HOTEL MANAGEMENT SYSTEM

Computer Science (083) Project

Developed By

MAYANK SAHAI
12th E

Index

Certificate

This is to certify BANK MANAGEMENT SYSTEM Computer Science project is developed by **MAYANK SAHAI** under my supervision in the session 2024-2025.

The work done by him is original.

_____ Computer Science Teacher

_____ EXTERNAL EXAMINER

Date: _____

Acknowledgement

I express our immense gratitude to our Computer Science teacher POOJA KHARE for her intellectual vigour and generously given support that has been invaluable in escalating our determination to reach the goal of writing this project successfully.

I can hardly find appropriate words to express our obligations and gratefulness to the Principal.

I also feel immense pleasure in recording deep sense of indebtedness, gratitude and sincere thanks to all fellow group mates for their help, company and hard work.

I are especially indebted to our parents for their sincere love, moral support and spontaneous encouragement throughout the entire period of this work.

Thank you!

Project Synopsis

Introduction and Need

- This project is all about software for the Hotel management system.

- The Hotel Management System is a software application designed to streamline hotel operations. It provides functionalities such as room management, customer management, and booking management. The system enhances efficiency, reduces errors, and improves user experience.

AIM

- The objective of this project is to let us apply programming knowledge into a real- world situation/problem and expose how programming skills help in developing a good software.

Idea of the Project

- **Challenges with Manual Systems:** Traditional hotel management systems are often inefficient, leading to errors in booking, room allocation, and customer data management.

- Technological Advancements: The rise of digital technologies has created an opportunity to improve the efficiency and accuracy of hotel operations through automation.

- Need for Real-Time Data: Hotels require systems that can track room availability, bookings, and customer data in real-time, ensuring smoother operations and better decision-making.

- **Growing Customer Expectations:** Customers expect a seamless, fast, and personalized experience during booking, check-in, and check-out, which can be better managed with digital systems.

- Industry Trends: As the hospitality industry grows, there's an increasing demand for scalable solutions that can manage multiple locations and provide insightful business analytics.

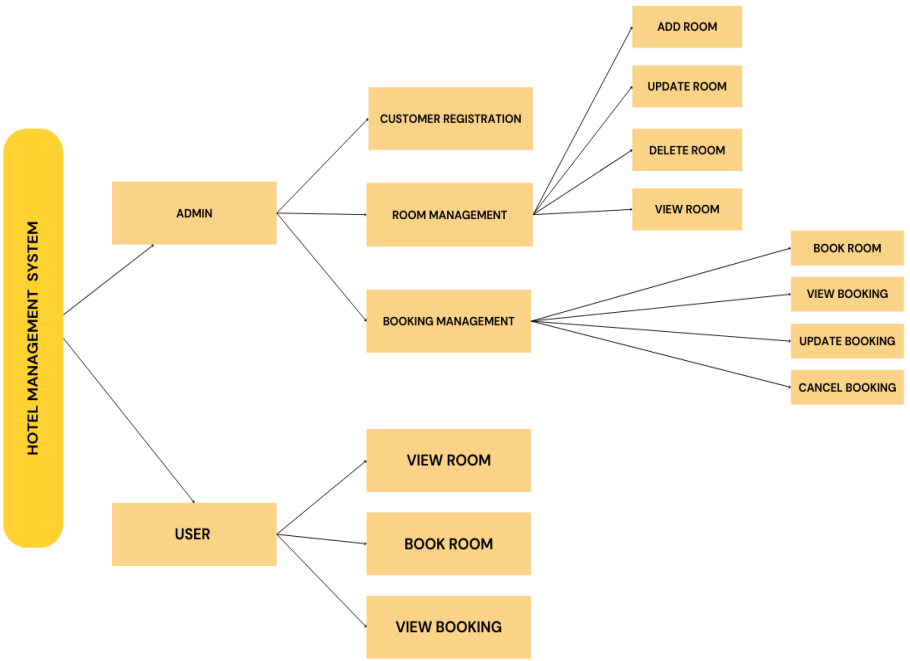
Plan For Implementation

The system uses a MySQL database to manage data. Key features include:

- Room Management: Add, update, and delete room records.

- Customer Management: Register and view customer details.

- Booking Management: Issue and view booking records



Validation and Add on Features

- If a user enters an invalid input, the system will prompt them to reattempt, ensuring a seamless experience. The program is designed to be intuitive and user-centric, making it easy to navigate and interact with. Moreover, the code includes specific features designed to cater to unique requirements, which are clearly detailed and implemented.

