Response: Sure, which menu category would you like to see? We have Breakfast, Veg Starters, Non-Veg Starters, Veg Main Course, Non-Veg Main Course, D
esserts, Drinks, Breads, and Miscellaneous.

--------------------------------------------------
Guest: list of starters and drinks
[System: Resort Manager is thinking... -]

Resort Response: Here are the menus for Veg Starters, Non-Veg Starters, and Drinks:

Veg Starters:
Paneer Tikka: ₹220.0
Veg Manchurian: ₹180.0
Hara Bhara Kabab: ₹200.0
Crispy Corn: ₹160.0
Aloo Chaat: ₹150.0
Paneer Pakora: ₹170.0
```



```
Soft Drinks: ₹60.0
Buttermilk: ₹60.0
Lassi: ₹80.0
Fresh Juice: ₹100.0
Mineral Water: ₹40.0

--------------------------------------------------
Guest: I'll go with corn chees ball (1 serving), 2 prawn tempura and 3 lime soda along with one lassi.
[System: Resort Manager is thinking... -]

Resort Response: Could you please provide your room number and guest ID so I can verify your stay and place the order?

--------------------------------------------------
Guest: room 208, Guest ID is 1009
[System: Resort Manager is thinking... -]

Resort Response: I am sorry, but the guest ID you provided does not match our records for room 208. Could you please double-check the guest ID?

--------------------------------------------------
Guest: room 208 and guest id is G-1010
[System: Resort Manager is thinking... -]

Resort Response: Thank you for verifying your information. How may I assist you?

--------------------------------------------------
Guest: What will we chage for the meal.
[System: Resort Manager is thinking... -]

Resort Response: I am having trouble processing the order at the moment. Please try again.

--------------------------------------------------
Guest: What is the charge for the meal
[System: Resort Manager is thinking... -]

Resort Response: I am very sorry, I am having trouble calculating the total at the moment. Can you please try again later?

--------------------------------------------------
Guest: Exit
Goodbye! Have a pleasant stay.

