CHINMAYA VIDYALAYA
ANNA NAGAR

[COMPUTER SCIENCE PROJECT]

LODGING MANAGEMENT SYSTEM

Harish Anand
Sabarish B
Vishal H
[Class XII-B]

ACKNOWLEDGEMENT

I would like to express my gratitude towards our Madam Principal Mrs. V Gowri Lakshmi, Madam Vice Principal Mrs. D Lakshmi Prabha and our Computer Science teacher Mrs. Gnanavadivu for giving us the opportunity to work on our project titled "LODGING MANAGEMENT SYTEM", and providing the needful guidance.

I would also like to thank the members of my group who showed support, co-operation and provided various suggestions to improve the quality of the project.

I would also like to thank my parents and friends who helped me in finalizing this project within the allotted time frame.

EXAM ROLL NO: ___________________

CHINMAYA VIDYALAYA, ANNANAGAR

Plot No. 5063A , Z-Block , Belly Area ,
Anna Nagar , Chennai-600 040

BONAFIDE CERTIFICATE

This is to certify that ____________________of class XII__ has performed the
__________________________ project titled ______________________________at
CHINMAYA VIDYALAYA , Anna Nagar during the academic year 2020-21.

Submitted for Practical Examination held on ____________ at CHINMAYA
VIDYALAYA, Anna Nagar.

Teacher in-charge:

Signature of the Principal   Signature of External Examiner

TABLE OF CONTENTS

WHY PYTHON?:

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Reason for increasing popularity

1. Emphasis on code readability, shorter codes, ease of writing
2. Programmers can express logical concepts in fewer lines of code in comparison to languages such as C++ or Java.
3. Python supports multiple programming paradigms, like object-oriented, imperative and functional programming or procedural.
4. There exist inbuilt functions for almost all of the frequently used concepts.
5. Philosophy is “Simplicity is the best”.

LANGUAGE FEATURES

•Interpreted

•There are no separate compilation and execution steps like C and C++.

•Directly run the program from the source code.

•Internally, Python converts the source code into an intermediate form called bytecodes which is then translated into native language of specific computer to run it.

•No need to worry about linking and loading with libraries, etc.

•Platform Independent

•Python programs can be developed and executed on multiple operating system platforms.

•Python can be used on Linux, Windows, Macintosh, Solaris and many more.

•Free and Open Source; Redistributable

•High-level Language

•In Python, no need to take care about low-level details such as managing the memory used by the program.

•Simple

•Closer to English language;Easy to Learn

•More emphasis on the solution to the problem rather than the syntax

•Embeddable

•Python can be used within C/C++ program to give scripting capabilities for the program's users.

•Robust:

•Exceptional handling features

•Memory management techniques in built

•Rich Library Support

•The Python Standard Library is vary vast.

•Known as the "batteries included" philosophy of Python ; It can help do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.

•Besides the standard library, there are various other high-quality libraries such as the Python Imaging Library which is an amazingly simple image manipulation library.

****

PROJECT DESIGN:

ABOUT THE PROJECT:

The aim is to provide an efficient, all-encompassing system of hotel/lodging management. This project was done with the perspective of distribution to a lodging firm, to be implemented as an appendage to their operations. It is to be used by employees of the company, and can be integrated into a website, where particular features can be accessed by the customer. Each lodge consists rooms of 3 tiers with varying prices and facilities accordingly. A hierarchical approach to the operations is used, where in certain functions of the code can only be accessed by those with clearance, for this instance, a password is used to split the operations. A system of booking rooms, checking, “front-desk” operations and Management operations are implemented. Navigation of said functions are done through a concise GUI for a fast UX.

Functions of the project:

- The structure of the code is based on a sequential system of function call used to navigate the text-based UI. The entirety of the code, including function definitions are placed in a MAIN_UI () function defined initially. This, allows for the MAIN_UI () function to be called inside the it, thus making a simple sequence of UI components that can be navigated back and forth and ends the program only on the user's discretion.
- The first frame of the UI sequence is used to specify the user's intent with the application, i.e., if they are a client or an employee. The client only has access to the bookRoom () function, while the employees have access to the employee portal, which brings us to the second frame of the UI.
- The employeePortal () function consists of functions that perform operations for company's perspective, where actions are divided into hierarchical levels of clearance, between higher management and other employees. It also makes use of the aforementioned code structure where the MAIN_UI () function can be called to navigate back and between frames.

