COMPUTER SCIENCE

TOPIC OF PROJECT-:
HOTEL MANAGEMENT SYSTEM

GROUP MEMBERS-

1) MALYA SHUKLA (XII-A)
2) CHAHAL CHANDIOK (XII-F)
3) ADWITYA CHAKARBORTHY (XII-A)

CLASS: XII

SCHOOL: Modern Vidya Niketan sector-43 Faridabad

CERTIFICATE-

THIS IS TO CERTIFY THAT THE COMPUTER SCIENCE PROJECT ON "**HOTEL MANAGEMENT**" HAS BEEN SUCCESSFULLY COMPLETED BY **MALYA SHUKLA, CHAHAL CHANDIOK AND ADWITYA CHAKARBORTHY** OF CLASS XII UNDER THE GUIDANCE OF **Mrs. MANDEEP WALIA** DURING THE ACADEMIC YEAR 2021-22.

_____ _____

INTERNAL EXAMINER PRINICIPAL

_____ _____

DATE EXTERNAL EXAMINER

ACKNOWLEGMENT

I would like to express my special thanks of gratitude to my teacher **Mrs. Mandeep Walia** as well as my principal ma'am who gave us golden opportunity to do this project of Computer Science, which also helped me in doing a lot of research and I came to know new things about it. Without their help, guidance and support it would be impossible to complete this project.

Secondly, I would also like to thank my parents and friends who helped me a lot in finishing this project within limited time.

I am making this project not only for marks but also to increase my knowledge.

Once again thanks to all who helped me in doing this project

INDEX

INTRODUCTION TO PROJECT-

Project title “HOTEL MANAGEMENT” (a project for keeping customers record and also calculate customer’s bill slips and manager’s salary).

The name of project is “HOTEL MANAGEMENT”. The objective of the project is to computerize the system of hotel. This project not only keeps the record of various people like customers, manager etc. but also reduces the extensive paperwork in the present system. It will make the system more versatile and user friendly. It also calculates the proper billing slip of high level and middle-class customers. This project is based on description about the structure of HOTEL MANAGEMENT SYSTEM.

This project contains: -

- Keeping the record of all customers, managers etc.
- Maintains proper list of all people.
- Generating proper bill slip
- Gaming department
- Restaurant bill

OBJECTIVE OF THE PROJECT –

The project "**HOTEL MANAGEMENT SYSTEM**" is aimed to develop to maintain the day-to-day state of admission/vacation of residents, List of workers, list of bills etc.

The main objective of the hotel: -

- ✓ **Records of salary structure of the employees of hotel by billing approach. Keeping records of admission of resident.**
- ✓ **Keeping user satisfaction as utmost priority.**
- ✓ **Scheduling the allotment of user with room to make it convenient for user.**
- ✓ **Keeping records of user registration details accurately arranged order so that the treatment of Customers becomes quick and satisfactory.**
- ✓ **Keeping details about the users, their needs and payment detail report etc.**
- ✓ **Keeping the best hotel facilities.**
- ✓ **Assuring Safe and Healthy Environment.**

FLOW OF PROJECT -

Our project is based on Hotel Management. The project consists of 6 modules-

1. ROOM ENQUIRY AND RESERVATION
2. CUSTOMER IDENTIFICATION
3. CHECK IN/CHECK OUT
4. RESTAURANT
5. BILLS AND PAYMENT MODULE
6. GAMING
7. REPORTS

ROOM ENQUIRY AND RESERVATION MODULE- This module deals with the **reservation enquiry**. During reservation, **the details of customers, type of room required, and number of rooms required** are fed into the system. Once these information's are entered**, the system searches for unoccupied rooms and display the result**. In reservation enquiry, customer can get the information such as rent of rooms and details of rooms available.

CUSTOMER IDENTIFICATION- This module deals with the customer details. Name, age, address, contact details and identity certificates that include legal documents like Aadhar card, passport, driver's license. These details are essential when checking in an hotel, to prevent **frauds** or **identity misuse**. Also people checking in, are above a certain age to do so.

CHECK IN/CHECK OUT MODULE- This module deals with the **reservation check-in** (i.e. allocating rooms for the resident who has already reserved) and **Direct check-in** (i.e. checking in without reservation), handling **checkout** of guest and bill payment.

RESTAURANT- This module deals with the availability of inventory. Options available in the **restaurant** (buffet system, veg or non-veg). This makes the customer familiar with what all is available that he/she can have access to.

