PM SHRI KENDRIYA VIDYALAYA, BERHAMPUR

Computer Science Project

2023-24

Topic: Hotel Management System

Submitted By: AKASH KUMAR BEHERA (XII -'A')

Roll No.:

Guided By: Mr. SAROJ KANTA MISRA(PGT CS)

CERTIFICATE

This is to certify that **Akash Kumar Behera** of class: **XII A** of **PM SHRI KENDRIYA VIDYALAYA BERHAMPUR** has done his project on **HOTEL MANAGEMENT SYSTEM** under my supervision. He has taken interest and has shown at most sincerity in completion of this project.

I certify this project up to my expectation & as per guidelines issued by **CBSE, NEW DELHI.**

Internal Examiner *External Examiner*

Principal

ACKNOWLEDGMENT

It is with pleasure that I acknowledge my sincere gratitude to our teacher, ***MR. S.K. MISRA*** who taught and undertook the responsibility of teaching the subject computer science. I have been greatly benefited from his classes. I am especially indebted to our Principal ***MR.SHIVAPRIYA DASH*** who has always been a source of encouragement and support and without whose inspiration this project would not have been a successful I would like to place on record heartfelt thanks to him.

Finally, I would like to express my sincere appreciation for all the other students for my batch their friendship & the fine time that we all shared together.

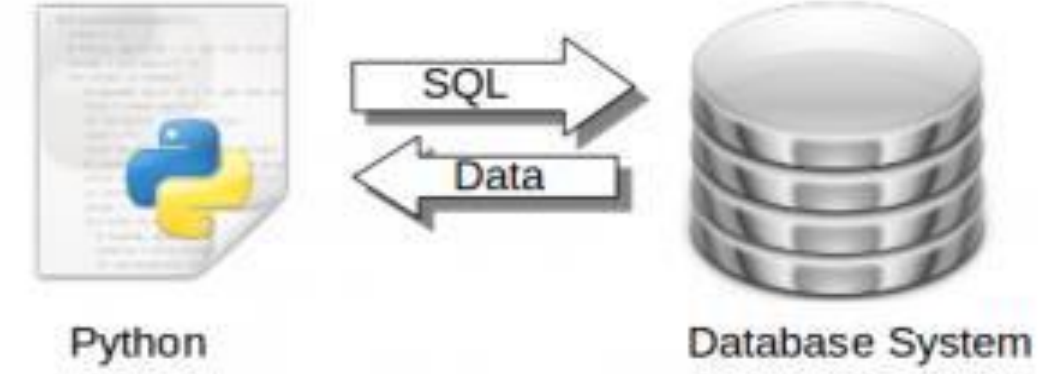
HARDWARE AND SOFTWARE REQUIRED

HARDWARE

1. PC
2. MOBILE PHONE

SOFTWARE

1. PYTHON (latest version)
2. MYSQL
3. PYTHON-MYSQL CONNECTOR



CONTENTS

HOTEL

A hotel is a commercial establishment that provides lodging, accommodation, and other services to travelers or tourists. Hotels typically offer a range of rooms or suites with varying levels of amenities and services. These establishments can vary widely in size and style, from small boutique hotels to large luxury resorts.

Key features of hotels include:

1. Accommodation: Hotels provide rooms or suites for guests to stay overnight or for an extended period.

2. Services: In addition to lodging, hotels often offer various services such as room service, housekeeping, concierge, and facilities like restaurants, gyms, swimming pools, conference rooms, and more.

3. Classification: Hotels are often classified based on factors like their size, amenities, and overall quality. Common classifications include budget/economy hotels, mid-range hotels, and luxury hotels.

4. Booking: Guests typically make reservations to secure their accommodations in advance. This can be done through various means, including online booking platforms, travel agencies, or directly with the hotel.

5. Hospitality Industry: Hotels are an integral part of the hospitality industry, which encompasses businesses that provide services to travelers and customers seeking leisure and comfort.

Overall, hotels play a crucial role in the travel and tourism industry, offering a temporary home away from home for individuals and groups seeking accommodation during their travels.

INTRODUCTION

The Hotel Management System (HMS) is a comprehensive software solution designed to streamline and optimize the operations of hotels and hospitality establishments. This integrated system combines various modules to efficiently manage different aspects of hotel functions, from reservation and guest services to billing and inventory management.

The reservation module of the HMS enables seamless booking processes, allowing guests to make reservations online or through the front desk. It maintains a centralized database of room availability, ensuring accurate and up-to-date information. The system also facilitates quick check-ins and check-outs, enhancing the overall guest experience.

Efficient guest services are a hallmark of the HMS, which includes features such as room service management, housekeeping, and personalized guest preferences. This ensures a high level of customer satisfaction and loyalty. Additionally, the system automates routine tasks, freeing up staff to focus on providing exceptional service.

Financial management is another key component, encompassing billing, invoicing, and reporting. The HMS generates detailed financial reports, helping hotel owners and managers make informed decisions to maximize profitability. It also integrates with point-of-sale systems, managing various revenue streams such as restaurants, bars, and spa services.

Inventory management is crucial for maintaining optimal stock levels in areas like housekeeping supplies and food and beverage items. The HMS tracks inventory, automates reordering processes, and minimizes wastage, contributing to cost efficiency.