- The **Client-side** system used a [bookRoom ()] function to display available branches, display the details of the rooms of the branch (room no., tiers) and finally procced to enter their details to finalize booking, all in succession through a sequential UI system. The client is asked to enter their name, date of booking and number of days of stay. A bill for the booking will be produced at finalization confirming the booking details and the relative pricing. The pricing is based on tier of the booked rooms, the tier is displayed during the booking process with their respective room numbers. The booking details are entered into the database with a full set of information, to the time and date of booking completion [done using the datetime module].

- The databases for the rooms in each branch are updated to depict occupation, and used to retrieve desired information about the room, i.e., tiers (related to room's pricing and facilities). The bookings databases present for the branch is appended each time a booking is made, with the aforementioned data set.

- The **company/server-side** of the system is where most of project's functionalities lie. Sequential frames of UIs are used to navigate the project, the employee portal UI is opened after a password is entered. Here, employee actions are divided into sections pertaining to each. Room Services, Reception and Higher management functions comprise the employee UI.

- Rooms requiring assistance are checked from the database, and employees using the function can serve the room mentioned when the function is called. Once served, the employees can use the system to update the database.

- The receptionist menu presents the user with access to the check-in and check-out functions, allows them to check room occupancy upon request, and retrieve booking details; converting them into a printable format. Checking clients out of rooms is done only through the Reception UI where the checkout () function is called. The function updates the respective databases upon checkout, check extended stays and adds penalties accordingly Receptionists can also make use of certain functions and check room occupancy [checkRoomOccupancy_RecpAction ()]. It is meant to be used as a means for the employee to look for empty rooms on short notice or upon the customer's request.

- Employees, through the reception portal, can also retrieve booking data from the database for referrals for their specific branch. The retrieved consisting of a list of completed bookings, can be converted into a printable format using the bookingDetails_ForPrint () function. The function converts the data retrieved from the database into a list of tuples that are then written onto a spreadsheet using the CSV module. Employees under higher management have access to booking info for all branches.

- Booking Data is stored in two forms to help referrals and are meant to be used a ledger of sorts, a database comprising details for bookings done on all branches and one for each branch. Employees with Higher Management clearances can access these functions, using them to add new room to the databases of each branch. They are also able to print out the data set for bookings done on all branches for referral.

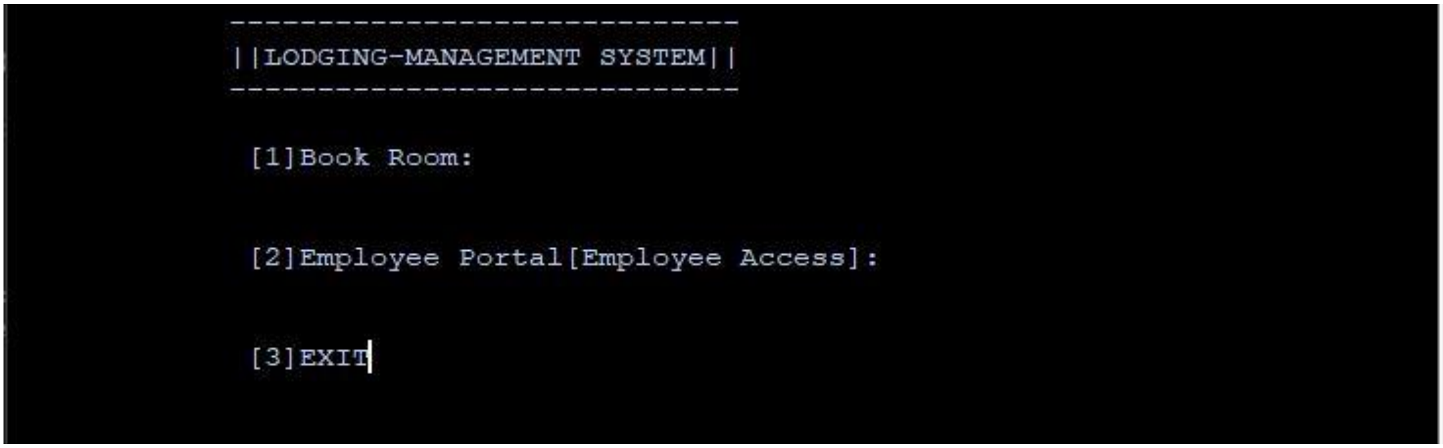
- Higher Management have access to the booking data for all branches. Employees with higher management clearance(password) also have access to the financial records of all the branches. Data consisting of room numbers and prices are stores in a separate database. Employees can use the features of this function to avail the complete data for each branch or all. They can also avail cumulative profits and booking numbers for calculations. The Financial_Records_UI () function contains the mentioned features. This function only collects financial data brought in through hotel bookings.

