HOTEL MANAGEMENT SYSTEM

**BY:RIYA VARDHAN, ADITI NAITHANI**

**SABITA**





**CERTIFICATE**

This is to be certified that \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of Class XII-A has successfully completed the project of ‘HOTEL MANAGEMENT SYSTEM’ in Computer Science (083) in the Academic Year 2019-20.

Registration Number:

Date:

***External Examiner*** *Teacher In-Charge*

***Mr. Sunil Kumar T***

***PGT (Computer Science)***

***Kendriya Vidyalaya No.1, Panambur***

***Mangaluru – 575010***

*Principal*

***Kendriya Vidyalaya No.1, Panambur***

***Mangaluru - 575010***

**Hotel Management System**

**By team members:**

* Riya Vardhan
* Aditi Naithani
* Sabita

We would like to express our special thanks of gratitude to our teacher **MR. SUNIL KUMAR T** (PGT COMPUTER SCIENCE) as well as our principal **MR. Ganesh Suryakant Indrale** sir who gave us the golden opportunity to do this wonderful project on the topic **Hotel Management System**, which also helped us in doing a lot of Research and we came to know about so many new things. We also thank **the staff of Moti Mahal Hotel, Falnir Road Mangaluru-575001** who guided and helped us with this project.

Secondly, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

HARDWARE AND SOFTWARE REQUIREMENTS

**HARDWARE REQUIRED**

* A processing system capable of running Windows operating systems.
* A minimum 64 MB RAM (desirable: 128 MB) at server.
* Minimum 60 MB free space for files.
* A CD ROM drive.

**SOFTWARE REQUIRED**

* Windows 8 or above
* Wamp (32 or 64 bit)
* Microsoft Word / Microsoft PowerPoint

OBJECTIVES OF THE PROJECT

Over the years, we have developed in the field of technology and therefore our ways of storing information has changed. This project aims the goal of providing aneffective method of datamanagement, keeping our focus on the hotel management system.

* A computer-based management system to handle all the primary information to calculate the monthly statements. A separate database is maintained to handle the details required for correct and accurate calculations and output.
* Introduces effectiveness in the various activities of information updation, maintenance and search operations.
* The information maintained can be accessed by authorized users only ensuring high security.
* Easy updation enables to maintain up-to-date records.

INTRODUCTION TO THE PROJECT HOTEL MANAGEMENT SYSTEM

The Hotel Management system is to generalize and simplify the monthly or day to day activities of Hotel like Room activities, check in of New Customer, check out of the customer, assigning a room according to customer requirement, and finally compute the bill etc. which must be performed repeatedly on regular basis. To provide efficient, fast, reliable and user-friendly system is the basic motto behind this exercise. And show the Hotel room facilities, Book room. The main aim of the entire activity is to automate the process of day to day activities of Hotel like room activities, admission of a new customer, assign a room according to customer’s demand, checkout of a computer and releasing the room and finally compute the bill etc.

ABOUT THE EXISTING HOTEL MANAGEMENT SYSTEM

The existing system being used by the hotel we surveyed and analyzed was ‘Axis Rooms’ Managers. Its main intention is to increase and improve Hotel Business Distribution using break through technological solution with a vision to help clients transition to better with their business practices.

It is a well built, efficient and transparent software and has lived up to its expectations to reduce the strenuous tasks that take place in the process of Hotel Management.

THE LIMITATIONS OF THE PROPOSED SYSTEM

* Our proposed system was ‘Axis Rooms Channel Manager’ which was being used in Moti Mahal Hotel.
* After talking to the hotel staff about the software and a performing a system study, we have concluded that there are certain drawbacks:

1. Inability to store large amount of information with modifications.
2. Less security of the details of the login facilities of the individual hotel staff.

With our system, we would try to remove these drawbacks and limitations and make the program of hotel management work effectively.

**PURPOSE OF OUR PROJECT**

* The main purpose of our exercise is to perform each employee's activity in computerized way rather than manually which is time consuming
* The entire activity is to automate the process of day to day activities of hotel like:
* Room activities very safely.
* Assign a room according to customer's check -out of a computer
* Releasing the room and finally computer the bill in short time.

**DEPARTMENTS OF A HOTEL**

* Accounting Department
* Housekeeping Department
* Engineering Department
* Front Office Department
* Human Resource Department
* Security department
* Food and Beverage Department

MAIN OBJECTIVE IS TO AUTOMATE PROCESS OF DAILY ACTIVITIES LIKE:

1. Room activities
2. Addition of a new customer
3. Assignment of room as peer customer’s demands
4. Check out and releasing the room
5. Computing bill
6. Advanced online bookings
7. Online cancellation
8. List of regular customers and their activities

FEATURES

Becomes easy to find the details about:

* Availability of unoccupied rooms
* Past customer record
* Overall records of the customers
* Easy to know how much services taken by the customer
* Bill generation process will become easier

Easy generation of various types of reports such as:

* Service report
* Bill reports
* Customer reports

**EXISTING FEATURES**

"HOTEL MANAGEMENT SYSTEM" has been designed to computerize the following functions that are performed by the system:

* Room detail function
* Opening a new room
* Modification to room assigned
* Check -in and check-out function
* Sign up of new customer
* Room assigned related to customer's need
* Statement of customer details
* Room details

**INDIVIDUALL CUSTOMER REPORT**

1. Registration Module

**Inputs:-**

The basic customer registration details will be asked by the system.

**Outputs :-**

It immediately displays and stores the details of the customer.

1. Room Selection Module

**Inputs:-**

All the important inputs for choosing the perfect and desired rooms for the customer are asked.

**Output :-**

This will generate the room type and room number as per the customer’s convenience. Moreover, the number of vacancies or ‘no vacancy’ will be printed while booking. Finally, the cost of the accommodation is displayed and stored for the final bill.

(3) Meals Module

**Inputs :-**

The customer is asked the room number for accessing the food ordering facility. Further, the type of meal (breakfast/lunch/snacks/dinner) is asked by the system and henceforth the menu is displayed and the food can be easily order by inputting the appropriate food code mentioned in the menu itself.

**Outputs :-**

The bill for the current meal will be immediately prepared.

(4) Additional Services and Facility Module

**Inputs :-**

Once the customer inputs room number, he can avail the additional facilities listed in the menu and avail any of it by choosing his desired facility.

**Outputs :-**

This will immediately prepare and add these charges into the customer’s bill and display it.

(5) Event Organization Module

**Inputs :-**

Any existing or new customer can book rooms for organizing events. The size of the hall is asked and booked accordingly.

**Outputs :-**

Immediately prints all the necessary details for the organizer to pre–plan his event and prepares the bill immediately.

(6) Staff Details Module

**Inputs :-**

This module stores information for the existing or new staff that joins the hotel management committee.

**Output :-**

This immediately stores and displays the information of individual staff member. This is beneficial for the administrators to provide salary and convey messages to them.

(7) Bill module

**Inputs :-**

After asking the room number, it will locate the customer’s details and prepare the bill by summing up all the facilities, accommodation and meals availed from the hotel. Henceforth, the final checkout of the customer will be done through the administrator’s module.

**Outputs :-**

The final bill will be made after the required room number has been entered. If a customer has taken multiple rooms, separate bills for each room will be automatically made.