(venv) C:\Users\Revanth\resort_ai_system>
```

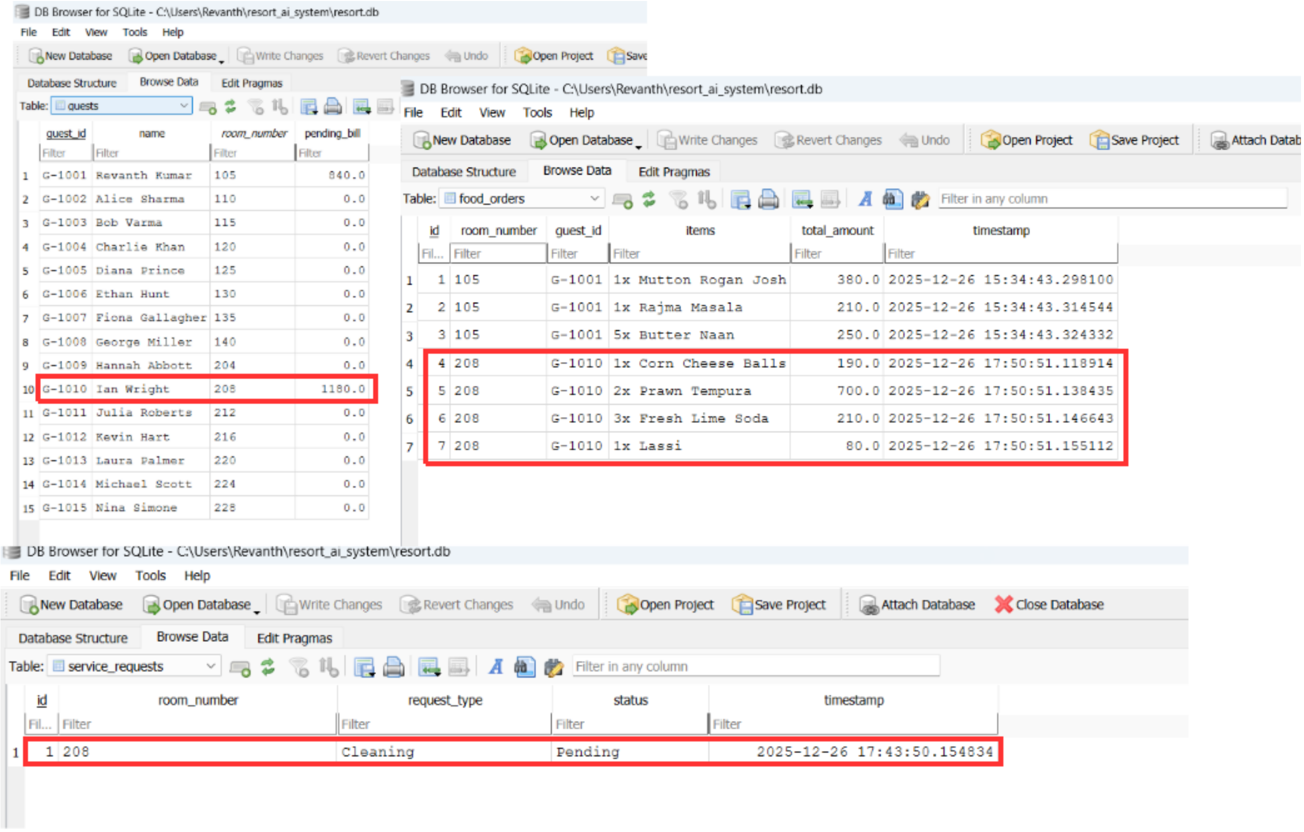


Table: guests

	guest_id	name	room_number	pending_bill
1	G-1001	Revanth Kumar	105	840.0
2	G-1002	Alice Sharma	110	0.0
3	G-1003	Bob Varma	115	0.0
4	G-1004	Charlie Khan	120	0.0
5	G-1005	Diana Prince	125	0.0
6	G-1006	Ethan Hunt	130	0.0
7	G-1007	Fiona Gallagher	135	0.0
8	G-1008	George Miller	140	0.0
9	G-1009	Hannah Abbott	204	0.0
10	G-1010	Ian Wright	208	1180.0
11	G-1011	Julia Roberts	212	0.0
12	G-1012	Kevin Hart	216	0.0
13	G-1013	Laura Palmer	220	0.0
14	G-1014	Michael Scott	224	0.0
15	G-1015	Nina Simone	228	0.0

Table: food_orders

	id	room_number	guest_id	items	total_amount	timestamp
1	1	105	G-1001	1x Mutton Rogan Josh	380.0	2025-12-26 15:34:43.298100
2	2	105	G-1001	1x Rajma Masala	210.0	2025-12-26 15:34:43.314544
3	3	105	G-1001	5x Butter Naan	250.0	2025-12-26 15:34:43.324332
4	4	208	G-1010	1x Corn Cheese Balls	190.0	2025-12-26 17:50:51.118914
5	5	208	G-1010	2x Prawn Tempura	700.0	2025-12-26 17:50:51.138435
6	6	208	G-1010	3x Fresh Lime Soda	210.0	2025-12-26 17:50:51.146643
7	7	208	G-1010	1x Lassi	80.0	2025-12-26 17:50:51.155112

Table: service_requests

	id	room_number	request_type	status	timestamp
1	1	208	Cleaning	Pending	2025-12-26 17:43:50.154834

Figure 3: Instance when a guest asks for room service, which is updated on the database, followed by a food order where the agent first refuses to take the order because of a mismatch in the guest ID and the room number. Later, the order is accepted by the agent, who also updates the database.

3.4 Conclusion

By bridging the gap between complex resort databases and user-centric communication, this system demonstrates a scalable, secure, and data-driven approach to modern hospitality management.