HOTEL

MANAGEMENT SYSTEM



Computer Science (083) Project

Developed By

MAYANK SAHAI  
12th E

Index

|  |  |  |
| --- | --- | --- |
| Sno | Description | Pageno |
| 1 | Certificate | 3 |
| 2 | Acknowledgement & References | 4 |
| 3 | Project Synopsis | 5 |
| 4 | Source Code | 9 |
| 5 | Output Screen | 15 |
| 6 | Hardware & Software requirement | 18 |
| 7 | Biblography | 19 |

# **Certificate**

## This is to certify BANK MANAGEMENT SYSTEM

Computer Science project is developed by **MAYANK SAHAI** undermy 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.

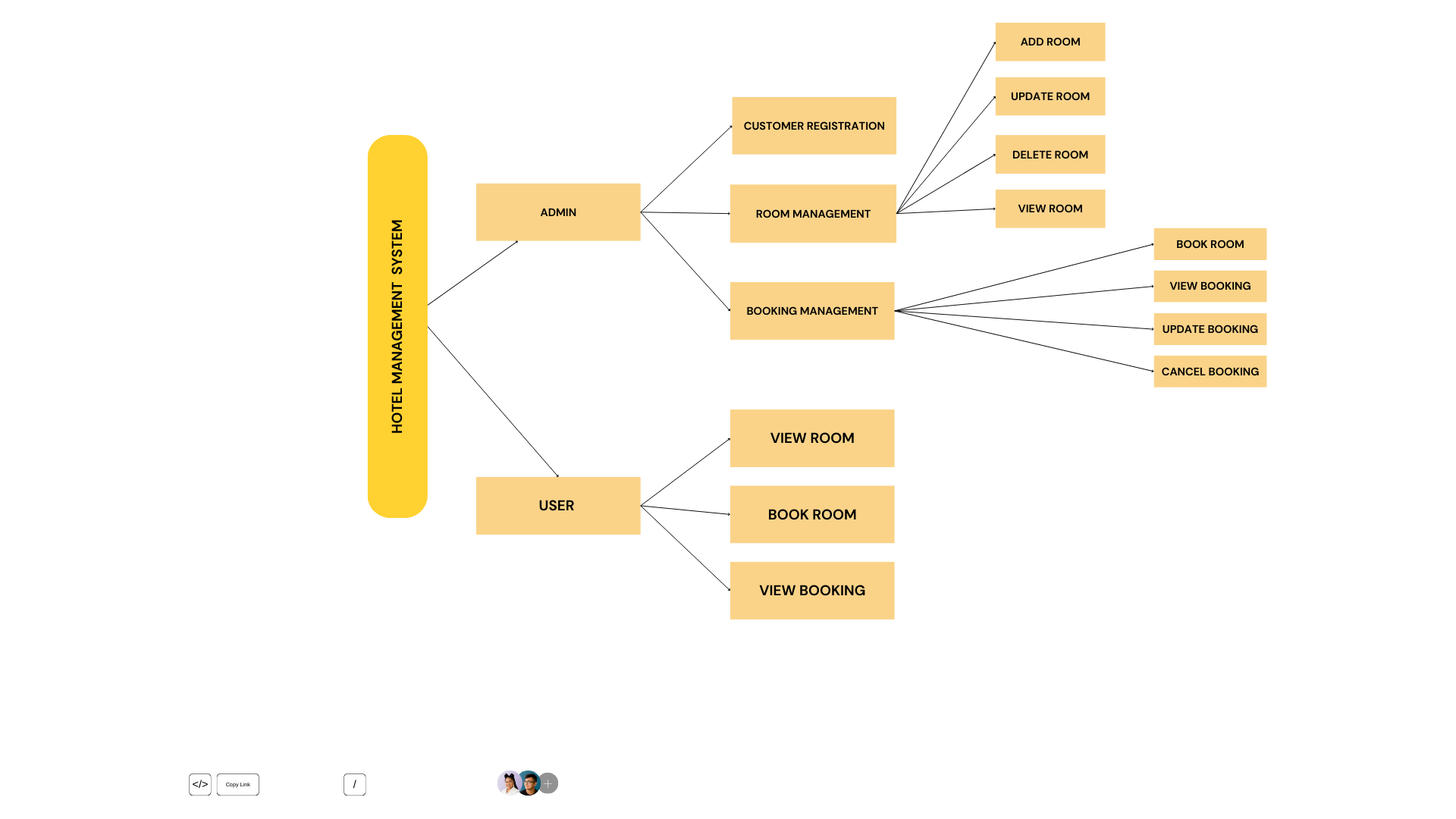
**Plan For Implementation**

**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:

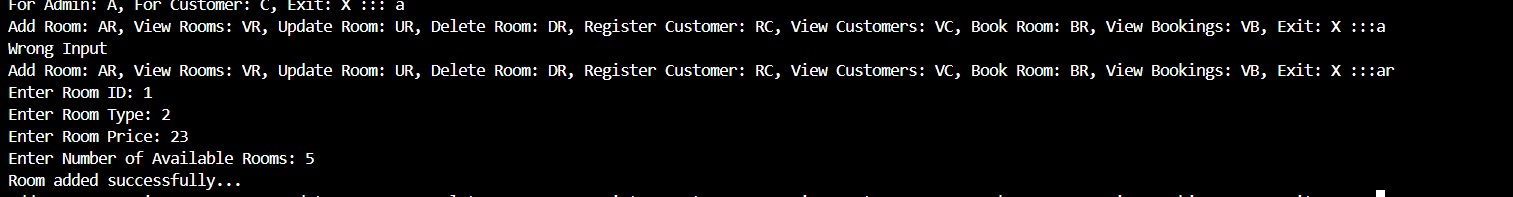
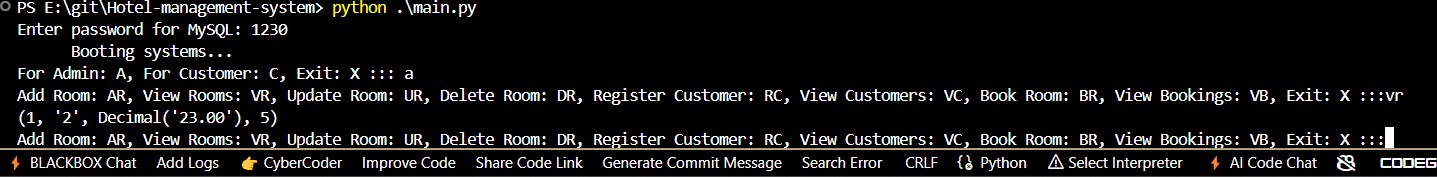
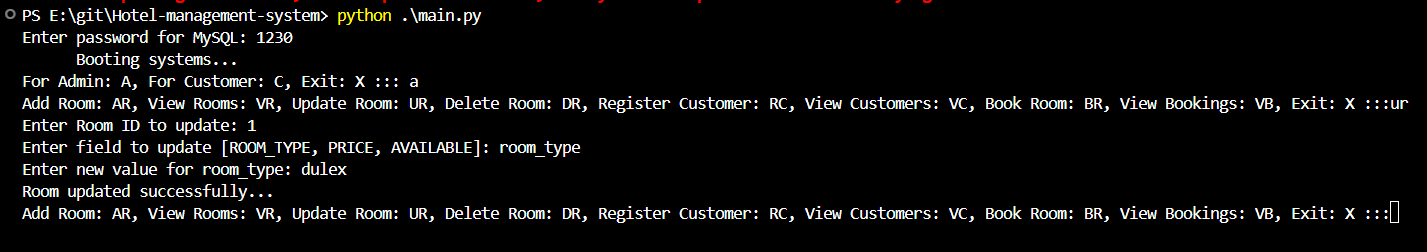
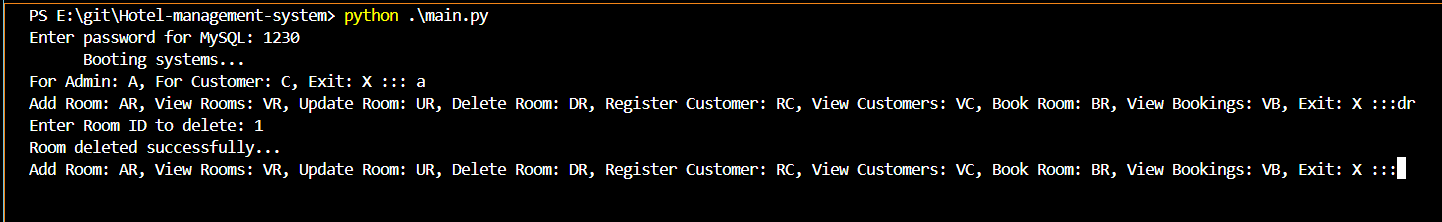
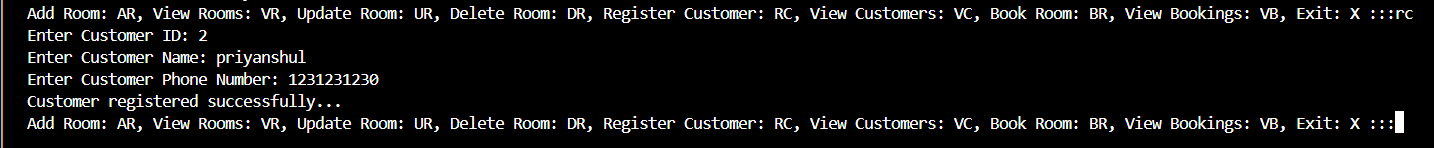
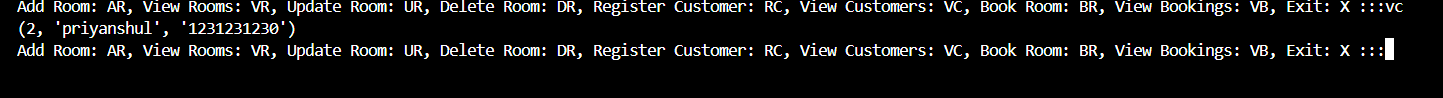
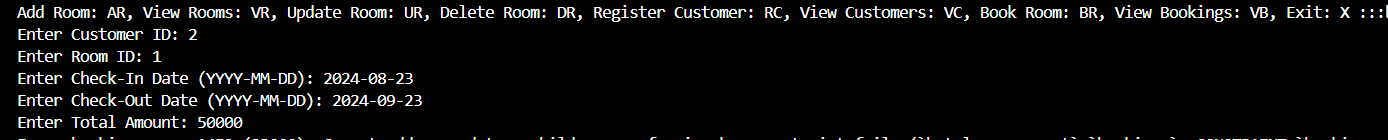
            print("Wrong Input")