Security features, including user access controls and data encryption, safeguard sensitive information and ensure compliance with privacy regulations. The system also aids in marketing efforts through guest relationship management (CRM) tools, allowing hotels to personalize promotions and loyalty programs.

In conclusion, the Hotel Management System is an indispensable tool for modern hotels, enhancing operational efficiency, improving guest satisfaction, and ultimately contributing to the success of hospitality businesses.

Key features of a Hotel Management System typically include:

1. ***Reservation Management: Allows the hotel staff to efficiently handle room bookings, cancellations, and modifications. It helps in managing room availability, rates, and guest preferences.***
2. ***Front Desk Operations: Enables the front desk staff to check-in/check-out guests, generate room keys, and manage guest information. It may also include features like guest folio creation and invoice generation.***
3. ***Billing and Invoicing: Handles the financial aspects of guest stays, including room charges, additional services, and taxes. It generates invoices and receipts for guest.***
4. ***Inventory Management : Tracks and manages hotel inventory, including housekeeping supplies ,linens, and other materials. This helps in maintaining optimal stock levels and preventing shortages.***
5. ***Housekeeping Management: Streamlines housekeeping tasks such as room cleaning schedules, maintenance requests, and inventory restocking. It ensures rooms are prepared efficiently for incoming guests.***

6. ***Reporting and Analytics**: Provides detailed reports and analytics on various aspects of hotel operations, allowing management to make informed decisions. Reports may cover occupancy rates, revenue, and guest demographics.*
7. ***Point of Sale (POS)**:Manages the hotel's various Revenue - generating outlets such as restaurants, bars, spa, and other services. It helps in tracking sales, inventory, and generating bills.*
8. ***Guest Relationship Management (CRM)**: Helps in building and maintaining relationships with guests by managing guest profiles, preferences, and feedback. This can lead to personalized services and improved guest satisfaction.*
9. *Security and Access Control:Ensures the security of guest data and property by implementing access controls and monitoring systems. It may include features like key card access and surveillance.*
10. *Channel Management: Manages the distribution of room inventory across various online booking channels to optimize occupancy rates and maximize revenue.*

11. ***Implementing a Hotel Management System not only enhances operational efficiency but also contributes to a better overall guest experience. The system can be tailored to meet the specific needs of different types and sizes of hospitality establishments, including hotels, resorts, motels, and bed-and-breakfasts.***