MODULES USED:

- mysql.connector
- datetime(date and timedelta functions)
- CSV
- tabulate

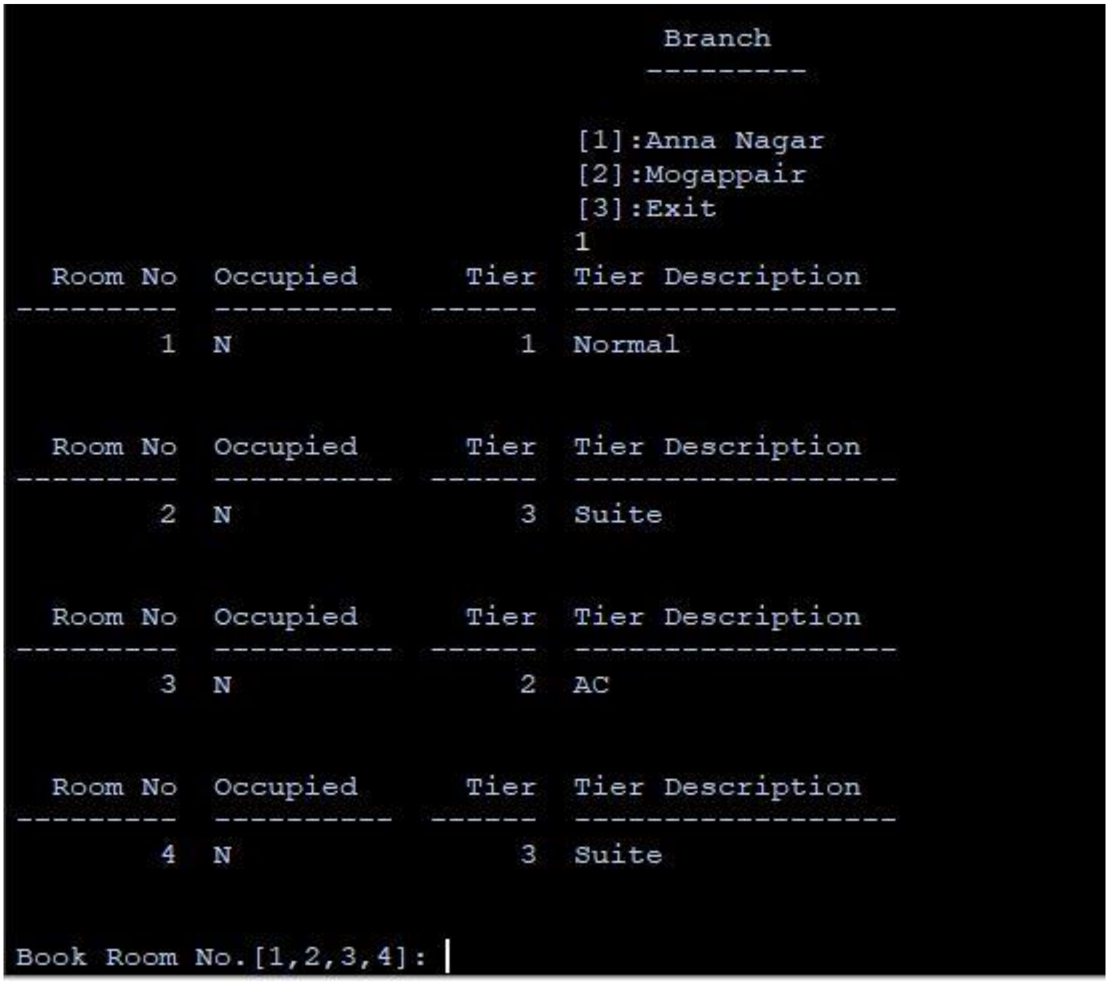
MENUS:

LODGING UI MENU:



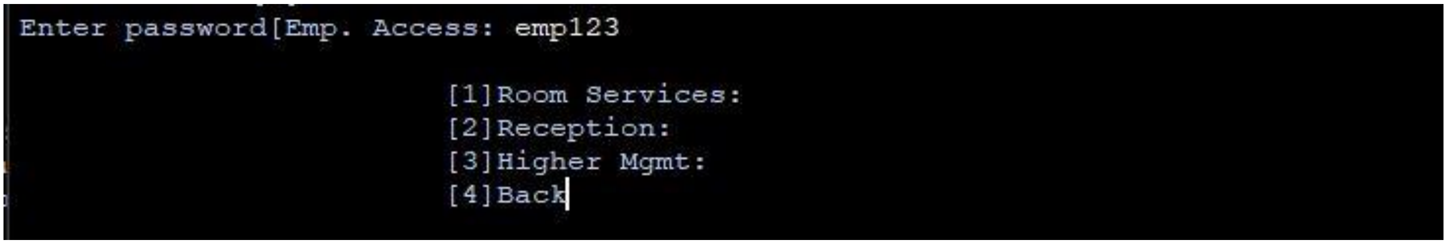
```
             -----------------------------
             ||LODGING-MANAGEMENT SYSTEM||
             -----------------------------

              [1]Book Room:


              [2]Employee Portal[Employee Access]:


              [3]EXIT
```

BOOKING MENU:



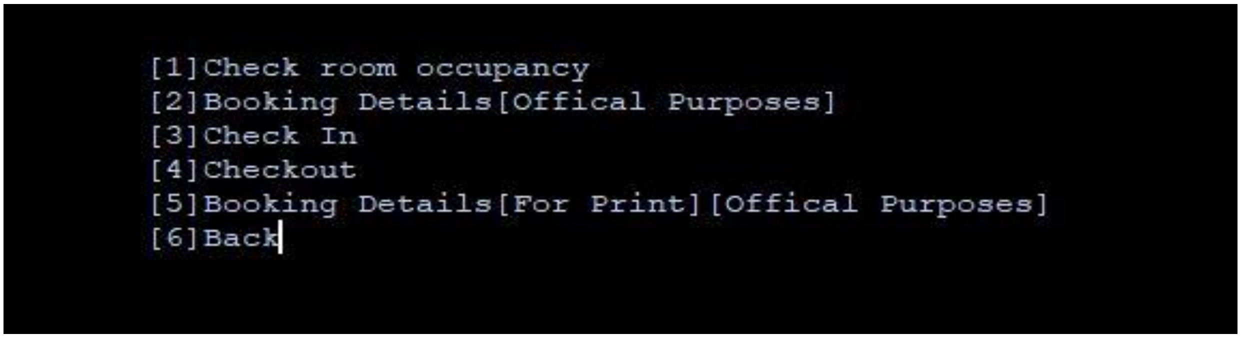
```
                                        Branch
                                       ---------

                                   [1]:Anna Nagar
                                   [2]:Mogappair
                                   [3]:Exit
                                   1
  Room No  Occupied      Tier  Tier Description
---------  ----------  ------  ------------------
        1  N                1  Normal


  Room No  Occupied      Tier  Tier Description
---------  ----------  ------  ------------------
        2  N                3  Suite


  Room No  Occupied      Tier  Tier Description
---------  ----------  ------  ------------------
        3  N                2  AC


  Room No  Occupied      Tier  Tier Description
---------  ----------  ------  ------------------
        4  N                3  Suite


Book Room No.[1,2,3,4]:
```