if \_\_name\_\_ == "\_\_main\_\_":

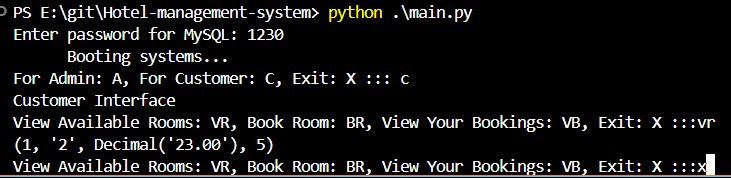
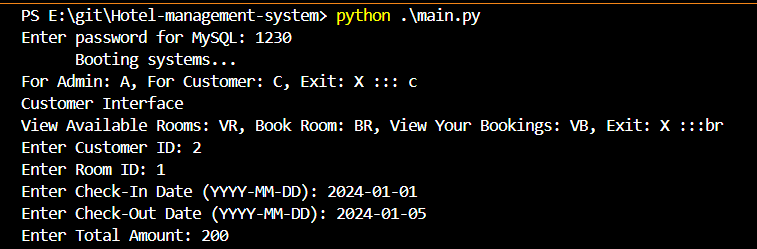
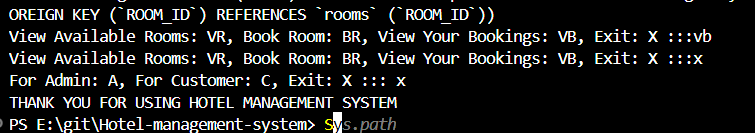
    main()

# **OUTPUT**

 Admin Controls

* **ADD ROOM**
* **VIEW ROOM**
* **UPDATE ROOM**
* **DELETE ROOM **
* **ADD CUSTOMER **
* **VIEW CUSTOMER**
* **BOOK ROOM**

 **User Controls**

* **VIEW ROOM**
* **BOOK ROOM**
* **VIEW YOUR BOOKINGS**
* **EXIT**

|  |
| --- |
|  |
| 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