HOTEL MANAGEMENT SYSTEM

Core Functionality Overview

The project is organized into smaller functions each responsible for a specific hotel task. These functions interact with the SQLite database to store and retrieve persistent data and provide a simple text-based user interface.

Functions Explained

show_vacant_rooms()

- Connects to the SQLite database.
- Retrieves all room numbers marked with status 'vacant' from the ROOMS table.
- Prints the list of vacant rooms or a message if none are available.
- Closes the database connection afterward.

checkin()

- Calls show_vacant_rooms() to display available rooms.
- Prompts the user to enter a room number for check-in.
- Checks if the room exists and if it is vacant.
- If valid, it asks for the customer's name and their check-in/check-out dates.
- Inserts a new record into the CUSTOMERS table with the booking details.
- Updates the room status to 'occupied' in the ROOMS table.
- Commits changes and closes the connection.
- Confirms the successful check-in to the user.

food_order()

- Asks for the room number placing the food order.
- Verifies if the room is currently occupied.
- Displays a predefined food menu with items and prices.
- Repeatedly prompts the user to input food items and quantities until 'done' is entered.
- Checks if ordered items are in the menu and the quantities are valid positive integers.

- Inserts each food order row into the RESTAURANT table linked to that room.
- Commits and closes the connection, confirming each added order.

checkout()

- Requests the customer name for checkout.
- Retrieves the room number and check-in/out dates from CUSTOMERS table.
- Calculates the number of days stayed, defaulting to 1 if invalid.
- Uses a predefined room rate (10,000 Rs per day) to compute room charges.
- Retrieves all food orders for the room from the RESTAURANT table.
- Calculates the total food bill.
- Uses PrettyTable to display formatted tables:
    - Customer details including name, room, dates, and duration of stay.
    - Food order details with individual and total prices.
- Displays the final bill comprising room and food charges.
- Deletes the customer and associated food orders from tables.
- Marks the room as vacant again.
- Commits changes and notifies that checkout is complete.

main_menu()

- A continuous loop presents a user menu with options:
    - Show vacant rooms
    - Check-in guest
    - Place food order
    - Checkout guest
    - Exit the system
- Based on user choice, calls the respective functions.
- Handles invalid choices gracefully.
- Terminates the program when the user selects exit.

Additional Notes

- The database tables are ROOMS (room info and status), CUSTOMERS (guest bookings), and RESTAURANT (food orders).
- The food menu is stored as a Python dictionary mapping item names to prices.
- Date inputs follow the YYYY-MM-DD format and are parsed using Python's datetime module.
- Room charges are fixed but can be configured by changing the rate in the code.
- Input validation is done for room existence, occupancy status, and food order quantities.
- The modular function-based design enhances clarity and maintainability.
- The system operates purely via a command-line interface but could be extended for GUI use.
- PrettyTable provides neat, readable tables to present billing summaries to users.