(8) Admin module and Login

**Input:-**

The username and password is asked. If the administrator is able to login successfully, he can perform multiple operations. The operations of modifying, overviewing and deleting customer of staff details and checking out a customer.

**Output :-**

Once the administrator types the correct username and password and the selection of the activity it wants to perform, the system opens an interface through which it can control the particular activity that it has entered.

**SOURCE CODE**:

#HOTEL MANAGEMENT SYSTEM

import random as r

from datetime import date

import mysql.connector as sqlc

conn=sqlc.connect(host='localhost',user='riyavardhan',password='riyav')

a=conn.cursor()

try:

q1="CREATE DATABASE MOTI\_MAHAL\_HOTELS"

a.execute(q1)

conn.commit()

q2="USE MOTI\_MAHAL\_HOTELS"

a.execute(q2)

conn.commit()

q3='CREATE TABLE REGISTRATION\_DETAILS(DOB DATE,REG\_NO INT(6) PRIMARY KEY NOT NULL,FNAME VARCHAR(20),LNAME VARCHAR(20),CITIZENSHIP VARCHAR(30),NATIONAL\_IDNO VARCHAR(20),GENDER VARCHAR(1),PHONE\_NO VARCHAR(13),ADDRESS VARCHAR(50),EMAIL VARCHAR(30),NO\_MEMBERS INT(3))'

a.execute(q3)

conn.commit()

q4="CREATE TABLE ROOM\_DETAILS(ROOM\_NO INT(3) PRIMARY KEY,POSITION VARCHAR(10) DEFAULT 'VACANT')"

a.execute(q4)

Q1='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(101)'

a.execute(Q1)

Q2='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(102)'

a.execute(Q2)

Q3='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(103)'

a.execute(Q3)

Q4='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(104)'

a.execute(Q4)

Q5='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(105)'

a.execute(Q5)

Q6='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(201)'

a.execute(Q6)

Q7='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(202)'

a.execute(Q7)

Q8='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(203)'

a.execute(Q8)

Q9='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(204)'

a.execute(Q9)

Q10='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(205)'

a.execute(Q10)

Q11='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(301)'

a.execute(Q11)

Q12='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(302)'

a.execute(Q12)

Q13='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(303)'

a.execute(Q13)

Q14='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(304)'

a.execute(Q14)

Q15='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(305)'

a.execute(Q15)

Q16='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(401)'

a.execute(Q16)

Q17='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(402)'

a.execute(Q17)

Q18='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(403)'

a.execute(Q18)

Q19='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(404)'

a.execute(Q19)

Q20='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(405)'

a.execute(Q20)

Q21='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(501)'

a.execute(Q21)

Q22='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(502)'

a.execute(Q22)

Q23='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(503)'

a.execute(Q23)

Q24='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(504)'

a.execute(Q24)

Q25='INSERT INTO ROOM\_DETAILS(ROOM\_NO)VALUES(505)'

a.execute(Q25)

conn.commit()

q8="CREATE TABLE STAFF(STAFF\_NO INT(4) PRIMARY KEY,NAME VARCHAR(30),MOBILE\_NO VARCHAR(20),GENDER VARCHAR(1),DESIGNATION VARCHAR(20),SALARY INT(8),NATID\_NO VARCHAR(20),EMAIL\_ID VARCHAR(30),ACC\_NO VARCHAR(20))"

a.execute(q8)

conn.commit()

q9="CREATE TABLE BILL(ROOM\_NO INT(6) REFERENCES REGISTRATION\_DETAILS(ROOM\_NO),ACCOMMODATION INT(6) DEFAULT 0,MEALS INT(6) DEFAULT 0,ADDON INT(6) DEFAULT 0,EVEORG INT(6) DEFAULT 0,TAX INT(6) DEFAULT 0,TOTAL INT(8) DEFAULT 0)"

a.execute(q9)

conn.commit()

q91='INSERT INTO BILL(ROOM\_NO)VALUES(101)'

a.execute(q91)

conn.commit()

q92='INSERT INTO BILL(ROOM\_NO)VALUES(102)'

a.execute(q92)

conn.commit()

q93='INSERT INTO BILL(ROOM\_NO)VALUES(103)'

a.execute(q93)

conn.commit()

q94='INSERT INTO BILL(ROOM\_NO)VALUES(104)'

a.execute(q94)

conn.commit()

q95='INSERT INTO BILL(ROOM\_NO)VALUES(105)'

a.execute(q95)

conn.commit()

q96='INSERT INTO BILL(ROOM\_NO)VALUES(201)'

a.execute(q96)

conn.commit()

q97='INSERT INTO BILL(ROOM\_NO)VALUES(202)'

a.execute(q97)

conn.commit()

q98='INSERT INTO BILL(ROOM\_NO)VALUES(203)'

a.execute(q98)

conn.commit()

q99='INSERT INTO BILL(ROOM\_NO)VALUES(204)'

a.execute(q99)

conn.commit()

q100='INSERT INTO BILL(ROOM\_NO)VALUES(205)'

a.execute(q100)

conn.commit()

q101='INSERT INTO BILL(ROOM\_NO)VALUES(301)'

a.execute(q101)

conn.commit()

q102='INSERT INTO BILL(ROOM\_NO)VALUES(302)'

a.execute(q102)

conn.commit()

q103='INSERT INTO BILL(ROOM\_NO)VALUES(303)'

a.execute(q103)

conn.commit()

q104='INSERT INTO BILL(ROOM\_NO)VALUES(304)'

a.execute(q104)

conn.commit()

q105='INSERT INTO BILL(ROOM\_NO)VALUES(305)'

a.execute(q105)

conn.commit()

q106='INSERT INTO BILL(ROOM\_NO)VALUES(401)'

a.execute(q106)

conn.commit()

q107='INSERT INTO BILL(ROOM\_NO)VALUES(402)'

a.execute(q107)

conn.commit()

q108='INSERT INTO BILL(ROOM\_NO)VALUES(403)'

a.execute(q108)

conn.commit()

q109='INSERT INTO BILL(ROOM\_NO)VALUES(404)'

a.execute(q109)

conn.commit()

q110='INSERT INTO BILL(ROOM\_NO)VALUES(405)'

a.execute(q110)

conn.commit()

q111='INSERT INTO BILL(ROOM\_NO)VALUES(501)'

a.execute(q111)

conn.commit()

q112='INSERT INTO BILL(ROOM\_NO)VALUES(502)'

a.execute(q112)

conn.commit()

q113='INSERT INTO BILL(ROOM\_NO)VALUES(503)'

a.execute(q113)

conn.commit()

q114='INSERT INTO BILL(ROOM\_NO)VALUES(504)'

a.execute(q114)

conn.commit()

q115='INSERT INTO BILL(ROOM\_NO)VALUES(505)'

a.execute(q115)

conn.commit()

except:

Q='USE MOTI\_MAHAL\_HOTELS'

a.execute(Q)

conn.commit()

conn.close()

