HOTEL RESERVATION SYSTEM

NAME - AANANDI GOEL

SAP ID - 590026126

Abstract

This project presents a console-based implementation of a basic hotel reservation system using the C programming language. The system allows users to check room availability, make bookings, and view current reservations. This project is ideal for learning structured programming, including function separation, file handling (for data persistence), arrays, conditional logic, and user interaction. The system is designed to be simple, efficient, and easy to extend with new features like room cancellation or a graphical interface.

1. Problem Definition

1.1 Overview

The Hotel Reservation System Project is a console-based application that simulates basic hotel management functionality. Users can check room availability, book a room, and view existing reservations. The project uses C programming to implement structures for rooms and reservations, file handling for persistent data storage, and functions for modular design.

1.2 Objectives

• To design a room and reservation management system using C programming.

• To implement a modular program structure using header files, multiple C files, and function definitions.

• To provide an interactive console interface for making and viewing reservations.

• To demonstrate file handling (reading/writing to save reservation data).

2. System Design

2.1 System Architecture

The architecture follows a Modular Layered Structure.

• Presentation Layer: Handles user interaction.

• Business Logic: Core reservation logic.

• Data Access: Manages room and reservation data.

2.2 Data Structures

Room Structure (struct Room)

- roomNumber
- roomType
- pricePerNight
- isAvailable

Reservation Structure (struct Reservation)

- reservationID
- roomNumber
- guestName
- checkInDate, checkOutDate
- totalBill

3. Implementation Details

3.1 Key Features

• 10 predefined rooms.
• Modular multi-file architecture.
• Room availability checking.
• Room booking.
• View all reservations.
• File-based persistent storage.
• Clean output formatting.

4. Testing & Results

Test Case 1: Successful Booking – Passed

Test Case 2: No Room Available – Passed

Test Case 3: View Reservations – Passed

Test Case 4: Data Persistence – Passed

5. Conclusion & Future Work

5.1 Conclusion

The system successfully meets all objectives using C with modular programming, data structures, and file handling.

5.2 Future Work

• Add check-out/cancellation.

• Use dynamic memory allocation.

• Advanced date/time handling.