PYTHON CODES

```
import mysql.connector
from tabulate import tabulate
import random
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
y=a.cursor()

#all details for admin
#to show employee details
def emp_details():
    a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
    y=a.cursor()
    q="select * from employees"
    y.execute(q)
    r = y.fetchall()
    columns = [i[0] for i in y.description]
    print(tabulate(r, headers=columns, tablefmt="fancy_grid"))

#to show customer details
Defcustomdet():
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")


    y=a.cursor()
    x="select * from booking"
    y.execute(x)
    r = y.fetchall()
    columns = [i[0] for i in y.description]
    print(tabulate(r, headers=columns, tablefmt="fancy_grid"))
    a.commit()

#to show room details(all rooms,vacant rooms,booked rooms)
def room_details():
    while True:
        print("**********ROOM DETAILS**********")
        print("1.Show Rooms")
        print("2.Rooms Vacant")
        print("3.Rooms Booked")
        print("FOR EXIT ENTER ANY NO.: ")
        ch=int(input("Enter your choice: "))
        if ch==1:
            show_rooms()
        elif ch==2:
            room_vacant()
        elif ch==3:
            rooms_booked()

          else:
              print("INVALID INPUT")
              break
```

```
#to show all rooms
def showrooms():
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
    y=a.cursor()
    y.execute("select room_type,prices,count(*) from rooms group by room_type,prices;")
    r = y.fetchall()
    columns = [i[0] for i in y.description]
    print(tabulate(r, headers=columns, tablefmt="fancy_grid"))

#to show room vacant
def  room_vacant():
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
    y=a.cursor()
    av="Available"
    z="select * from rooms where Status ='{}'".format(av)
    y.execute(z)
    r = y.fetchall()

 columns = [i[0] for i in y.description]
    print(tabulate(r, headers=columns, tablefmt="fancy_grid"))



#to show booked rooms
def rooms_booked():
    a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
    y=a.cursor()
    bk="Booked"
    x="select * from rooms where Status ='{}'".format(bk)
    y.execute(x)
    r = y.fetchall()
    columns = [i[0] for i in y.description]
    print(tabulate(r, headers=columns, tablefmt="fancy_grid"))

#to show restaurant details
def restaurant_details():
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
    y=a.cursor()
    z="select*from orders"
    y.execute(z)
    r = y.fetchall()
    columns = [i[0] for i in y.description]
    print(tabulate(r, headers=columns, tablefmt="fancy_grid"))
```

```
#to show all feedback
def fedback():
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
    y=a.cursor()
    x="select * from fdback"
    y.execute(x)
    x = y.fetchall()
    columns = [i[0] for i in y.description]
    print(tabulate(x, headers=columns, tablefmt="fancy_grid"))

#restaurant
def restaurant():
#VIEW MENU
    def menu():
            a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_su
nset")
        y=a.cursor()
        b="select * from menu".format()
        y.execute(b)
        menu=y.fetchall()
        columns = [i[0] for i in y.description]
        print(tabulate(menu, headers=columns, tablefmt="fancy_grid"))
        if len(menu)>0:
            print("Available")
        a.commit()

        yn=int(input("Do uou want to order an item ?type(1 for yes/2 for back to main page):"))
        if yn ==1:
            b_order()
        elif yn==2:
            print("THANK YOU")
            print("YOU HAVE BEEN REDIRECTED TO MAIN PAGE")
            customer_slot()


#BOOKING ORDER
def b_order():
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")

        y=a.cursor()
        Id=int(input("ENTER DISH NO.: "))
        Quantity=int(input("ENTER QUANTITY: "))
        Name=input("ENTER YOUR NAME: ")
        Mobile_No=int(input("Enter mobilr no."))
        Address=input("Enter Address:")
        f=("select * from menu where Dish_ID={}").format(Id)
        y.execute(f)
        x=y.fetchall()
        itn=x[0][1]
        ip=x[0][3]
        tp=ip*Quantity
```

```
        ins="insert into
orders(ID,Name,Quantity,Item_Price,Total_Price,Mobile_No,Adress)values({},'{}',{},{},{},{},'{}')".format(Id,
itn,Quantity,ip,tp,Mobile_No,Address)
        y.execute(ins)
        print("THANKS FOR THE ORDER","\n\n","YOUR ORDER HAS BEEN ORDERED
SUCCESSFULLY","\n\n")
        print("YOU HAVE BEEN REDIRECTED TO THE MAIN PAGE")
        a.commit()

#VIEW ORDER
def vorders():
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
        y=a.cursor()
        m=int(input("Enter your number :"))
        n="select * from orders where Mobile_No={} ".format(m)
        print("\n","YOUR RECENT ORDERS","\n")
        y.execute(n)
        o=y.fetchall()
        columns = [i[0] for i in y.description]
        print(tabulate(o, headers=columns, tablefmt="fancy_grid"))
        for i in o:
            p="select * from menu,orders where Mobile_No={} and menu.Dish_ID=orders.ID".format(m)
            y.execute(p)
            q=y.fetchall()
            a.commit()

 #cancel order
    def corder():
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
        y=a.cursor()
        x=int(input("enter your number:"))
        s="delete from orders where Mobile_No={}".format(x)
        y.execute(s)
        print("\n\n","YOUR ORDER HAS BEEN CANCELLED")
        print("YOU HAVE BEEN REDIRECTED TO THE MAIN PAGE","\n\n")
        a.commit()

    #feedback
    def fdback():
a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
        y=a.cursor()
        fdn=input("Enter your name:")
        print("Write something about us...")
        fdi=input()
        x="insert into fdback values('{}','{}')".format(fdn,fdi)
        y.execute(x)
        print("\n\n")
        print("THANKYOU FOR YOUR FEEDBACK")
        print("\n")
        print("YOU HAVE BEEN REDIRECTED TO THE MAIN PAGE")
        a.commit()
```

```
    #welcome
    def start1():
        while True:
            print("\n")
            print("1. VIEW MENU")
            print("2. VIEW YOUR ORDERS")
            print("3. CANCEL ORDER")
            print("4. FEEDBACK")
            print("5. EXIT")
            ch1=int(input(" enter your choice:"))
            if ch1==1:
                menu()
            elif ch1==2:
                    vorders()
            elif ch1==3:
                corder( )
            elif ch1==4:
                fdback()
            elif ch1==5:
                    break

            else:
                print("\n","INVALID CHOICE"," \n" ,"TRY AGAIN."," \n")

    start1()

#booking rooms section
# Create the table if not exists
create_table = "CREATE TABLE IF NOT EXISTS booking (Booking_ID int(10) PRIMARY
KEY,Room_Type varchar(20) not null,Guest_Name VARCHAR(255),Phone_number varchar(15) not null,
Room_Number int(5) not null, Check_In_Date DATE, Check_Out_Date DATE)"
y.execute(create_table)
def booking_id():
    return random.randint(10000, 99999)
#to book room
def book_room(guest_name, ph_no,ro_no, check_in_date, check_out_date, td1, pr):
    try:
        a = mysql.connector.connect(host="localhost", user="root", password="admin",
database="hotel_sunset")
        y = a.cursor()
        b_id = booking_id()
        # Fetch available rooms
        c = "SELECT * FROM rooms WHERE Status = 'Available' and room_no={}".format(ro_no)
        y.execute(c)
        d = y.fetchall()
        if not d:
            print("No available rooms.")
            return None
        e = d[0]
        # Update room status to 'booked'
        update_query = "UPDATE rooms SET Status = 'Booked' WHERE room_no = {}".format(ro_no)
        y.execute(update_query)
```

```
        # Insert booking record
        ins = "INSERT INTO booking (Booking_ID, Room_Type, Guest_Name, Phone_number, Room_Number, Check_In_Date, Check_Out_Date, Total_Days, Price)VALUES ({}, '{}', '{}', {}, {}, '{}', '{}', {}, '{}')".format(b_id, e[1], guest_name, ph_no, ro_no, check_in_date, check_out_date, td1, pr)
        y.execute(ins)
        print("Room booked successfully! Room Number: ", ro_no)
        return b_id
    except:
        print("Error")
    finally:
        a.commit()

#to book delux room
def book_delux_room():
    a = mysql.connector.connect(host="localhost", user="root", password="admin", database="hotel_sunset")
    y = a.cursor()
    try:
        z=random.randint(101, 111)
        ro_no = z
        g_name = input("Enter guest name: ")
        ph_no = input("Enter your phone number: ")
        in_date = input("Enter check-in date (YYYY-MM-DD): ")
        out_date = input("Enter check-out date (YYYY-MM-DD):")
        total_days_query = "SELECT DATEDIFF('{}', '{}')".format(out_date, in_date)
        y.execute(total_days_query)
        td1 = y.fetchone()[0]
        pr = 15000 * td1
        booking_id = book_room(g_name, ph_no,ro_no, in_date, out_date, td1, pr)
        # Display booking history for the specific Booking_ID
        if booking_id is not None:
            q = "SELECT * FROM booking WHERE Booking_ID = {}".format(booking_id)
            y.execute(q)
            print("\nBooking History for Booking_ID {}: ".format(booking_id))
            x = y.fetchall()
            columns = [i[0] for i in y.description]
            print(tabulate(x, headers=columns, tablefmt="fancy_grid"))
    except :
        print("Error")

#to book double room
def book_double_room():
    a = mysql.connector.connect(host="localhost", user="root", password="admin", database="hotel_sunset")
    y = a.cursor()
    try:
        z=random.randint(201,211)
        ro_no = z
        g_name = input("Enter guest name: ")
        ph_no = input("Enter your phone number: ")
        in_date = input("Enter check-in date (YYYY-MM-DD): ")
        out_date = input("Enter check-out date (YYYY-MM-DD):")
```

```
      total_days_query = "SELECT DATEDIFF('{}', '{}')".format(out_date, in_date)
       y.execute(total_days_query)
       td1 = y.fetchone()[0]
       pr = 25000 * td1
       booking_id = book_room(g_name, ph_no,ro_no, in_date, out_date, td1, pr)
       # Display booking history for the specific Booking_ID
       if booking_id is not None:
          q = "SELECT * FROM booking WHERE Booking_ID = {}".format(booking_id)
          y.execute(q)
          print("\nBooking History for Booking_ID {}: ".format(booking_id))
          x = y.fetchall()
          columns = [i[0] for i in y.description]
          print(tabulate(x, headers=columns, tablefmt="fancy_grid"))

    except :
       print("Error")

#to book king room
def book_king_room():
    a = mysql.connector.connect(host="localhost", user="root", password="admin",
database="hotel_sunset")
    y = a.cursor()
    try:
       z=random.randint(301,311)
       ro_no = z
       g_name = input("Enter guest name: ")
       ph_no = input("Enter your phone number: ")
       in_date = input("Enter check-in date (YYYY-MM-DD): ")
       out_date = input("Enter check-out date (YYYY-MM-DD):")
       total_days_query = "SELECT DATEDIFF('{}', '{}')".format(out_date, in_date)
       y.execute(total_days_query)
       td1 = y.fetchone()[0]
       pr = 40000 * td1
       booking_id = book_room(g_name, ph_no,ro_no, in_date, out_date, td1, pr)


       # Display booking history for the specific Booking_ID
       if booking_id is not None:
          q = "SELECT * FROM booking WHERE Booking_ID = {}".format(booking_id)
          y.execute(q)
          print("\nBooking History for Booking_ID {}: ".format(booking_id))
          x = y.fetchall()
          columns = [i[0] for i in y.description]
          print(tabulate(x, headers=columns, tablefmt="fancy_grid"))
    except :
       print("Error")

#to book balcony rooms
def book_balcony_room():
    a = mysql.connector.connect(host="localhost", user="root", password="admin",
database="hotel_sunset")
    y = a.cursor()
```

```
    try:
        z=random.randint(401,411)
        ro_no = z
        g_name = input("Enter guest name: ")
        ph_no = input("Enter your phone number: ")

        in_date = input("Enter check-in date (YYYY-MM-DD): ")
        out_date = input("Enter check-out date (YYYY-MM-DD):")
        total_days_query = "SELECT DATEDIFF('{}', '{}')".format(out_date, in_date)
        y.execute(total_days_query)
        td1 = y.fetchone()[0]
        pr = 45000 * td1
        booking_id = book_room(g_name, ph_no,ro_no, in_date, out_date, td1, pr)
        # Display booking history for the specific Booking_ID
        if booking_id is not None:
            q = "SELECT * FROM booking WHERE Booking_ID = {}".format(booking_id)
            y.execute(q)
            print("\nBooking History for Booking_ID {}: ".format(booking_id))
            x = y.fetchall()
            columns = [i[0] for i in y.description]
            print(tabulate(x, headers=columns, tablefmt="fancy_grid"))
    except :
        print("Error")