#A module storing the registory details

def registration\_details():

conn=sqlc.connect(host='localhost',user='riyavardhan',password='riyav',database='MOTI\_MAHAL\_HOTELS')

a=conn.cursor()

today=date.today()

d=today.strftime("%B %d, %Y")

print()

print("Today",d,'is your date of booking')

reg\_no=r.randint(100000,999999)

print()

print('The registration number generated for you by the system is',reg\_no)

print()

print('Kindly take a note of this registration number as it may be required for completion of other booking procedures later on')

print()

fname=input("First name:")

lname=input("Last name:")

citz=input("Citizenship:")

natid\_no=input("National ID card number:")

gender=input("Gender(M/F):")

ph\_no=int(input("Phone number:"))

address=input("Residential address:")

email=input("Email ID:")

nom=int(input("No.of members:"))

q="INSERT INTO REGISTRATION\_DETAILS VALUES('{}',{},'{}','{}','{}','{}','{}','{}','{}','{}',{})".format(today,reg\_no,fname,lname,citz,natid\_no,gender,ph\_no,address,email,nom)

a.execute(q)

conn.commit()

print("Your details have been registered")

conn.close()

#A module for processing and selecting the ideal room

def room\_selection():

conn=sqlc.connect(host='localhost',user='riyavardhan',password='riyav',database='MOTI\_MAHAL\_HOTELS')

a=conn.cursor()

print('Have a look through the packages we offer for accommodation and other purposes')

print('Press any key to continue...')

input()

print('(1) Suite rooms (double bed,AC) : Rs.7000 per day')

print('(2) Premium rooms (double bed with AC) : Rs.6000 per day')

print('(3) Premium rooms (single bed with AC) : Rs.4000 per day')

print('(4) Premium rooms (double bed without AC) : Rs.3000 per day')

print('(5) Premium rooms (single bed without AC) : Rs.2000 per day')

print()

d=input('Are you sure you want to continue with purchasing our packages(Y/N):')

print()

if d[0].lower()=='y':

global nor

nor=int(input('Enter the number of rooms you want:'))

c=0

tc=0

for i in range(nor):

ch=int(input("Enter your choice of room type(1,2,3,4 or 5):"))

print()

if ch==1:

try:

tc+=7000

room="SELECT\*FROM ROOM\_DETAILS WHERE POSITION='VACANT' AND ROOM\_NO BETWEEN 101 AND 105"

a.execute(room)

d=a.fetchall()

print('Your room number is',d[0][0])

room\_no=d[0][0]

q='UPDATE ROOM\_DETAILS SET POSITION="OCCUPIED" WHERE ROOM\_NO={}'.format(room\_no)

a.execute(q)

conn.commit()

except:

print('Your desirable room is not vacant. Kindly try for some other room type')

elif ch==2:

try:

tc+=6000

room="SELECT\*FROM ROOM\_DETAILS WHERE POSITION='VACANT' AND ROOM\_NO BETWEEN 201 AND 205"

a.execute(room)

d=a.fetchall()

print('Your room number is',d[0][0])

room\_no=d[0][0]

q='UPDATE ROOM\_DETAILS SET POSITION="OCCUPIED" WHERE ROOM\_NO={}'.format(room\_no)

a.execute(q)

conn.commit()

except:

print('Your desirable room is not vacant. Kindly try for some other room type')

elif ch==3:

try:

tc+=4000

room="SELECT\*FROM ROOM\_DETAILS WHERE POSITION='VACANT' AND ROOM\_NO BETWEEN 301 AND 305"

a.execute(room)

d=a.fetchall()

print('Your room number is',d[0][0])

room\_no=d[0][0]

q='UPDATE ROOM\_DETAILS SET POSITION="OCCUPIED" WHERE ROOM\_NO={}'.format(room\_no)

a.execute(q)

conn.commit()

except:

print('Your desirable room is not vacant. Kindly try for some other room type')

elif ch==4:

try:

tc+=3000

room="SELECT\*FROM ROOM\_DETAILS WHERE POSITION='VACANT' AND ROOM\_NO BETWEEN 401 AND 405"

a.execute(room)

d=a.fetchall()

print('Your room number is',d[0][0])

room\_no=d[0][0]

q='UPDATE ROOM\_DETAILS SET POSITION="OCCUPIED" WHERE ROOM\_NO={}'.format(room\_no)

a.execute(q)

conn.commit()

except:

print('Your desirable room is not vacant. Kindly try for some other room type')

elif ch==5:

try:

tc+=2000

room="SELECT\*FROM ROOM\_DETAILS WHERE POSITION='VACANT' AND ROOM\_NO BETWEEN 501 AND 505"

a.execute(room)

d=a.fetchall()

print('Your room number is',d[0][0])

room\_no=d[0][0]

q='UPDATE ROOM\_DETAILS SET POSITION="OCCUPIED" WHERE ROOM\_NO={}'.format(room\_no)

a.execute(q)

conn.commit()

except:

print('Your desirable room is not vacant. Kindly try for some other room type')

else:

print('OOPS! You chose an invalid choice')

print()

print('The specificed details of your choice of room has been made and stored')

global rooms\_cost

rooms\_cost=tc

Q='UPDATE BILL SET ACCOMMODATION={} WHERE ROOM\_NO={}'.format(rooms\_cost,room\_no)

a.execute(Q)

conn.commit()

else:

print('''You can also call us on our toll free number 9482843844 to get further customisation of your choice.

Exiting...''')

conn.close()

#A module for ordering food

def meals():

conn=sqlc.connect(host='localhost',user='riyavardhan',password='riyav',database='MOTI\_MAHAL\_HOTELS')

a=conn.cursor()

global meals\_cost

room\_no=int(input("Room number:"))

ch=int(input('''The orders will be taken one by one

What would you like:

1.Breakfast

2.Lunch

3.Snacks

4.Dinner

Enter your choice(1,2,3 or 4):'''))

bc=lc=sc=dc=0

if ch==1:

print()

f=open('BREAKFAST.txt','r')

S=f.read()

print(S)

L=[70,120,60,50,45,55,30,75,80,60]

i='yes'

try:

while i[0].lower()=='y':

print()

icode=int(input('Enter the food code of the item you want:'))

bc+=L[icode-1]

meals\_cost+=L[icode-1]

i=input('Do you want to have more items(YES/NO):')

print('The total cost for breakfast is',bc)

except:

print('Invalid food code...')

f.close()

elif ch==2:

print()

f=open('LUNCH.txt','r')

S=f.read()

print(S)

L=[120,125,145,120,120,200,250,150,175,250,175,300,350,750,125]

i='yes'

try:

while i[0].lower()=='y':

print()

icode=int(input('Enter the food code of the item you want:'))

lc+=L[icode-1]

meals\_cost+=L[icode-1]

i=input('Do you want to have more items(YES/NO):')

print('The total cost for lunch is',lc)

except:

print('Invalid food code...')

f.close()

elif ch==3:

print()

f=open('SNACKS.txt','r')

S=f.read()

print(S)

L=[40,30,50,20,25,65,35,25,40,50,155,125,110,125,165]

i='yes'

try:

while i[0].lower()=='y':

print()

icode=int(input('Enter the food code of the item you want:'))

sc+=L[icode-1]

meals\_cost+=L[icode-1]

i=input('Do you want to have more items(YES/NO):')

print('The total cost for snacks is',sc)

except:

print('Invalid food code...')

f.close()

elif ch==4:

print()

f=open('DINNER.txt','r')

S=f.read()

print(S)

L=[135,75,80,225,150,185,160,175,85,50]

i='yes'

try:

while i[0].lower()=='y':

print()

icode=int(input('Enter the food code of the item you want:'))

dc+=L[icode-1]

meals\_cost+=L[icode-1]

i=input('Do you want to have more items(YES/NO):')

print('The total cost for dinner is',dc)

except:

print('Invalid food code...')

f.close()

else:

print('Invalid choice')

q='UPDATE BILL SET MEALS={} WHERE ROOM\_NO={}'.format(meals\_cost,room\_no)

a.execute(q)

conn.commit()

conn.close()

#A module to select additional provisions of the hotel

def addon\_services():

conn=sqlc.connect(host='localhost',user='riyavardhan',password='riyav',database='MOTI\_MAHAL\_HOTELS')

a=conn.cursor()

room\_no=int(input('Room number:'))

global ct

print('''Following are the services we provide in addition to our normal facilities.