BILLS AND PAYMENT MODULE- This module deals with the generation and tracking of bills and payments made by the guest. The bills are

classified into **lodging bills** and **Restaurant Bill**. The Lodging bill is calculated using the **check- in and check-out** details of the person. **The Restaurant Bill** is generated based on the food items consumed by the guest during the stay in the hotel.

GAMING- This module deals with the available gaming services in the hotel, which include various sports like **badminton, basketball, and tennis**. These sports become a necessity when spending time with families and friends. Though everything has to be accessed through a certain payment.

REPORT-This module deals with the **generation of the reports** for various modules. The customer list can be generated. **Room status list can be retrieved for reference**. **The check-in and check-out registers** can also be retrieved for any future queries.

FUNCTIONS AND MODULES-

Implementation module-

1. ADMIN
2. RESIDENTS (USER)

ADMIN MODULE: This module provides administrator related functionalities. Administrator can view the registered user and payment model. In this module admin can upload the information about the hotels such as hotel facilities, payment details, residents' allotment details and allotment status. Admin will give the responses to users based on requirements always checking the hotel strength and availability of rooms, and salary payment for workers. The main goal of the hotel is to provide security maintenance.

USER MODULE: This module is about users of this portal. By using this module user can lodge any complaint about process. The user must be registered with the system. By using this login id, user will log on to this portal and do all transactions which are assigned to them. User registration details must be kept secret. After that he/she will get personal ID, password and through that contact only the admin user can send a room allotment request to the admin.

Recommended System Requirements

Processors: Intel® CoreTM i3 processor 4300M at 2.60 GHz.

Disk space: 2 to 4 GB.

Operating systems: Windows® 10, and UBUNTU.

Python Versions: 3.8 or Higher.

Minimum System Requirements

Processors: Intel Atom® processor or Intel® CoreTM i3 processor.

Disk space: 1 GB.

Operating systems: Windows 10 or later, and UBUNTU.

Python Versions: 2.7.X, 3.6.X.

Prerequisites before installing MySQL Connector Python

You need root or administrator privileges to perform the installation process.

Python must be installed on your machine.

Note: – MySQL Connector Python requires python to be in the system's PATH. Installation fails if it doesn't find Python.

On Windows, If Python doesn't exist in the system's PATH, please manually add the directory containing python.exe yourself.

HARDWARE SYSTEM CONFIGURATION-

Speed -1.1GHz

RAM - 256MB (min)

Hard Disk -20 GB

Floppy Drive 1.44 MB

Keyboard - Standard window Keyboard

Mouse- Two or Three Button Mouse

Monitor – SVGA

BIBLIOGRAPHY-

- The hotel management system
- studymafia.org
- freeprojectz.com
- Geekforgeeks.com
- Programiz.com
- Tutorialspoint.com

CONCLUSION-

The conclusion of this project is that Hotel management system is a computerized management system. This system keeps the records of hardware assets besides software of this organization. The proposed system will keep a track of Workers, Residents', Accounts, and generation of report regarding the present status. This project has GUI based software that will help in storing, updating, and retrieving the information through various user-friendly menu-driven modules. The project "Hotel Management System" is aimed to develop the day-to-day state of admission/Vacation of residents, payment details etc. Main objective of this project is to provide solution for hotel to manage, because most of there work is using computerized process. This software application will help admin to handle customer's information, room allocation details, payment details, billing information, etc. Detailed explanation about modules and design are provided in project documentation. The existing system is a manually maintained system. All the Hotel records are to be maintained for the details of each customer, Fee details, Room Allocation, Attendance etc. All these details are entered and retrieved manually, because of this there are many disadvantages like Time Consuming, updating process, inaccuracy of data. For avoiding this we introduced or proposed a new system in the computerized version of the existing system. Provides easy and quick access over the data.