#to book cavana room
def book_cavana():
    a = mysql.connector.connect(host="localhost", user="root", password="admin",
database="hotel_sunset")
    y = a.cursor()
    try:
        z=random.randint(501,506)
        ro_no = z
        g_name = input("Enter guest name: ")
        ph_no = input("Enter your phone number: ")
        in_date = input("Enter check-in date (YYYY-MM-DD): ")
        out_date = input("Enter check-out date (YYYY-MM-DD):")
        total_days_query = "SELECT DATEDIFF('{}', '{}')".format(out_date, in_date)
        y.execute(total_days_query)
        td1 = y.fetchone()[0]
        pr = 90000 * td1
        booking_id = book_room(g_name, ph_no,ro_no, in_date, out_date, td1, pr)
        # Display booking history for the specific Booking_ID
        if booking_id is not None:
            q = "SELECT * FROM booking WHERE

Booking_ID = {}".format(booking_id)
            y.execute(q)
            print("\nBooking History for Booking_ID {}: ".format(booking_id))
            x = y.fetchall()
            columns = [i[0] for i in y.description]
            print(tabulate(x, headers=columns, tablefmt="fancy_grid"))
```

```
    except :
        print("Error")

#user choice
def bookings():
    try:
        a = mysql.connector.connect(host="localhost", user="root", password="admin",
database="hotel_sunset")
        y = a.cursor()
        z="select * from book_rooms"
        y.execute(z)
        x = y.fetchall()
        columns = [i[0] for i in y.description]
        print(tabulate(x, headers=columns, tablefmt="fancy_grid"))
        roomchoice = int(input("Enter Your Option : "))
        if roomchoice == 1:

            book_delux_room()
        elif roomchoice == 2:
            book_double_room()
        elif roomchoice == 3:
            book_king_room()
        elif roomchoice == 4:
            book_balcony_room()
        elif roomchoice == 5:
            book_cavana()
        else:
            print("Sorry, May Be You Are Giving Me Wrong Input, Please Try Again !!! ")
    except:
        print("Error")
    finally:
        y.close()
        a.close()

#gaming section
def gaming():
    print("1. Table Tennis -----> 15000 Rs./HR")
    print("2. Bowling -----> 10000 Rs./HR")
    print("3. Snooker -----> 25000Rs./HR")
    print("4. VR World Gaming -----> 40000 Rs./HR")
    print("5. Video Games -----> 35000 Rs./HR")
    print("6. Swimming Pool Games -----> 50000Rs./HR")
    print("7. Exit")
    game=int(input("Enter What Game You Want To Play : "))
    hour=int(input("Enter No Of Hours You Want To Play : "))
    if game==1:
        print("YOU HAVE SELECTED TO PLAY : Table Tennis")
        gamingbill = hour * 15000
        price=print("Total price = ",gamingbill,"Rs.")
    elif game==2:
        print("YOU HAVE SELECTED TO PLAY : Bowling")
        gamingbill = hour * 10000
        price=print("Total price = ",gamingbill,"Rs.")
```

```
    elif game==3:
        print("YOU HAVE SELECTED TO PLAY : Snooker")
        gamingbill = hour * 25000
        price=print("Total price = ",gamingbill,"Rs.")
    elif game==4:
        print("YOU HAVE SELECTED TO PLAY : VR World Gaming")
        gamingbill = hour * 40000
        price=print("Total price = ",gamingbill,"Rs.")
    elif game==5:
        print("YOU HAVE SELECTED TO PLAY :Video Games")
        gamingbill = hour * 35000
        price=print("Total price = ",gamingbill,"Rs.")
    elif game ==6:
        print("YOU HAVE SELECTED TO PLAY : Swimming Pool Games")
        gamingbill = hour *50000
        price=print("Total price = ",gamingbill,"Rs.")
    else:
        print("Sorry ,May Be You Are Giving Me Wrong Input, Please Try Again !!! ")

#feedback to be asked by user
def feedback():
    a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
    y=a.cursor()
    fdn=input("Enter your name:")
    print("Write something about us...")
    fdi=input()
    x="insert into fdback values('{}','{}')".format(fdn,fdi)
    y.execute(x)
    print("\n\n")
    print("THANKYOU FOR YOUR FEEDBACK")
    print("\n")
    print("YOU HAVE BEEN REDIRECTED TO THE MAIN PAGE")
    a.commit()


#for admin
def admin_slot():
    while True:
        print("*********WELCOME ADMIN*********")
        print("1.Employees Details")
        print("2.Coustomer Details")
        print("3.Room Details")
        print("4.Feedback")
        print("5.Restaurant Details")
        print("6.Exit")
        a=int(input("enter your choice"))
        if a==1:
            emp_details()
        elif a==2:
            customdet()
        elif a==3:
            room_details()
```

```
        elif a==4:
            fedback()
        elif a==5:
            restaurant_details()
        elif a==6:
            break
        else:
            print("\n\n","INVALID CHOICE","\n","TRY AGAIN")
#for customer
def customer_slot():
    while True:
        print("*************NAMASTE*************")
        print("1.RESTAURANT")
        print("2.BOOK ROOMS")
        print("3.GAMING")
        print("4.FEEDBACK")
        print("5.EXIT")
        a=int(input("enter your choice"))
        if a==1:
            restaurant()
        elif a==2:
            bookings()
        elif a==3:
            gaming()
        elif a==4:
            feedback()
        elif a==5:
            break

        else:
            print("\n\n","INVALID CHOICE")

#first interface
while True:
    print("*********WELCOME TO HOTEL SUNSET*********")
    print("1.admin")
    print("2.customer")
    print("3.exit")
    a=int(input("who are you"))
    if a==1:
        def admin_login(a, username, password):

a=mysql.connector.connect(host="localhost",user="root",password="admin",database="hotel_sunset")
            y=a.cursor()
            try:

 # Check if the provided username and password match an admin record
                query = "SELECT * FROM users WHERE username = %s AND password = %s"
                y.execute(query, (username, password))
                admin_result = y.fetchone()
                if admin_result:
                    print("Login successful. Welcome, Admin!")
```

```
    else:
            print("Invalid username or password. Please try again.")
        except Exception as e:
            print(f"Error: {e}")
        a.close()
    admin_username = input("Enter Admin Username: ")
    admin_password = input("Enter Admin Password: ")
    admin_login(a, admin_username, admin_password)
    admin_slot()
elif a==2:
    customer_slot()
elif a==3:
    break
```