1. Laundry (Rs.1225)

2. Spa (Rs.3000)

3. Room cleaning (Free)

4. Ironing service (Amount depends upon the clothing items)

5. Swimming (Free, Timings(7 AM - 7 PM))

6. Sports (Free)

7. Gym (Rs.500 )

''')

i='y'

while i[0].lower()=='y':

ads=int(input('Choose a service(1,2,3,4,5,6 or 7):'))

if ads==1:

ct+=1225

elif ads==2:

ct+=3000

elif ads==3:

ct+=0

elif ads==4:

ct+=1556

elif ads==5:

ct+=0

elif ads==6:

ct+=0

elif ads==7:

ct+=500

q="INSERT INTO ADDON\_SERVICE(ROOM\_NO,SERVICE\_NO) VALUES({},'{}')".format(room\_no,ads)

print()

i=input('Do you want to choose more facilities? [Y/N]:')

print()

global cost\_addon

cost\_addon=ct

q='UPDATE BILL SET ADDON={} WHERE ROOM\_NO={}'.format(cost\_addon,room\_no)

a.execute(q)

conn.commit()

print('Thank you for choosing our services! We are happy helping you !')

conn.close()

#A module for event organisers to use hotel halls

def eventorg():

conn=sqlc.connect(host='localhost',user='riyavardhan',password='riyav',database='MOTI\_MAHAL\_HOTELS')

a=conn.cursor()

room\_no=int(input('Room number:'))

eve=input('Name of the event to be organised:')

name=input('Organiser name:')

org=input('Organisation name:')

idno=input('National ID card number:')

date=input('Enter the date or organisation in [YYYY-MM-DD] format:')

reg\_no=r.randint(100000,999999)

global tce

hall=int(input('''What hall size do you require?

1. Large

2. Medium

3. Small

Your choice(1,2,3):'''))

print('The hall number',hall,'has been allotted to you')

global tc

d=int(input('Number of days you would require the hall:'))

if hall==1:

print('Your large-sized hall has been booked')

c=15000

tce+=c\*d

print('The cost of the hall booking is',tce)

elif hall==2:

print('Your medium-sized hall has been booked')

c=12000

tce+=c\*d

print('The cost of the hall booking is',tce)

elif hall==3:

print('Your small-sized hall has been booked')

c=10000

tce+=c\*d

print('The cost of the hall booking is',tce)

else:

print('Oops! You entered an invalid choice')

q='UPDATE BILL SET EVEORG={} WHERE ROOM\_NO={}'.format(tce,room\_no)

a.execute(q)

conn.commit()

conn.close()

#A module to exclusively store the deatils of staff members

def staff():

conn=sqlc.connect(host='localhost',user='riyavardhan',password='riyav',database='MOTI\_MAHAL\_HOTELS')

a=conn.cursor()

sno=r.randint(1000,9999)

print('Staff number:',sno)

name=input('Staff name:')

mno=input('Mobile number:')

gender=input('Gender(M/F):')

desg=input('Designation:')

salary=int(input('Salary of the staff:'))

natid\_no=input('National ID Card number:')

eid=input('Email ID:')

accno=input('Account number:')

qu="INSERT INTO STAFF VALUES({},'{}','{}','{}','{}',{},'{}','{}','{}')".format(sno,name,mno,gender,desg,salary,natid\_no,eid,accno)

a.execute(qu)

conn.commit()

print()

print('Your details have been entered.')

conn.close()

#A module to generate the final bill

def bill():

conn=sqlc.connect(host='localhost',user='riyavardhan',password='riyav',database='MOTI\_MAHAL\_HOTELS')

a=conn.cursor()

r\_no=int(input('Room number:'))

print('Processing details...')

print()

print()

Q='UPDATE BILL SET TAX=0.145\*(ACCOMMODATION+MEALS+ADDON+EVEORG) WHERE ROOM\_NO={}'.format(r\_no)

a.execute(Q)

conn.commit()

Q2='UPDATE BILL SET TOTAL=1.145\*(ACCOMMODATION+MEALS+ADDON+EVEORG) WHERE ROOM\_NO={}'.format(r\_no)

a.execute(Q2)

conn.commit()

q='SELECT\*FROM BILL WHERE ROOM\_NO={}'.format(r\_no)

a.execute(q)

data=a.fetchall()

print('\t\t\t MOTI MAHAL HOTELS')

print('\t\t\t\t BILL')

print('Room number :',data[0][0])

print()

print('Accommodation :',data[0][1])

print('Meals :',data[0][2])

print('Additional service :',data[0][3])

print('Event organisation :',data[0][4])

print('Tax (GST) :',data[0][5])

print('Total :',data[0][6])

print()

print('Thank you for choosing Moti Mahal for your stay! We hope you visit again!')

conn.close()

#\_main\_#

print()

print('\t\t\t\t WELCOME TO THE MOTI MAHAL HOTELS')

print('\t\t\t\t---------------------------------')

while True:

print('Are you :')

print('1. A Customer')

print('2. An administrator')

print('3. A staff')

print('4. About')

print('5. Exit the Moti mahal page')

ch=int(input("Enter your choice(1,2,3,4 or 5):"))

if ch==1:

print()

print('---------------------------------------------------')

print(' CUSTOMER PAGE ')

print('---------------------------------------------------')

print('Dear Customer! Welcome to the Moti Mahal Hotel!')

print('Kindly register into our page to buy packages')

c=input('Do you want to continue with the registration (yes/no):')

if c[0].lower()=='y':

registration\_details() #Calling of function

print('\n'\*3)

input('Press any key to continue . . .')

print()

room\_selection() #Calling of function

input('Press any key to continue . . .')

print()

print('Your registration and the required type of room has been successfully reserved for you.')

print('You may now start with exploring and availing the facilities in our hotel')

print()

input('Press any key to continue . . .')

print()

meals\_cost=0

ct=0

tce=0

while True:

s=int(input('''Which of the following services you would like to avail...

