DELHI PUBLIC SCHOOL MATHURA ROAD

ACADEMIC YEAR : 2020-21

PROJECT REPORT ON

HOTEL MANAGEMENT SYSTEM

ROLL NO	**:**	
NAME	**:**	**MOHAMMAD DANISH**
CLASS	**:**	**XII**
SUBJECT	**:**	**COMPUTER SCIENCE**
SUB CODE	**:**	**083**

DELHI PUBLIC SCHOOL MATHURA ROAD

CERTIFICATE

This is to certify that MOHAMMAD DANISH CBSE Roll No:_____ has successfully completed the project Work entitled **EMPLOYEE MANAGEMENT SYSTEM** in the subject Computer Science (083) laid down in the regulations of CBSE for the purpose of Practical Examination in Class XII to be held in DELHI PUBLIC SCHOOL, MATHURA ROAD on_____.

(Monica Sahni)
HOD Comp Sci

TABLE OF CONTENTS [T O C]

ACKNOWLEDGEMENT

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

My sincere thanks to **Ms. Monica Sahni**, Teacher In-charge, A guide, Mentor all the above a friend, who critically reviewed my project and helped in solving each and every problem, occurred during implementation of the project

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

PROJECT ON HOTEL MANAGEMENT SYSTEM

INTRODUCTION

Hotel Management System deals with the maintenance of a guest's bill during one's stay at the hotel and withal the allocation of rooms for them. This software will be used mainly by the receptionist who will be the first staff member a guest conventionally sees on ingress and additionally the last one afore one leaves. The receptionist can utilize this software to allocate rooms to the newly arrived guests based on their budget requisites. The room number will be then engendered and given to the guest along with a unique customer ID. Any restaurant, laundry, recreational activity bills will be accounted by the receptionist of particular facility. The entire bill can then be paid by the guest at the time one wishes to depart from the hotel.

OBJECTIVES OF THE PROJECT

The objective of this project is to let the students apply the programming knowledge into a real-world situation/problem and exposed the students how programming skills helps in developing a good software.

- Ascertain that the software can run on any given platform.
- Facilely maintain the details of all the guests who have stayed at the hotel
- Make Reservation for each guest, if any room is available in hotel.
- Ascertain to have a user-amicable interface so that users are drawn into utilizing the software again.

PROPOSED SYSTEM:

HOTEL MANAGEMENT SYSTEM is a management system where in we schedule and reserve rooms for various guests. The Hotel has rooms and rooms have different seats. Only one room is reserved per guest and only if total seats required by guest matches the number of seats, the room is reserved to him/her.

The Structure of the Software is described as:

- When a new guest arrives, he is added to system and his profile is maintained for further processing.
- The guest can make a reservation on a particular date.
- The reservation is accepted by system only if the seats requested by guest can be served on that date, otherwise it is denied.
- If a guest demands reservation on any particular date and if the same guest has already any reservation on that day, then admij is notified about it and it is prompted that if him/her would change, delete or update reservation.
- The reservation is accepted on a particular date only if room is available on that date. Rooms may be occupied by other guests on same date and if no room is available on that day then request is denied

DIFFERENT PHASES OF PROJECT:

1. **ADD ROOM:** A new Room is added to the system. The room is identified by room number which serves as its unique key. The room has seats and total number are seats are also provided with this new room entry.
2. **ADD GUEST:** New Guest on arrival is added to the system. The Guest is identified uniquely by system generated key. Same named guests are allowed into the system, as uniqueness is guarenteed by key.
3. **ADD RESERVATION:** The reservation for a room is added using this phase of app. The guest can make reservation by providing a particular date and number of seats needed on that date. Admin then checks the availability of reservation and notifies the guest if reservation can be made.
4. **GUEST STATUS:** Admin can check the status of guest. The name and id related to particular guest can be found using this module.