MYSQL DATABASE

* ***All tables used:-***

```
mysql> use hotel_sunset;
Database changed
mysql> show tables;
+-------------------------+
| Tables_in_hotel_sunset |
+-------------------------+
| booking                 |
| employees               |
| fdback                  |
| menu                    |
| orders                  |
| room_details            |
| rooms                   |
| users                   |
+-------------------------+
8 rows in set (0.00 sec)
```

* ***Describing all tables:-***

1. ***Booking:***

```
mysql> desc booking;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Booking_ID    | int         | NO   | PRI | NULL    |       |
| Room_Type     | varchar(20) | NO   |     | NULL    |       |
| Guest_Name    | varchar(200)| NO   |     | NULL    |       |
| Phone_Number  | varchar(14) | NO   |     | NULL    |       |
| Room_Number   | int         | NO   |     | NULL    |       |
| Check_In_Date | date        | YES  |     | NULL    |       |
| Check_Out_Date| date        | YES  |     | NULL    |       |
| Total_Days    | int         | NO   |     | NULL    |       |
| Price         | varchar(30) | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
9 rows in set (0.03 sec)
```

2. ***Employees:***

```
mysql> desc employees;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| emp_id     | int         | NO   | PRI | NULL    |       |
| emp_name   | varchar(90) | NO   |     | NULL    |       |
| gender     | varchar(10) | NO   |     | NULL    |       |
| emp_age    | int         | NO   |     | NULL    |       |
| shift      | varchar(20) | YES  |     | NULL    |       |
| shift_hour | varchar(35) | NO   |     | NULL    |       |
| salary     | int         | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

3. ***Feedback:***

```
mysql> desc fdback;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| Name     | varchar(100) | NO   |     | NULL    |       |
| Feedback | varchar(1000)| NO   |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

4. ***Menu:***

```
mysql> desc menu;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Dish_ID   | int         | NO   | PRI | NULL    |       |
| Dish_Name | varchar(30) | NO   |     | NULL    |       |
| Dish_Type | varchar(20) | NO   |     | NULL    |       |
| Price     | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

5. ***Orders:***

```
mysql> desc menu;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Dish_ID   | int         | NO   | PRI | NULL    |       |
| Dish_Name | varchar(30) | NO   |     | NULL    |       |
| Dish_Type | varchar(20) | NO   |     | NULL    |       |
| Price     | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

6. ***Rooms:***

```
mysql> desc rooms;
+-----------+-------------+------+-----+--------------------+------------------+
| Field     | Type        | Null | Key | Default            | Extra            |
+-----------+-------------+------+-----+--------------------+------------------+
| room_no   | int         | NO   | PRI | NULL               |                  |
| room_type | varchar(30) | NO   |     | NULL               |                  |
| prices    | int         | NO   |     | NULL               |                  |
| Status    | varchar(20) | YES  |     | _cp850\'Available\' | DEFAULT_GENERATED |
+-----------+-------------+------+-----+--------------------+------------------+
4 rows in set (0.00 sec)
```

7. ***Room Details:***

```
mysql> desc room_details;
+------------+------------+------+-----+---------+-------+
| Field      | Type       | Null | Key | Default | Extra |
+------------+------------+------+-----+---------+-------+
| room_no    | int        | YES  |     | NULL    |       |
| room_type  | varchar(98) | YES  |     | NULL    |       |
| room_vacant | int        | YES  |     | NULL    |       |
| price      | int        | YES  |     | NULL    |       |
+------------+------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

8. ***Users:***

```
mysql> desc users;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(20) | NO   |     | NULL    |       |
| password | varchar(10) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

OUTPUTS

- ***First interface with admin login:***

```
**********WELCOME TO HOTEL SUNSET**********
1.admin
2.customer
3.exit
who are you1
Enter Admin Username: akash
Enter Admin Password: akash2007
Login successful. Welcome, Admin!
```

- ***Customer login:***

```
**********WELCOME TO HOTEL SUNSET**********
1.admin
2.customer
3.exit
who are you2
***************NAMASTE***************
1.RESTAURANT
2.BOOK ROOMS
3.GAMING
4.FEEDBACK
5.EXIT
enter your choice
```