1. Order food

2. Avail additional facilities

3. Organise an event

4. Exit

Your choice(1,2,3 or 4):'''))

if s==1:

print('----------------------------------------------------')

print(' FOOD ORDERS ')

print('----------------------------------------------------')

meals() #Calling of function

print()

input('Press any key to continue . . .')

print()

elif s==2:

print('----------------------------------------------------')

print(' ADDITIONAL FACILITIES ')

print('----------------------------------------------------')

addon\_services() #Calling of function

print()

input('Press any key to continue . . .')

print()

elif s==3:

print('----------------------------------------------------')

print(' EVENT ORGANISATION ')

print('----------------------------------------------------')

eventorg() #Calling of function

print()

input('Press any key to continue . . .')

print()

elif s==4:

print('Exiting . . . .')

break

input('Press any key to continue . . .')

print()

else:

print('Invalid choice')

input('Press any key to continue . . .')

print()

print()

print('Thank you for chosing our services! You may now proceed for the billing')

input('Press any key to continue . . .')

print()

print('------------------------------------------------------')

print(' BILL ')

print('------------------------------------------------------')

for i in range(nor):

print('\_\_\_\_\_\_\_\_\_\_\_BILL',(i+1),'\_\_\_\_\_\_\_\_\_\_')

bill() #Calling of function

print()

print('We hope to see you again')

print('\n'\*5)

input('Press any key to continue . . .')

else:

print('Thank you for visiting our page! We hope to provide you services in the future.')

input('Press any key to continue . . .')

elif ch==2:

#Administrator module

print('--------------------------------------------------')

print(' ADMINISTRATOR PAGE ')

print('--------------------------------------------------')

print('LOGIN')

user=input('Admin username:')

password=input('Admin password:')

try:

c=sqlc.connect(host='localhost',user= user,password= password,database='MOTI\_MAHAL\_HOTELS')

a=c.cursor()

while True:

print('''What operations would you like to perform?

1. Modify customer details as per their requirement

2. Overviewing the customer's bill

3. Overviewing the staff details

4. Modifying the staff details

5. Delete a staff record

6. Delete a customer record

7. Check out a customer

8. Exit

''')

s=int(input('Your choice(1,2,3,4,5,6 or 7):'))

if s==1:

print()

print('\t\t\t CUSTOMER DATA MODIFICATION')

print()

try:

#Updation of registration details

reg\_no=int(input('Enter the registration number of the customer whose details are to be modified:'))

print('Kindly enter the new or modified details of the customer.')

fname=input("First name:")

q1="UPDATE REGISTRATION\_DETAILS SET FNAME='{}' WHERE REG\_NO={}".format(fname,reg\_no)

a.execute(q1)

c.commit()

lname=input("Last name:")

q2="UPDATE REGISTRATION\_DETAILS SET LNAME='{}' WHERE REG\_NO={}".format(lname,reg\_no)

a.execute(q2)

c.commit()

citz=input("Citizenship:")

q3="UPDATE REGISTRATION\_DETAILS SET CITIZENSHIP='{}' WHERE REG\_NO={}".format(citz,reg\_no)

a.execute(q3)

c.commit()

natid\_no=input("National ID card number:")

q4="UPDATE REGISTRATION\_DETAILS SET NATIONAL\_IDNO='{}' WHERE REG\_NO={}".format(natid\_no,reg\_no)

a.execute(q4)

c.commit()

gender=input("Gender(M/F):")

q5="UPDATE REGISTRATION\_DETAILS SET GENDER='{}' WHERE REG\_NO={}".format(gender,reg\_no)

a.execute(q5)

c.commit()

ph\_no=int(input("Phone number:"))

q6="UPDATE REGISTRATION\_DETAILS SET PHONE\_NO='{}' WHERE REG\_NO={}".format(ph\_no,reg\_no)

a.execute(q6)

c.commit()

address=input("Residential address:")

q7="UPDATE REGISTRATION\_DETAILS SET ADDRESS='{}' WHERE REG\_NO={}".format(address,reg\_no)

a.execute(q7)

c.commit()

email=input("Email ID:")

q8="UPDATE REGISTRATION\_DETAILS SET EMAIL='{}' WHERE REG\_NO={}".format(email,reg\_no)

a.execute(q8)

c.commit()

nm=int(input("No.of members:"))

q9="UPDATE REGISTRATION\_DETAILS SET NO\_MEMBERS={} WHERE REG\_NO={}".format(nm,reg\_no)

a.execute(q9)

c.commit()

print()

print('Details have been successfully updated !')

input('Press any key to continue . . .')

print()

except:

print('An unexpected connection error occured! Kindly try again!')

input('Press any key to continue . . .')

print()

elif s==2:

try:

print()

print('\t\t\t CUSTOMER BILL OVERVIEW')

print()

room\_no=int(input('Enter the room number of the customer whose billing details you want to view:'))

print()

q='SELECT\*FROM BILL WHERE ROOM\_NO={}'.format(room\_no)

a.execute(q)

data=a.fetchall()

print('Accommodation :',data[0][1])

print('Food :',data[0][2])

print('Additional service:',data[0][3])

print('Event organisation:',data[0][4])

print()

print('GST :',data[0][5])

print('Total :',data[0][6])

print()

input('Press any key to continue . . .')

print()

except:

print('An unexpected error occured! Kindly try again!')

print()

input('Press any key to continue . . .')

print()

elif s==3:

try:

print()

print('\t\t\t OVERVIEWING THE STAFF DETAILS')

sno=int(input('Enter the staff number whose details you want to view:'))

q='SELECT\*FROM STAFF WHERE STAFF\_NO={}'.format(sno)

a.execute(q)

data=a.fetchall()

print('Staff number :',data[0][0])

print('Staff name ;',data[0][1])

print('Mobile number :',data[0][2])

print('Gender :',data[0][3])

print('Designation :',data[0][4])

print('Salary :',data[0][5])

print('National ID no. :',data[0][6])

print('Email ID :',data[0][7])

print('Account number :',data[0][8])

print()

input('Press any key to continue . . .')

print()

except:

print('An unexpected error occured or the staff number not found! Kindly try again!')

print()

input('Press any key to continue . . .')

print()

elif s==4:

try:

print()

print('\t\t\t STAFF DATA MODIFICATION')

reg\_no=int(input('Enter the staff number whose details are to be modified:'))

print('Kindly enter the new or modified details of the staff.')

name=input("Staff name:")

q1="UPDATE STAFF SET NAME='{}' WHERE STAFF\_NO={}".format(name,reg\_no)

a.execute(q1)

c.commit()

mno=int(input('Mobile number:'))

q2="UPDATE STAFF SET MOBILE\_NO={} WHERE STAFF\_NO={}".format(mno,reg\_no)

a.execute(q2)

c.commit()

gender=input("Gender(M/F):")

q3="UPDATE STAFF SET GENDER='{}' WHERE STAFF\_NO={}".format(gender,reg\_no)

a.execute(q3)

c.commit()

desg=input('Designation:')

q4="UPDATE STAFF SET DESIGNATION='{}' WHERE STAFF\_NO={}".format(desg,reg\_no)

a.execute(q4)

c.commit()

sal=int(input("Salary:"))

q5="UPDATE STAFF SET SALARY={} WHERE STAFF\_NO={}".format(sal,reg\_no)

a.execute(q5)

c.commit()

natid\_no=input("National ID card number:")

q6="UPDATE STAFF SET NATID\_NO='{}' WHERE STAFF\_NO={}".format(natid\_no,reg\_no)

a.execute(q6)

c.commit()

email=input("Email ID:")

q7="UPDATE STAFF SET EMAIL\_ID='{}' WHERE STAFF\_NO={}".format(email,reg\_no)

a.execute(q7)

c.commit()

acno=input('Account number:')

q8="UPDATE STAFF SET ACC\_NO='{}' WHERE STAFF\_NO={}".format(acno,reg\_no)

a.execute(q8)

c.commit()

print()

print('Details have been successfully updated !')