5. **ROOM STATUS:** The details about room can be checked by using this module. Details include Room Number and total seats are available in this phase.
6. **EDIT GUEST:** Guest name can be edited in this module, however guest id remains same and continues to serve as unique identifier.
7. **EDIT ROOM:** Room details can be edited also, like room no, total seats. This module provides the functionality.
8. **DELETE GUEST:** Guest can be deleted at any by this functionality of the module. All reservation associated with this guest are deleted along with the deleted guest.
9. **DELETE ROOM:** Room can be dropped anytime and reservation including this room are also deleted. New rooms should be reserved for those reservations.
10. **RESERVATION STATUS:** Admin can anytime check the reservations made by guests. Using this functionality admin is aware of all the reservations in the database.

DATABASE STRUCTURE

The structure of database for the project is as:

1. **ROOM TABLE:** The table holds all the information about rooms. It has fields ‘room_no’ and ‘seats’. The room_no serves as unique identifier and seats are total seats available to this room.
2. **GUEST TABLE:** Holds information about guests. The guests are identified by system generated keys which implies that duplicate named guests can also be inserted into this table.
3. **RESERVATIONS TABLE:** This table holds information about all the reservations made. The table has four field i.e ‘c_id’,’date’,’room_no’ and ’seats’. The foreign keys ‘c_id’ refers to ‘guest_id’ of Guest Table and ‘room_no’ refers to ‘room_no’ of Room Table. The operation on foreign key is specified as on_delete =CASCADE, which ,means if we delete room or guest all their reservations get deleted automatically.

SOURCE CODES:

1. ADD ROOM:

```
import mysql.connector as connector


def addroom():
    conn = connector.connect(host="localhost", user="root",
                             password=          database="HOTEL")
    temp_cursor = conn.cursor()
    print("\n")
    print("-"*60)
    print("\n\n\tADD ROOM MENU ")
    print("-"*60)
    room_no = input("\n\tEnter Room no : ")
    seats = input("Total seats in above room : ")
    query = f"INSERT INTO ROOM (room_no,seats) VALUES({room_no},{seats})"
    temp_cursor.execute(query)
    conn.commit()
    print("\n\n\tRoom ADDED!")
    print("\n\n\t Returned to Main Menu")
    conn.close()
```

2. ADD GUEST:

```
import mysql.connector as connector
from prettytable import PrettyTable


def addcustomer():
    conn = connector.connect(host="localhost", user="root",
                             password=          database="HOTEL")
    print("\n")
    print("-"*60)
    print("\n\n\tADD CUSTOMER MENU ")
    print("-"*60)
    temp_cursor = conn.cursor()
    name = input("\n\tEnter Customer Name : ")
    query = "INSERT INTO CUSTOMER(c_name) VALUES('{}')".format(name)
    temp_cursor.execute(query)
    conn.commit()
    print("-"*60)
    print("\n\tCustomer ADDED!")
    print("\n\n\t Returned to Main Menu")
    conn.close()
```