EMPLOYEE PORTAL:



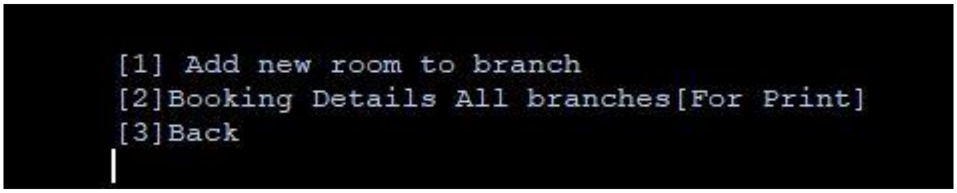
```
Enter password[Emp. Access: emp123

                              [1]Room Services:
                              [2]Reception:
                              [3]Higher Mgmt:
                              [4]Back
```

RECEPTION MENU:



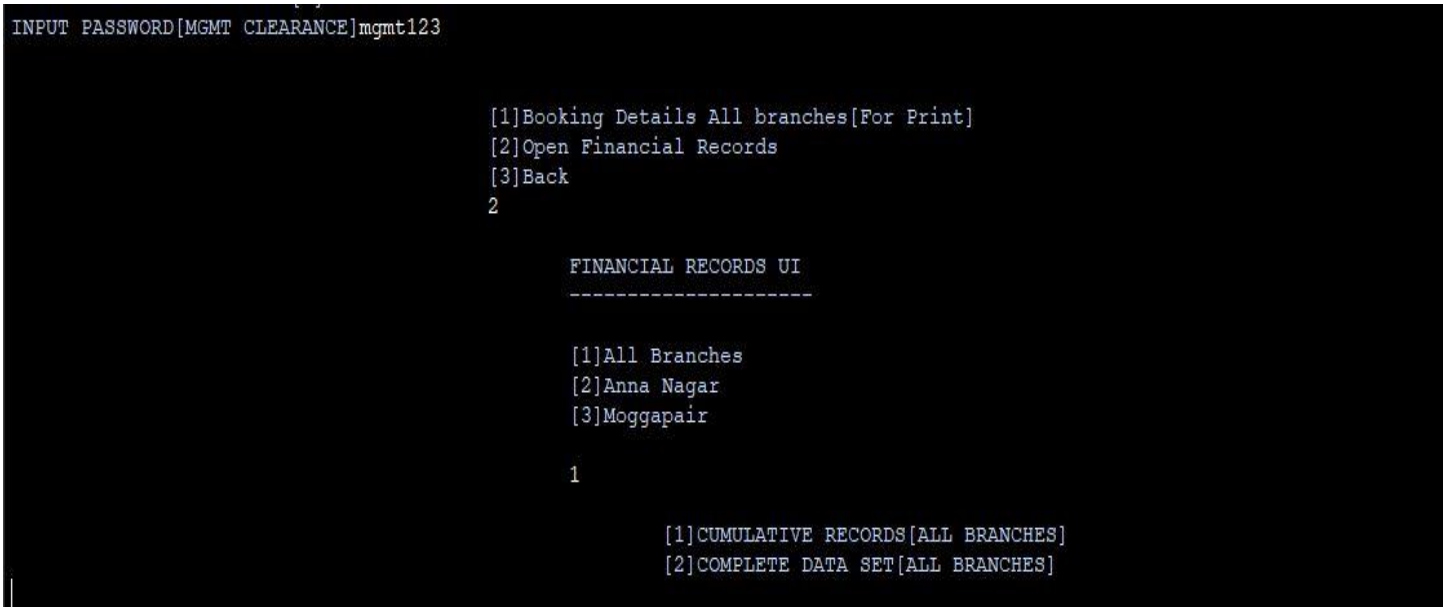
```
          [1]Check room occupancy
          [2]Booking Details[Offical Purposes]
          [3]Check In
          [4]Checkout
          [5]Booking Details[For Print][Offical Purposes]
          [6]Back
```