print()

input('Press any key to continue . . .')

print()

except:

print('An unexpected error occured or the record could not be found! Kindly try again!')

print()

input('Press any key to continue . . .')

print()

elif s==5:

try:

print()

print('\t\t\t STAFF DATA RECORD DELETION')

stno=int(input('Enter staff number whose record you want to delete:'))

q="DELETE FROM STAFF WHERE STAFF\_NO={}".format(stno)

a.execute(q)

c.commit()

print('Record of the staff has been deleted successfully')

print()

input('Press any key to continue . . .')

print()

except:

print('An unexpected error occured or the record could not be found! Kindly try again!')

print()

input('Press any key to continue . . .')

print()

elif s==6:

try:

print()

print('\t\t\t CUSTOMER DATA RECORD DELETION')

cno=int(input('Enter registration number whose record you want to delete:'))

q="DELETE FROM REGISTRATION\_DETAILS WHERE REG\_NO={}".format(cno)

a.execute(q)

c.commit()

print('Record of the customer has been deleted successfully')

print()

input('Press any key to continue . . .')

print()

except:

print('An unexpected error occured or the record could not be found! Kindly try again!')

print()

input('Press any key to continue . . .')

print()

elif s==7:

print('\t\t\t\t CUSTOMER CHECKOUT')

rno=int(input('Room number:'))

q="UPDATE ROOM\_DETAILS SET POSITION='VACANT' WHERE ROOM\_NO={}".format(rno)

a.execute(q)

c.commit()

print('Customer has been checked out')

print()

input('Press any key to continue . . .')

print()

elif s==8:

print()

print('EXITING...')

break

input('Press any key to continue . . .')

c.close()

except:

print('KINDLY CHECK YOUR USERNAME AND PASSWORD. TRY AGAIN!')

input('Press any key to continue . . .')

print()

elif ch==3:

print()

print('--------------------------------------------------')

print(' STAFF PAGE ')

print('--------------------------------------------------')

print('Thank you for showing interest in joining our community')

print()

staff()

print()

input('Press any key to continue . . .')

print()

elif ch==4:

print()

print('--------------------------------------------------')

print(' THE CREATORS ')

print('--------------------------------------------------')

print('1. Riya Vardhan')

print('2. Aditi Naithani')

print('3. Sabita Rani')

print()

input('Press any key to continue . . .')

print()

elif ch==5:

print('Exiting permanantly from HOME . . .')

print()

break

input('Press any key to continue . . .')

print()

else:

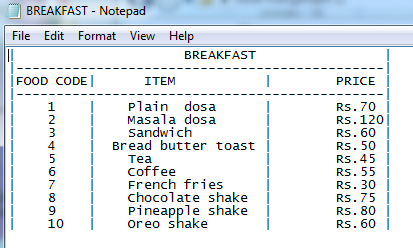
print('An invalid choice!')

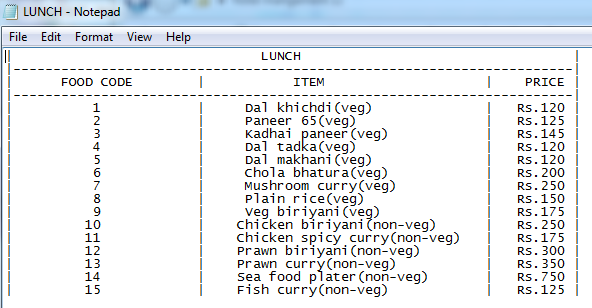
print()

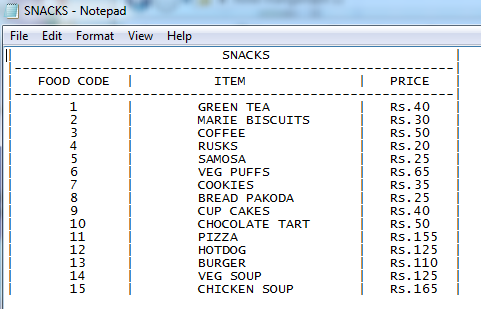
input('Press any key to continue . . .')

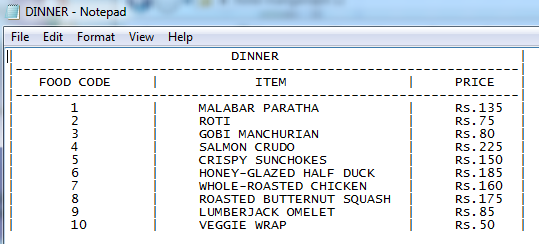
print()