3. ADD RESERVATION:

```
import mysql.connector as connector
import sys
from prettytable import PrettyTable


def addreservation():
    conn = connector.connect(host="localhost", user="root",
                             password=          , database="HOTEL")

    temp_cursor = conn.cursor()
    temp_cursor.execute("SELECT * from CUSTOMER")
    customers = []
    for i in temp_cursor:
        customers.append(i)
    print("\n")
    print("-"*60)
    print("\tRESERVATION MENU ")
    print("-"*60)
    table = PrettyTable()
    table.field_names = [' ', 'Names']
    print("\n\n\tSelect a Customer whose Reservation is to made!")
    j = 1
    for i in customers:
        table.add_row([f"Press {j} for", f" {i[0]}"])
        j += 1
    print(table)
    customer_id = int(input("\n\n\tPlease Select one Among above : "))
    try:
        c_id = customers[customer_id-1][1]
    except Exception:
        print("\n\n\tWrong Selection Returned!")
    # print(c_id)
    date = input("\n\n\tPlease enter date of Reservation format dd/mm/yyyy : ")
    # print(date)

    # CHECK IF ALREADY RESERVED OR NOT ON THIS DATE
    query = "SELECT * FROM RESERVATION WHERE date='{}' and c_id = {}".format(
        date, c_id)
    temp_cursor.execute(query)
    flag = 0
    update = 0
    for i in temp_cursor:
        flag = 1
        print(
            "\n\n\tRESERVATION ALREADY MADE on {} for {} seats".format(date, i[3]))
        print("\n\n\tWould u like to update Reservation? Press 1 else press 0: ")
        update = int(input("\n\t ->"))
        if update != 1:
            print("\n\nRESERVATION MAINTAINED")
            return
        break
    seats = int(input("\n\n\tSeats Required : "))
```

```
# SELECT A ROOM ON BEST FIT BASIS
rooms = []
temp_cursor.execute("SELECT * from ROOM ORDER BY seats asc")
for i in temp_cursor:
    # print(i)
    rooms.append(i)
room_no = ''
for i in rooms:
    query = "SELECT * FROM RESERVATION WHERE room_no = '{}' and date = '{}' LIMIT 1".format(
        i[0], date)
    temp_cursor.execute(query)
    row = temp_cursor.fetchone()
    if row is not None:
        continue
    if i[1] >= seats:
        room_no = i[0]
        break

# IF UPDATION IS NOT POSSIBLE:
if update == 1 and room_no == '':
    print("\n\n\tReservation can't be updated on {} for {} seats, Would You like to maintain previous r
    maintain = int(input('\n\n\t->'))
    if maintain != 1:
        query = "DELETE FROM RESERVATION where c_id = {} and date ='{}'".format(
            c_id, date)
        temp_cursor.execute(query)
        conn.commit()
        print("\n\n\tRESERVATION DELETED!")
        return
    else:
        print("\n\n\tPREVIOUS RESERVATION MAINTAINED!")
        return

# MAKE A RESERVATION
if room_no:
    #c_id, room_no, seats
    if update == 1:
        query = "UPDATE RESERVATION SET room_no = '{}' , seats = {} where c_id={} and date='{}'".format
            room_no, seats, c_id, date)
    else:
        query = "INSERT INTO RESERVATION(c_id,date,room_no,seats) VALUES({},'{}','{}',{})".format(
            c_id, date, room_no, seats)
    # print(query)
    temp_cursor.execute(query)
    conn.commit()
    if update == 1:
        print("\n\n\tReservation Updated for {} on {}".format(
            customers[customer_id-1][0], date))
    else:
        print("\n\n\tReservation made for {} on {}".format(
            customers[customer_id-1][0], date))
```

```
    # IF UPDATION IS NOT POSSIBLE:
    if update == 1 and room_no == '':
        print("\n\n\tReservation can't be updated on {} for {} seats, Would You like to maintain previous r
        maintain = int(input('\n\n\t->'))
        if maintain != 1:
            query = "DELETE FROM RESERVATION where c_id = {} and date ='{}'".format(
                c_id, date)
            temp_cursor.execute(query)
            conn.commit()
            print("\n\n\tRESERVATION DELETED!")
            return
        else:
            print("\n\n\tPREVIOUS RESERVATION MAINTAINED!")
            return

    # MAKE A RESERVATION
    if room_no:
        #c_id, room_no, seats
        if update == 1:
            query = "UPDATE RESERVATION SET room_no = '{}' , seats = {} where c_id={} and date='{}'".format
                room_no, seats, c_id, date)
        else:
            query = "INSERT INTO RESERVATION(c_id,date,room_no,seats) VALUES({},'{}','{}',{})".format(
                c_id, date, room_no, seats)
        # print(query)
        temp_cursor.execute(query)
        conn.commit()
        if update == 1:
            print("\n\n\tReservation Updated for {} on {}".format(
                customers[customer_id-1][0], date))
        else:
            print("\n\n\tReservation made for {} on {}".format(
                customers[customer_id-1][0], date))

    else:
        print("\n\n\tNo room available on this date, Please select a different date or different number of
    conn.close()


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

4. GUEST STATUS:

```
import mysql.connector as connector
from prettytable import PrettyTable


def customers():
    table = PrettyTable()
    print("\n")
    print("-"*60)
    print("\tCUSTOMERS STATUS ")
    print("-"*60)
    table.field_names = ['Name', 'ID']
    conn = connector.connect(host="localhost", user="root",
                             password=          database="HOTEL")
    temp_cursor = conn.cursor()
    query = "SELECT * from CUSTOMER"
    temp_cursor.execute(query)
    for i in temp_cursor:
        table.add_row(i)
    print(table)
    conn.close()
```

5. ROOM STATUS:

```
import mysql.connector as connector
from prettytable import PrettyTable


def rooms():
    table = PrettyTable()
    print("\n")
    print("-"*60)
    print("\tROOMS STATUS ")
    print("-"*60)
    table.field_names = ['Room No', 'Seats']
    conn = connector.connect(host="localhost", user="root",
                             password=          , database="HOTEL")
    temp_cursor = conn.cursor()
    query = "SELECT * from ROOM"
    temp_cursor.execute(query)
    for i in temp_cursor:
        table.add_row(i)
    print(table)
    conn.close()
```

6. EDIT GUEST:

```
def edit_customer():
    conn = connector.connect(host="localhost", user="root",
                             password=        , database="HOTEL")

    temp_cursor = conn.cursor()
    table = PrettyTable()
    table.field_names = ['  ', 'Customer Name']
    query = "SELECT * from CUSTOMER"
    temp_cursor.execute(query)
    print("\n")
    print("-"*60)
    print("\tEDIT CUSTOMER MENU ")
    print("-"*60)
    print("\n\n\tSELECT ROOMS : ")
    print("\n\n\tPlease Select the Customer Which You want To Edit : ")
    j = 1
    customers = []
    for i in temp_cursor:
        table.add_row([f"Press {j} ", f" {i[0]}"])
        j += 1
        customers.append(i)
    print(table)
    try:
        choice = int(input(
            "\n\n\tEnter the Number You Want to Edit of Any Other Key To Return to Main Menu: "))
        name = input("\n\n\tEnter New Name : ")
        query = "UPDATE CUSTOMER SET c_name = '{}' WHERE c_id = {}".format(
            name, customers[choice-1][1])
        # print(query)
        temp_cursor.execute(query)
        conn.commit()
        print("\n\n\tCustomer Updated SUCCEFULLY!")
    except Exception:
        print("\n\n\tReturned!")
```

7. EDIT ROOM

```
import mysql.connector as connector
from prettytable import PrettyTable


def edit_room():
    conn = connector.connect(host="localhost", user="root",
                             password=          database="HOTEL")

    temp_cursor = conn.cursor()
    table = PrettyTable()
    table.field_names = [' ', 'Room No', 'Seats']
    temp_cursor = conn.cursor()
    query = "SELECT * from ROOM"
    temp_cursor.execute(query)
    print("\n")
    print("-"*60)
    print("\tEDIT ROOM MENU ")
    print("-"*60)
    print("\n\n\tSELECT ROOMS : ")
    j = 1
    rooms = []
    print("\n\n\tPlease Select Room which you want to edit! : ")
    for i in temp_cursor:
        table.add_row([f"Press {j} ", f" {i[0]}", f"{i[1]}"])
        j += 1
        rooms.append(i)
    print(table)
    try:
        choice = int(
            input("\n\n\tEnter Room No to Update or Any Other Key to Return : "))
        # print(rooms[choice-1][0])
        room_no = input("\n\tEnter New Room No : ")
        seats = int(input("\n\tEnter Total Seats in Room :  "))
        query = "UPDATE ROOM SET seats = {},room_no = '{}' WHERE room_no = '{}'".format(
            seats, room_no, rooms[choice-1][0])
        # print(query)
        temp_cursor.execute(query)
        conn.commit()
        print("\n\n\tRoom Updated SUCCEFULLY!")
    except Exception:
        print("\n\n\tReturned!")
```

8. DELETE GUEST:

```
import mysql.connector as connector
from prettytable import PrettyTable


def deletecustomer():
    print("\n\tDELETE CUSTOMER MENU")
    table = PrettyTable()
    table.field_names = ['  ', 'Customer Name']
    conn = connector.connect(host="localhost", user="root",
                             password=[redacted] database="HOTEL")
    temp_cursor = conn.cursor()
    query = "SELECT * from CUSTOMER"
    temp_cursor.execute(query)
    print("\n\tSELECT CUSTOMER : ")
    customers = []
    j = 1
    for i in temp_cursor:
        table.add_row([f"Press {j} ", f" {i[0]}"])
        j += 1
        customers.append(i)
    print(table)
    # print(customers[choice-1][0])
    try:
        choice = int(input("Enter ID of Customer to DELETE : "))
        query = "DELETE FROM CUSTOMER WHERE c_id = {}".format(
            customers[choice-1][1])
        # print(query)
        temp_cursor.execute(query)
        conn.commit()
        print("\n\n\tCUSTOMER DELETED SUCCEFULLY!\n")
    except Exception:
        print("\n\n\tReturned!\n")
```

9. DELETE ROOM:

```
import mysql.connector as connector
from prettytable import PrettyTable


def deleteroom():
    print("\n\t DELETE ROOM MENU")
    table = PrettyTable()
    table.field_names = [' ', 'Room No']
    conn = connector.connect(host="localhost", user="root",
                             password=          database="HOTEL")
    temp_cursor = conn.cursor()
    query = "SELECT * from ROOM"
    temp_cursor.execute(query)
    print("\n\t SELECT ROOMS : ")
    j = 1
    rooms = []
    for i in temp_cursor:
        table.add_row([f"Press {j} ", f" {i[0]}"])
        j += 1
        rooms.append(i)
    print(table)
    try:
        choice = int(
            input("Enter Room No. to delete Or Any Key to return  : "))
        # print(rooms[choice-1][0])
        query = "DELETE FROM ROOM WHERE room_no = '{}'".format(
            rooms[choice-1][0])
        # print(query)
        temp_cursor.execute(query)
        conn.commit()
        print("\n\n\tRoom DELETED SUCCEFULLY!\n")
    except Exception:
        print("\n\n\tReturned!\n")
```

10. RESERVATION STATUS:

```
import mysql.connector as connector
from prettytable import PrettyTable


def reservationstatus():
    table = PrettyTable()
    table.field_names = ['Name',
                         'Date of Reservation', 'Room Alloted', 'Seats']
    conn = connector.connect(host="localhost", user="root",
                             password=          database="HOTEL")
    temp_cursor = conn.cursor()
    query = "SELECT * from RESERVATION"
    temp_cursor.execute(query)
    reservations = []
    for i in temp_cursor:
        reservations.append(i)
    j = 0
    for i in reservations:
        query = "SELECT c_name FROM CUSTOMER WHERE c_id ={} LIMIT 1".format(
            i[0])
        temp_cursor.execute(query)
        name = temp_cursor.fetchone()
        reservations[j] = [name[0]]+list(i[1:])
        j += 1
    table.add_rows(reservations)
    print(table)
    conn.close()
```

OUTPUT:

1. ADD ROOM:

```
************************************************************


------------------------------------------------------------


        ADD ROOM MENU
------------------------------------------------------------

        Enter Room no : 106
Total seats in above room : 3


        Room ADDED!