CODING-

```
import mysql.connector
myConnection = mysql.connector.connect(host="localhost", user="root", passwd="sql")
cur = myConnection.cursor()


# TO CREATE TABLES IN THE DATABASE
def Database_Creation():
    try:
        my_query = "create table if not exists cust_details(\
        CID int primary key not null AUTO_INCREMENT,\
        C_NAME varchar(30) not null,\
        C_ADDRESS varchar(30),\
        C_COUNTRY varchar(30),\
        C_CONTACT char(10))"
        cur.execute(my_query)

        my_query = "create table if not exists Booking_Rec(\
        CID int primary key,\
        CHECK_IN_DATE date,\
        CHECK_OUT_DATE date)"
        cur.execute(my_query)

        my_query = "create table if not exists Room_Rent(\
        CID int primary key,\
        RoomChoice int,\
        NoOfRooms int,\
        Days int,\
        RoomNo text,\
        RoomRent int)"
        cur.execute(my_query)

        my_query = "create table if not exists Restaurant(\
        CID int primary key not null,\
        Cuisine int(1),\
        Quantity int,\
        Bill int)"
        cur.execute(my_query)

        my_query = "create table if not exists Gaming(\
        CID int primary key not null,\
        Games varchar(30),\
        Hours varchar(30),\
        Gaming_Bill int)"
        cur.execute(my_query)

        my_query = "create table if not exists Total_Bill(\
        CID int primary key,\
        C_NAME varchar(30),\
        grand_total int,\
        RoomRent int,\
        RestaurantBill int,\
        GamingBill int)"
        cur.execute(my_query)

    except Exception as e:
        print(e)
```

```
def User_Entry():
    C_NAME = input("Enter Customer Name: ").upper()
    C_ADDRESS = input("Enter Customer Address: ").upper()
    C_COUNTRY = input("Enter Customer Country: ").upper()
    C_CONTACT = int(input("Enter Customer Contact Number: "))
    cur.execute("insert into cust_details (C_NAME, C_ADDRESS, C_COUNTRY, C_CONTACT) values('{}','{}','{}',{})".
                format(C_NAME, C_ADDRESS, C_COUNTRY, C_CONTACT))
    myConnection.commit()
    CID=cur.lastrowid
    Booking_Rec(CID)
    Room_Rent(CID)
    cur.execute("insert into RESTAURANT values({},'{}',{},{})".format(CID, 0, 0, 0))
    cur.execute("insert into Gaming values({},{},{},{})".format(CID, 0, 0, 0))
    myConnection.commit()


def Booking_Rec(CID):
    checkin = input("\nEnter Customer Check-IN date[YYYY-MM-DD]: ")
    checkout = input("Enter Customer Check-OUT date[YYYY-MM-DD]: ")
    cur.execute("insert into Booking_Rec values({}, '{}','{}')".format(CID, checkin, checkout))
    myConnection.commit()
    print("CHECK-IN and CHECK-OUT entry added successfully")

def Room_Rent(CID):
    print("\n ##### We have The Following Rooms For You #####")
    print("1.UltraRoyal------>10000Rs.")
    print("2.Royal ---------->5000Rs.")
    print("3.Elite ---------->3500Rs.")
    print("4.Budget---------->2500Rs.")
    RoomChoice = int(input("Enter the type of room you want: "))
    Noofrooms = int(input("Enter the number of rooms you want: "))
    RoomNo = input("Enter Customer's Room no: ")
    cur.execute('select DATEDIFF(CHECK_OUT_DATE, CHECK_IN_DATE) from booking_rec where CID="{}"'.format(CID))
    Days = cur.fetchall()[0][0]
    if RoomChoice == 1:
        Roomrent = Days * 10000 * Noofrooms
        print("Ultra Royal RoomRent:", Roomrent)
    elif RoomChoice == 2:
        Roomrent = Days * 5000 * Noofrooms
        print("Royal Room Rent:", Roomrent)
    elif RoomChoice == 3:
        Roomrent = Days * 3500 * Noofrooms
        print("Elite Room Rent:", Roomrent)
    elif RoomChoice == 4:
        Roomrent = Days * 2500 * Noofrooms
        print("Budget Room Rent:", Roomrent)
    else:
        print("Sorry, You are giving wrong inputs,Try Again!!")
        return
    cur.execute("insert into Room_Rent values({}, {}, {}, {}, '{}', {})".
                format(CID, RoomChoice, Noofrooms, Days, RoomNo, int(Roomrent)))
    myConnection.commit()
    print("Thank You, Your", Noofrooms, "Room(s) Has Been Booked For:", Days, "Days")
    print("Your Total Room Rent is: Rs.", Roomrent)
```

```
def Restaurant():
    customer = search_cust()
    if customer[0]:
        print("1.Vegetarian Combo ------------------------->300Rs.")
        print("2.Non-Vegetarian Combo --------------------->500Rs.")
        print("3.Vegetarian and Non-Vegetarian Combo ------>750Rs.")
        print("4.Exit")
        ch_dish = int(input("Enter Your Choice Of Cusine:"))
        q = int(input("Enter Quantity:"))
        if ch_dish == 1:
            print("YOU HAVE ORDERED VEGETARIAN COMBO")
            RestaurantBill = q * 300
        elif ch_dish == 2:
            print("YOU HAVE ORDERED NON VEGETARIAN COMBO")