**FILES USED**







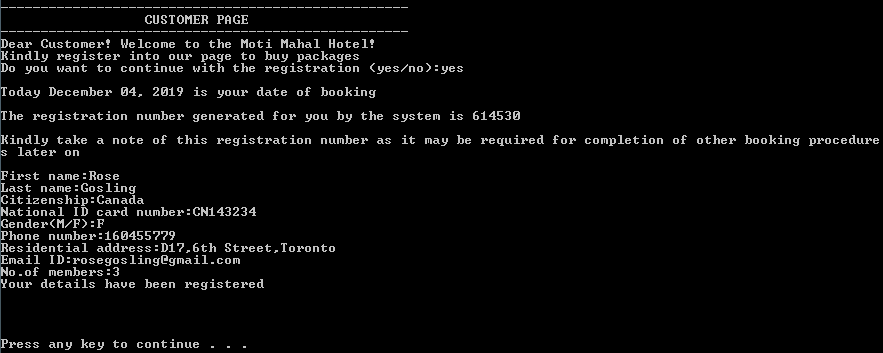


**OUTPUTS :**

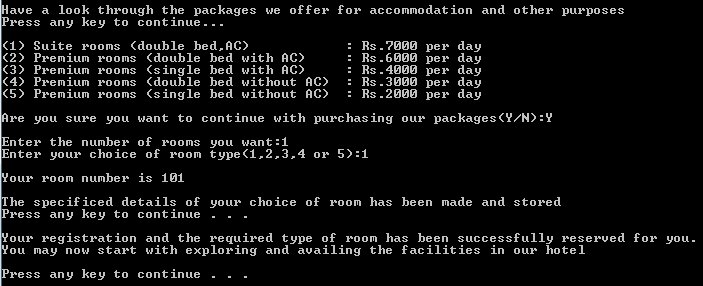
The Main Menu



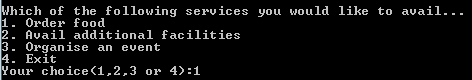
Customer Registration



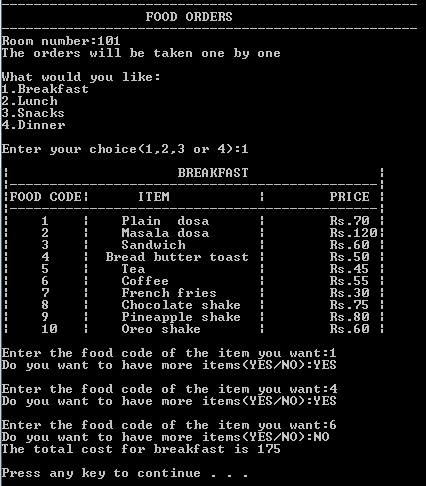
Room selection

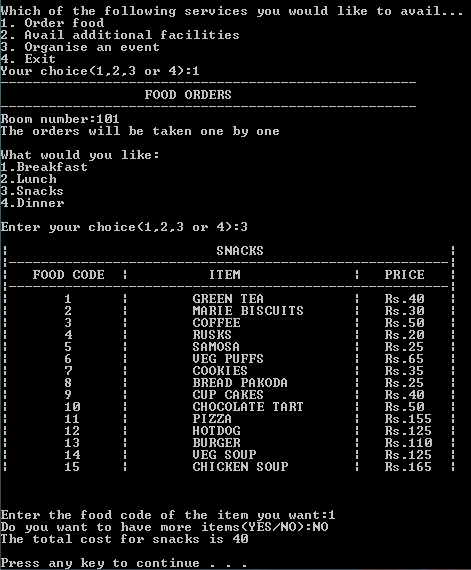


Sub-Menu to avail services

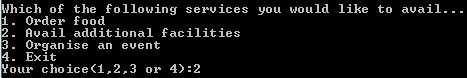


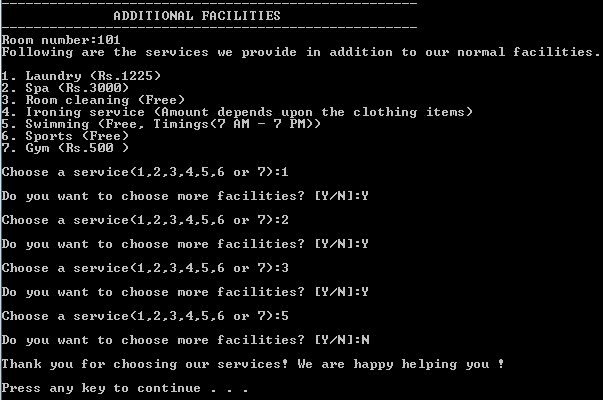
Placing food orders



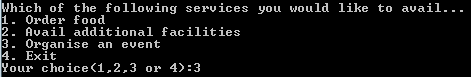


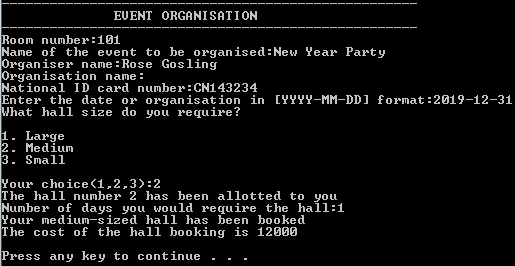
Accessing additional facilities



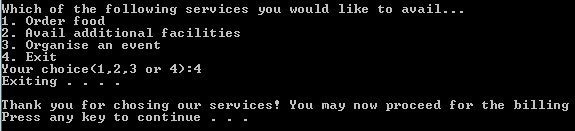


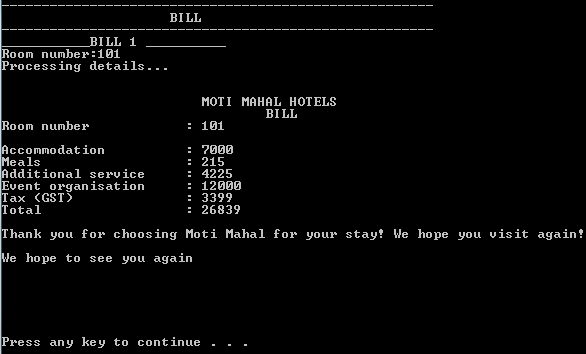
Event Organization and Hall booking





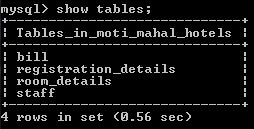
Automated Billing





The automatic creation and use of database and tables

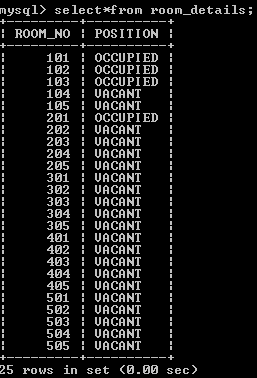


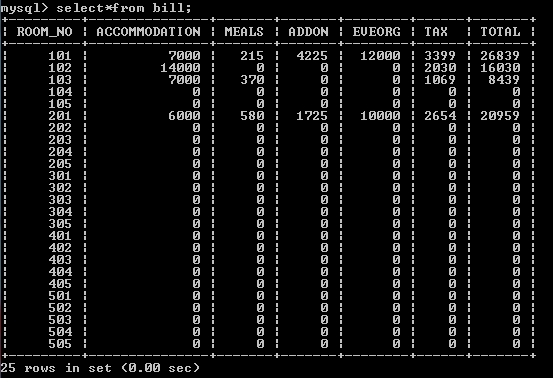


The storage of customer data and activities

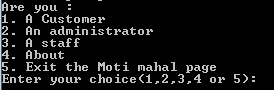


Registration data entered

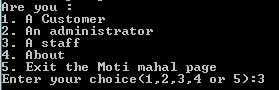




Directed to the Main Menu after the Billing automatically



Staff details entry

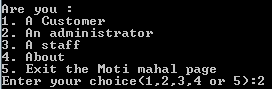


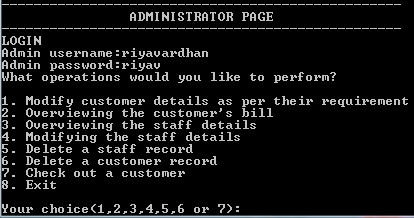


Storage of staff data



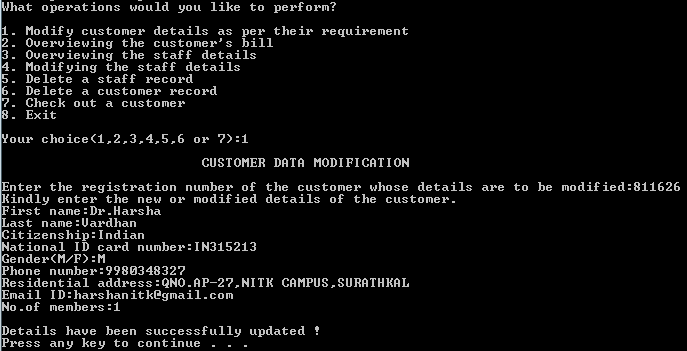
Administrator’s Login



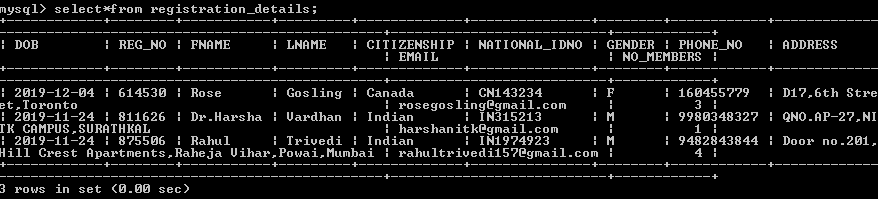


Administrator’s functions:-

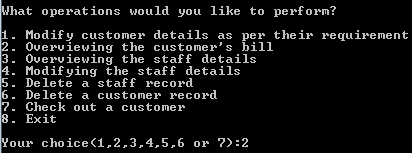
1. Customer Details’ Modification

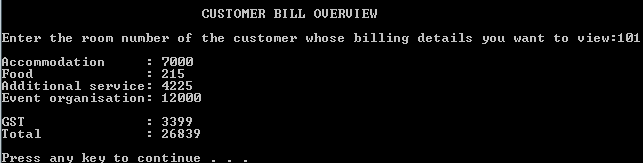


Modification in the Database

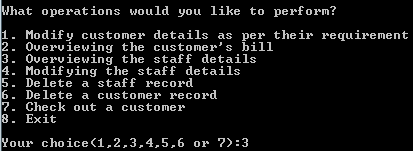


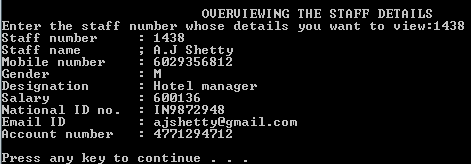
1. Customer Bill Overview



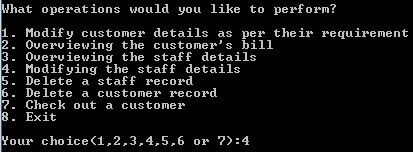


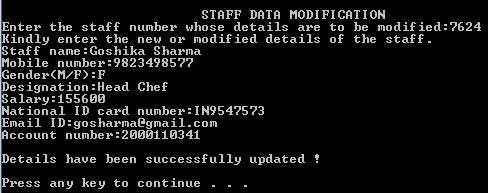
1. Staff Detail Overview





1. Staff Detail Modification

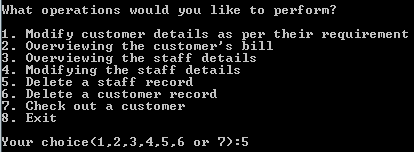




Modification in the Database

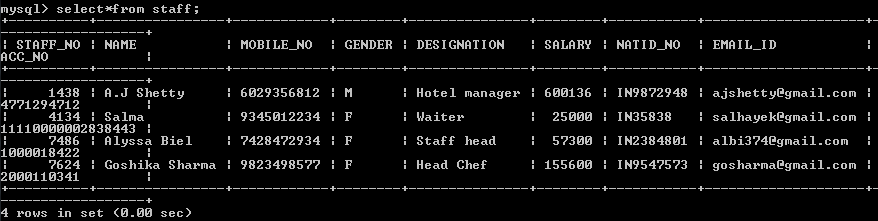


1. Staff Record Deletion





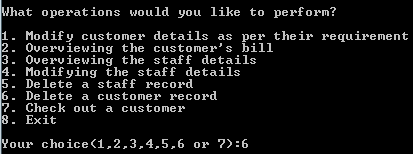
Modification in the Database

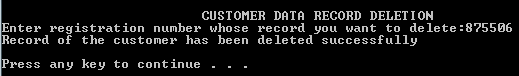