         Returned to Main Menu
************************************************************



         Press 'C' to continue! Any other key to Exit
------------------------------------------------------------
```

2. ADD GUEST

```
------------------------------------------------------------


        ADD CUSTOMER MENU
------------------------------------------------------------

        Enter Customer Name : DANISH
------------------------------------------------------------

        Customer ADDED!


         Returned to Main Menu
************************************************************



         Press 'C' to continue! Any other key to Exit
------------------------------------------------------------
```

3. ADD RESERVATION:

```
************************************************************


------------------------------------------------------------
        RESERVATION MENU
------------------------------------------------------------


        Select a Customer whose Reservation is to made!
+-------------+---------+
|             |  Names  |
+-------------+---------+
| Press 1 for |   ZAKIR |
| Press 2 for |  SUHAIL |
| Press 3 for |  ZUBAIR |
| Press 4 for |  ISHFAQ |
| Press 5 for |  ZUBAIR |
| Press 6 for |   SAKIB |
| Press 7 for |  ISHFAQ |
| Press 8 for |  Danish |
| Press 9 for |  DANISH |
+-------------+---------+


        Please Select one Among above : 9


        Please enter date of Reservation format dd/mm/yyyy : 12/05/2021


        Seats Required : 2


        Reservation made for DANISH on 12/05/2021
************************************************************


         Press 'C' to continue! Any other key to Exit
------------------------------------------------------------
```

4. GUEST STATUS:

```
************************************************************


------------------------------------------------------------
        CUSTOMERS STATUS
------------------------------------------------------------
+--------+----+
|  Name  | ID |
+--------+----+
| ZAKIR  | 4  |
| SUHAIL | 6  |
| ZUBAIR | 7  |
| ISHFAQ | 8  |
| ZUBAIR | 9  |
| SAKIB  | 10 |
| ISHFAQ | 11 |
| Danish | 12 |
| DANISH | 13 |
+--------+----+
************************************************************


         Press 'C' to continue! Any other key to Exit
------------------------------------------------------------
```

5. ROOM STATUS

```
************************************************************


------------------------------------------------------------
        ROOMS STATUS
------------------------------------------------------------
+---------+-------+
| Room No | Seats |
+---------+-------+
|   104   |   1   |
|   105   |   4   |
+---------+-------+
************************************************************


         Press 'C' to continue! Any other key to Exit
------------------------------------------------------------
```

6. DELETE GUEST

```
************************************************************

        DELETE CUSTOMER MENU

        SELECT CUSTOMER :
+----------+---------------+
|          | Customer Name |
+----------+---------------+
| Press 1  |      ZAKIR    |
| Press 2  |     SUHAIL    |
| Press 3  |     ZUBAIR    |
| Press 4  |     ISHFAQ    |
| Press 5  |     ZUBAIR    |
| Press 6  |      SAKIB    |
| Press 7  |     ISHFAQ    |
| Press 8  |     Danish    |
| Press 9  |     DANISH    |
+----------+---------------+
Enter ID of Customer to DELETE : 8


        CUSTOMER DELETED SUCCEFULLY!

************************************************************


         Press 'C' to continue! Any other key to Exit
------------------------------------------------------------
```

```
 ------------------------------------------------------------
         CUSTOMERS STATUS
 ------------------------------------------------------------
 +--------------+----+
 |     Name     | ID |
 +--------------+----+
 | ZUBAIR AKBAR | 4  |
 |    SUHAIL    | 6  |
 |    ZUBAIR    | 7  |
 |    ISHFAQ    | 8  |
 |    ZUBAIR    | 9  |
 |    SAKIB     | 10 |
 |    ISHFAQ    | 11 |
 |    DANISH    | 13 |
 +--------------+----+
 ************************************************************


          Press 'C' to continue! Any other key to Exit
 ------------------------------------------------------------
```

7. DELET ROOM:

```
************************************************************

         DELETE ROOM MENU

         SELECT ROOMS :
+----------+---------+
|          | Room No |
+----------+---------+
| Press 1  |    104  |
| Press 2  |    105  |
| Press 3  |    106  |
+----------+---------+
Enter Room No. to delete Or Any Key to return  : 3


        Room DELETED SUCCEFULLY!