            RestaurantBill = q * 500
        elif ch_dish == 3:
            print("YOU HAVE ORDERED VEGETARIAN AND NON VEGETARIAN BOTH")
            RestaurantBill = q * 800
        elif ch_dish == 4:
            return
        else:
            print("Sorry,You are giving wrong inputs, try again!!")
            return
        cur.execute("update restaurant set Cuisine={}, Quantity={}, Bill={} where CID={}".
                    format(ch_dish, q, int(RestaurantBill), customer[1][0]))
        myConnection.commit()
        print("Your total Bill Amount is: Rs.", RestaurantBill)
        print("\n****WE HOPE YOU WILL ENJOY YOUR MEAL****")
    else:
        print("\nSorry, No Such Customer Found")

def Gaming():
    customer = search_cust()
    if customer[0]:
        print("1.Table Tennis ------------->150Rs./HR")
        print("2.Bowling ------------------->100Rs./HR")
        print("3.Snooker ------------------->250Rs./HR")
        print("4.VR World Gaming ---------->400Rs./HR")
        print("5.Video Games -------------->300Rs./HR")
        print("6.Swimming Pool Games ------>350Rs./HR")
        print("7.Exit")
        Games = int(input("Enter what game you want to play: "))
        Hours = float(input("Enter No. of Hours you want to play: "))
        if Games == 1:
            print("YOU HAVE SELECTED TABLE TENNIS")
            GamingBill = Hours * 150
        elif Games == 2:
            print("YOU HAVE SELECTED BOWLING")
            GamingBill = Hours * 100
        elif Games == 3:
            print("YOU HAVE SELECTED SNOOKER")
            GamingBill = Hours * 250
        elif Games == 4:
            print("YOU HAVE SELECTED VR WORLD GAMING")
            GamingBill = Hours * 400
        elif Games == 5:
            print("YOU HAVE SELECTED VIDEO GAMES")
            GamingBill = Hours * 300
        elif Games == 6:
            print("YOU HAVE SELECTED SWIMMING POOL GAMES")
            GamingBill = Hours * 350
        elif Games == 7:
            return
        else:
            print("Sorry wrong Input")
            return
        cur.execute("update gaming set Games={}, Hours={}, Gaming_Bill={} where CID={}".
                    format(Games, Hours, int(GamingBill) + 1, customer[1][0]))
        myConnection.commit()
        print("YOUR TOTAL GAMING BILL IS: Rs.", int(GamingBill) + 1, "For:", Hours, "Hours")
        print("\n ***** WE HOPE YOU ENJOY YOUR GAME *****")

    else:
        print("\nSorry, No Such Customer Found")
```

```
def Amount():
    customer = search_cust()
    if customer[0]:
        C_NAME = customer[1][1]
        cur.execute('select RoomRent from room_rent where CID={}'.format(customer[1][0]))
        RoomRent = cur.fetchone()[0]
        cur.execute('select Bill from restaurant where CID={}'.format(customer[1][0]))
        RestaurantBill = cur.fetchone()[0]
        cur.execute('select Gaming_Bill from gaming where CID={}'.format(customer[1][0]))
        GamingBill = cur.fetchone()[0]
        grandTotal = RoomRent + RestaurantBill + GamingBill
        cur.execute("insert into total_bill values({},'{}',{},{},{},{})".format(customer[1][0], C_NAME, grandTotal, RoomRent, RestaurantBill, GamingBill))
        myConnection.commit()
        print("\n ***** VENTIAN HOTEL ***** CUSTOMER BILLING*******")
        print("CUSTOMER NAME:", C_NAME)
        print("ROOMRENT: Rs.", RoomRent)
        print("RESTAURANT BILL: Rs.", RestaurantBill)
        print("GAMING BILL: Rs.", GamingBill)
        print("_____")
        print("TOTAL AMOUNT:Rs.", grandTotal)
    else:
        print("Sorry, No Such Customer Found")
    phone = int(input("Enter Customer's Phone No.: "))
    cur.execute("select * from Cust_Details where C_CONTACT={}".format(phone))
    rl = cur.fetchall()
    if rl:
        if display:
            print(rl[0])
        return True, rl[0]
    else:
        if display:
            print("Record not found,Try Again!")
        return False, rl

cur.execute("create database if not exists HOTEL")
cur.execute("use HOTEL")
Database_Creation()
while True:
    print("\n\n_____WELCOME TO VENETIAN HOTEL MANAGEMENT_____")
    print("_____ATITHI DEVO BHAVA_____")
    print("1---->NEW CUSTOMER")
    print("2---->RESTAURANT BILL")
    print("3---->GAMING BILL")
    print("4---->DISPLAY CUSTOMER DETAILS")
    print("5---->GENERATE TOTAL BILL")
    print("6---->EXIT")
    ch = input("Enter you choice: ")
    if ch == '1':
        User_Entry()
    elif ch == '2':
        Restaurant()
    elif ch == '3':
        Gaming()
    elif ch == '4':
        search_cust(True)
    elif ch == '5':
        Amount()
    elif ch == '6':
        break
    else:
        print("SORRY WRONG INPUT, PLEASE TRY AGAIN")
```