(After deletion)

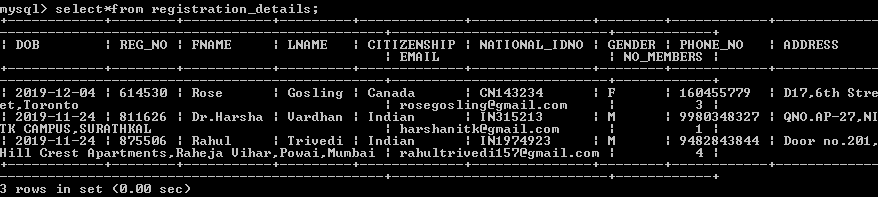


1. Customer Record Deletion

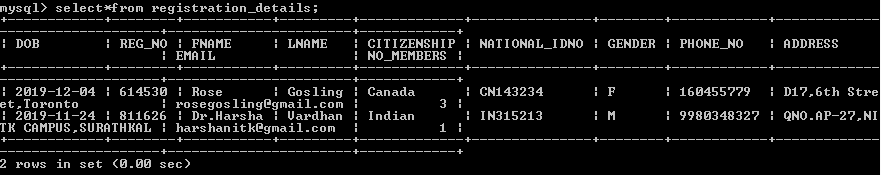




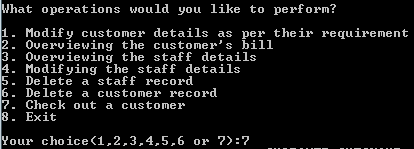
Modification in the Database



(After deletion)

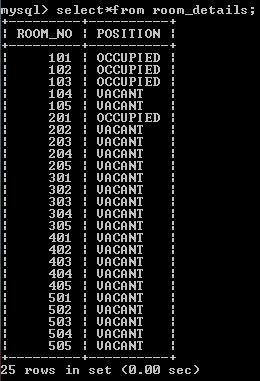


1. Customer Checkout

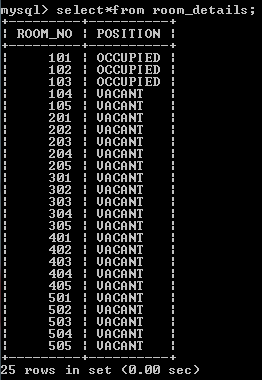




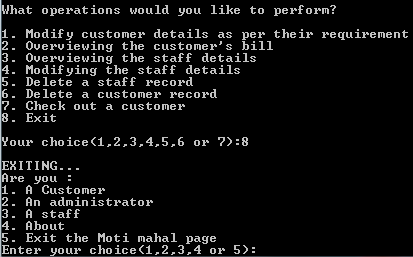
Modification in the Database



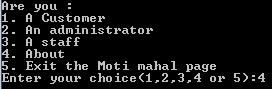
(After checkout)

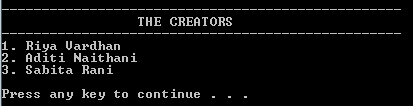


1. Exiting the Page and redirection to Main Menu

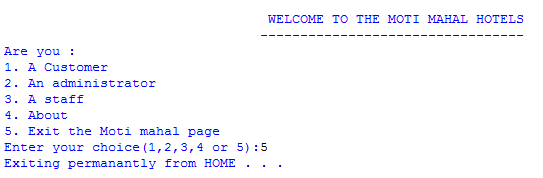


About the Creators





Exit from the Page permanently



**DRAWBACKS OF OUR PROJECT HOTEL MANAGEMENT SYSTEM:**

* Our project is only a humble venture to satisfy the needs to manage the records. Several user-friendly coding has also been adopted. We have removed the hardships that was being faced by the existing system.
* The proposed system has been made to reduce the time being spent on the existing system. The proposed system also helps in the detailed and smooth storage of the data and retrieval of the data.
* With this system, cab arranging facilities and regulation of other tourism facilities for the customers will not be possible. The details of the trip can be directly chosen from the customer’s selected package, but the tourism facilities cannot undergo any modifications after selection.
* The emergency facilities and services will not be made available online.

**FUTURE SCOPE OF THIS PROJECT**

With this proposed system, we can manage huge variation of data easily and thus helping us in carrying out the process of hotel management easily and effectively. Storage and modification of data can easily be done with this proposed system.

In future, the drawbacks of transport facilities arrangement and availability of emergency services and facilities online will be made possible. Also, more emphasis would be on making the code more efficient and compact for the future.

THANK

YOU