SOURCE CODE

```
iimport mysql.connector as pymysql
from datetime import datetime

passwrd = None
db = None
C = None

def base_check():
    check = 0
    db = pymysql.connect(host="localhost", user="root",
password=passwrd)
    cursor = db.cursor()
    cursor.execute('SHOW DATABASES')
    result = cursor.fetchall()
    for r in result:
        for i in r:
            if i == 'hotel_management':
                cursor.execute('USE hotel_management')
                check = 1
    if check != 1:
        create_database()

def table_check():
    db = pymysql.connect(host="localhost", user="root",
password=passwrd)
    cursor = db.cursor()
    cursor.execute('SHOW DATABASES')
    result = cursor.fetchall()
    for r in result:
        for i in r:
            if i == 'hotel_management':
                cursor.execute('USE hotel_management')
```

```
                cursor.execute('SHOW TABLES')
                result = cursor.fetchall()
                if len(result) <= 2:
                    create_tables()
                else:
                    print('      Booting systems...')

def create_database():
    try:
        db = pymysql.connect(host="localhost", user="root",
password=passwrd)
        cursor = db.cursor()
        cursor.execute("CREATE DATABASE IF NOT EXISTS
hotel_management")
        db.commit()
        db.close()
        print("Database 'hotel_management' created successfully.")
    except pymysql.Error as e:
        print(f"Error creating database: {str(e)}")

def create_tables():
    try:
        db = pymysql.connect(host="localhost", user="root",
password=passwrd, database="hotel_management")
        cursor = db.cursor()

        cursor.execute("""
            CREATE TABLE IF NOT EXISTS rooms (
                ROOM_ID INT PRIMARY KEY,
                ROOM_TYPE VARCHAR(255),
                PRICE DECIMAL(10, 2),
                AVAILABLE INT
            )
        """)

        cursor.execute("""
            CREATE TABLE IF NOT EXISTS customers (
                CUSTOMER_ID INT PRIMARY KEY,
                NAME VARCHAR(255),
                PHONE_NO VARCHAR(15)
            )
        """)

        cursor.execute("""
            CREATE TABLE IF NOT EXISTS bookings (
                BOOKING_ID INT AUTO_INCREMENT PRIMARY KEY,
                CUSTOMER_ID INT,
                ROOM_ID INT,
                CHECK_IN_DATE DATE,
                CHECK_OUT_DATE DATE,
                TOTAL_AMOUNT DECIMAL(10, 2),
                FOREIGN KEY (CUSTOMER_ID) REFERENCES
customers(CUSTOMER_ID),
```

```
                FOREIGN KEY (ROOM_ID) REFERENCES rooms(ROOM_ID)
            )
        """)

        db.commit()
        db.close()
        print("Tables 'rooms', 'customers', and 'bookings' created
successfully.")
    except pymysql.Error as e:
        print(f"Error creating tables: {str(e)}")

def add_room():
    room_id = int(input("Enter Room ID: "))
    room_type = input("Enter Room Type: ")
    price = float(input("Enter Room Price: "))
    available = int(input("Enter Number of Available Rooms: "))
    data = (room_id, room_type, price, available)
    sql = "INSERT INTO rooms (ROOM_ID, ROOM_TYPE, PRICE, AVAILABLE)
VALUES (%s, %s, %s, %s)"
    try:
        C.execute(sql, data)
        db.commit()
        print('Room added successfully...')
    except pymysql.Error as e:
        print(f"Error adding room: {str(e)}")

def view_rooms():
    C.execute("SELECT * FROM rooms")
    result = C.fetchall()
    for r in result:
        print(r)

def update_room():
    room_id = int(input("Enter Room ID to update: "))
    field = input("Enter field to update [ROOM_TYPE, PRICE, AVAILABLE]:
")
    new_value = input(f"Enter new value for {field}: ")
    if field == 'PRICE':
        new_value = float(new_value)
    elif field == 'AVAILABLE':
        new_value = int(new_value)
    sql = f"UPDATE rooms SET {field} = %s WHERE ROOM_ID = %s"
    try:
        C.execute(sql, (new_value, room_id))
        db.commit()
        print('Room updated successfully...')
