PROJECT REPORT
ON
Hotel Reservation System

INTRODUCTION:
The Hotel Reservation System is a software application designed to streamline the process of booking hotel rooms, managing guest data, and handling payments. It serves as a centralized platform for both hotel staff and guests, ensuring efficient operations and enhanced customer satisfaction. The system automates manual tasks, reduces errors, and provides real-time information about room availability, reservations, and payments.

This project aims to create a robust and user-friendly system that caters to the needs of modern hotels, enabling them to manage their operations effectively while providing a seamless experience for guests.

Planning of the Project

Objective:

- To develop a system that simplifies the process of room reservations, guest management, and payment processing.
- To provide real-time updates on room availability and reservation status.
- To generate invoices and maintain records for future reference.

Scope:
The system will handle room bookings, guest information, reservations, payments, and employee management.

It will provide functionalities for guests to book rooms, for staff to manage reservations, and for administrators to generate reports.

Target Users:
Guests: To book rooms and view reservation details.
Hotel Staff: To manage reservations, process payments, and update room status.
Administrators: To oversee operations and generate reports.

Working of the Project

Guest Interaction:

Guests can browse available rooms, check prices, and make reservations.
They can provide personal details (name, contact information, etc.) during the booking process.

Staff Interaction:

Staff can view and manage reservations, update room status, and process payments.
They can generate invoices and provide support to guests.

Administrator Interaction:

Administrators can oversee all operations, manage employees, and generate reports (e.g., revenue, occupancy rates).

System Flow:

A guest selects a room and makes a reservation.
The system updates room status to "booked" and stores the reservation details.
Upon check-in, the guest's details are verified, and payment is processed.
After check-out, the room status is updated to "available,"

Functionalities

For Guests:

Browse available rooms.
Make, modify, or cancel reservations.
View reservation details and invoices.

For Staff:

Manage room availability and status.
Process guest check-ins and check-outs.
Handle payments and generate invoices.

For Administrators:

Add, update, or delete room and employee details.
Generate reports (e.g., revenue, occupancy rates).
Monitor system performance and user activity.

Entities and Attributes

Rooms:

- room_id (Primary Key)
- room_number
- room_type (e.g., Deluxe, Suite)
- price_per_night
- status (e.g., available, occupied)

Guests:

- guest_id (Primary Key)
- first_name
- last_name
- email
- phone_number
- address

Reservations:

- reservation_id (Primary Key)
- guest_id (Foreign Key referencing Guests)
- room_id (Foreign Key referencing Rooms)
- check_in_date
- check_out_date
- total_cost
- status (e.g., confirmed, canceled)

Payments:

- payment_id (Primary Key)
- reservation_id (Foreign Key referencing Reservations)
- employee_id (Foreign Key referencing Employees)
- payment_date
- amount
- payment_method (e.g., credit card, cash)

Employees:

- employee_id (Primary Key)
- first_name
- last_name
- email
- phone_number
- role (e.g., receptionist, manager)

SQL CODE

```sql
-- Create Rooms Table
CREATE TABLE Rooms (
    room_id INT PRIMARY KEY IDENTITY(1,1),
    room_number VARCHAR(10) NOT NULL UNIQUE,
    room_type VARCHAR(50) NOT NULL,
    price_per_night DECIMAL(10,2) NOT NULL,
    status VARCHAR(10) DEFAULT 'available' CHECK (status IN ('available', 'occupied'))
);

-- Create Guests Table
CREATE TABLE Guests (
    guest_id INT PRIMARY KEY IDENTITY(1,1),
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    phone_number VARCHAR(15),
    address VARCHAR(255)
);

-- Create Employees Table
CREATE TABLE Employees (
    employee_id INT PRIMARY KEY IDENTITY(1,1),
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    phone_number VARCHAR(15),
    role VARCHAR(20) CHECK (role IN ('receptionist', 'manager', 'staff'))
);

-- Create Reservations Table
CREATE TABLE Reservations (
    reservation_id INT PRIMARY KEY IDENTITY(1,1),
    guest_id INT,
    room_id INT,
    check_in_date DATE NOT NULL,
    check_out_date DATE NOT NULL,
    total_cost DECIMAL(10,2) NOT NULL,
    status VARCHAR(10) DEFAULT 'confirmed' CHECK (status IN ('confirmed',
'canceled')),
    FOREIGN KEY (guest_id) REFERENCES Guests(guest_id),
    FOREIGN KEY (room_id) REFERENCES Rooms(room_id)
);

-- Create Payments Table
CREATE TABLE Payments (
    payment_id INT PRIMARY KEY IDENTITY(1,1),
    reservation_id INT,
    employee_id INT,
    payment_date DATE NOT NULL,
    amount DECIMAL(10,2) NOT NULL,
    payment_method VARCHAR(20) CHECK (payment_method IN ('credit card',
'cash', 'debit card')),
    FOREIGN KEY (reservation_id) REFERENCES Reservations(reservation_id),
    FOREIGN KEY (employee_id) REFERENCES Employees(employee_id)
);

-- Insert sample data into Rooms
INSERT INTO Rooms (room_number, room_type, price_per_night, status)
VALUES
('101', 'Deluxe', 150.00, 'available'),
('102', 'Suite', 200.00, 'available'),
('103', 'Standard', 100.00, 'occupied');

-- Insert sample data into Guests
INSERT INTO Guests (first_name, last_name, email, phone_number, address)
VALUES
('John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Main St'),
('Jane', 'Smith', 'jane.smith@example.com', '987-654-3210', '456 Elm St');

-- Insert sample data into Employees
INSERT INTO Employees (first_name, last_name, email, phone_number, role)
VALUES
```

```
('Alice', 'Johnson', 'alice.johnson@example.com', '555-123-4567', 'receptionist'),
('Bob', 'Williams', 'bob.williams@example.com', '555-987-6543', 'manager');

-- Insert sample data into Reservations
INSERT INTO Reservations (guest_id, room_id, check_in_date, check_out_date,
total_cost, status) VALUES
(1, 1, '2023-10-15', '2023-10-20', 750.00, 'confirmed'),
(2, 2, '2023-10-16', '2023-10-18', 400.00, 'confirmed');

-- Insert sample data into Payments
INSERT INTO Payments (reservation_id, employee_id, payment_date, amount,
payment_method) VALUES
(1, 1, '2023-10-14', 750.00, 'credit card'),
(2, 1, '2023-10-15', 400.00, 'cash');
```

QUIRES:

```
SELECT * FROM Rooms WHERE status = 'available';
SELECT * FROM Guests;
SELECT * FROM Reservations;
SELECT * FROM Payments;
SELECT * FROM Employees;
SELECT * FROM Reservations WHERE guest_id = 1;
SELECT * FROM Payments WHERE reservation_id = 1;
SELECT * FROM Rooms WHERE room_type = 'Deluxe';
SELECT * FROM Reservations WHERE status = 'confirmed';
SELECT * FROM Payments WHERE employee_id = 1;
```

Conclusion

The Hotel Reservation System is a comprehensive solution for managing hotel operations efficiently. By automating tasks like room bookings, guest management, and payment processing, the system reduces manual effort and enhances the overall guest experience. With well-defined entities, attributes, and functionalities, the system ensures smooth operations and provides valuable insights for decision-making.

This project is designed to meet the needs of modern hotels, making it an essential tool for improving productivity and customer satisfaction.