************************************************************



         Press 'C' to continue! Any other key to Exit
------------------------------------------------------------
```

```
------------------------------------------------------------
         ROOMS STATUS
------------------------------------------------------------
+---------+-------+
| Room No | Seats |
+---------+-------+
|   104   |   1   |
|   105   |   4   |
+---------+-------+
************************************************************



          Press 'C' to continue! Any other key to Exit
------------------------------------------------------------
```

8. EDIT GUEST:

```
**************************************************************

--------------------------------------------------------------
        EDIT CUSTOMER MENU
--------------------------------------------------------------


        SELECT ROOMS :


        Please Select the Customer Which You want To Edit :
+----------+---------------+
|          | Customer Name |
+----------+---------------+
| Press 1  |      ZAKIR    |
| Press 2  |     SUHAIL    |
| Press 3  |     ZUBAIR    |
| Press 4  |     ISHFAQ    |
| Press 5  |     ZUBAIR    |
| Press 6  |      SAKIB    |
| Press 7  |     ISHFAQ    |
| Press 8  |     DANISH    |
+----------+---------------+


        Enter the Number You Want to Edit of Any Other Key To Return to Main Menu: 1


        Enter New Name : ZUBAIR AKBAR


        Customer Updated SUCCEFULLY!
**************************************************************



         Press 'C' to continue! Any other key to Exit
--------------------------------------------------------------
```

```
+----------+---------------+
|          | Customer Name |
+----------+---------------+
| Press 1  |  ZUBAIR AKBAR |
| Press 2  |     SUHAIL    |
| Press 3  |     ZUBAIR    |
| Press 4  |     ISHFAQ    |
| Press 5  |     ZUBAIR    |
| Press 6  |      SAKIB    |
| Press 7  |     ISHFAQ    |
| Press 8  |     DANISH    |
+----------+---------------+


        Enter the Number You Want to Edit of Any Other Key To Return to Main Menu:
```

9. EDIT ROOM:

```
------------------------------------------------------------


        SELECT ROOMS :


        Please Select Room which you want to edit! :
+----------+---------+-------+
|          | Room No | Seats |
+----------+---------+-------+
| Press 1  |    104  |   1   |
| Press 2  |    105  |   5   |
+----------+---------+-------+


        Enter Room No to Update or Any Other Key to Return : 2

        Enter Total Seats in Room :  4


        Room Updated SUCCEFULLY!
************************************************************


************************************************************


------------------------------------------------------------
        EDIT ROOM MENU
------------------------------------------------------------


        SELECT ROOMS :


        Please Select Room which you want to edit! :
+----------+---------+-------+
|          | Room No | Seats |
+----------+---------+-------+
| Press 1  |    104  |   1   |
| Press 2  |    105  |   4   |
+----------+---------+-------+


        Enter Room No to Update or Any Other Key to Return :
```

10. RESERVATION STATUS:

```
************************************************************
+--------+--------------------+--------------+-------+
|  Name  | Date of Reservation | Room Alloted | Seats |
+--------+--------------------+--------------+-------+
| ZUBAIR |      12/02/2021     |     105      |   4   |
| ISHFAQ |      03/03/2021     |     105      |   4   |
| DANISH |      12/05/2021     |     105      |   2   |
+--------+--------------------+--------------+-------+
************************************************************


         Press 'C' to continue! Any other key to Exit
------------------------------------------------------------
```

BIBLIOGRAPHY

1. ***Computer science With Python - Class XII By : SumitaArora***
2. ***Website:*** **https://www.youtube.com**

***