- ***From restaurant view menu:***

```
enter your choice:1
```

Dish_ID	Dish_Name	Type	Price
1	Idli	Veg.	150
2	Vada	Veg.	150
3	Masala Dosa	Veg.	200
4	Plain Dosa	Veg.	150
5	Chole Bhature	Veg.	160
6	Upma	Veg.	130
7	Masala Upma	Veg.	180
8	Puri	Veg.	140
9	Halwa	Veg.	100
10	Aloo Chop	Veg.	160
11	Plain Rice	Veg.	240
12	Fried Rice	Veg.	260
13	Biryani	Veg.	300
14	Paneer Biryani	Veg.	340
15	Special Biryani	Non-Veg.	450
16	Chicken Biryani	Non-Veg.	400
17	Roti	Veg.	100
18	Tandoori Roti	Veg.	150
19	Plain Naan	Veg.	100
20	Masala Naan	Veg.	140
21	Butter Naan	Veg.	130
22	Paratha	Veg.	100
23	Lachha Paratha	Veg.	120
24	Methi Paratha	Veg.	150
25	Paneer Butter Masala	Veg.	240
26	Paneer Khadai	Veg.	260
27	Mushroom Chilli	Veg.	270
28	Mushroom Curry	Veg.	250
29	Chicken Butter Masala	Non-Veg.	300
30	Chicken Tikka Masala	Non-Veg.	350
31	Mutton Curry	Non-Veg.	320
32	Mix Veg Curry	Veg.	280
33	Iced Tea	Beverage	180
34	Masala Cold Drink	Beverage	160
35	Lemonade	Beverage	140
36	Soda Pop	Beverage	150
37	Butterscotch Icecream	Beverage	190
38	Vanilla Icecream	Beverage	160
39	Chocolate Icecream	Beverage	180
40	Water Bottle	Beverage	100

```
Available
Do uou want to order an item ?type(1 for yes/2 for back to main page):
```

❖ ***Order an item:***

```
Available
Do uou want to order an item ?type(1 for yes/2 for back to main page):1
ENTER DISH NO.: 26
ENTER QUANTITY: 4
ENTER YOUR NAME: akash
Enter mobilr no.969202396
Enter Address:address
THANKS FOR THE ORDER

 YOUR ORDER HAS BEEN ORDERED SUCCESSFULLY


YOU HAVE BEEN REDIRECTED TO THE MAIN PAGE
```

❖ ***View order:***

```
1. VIEW MENU
2. VIEW YOUR ORDERS
3. CANCEL ORDER
4. FEEDBACK
5. EXIT
 enter your choice:2
Enter your number :969202396

 YOUR RECENT ORDERS
```

ID	Name	Quantity	Item_price	Total_Price	Mobile_No	Adress
26	Paneer Khadai	4	260	1040	969202396	address

❖ ***Canceling an order:***

```
1. VIEW MENU
2. VIEW YOUR ORDERS
3. CANCEL ORDER
4. FEEDBACK
5. EXIT
 enter your choice:3
enter your number:9692023969


 YOUR ORDER HAS BEEN CANCELLED
YOU HAVE BEEN REDIRECTED TO THE MAIN PAGE
```

❖ ***Feedback of restaurant:***

```
1. VIEW MENU
2. VIEW YOUR ORDERS
3. CANCEL ORDER
4. FEEDBACK
5. EXIT
 enter your choice:4
Enter your name:akash
Write something about us...
very nice restaurant...


THANKYOU FOR YOUR FEEDBACK

YOU HAVE BEEN REDIRECTED TO THE MAIN PAGE
```

❖ ***Book a room:***

```
1.RESTAURANT
2.BOOK ROOMS
3.GAMING
4.FEEDBACK
5.EXIT
enter your choice2
```

Sl_No	Room_Type	Price
1	Deulex F	15000
2	Double Room	25000
3	King Room	40000
4	Balcony Room	45000
5	Cavana	90000

```
Enter Your Option : 1
Enter guest name: akash
Enter your phone number: 9692023969
Enter check-in date (YYYY-MM-DD): 2023-12-22
Enter check-out date (YYYY-MM-DD):2023-12-27
Room booked successfully! Room Number:  104

Booking History for Booking_ID 73073:
```

Booking_ID	Room_Type	Guest_Name	Phone_number	Room_Number	Check_In_Date	Check_Out_Date	Total_Days	Price
73073	Delux F	akash	9692023969	104	2023-12-22	2023-12-27	5	75000

❖ ***Gaming:***

```
**************NAMASTE**************
1.RESTAURANT
2.BOOK ROOMS
3.GAMING
4.FEEDBACK
5.EXIT
enter your choice3
1. Table Tennis -----> 150 Rs./HR
2. Bowling -----> 100 Rs./HR
3. Snooker -----> 250 Rs./HR
4. VR World Gaming -----> 400 Rs./HR
5. Video Games -----> 300 Rs./HR
6. Swimming Pool Games -----> 350 Rs./HR
7. Exit
Enter What Game You Want To Play : 4
Enter No Of Hours You Want To Play : 4
YOU HAVE SELECTED TO PLAY : VR World Gaming
Total Price:  1600
```

❖ ***Feedback for hotel:***

```
**************NAMASTE**************
1.RESTAURANT
2.BOOK ROOMS
3.GAMING
4.FEEDBACK
5.EXIT
enter your choice4
Enter your name:akash
Write something about us...
its a very nice hotel and had a nice stay.


THANKYOU FOR YOUR FEEDBACK
```