    except pymysql.Error as e:
        print(f"Error updating room: {str(e)}")

def delete_room():
    room_id = int(input("Enter Room ID to delete: "))
    sql = "DELETE FROM rooms WHERE ROOM_ID = %s"
    try:
```

```
        C.execute(sql, (room_id,))
        db.commit()
        print('Room deleted successfully...')
    except pymysql.Error as e:
        print(f"Error deleting room: {str(e)}")

def register_customer():
    customer_id = int(input("Enter Customer ID: "))
    name = input("Enter Customer Name: ")
    phone_no = input("Enter Customer Phone Number: ")
    data = (customer_id, name, phone_no)
    sql = "INSERT INTO customers (CUSTOMER_ID, NAME, PHONE_NO) VALUES
(%s, %s, %s)"
    try:
        C.execute(sql, data)
        db.commit()
        print('Customer registered successfully...')
    except pymysql.Error as e:
        print(f"Error registering customer: {str(e)}")

def view_customers():
    C.execute("SELECT * FROM customers")
    result = C.fetchall()
    for r in result:
        print(r)

def book_room():
    customer_id = int(input("Enter Customer ID: "))
    room_id = int(input("Enter Room ID: "))
    check_in_date = input("Enter Check-In Date (YYYY-MM-DD): ")
    check_out_date = input("Enter Check-Out Date (YYYY-MM-DD): ")
    total_amount = float(input("Enter Total Amount: "))
    data = (customer_id, room_id, check_in_date, check_out_date,
total_amount)
    sql = "INSERT INTO bookings (CUSTOMER_ID, ROOM_ID, CHECK_IN_DATE,
CHECK_OUT_DATE, TOTAL_AMOUNT) VALUES (%s, %s, %s, %s, %s)"
    try:
        C.execute(sql, data)
        db.commit()
        print('Room booked successfully...')
    except pymysql.Error as e:
        print(f"Error booking room: {str(e)}")

def view_bookings():
    C.execute("SELECT * FROM bookings")
    result = C.fetchall()
    for r in result:
        print(r)

def main():
    global passwrd
    passwrd = input("Enter password for MySQL: ")
```

```
    base_check()
    table_check()

    global db, C
    db = pymysql.connect(host="localhost", user="root",
password=passwrd, database="hotel_management")
    C = db.cursor()

    while True:
        log = input("For Admin: A, For Customer: C, Exit: X ::: ")
        if log.upper() == "A":
            while True:
                menu = input('''Add Room: AR, View Rooms: VR, Update
Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC,
Book Room: BR, View Bookings: VB, Exit: X :::''')
                if menu.upper() == 'AR':
                    add_room()
                elif menu.upper() == 'VR':
                    view_rooms()
                elif menu.upper() == 'UR':
                    update_room()
                elif menu.upper() == 'DR':
                    delete_room()
                elif menu.upper() == 'RC':
                    register_customer()
                elif menu.upper() == 'VC':
                    view_customers()
                elif menu.upper() == 'BR':
                    book_room()
                elif menu.upper() == 'VB':
                    view_bookings()
                elif menu.upper() == 'X':
                    break
                else:
                    print("Wrong Input")

        elif log.upper() == "C":
            print("Customer Interface")
            while True:
                customer_menu = input('''View Available Rooms: VR, Book
Room: BR, View Your Bookings: VB, Exit: X :::''')
                if customer_menu.upper() == 'VR':
                    view_rooms()
                elif customer_menu.upper() == 'BR':
                    book_room()
                elif customer_menu.upper() == 'VB':
                    view_bookings()
                elif customer_menu.upper() == 'X':
                    break
                else:
                    print("Wrong Input")

        elif log.upper() == "X":
```

```
            print("THANK YOU FOR USING HOTEL MANAGEMENT SYSTEM ")
            break
        else:
            print("Wrong Input")

if __name__ == "__main__":
    main()
```