OUTPUT-

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Lenovo\Downloads\Hotel Management (2).py ==========


_____WELCOME TO VENETIAN HOTEL MANAGEMENT_____
_____ATITHI DEVO BHAVA_____
1---->NEW CUSTOMER
2---->RESTAURANT BILL
3---->GAMING BILL
4---->DISPLAY CUSTOMER DETAILS
5---->GENERATE TOTAL BILL
6---->EXIT
Enter you choice: 1
Enter Customer Name: Ayush Sharma
Enter Customer Address: 761 LODHI ROAD
Enter Customer Country: INDIA
Enter Customer Contact Number: 9876789665

Enter Customer Check-IN date[YYYY-MM-DD]: 2021-01-11
Enter Customer Check-OUT date[YYYY-MM-DD]: 2021-01-17
CHECK-IN and CHECK-OUT entry added successfully

 ##### We have The Following Rooms For You #####
1.UltraRoyal------>10000Rs.
2.Royal ---------->5000Rs.
3.Elite ---------->3500Rs.
4.Budget---------->2500Rs.
Enter the type of room you want: 1
Enter the number of rooms you want: 2
Enter Customer's Room no: 213
Ultra Royal RoomRent: 120000
Thank You, Your 2 Room(s) Has Been Booked For: 6 Days
Your Total Room Rent is: Rs. 120000
```

```
_____WELCOME TO VENETIAN HOTEL MANAGEMENT_____
_____ATITHI DEVO BHAVA_____
1---->NEW CUSTOMER
2---->RESTAURANT BILL
3---->GAMING BILL
4---->DISPLAY CUSTOMER DETAILS
5---->GENERATE TOTAL BILL
6---->EXIT
Enter you choice: 2
Enter Customer's Phone No.: 9876789665
1.Vegetarian Combo ------------------------>300Rs.
2.Non-Vegetarian Combo --------------------->500Rs.
3.Vegetarian and Non-Vegetarian Combo ------>750Rs.
4.Exit
Enter Your Choice Of Cusine:3
Enter Quantity:2
YOU HAVE ORDERED VEGETARIAN AND NON VEGETARIAN BOTH
Your total Bill Amount is: Rs. 1600

****WE HOPE YOU WILL ENJOY YOUR MEAL****
```

```
_____WELCOME TO VENETIAN HOTEL MANAGEMENT_____
_____ATITHI DEVO BHAVA_____
1---->NEW CUSTOMER
2---->RESTAURANT BILL
3---->GAMING BILL
4---->DISPLAY CUSTOMER DETAILS
5---->GENERATE TOTAL BILL
6---->EXIT
Enter you choice: 3
Enter Customer's Phone No.: 9876789665
1.Table Tennis ------------->150Rs./HR
2.Bowling ------------------>100Rs./HR
3.Snooker ------------------>250Rs./HR
4.VR World Gaming ---------->400Rs./HR
5.Video Games -------------->300Rs./HR
6.Swimming Pool Games ------>350Rs./HR
7.Exit
Enter what game you want to play: 4
Enter No. of Hours you want to play: 2
YOU HAVE SELECTED VR WORLD GAMING
YOUR TOTAL GAMING BILL IS: Rs. 801 For: 2.0 Hours

 ***** WE HOPE YOU ENJOY YOUR GAME ****


_____WELCOME TO VENETIAN HOTEL MANAGEMENT_____
_____ATITHI DEVO BHAVA_____
1---->NEW CUSTOMER
2---->RESTAURANT BILL
3---->GAMING BILL
4---->DISPLAY CUSTOMER DETAILS
5---->GENERATE TOTAL BILL
6---->EXIT
Enter you choice: 4
Enter Customer's Phone No.: 9876789665
(1, 'AYUSH SHARMA', '761 LODHI ROAD', 'INDIA', '9876789665')