❖ *Viewing employee details:*

```
*********WELCOME ADMIN*********
1.Employees Details
2.Coustomer Details
3.Room Details
4.Feedback
5.Restaurant Details
6.Exit
enter your choicel
```

emp_id	emp_name	gender	emp_age	shift	shift_hour	salary
10001	Ricky	Male	29	Morning	7:00 am - 14:00 pm	86000
10002	Rajesh	Male	33	Morning	7:00 am - 14:00 pm	86000
10003	Divya	Female	29	Evening	14:00 pm - 22:00 pm	140000
10004	Roy	Male	37	Night	22:00 pm - 7:00 am	255000
10005	Riya	Female	29	Morning	7:00 am - 14:00 pm	86000
10006	Dinesh	Male	43	Evening	14:00 pm - 22:00 pm	140000
10007	Smruti	Female	31	Night	22:00 pm - 7:00 am	255000
10008	Varun	Male	27	Evening	14:00 pm - 22:00 pm	140000
10009	Asish	Male	36	Evening	14:00 pm - 22:00 pm	140000
10010	Sweta	Female	39	Morning	7:00 am - 14:00 pm	86000

❖ *Viewing customer details:*

```
*********WELCOME ADMIN*********
1.Employees Details
2.Coustomer Details
3.Room Details
4.Feedback
5.Restaurant Details
6.Exit
enter your choice2
```

Booking_ID	Room_Type	Guest_Name	Phone_number	Room_Number	Check_In_Date	Check_Out_Date	Total_Days	Price
45010	Balcony Room	barun	908123567	401	2023-12-24	2023-12-29	5	225000
71098	Double Room	ricky	965032456	205	2023-12-20	2023-12-15	-5	-125000
73073	Delux F	akash	9692023969	104	2023-12-22	2023-12-27	5	75000
86465	Cavana	kaustav	789346720	501	2023-12-24	2024-01-02	9	810000

❖ *Viewing room details:*

```
**********ROOM DETAILS**********
1.Show Rooms
2.Rooms Vacant
3.Rooms Booked
FOR EXIT ENTER ANY NO.:
Enter your choice: 1
```

room_type	prices	count(*)
Delux F	15000	10
Double Room	25000	10
Kings Room	40000	10
Balcony Room	45000	10
Cavana	90000	5

❖ ***Showing vacant rooms:***

```
**********ROOM DETAILS**********
1.Show Rooms
2.Rooms Vacant
3.Rooms Booked
FOR EXIT ENTER ANY NO.:
Enter your choice: 2
```

room_no	room_type	prices	Status
101	Delux F	15000	Available
102	Delux F	15000	Available
103	Delux F	15000	Available
105	Delux F	15000	Available
106	Delux F	15000	Available
107	Delux F	15000	Available
109	Delux F	15000	Available
110	Delux F	15000	Available
201	Double Room	25000	Available
202	Double Room	25000	Available
203	Double Room	25000	Available
204	Double Room	25000	Available
206	Double Room	25000	Available
207	Double Room	25000	Available
208	Double Room	25000	Available
209	Double Room	25000	Available
210	Double Room	25000	Available
301	Kings Room	40000	Available
302	Kings Room	40000	Available
303	Kings Room	40000	Available
304	Kings Room	40000	Available
305	Kings Room	40000	Available
306	Kings Room	40000	Available
307	Kings Room	40000	Available
308	Kings Room	40000	Available
309	Kings Room	40000	Available
310	Kings Room	40000	Available
402	Balcony Room	45000	Available
403	Balcony Room	45000	Available
404	Balcony Room	45000	Available
405	Balcony Room	45000	Available
406	Balcony Room	45000	Available
407	Balcony Room	45000	Available
408	Balcony Room	45000	Available
409	Balcony Room	45000	Available
410	Balcony Room	45000	Available
502	Cavana	90000	Available
503	Cavana	90000	Available
504	Cavana	90000	Available

❖ ***Showing booked rooms:***

```
1.Show Rooms
2.Rooms Vacant
3.Rooms Booked
FOR EXIT ENTER ANY NO.:
Enter your choice: 3
```

room_no	room_type	prices	Status
104	Delux F	15000	Booked
108	Delux F	15000	Booked
205	Double Room	25000	Booked
401	Balcony Room	45000	Booked
501	Cavana	90000	Booked

- ***Viewing feedbacks from both hotel and restaurant:***

```
*********WELCOME ADMIN*********
1.Employees Details
2.Coustomer Details
3.Room Details
4.Feedback
5.Restaurant Details
6.Exit
enter your choice4
```

Name	Feedback
akash	very nice restaurant...
akash	its a very nice hotel and had a nice stay.

- ***Viewing restaurant details:***

```
*********WELCOME ADMIN*********
1.Employees Details
2.Coustomer Details
3.Room Details
4.Feedback
5.Restaurant Details
6.Exit
enter your choice5
```

ID	Name	Quantity	Item_price	Total_Price	Mobile_No	Adress
26	Paneer Khadai	4	260	1040	969202396	address

REFERENCES

- *Class 12 CS Sumita Arora book*
- *Class 11 CS Sumita Arora book*
- *Mysql*
- *Python IDLE*

THANK YOU