OUTPUT

➢ Admin Controls

- **ADD ROOM**

```
For Admin: A, For Customer: C, Exit: X ::: a
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::a
Wrong Input
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::ar
Enter Room ID: 1
Enter Room Type: 2
Enter Room Price: 23
Enter Number of Available Rooms: 5
Room added successfully...
```

- **VIEW ROOM**

```
PS E:\git\Hotel-management-system> python .\main.py
Enter password for MySQL: 1230
      Booting systems...
For Admin: A, For Customer: C, Exit: X ::: a
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::vr
(1, '2', Decimal('23.00'), 5)
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::
BLACKBOX Chat  Add Logs  CyberCoder  Improve Code  Share Code Link  Generate Commit Message  Search Error  CRLF  Python  Select Interpreter  AI Code Chat  CODEG
```

- **UPDATE ROOM**

```
PS E:\git\Hotel-management-system> python .\main.py
Enter password for MySQL: 1230
      Booting systems...
For Admin: A, For Customer: C, Exit: X ::: a
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::ur
Enter Room ID to update: 1
Enter field to update [ROOM_TYPE, PRICE, AVAILABLE]: room_type
Enter new value for room_type: dulex
Room updated successfully...
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::
```

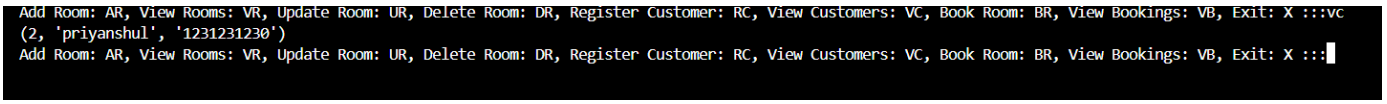
- **DELETE ROOM**

```
PS E:\git\Hotel-management-system> python .\main.py
Enter password for MySQL: 1230
      Booting systems...
For Admin: A, For Customer: C, Exit: X ::: a
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::dr
Enter Room ID to delete: 1
Room deleted successfully...
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::
```

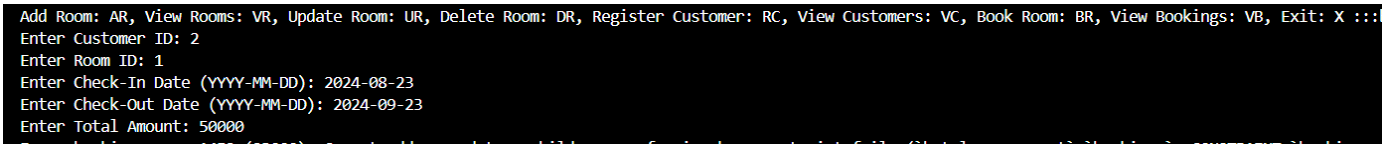
- **ADD CUSTOMER**

```
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::rc
Enter Customer ID: 2
Enter Customer Name: priyanshul
Enter Customer Phone Number: 1231231230
Customer registered successfully...
Add Room: AR, View Rooms: VR, Update Room: UR, Delete Room: DR, Register Customer: RC, View Customers: VC, Book Room: BR, View Bookings: VB, Exit: X :::
```

- ## VIEW CUSTOMER



- ## BOOK ROOM



➢ User Controls

- **VIEW ROOM**

```
PS E:\git\Hotel-management-system> python .\main.py
Enter password for MySQL: 1230
      Booting systems...
For Admin: A, For Customer: C, Exit: X ::: c
Customer Interface
View Available Rooms: VR, Book Room: BR, View Your Bookings: VB, Exit: X :::vr
(1, '2', Decimal('23.00'), 5)
View Available Rooms: VR, Book Room: BR, View Your Bookings: VB, Exit: X :::x
```

- **BOOK ROOM**

```
PS E:\git\Hotel-management-system> python .\main.py
Enter password for MySQL: 1230
      Booting systems...
For Admin: A, For Customer: C, Exit: X ::: c
Customer Interface
View Available Rooms: VR, Book Room: BR, View Your Bookings: VB, Exit: X :::br
Enter Customer ID: 2
Enter Room ID: 1
Enter Check-In Date (YYYY-MM-DD): 2024-01-01
Enter Check-Out Date (YYYY-MM-DD): 2024-01-05
Enter Total Amount: 200
```

- **VIEW YOUR BOOKINGS**

```
View Available Rooms: VR, Book Room: BR, View Your Bookings: VB, Exit: X :::vb
View Available Rooms: VR, Book Room: BR, View Your Bookings: VB, Exit: X :::
```

- **EXIT**

```
OREIGN KEY (`ROOM_ID`) REFERENCES `rooms` (`ROOM_ID`))
View Available Rooms: VR, Book Room: BR, View Your Bookings: VB, Exit: X :::vb
View Available Rooms: VR, Book Room: BR, View Your Bookings: VB, Exit: X :::x
For Admin: A, For Customer: C, Exit: X ::: x
THANK YOU FOR USING HOTEL MANAGEMENT SYSTEM
PS E:\git\Hotel-management-system> Sys.path
```

Hardware Requirement PC/Laptop/MacBook with Intel core/i3/i5/i7 or any equivalent With at least 2 GB RAM 10 MB free space on Hard Disk LCD/LED Operating System & Compiler MS Windows/Ubuntu/MacOS Python IDLE 3.x OR colab.research.google.com (gmail account) and MySQL 8.x

References

1. Classnotes

2. www.w3schools.com

3. www.geekforgeeks.com

4. Friends