_____WELCOME TO VENETIAN HOTEL MANAGEMENT_____
_____ATITHI DEVO BHAVA_____
1---->NEW CUSTOMER
2---->RESTAURANT BILL
3---->GAMING BILL
4---->DISPLAY CUSTOMER DETAILS
5---->GENERATE TOTAL BILL
6---->EXIT
Enter you choice: 5
Enter Customer's Phone No.: 9876789665
801

 ***** VENTIAN HOTEL ***** CUSTOMER BILLING*******
CUSTOMER NAME: AYUSH SHARMA
ROOMRENT: Rs. 120000
RESTAURANT BILL: Rs. 1600
GAMING BILL: Rs. 801
_____
TOTAL AMOUNT:Rs. 122401


_____WELCOME TO VENETIAN HOTEL MANAGEMENT_____
_____ATITHI DEVO BHAVA_____
1---->NEW CUSTOMER
2---->RESTAURANT BILL
3---->GAMING BILL
4---->DISPLAY CUSTOMER DETAILS
5---->GENERATE TOTAL BILL
6---->EXIT
Enter you choice: 6
>>>
```

MYSQL –

1) CUSTOMER DETAILS-

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 73
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
mysql> use Hotel;
Database changed
mysql> show tables;
+-----------------+
| Tables_in_hotel |
+-----------------+
| booking_rec     |
| cust_details    |
| gaming          |
| restaurant      |
| room_rent       |
| total_bill      |
+-----------------+
6 rows in set (0.25 sec)

mysql> select * from cust_details;
+-----+---------------+-------------------------+-----------+------------+
| CID | C_NAME        | C_ADDRESS               | C_COUNTRY | C_CONTACT  |
+-----+---------------+-------------------------+-----------+------------+
|   1 | AYUSH SHARMA  | 761 LODHI ROAD          | INDIA     | 9876789665 |
|   2 | ADITYA SINGH  | 432 STREET WON          | INDIA     | 9876543563 |
|   3 | RIYA JHA      | 870 MG ROAD             | INDIA     | 8765457686 |
|   4 | BELLY COOPER  | 21 LAKE ROAD WASHINGTON | USA       | 7432123453 |
+-----+---------------+-------------------------+-----------+------------+
4 rows in set (0.00 sec)
```

2) BOOKING RECORD-

```
mysql> select * from booking_rec;
+-----+---------------+----------------+
| CID | CHECK_IN_DATE | CHECK_OUT_DATE |
+-----+---------------+----------------+
|   1 | 2021-01-11    | 2021-01-17     |
|   2 | 2022-02-14    | 2022-02-20     |
|   3 | 2021-03-02    | 2021-03-05     |
|   4 | 2022-03-16    | 2022-03-20     |
+-----+---------------+----------------+
4 rows in set (0.01 sec)
```

3) GAMING BILL-

```
mysql> select * from gaming;
+-----+-------+-------+-------------+
| CID | Games | Hours | Gaming_Bill |
+-----+-------+-------+-------------+
|   1 | 4     | 2.0   |         801 |
|   2 | 1     | 1.0   |         151 |
|   3 | 0     | 0     |           0 |
|   4 | 0     | 0     |           0 |
+-----+-------+-------+-------------+
4 rows in set (0.00 sec)
```

4) RESTAURANT BILL-

```
mysql> select * from restaurant;
+-----+---------+----------+------+
| CID | Cuisine | Quantity | Bill |
+-----+---------+----------+------+
|   1 |       3 |        2 | 1600 |
|   2 |       3 |        2 | 1600 |
|   3 |       1 |        1 |  300 |
|   4 |       2 |        2 | 1000 |
+-----+---------+----------+------+
4 rows in set (0.00 sec)
```

5) TOTAL BILL-

```
mysql> select * from total_bill;
+-----+--------------+------------+---------+---------------+------------+
| CID | C_NAME       | grand_total | RoomRent | RestaurantBill | GamingBill |
+-----+--------------+------------+---------+---------------+------------+
|   1 | AYUSH SHARMA |     122401 |  120000 |          1600 |        801 |
|   2 | ADITYA SINGH |      31751 |   30000 |          1600 |        151 |
|   3 | RIYA JHA     |      15300 |   15000 |           300 |          0 |
|   4 | BELLY COOPER |      81000 |   80000 |          1000 |          0 |
+-----+--------------+------------+---------+---------------+------------+
4 rows in set (0.00 sec)
```