HIGHER MANAGEMENT MENU:



```
        [1] Add new room to branch
        [2]Booking Details All branches[For Print]
        [3]Back
```

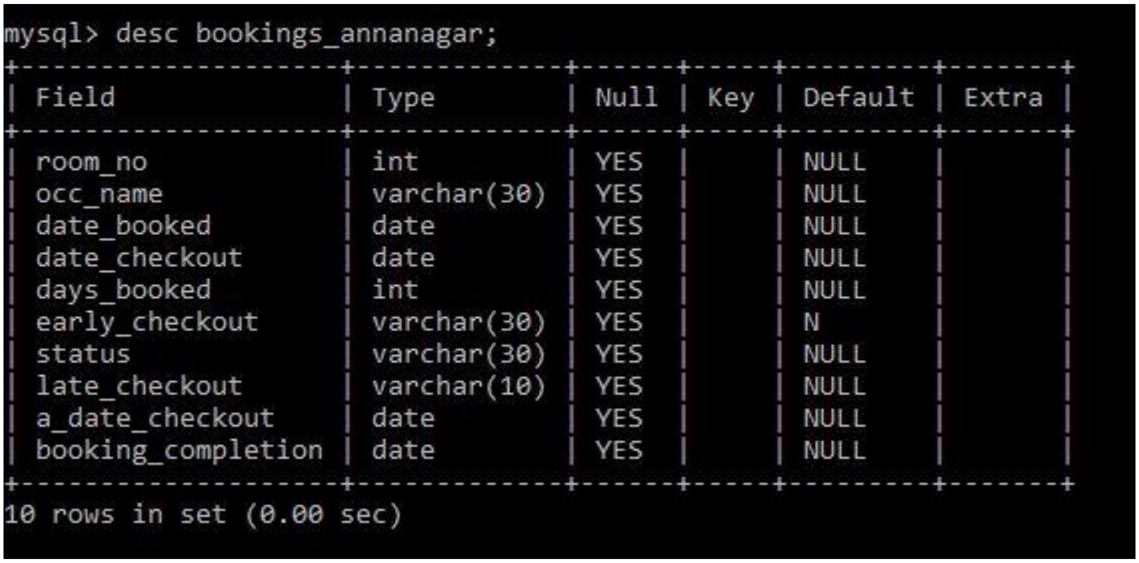
HIGHER MANAGEMENT/FINANCIAL RECORDS UI:



```
INPUT PASSWORD[MGMT CLEARANCE]mgmt123


                                        [1]Booking Details All branches[For Print]
                                        [2]Open Financial Records
                                        [3]Back
                                        2

                                               FINANCIAL RECORDS UI
                                               ---------------------

                                               [1]All Branches
                                               [2]Anna Nagar
                                               [3]Moggapair

                                               1

                                                       [1]CUMULATIVE RECORDS[ALL BRANCHES]
                                                       [2]COMPLETE DATA SET[ALL BRANCHES]
```

FILES/SQL TABLES USED:

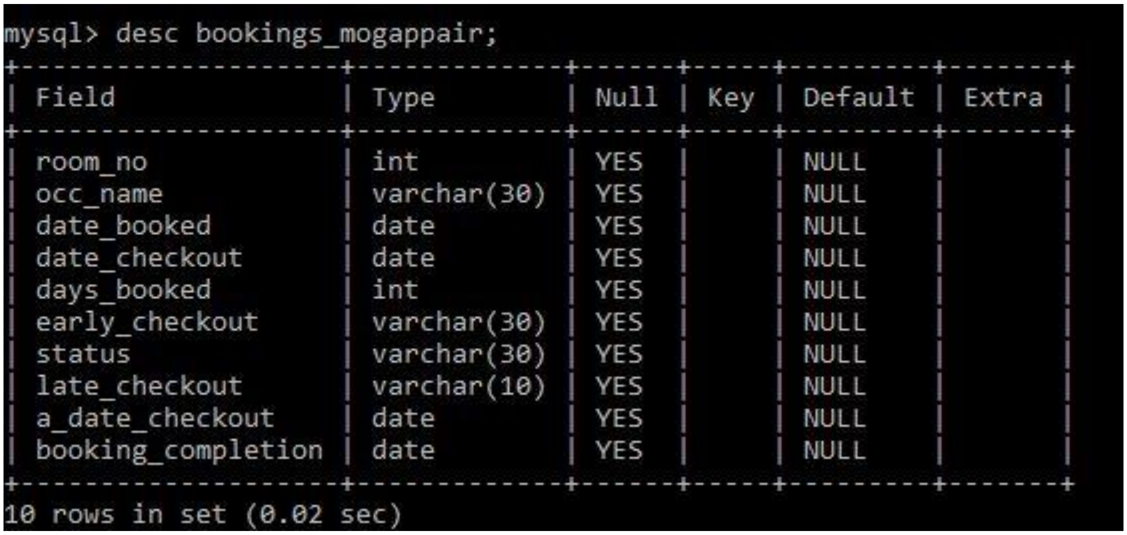
bookings_annanagar:



```
mysql> desc bookings_annanagar;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| room_no           | int         | YES  |     | NULL    |       |
| occ_name          | varchar(30) | YES  |     | NULL    |       |
| date_booked       | date        | YES  |     | NULL    |       |
| date_checkout     | date        | YES  |     | NULL    |       |
| days_booked       | int         | YES  |     | NULL    |       |
| early_checkout    | varchar(30) | YES  |     | N       |       |
| status            | varchar(30) | YES  |     | NULL    |       |
| late_checkout     | varchar(10) | YES  |     | NULL    |       |
| a_date_checkout   | date        | YES  |     | NULL    |       |
| booking_completion | date       | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

Consists of booking information the lodging firm’s Anna Nagar branch. Used as a list of all previous bookings for use of statistics, investigations if need be.

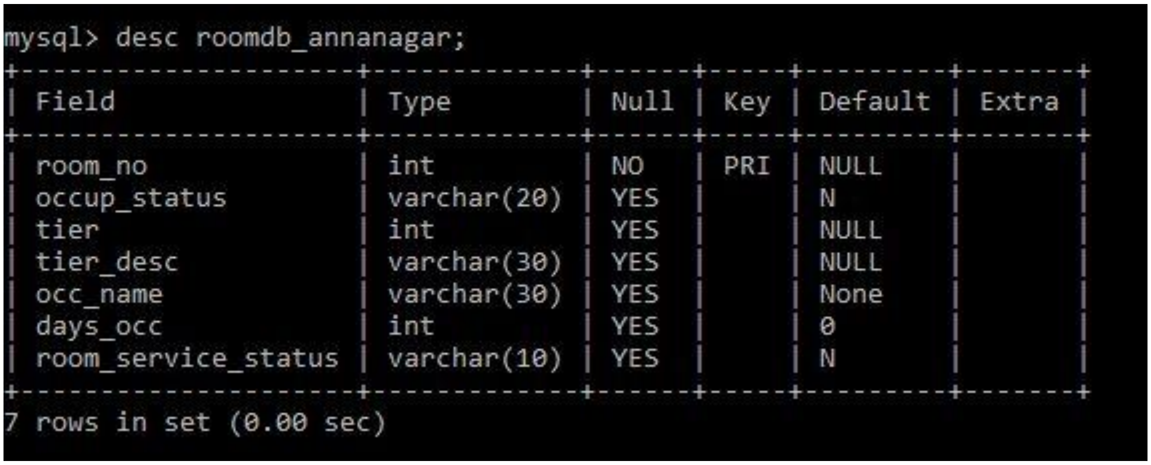
bookings_mogappair:



```
mysql> desc bookings_mogappair;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| room_no           | int         | YES  |     | NULL    |       |
| occ_name          | varchar(30) | YES  |     | NULL    |       |
| date_booked       | date        | YES  |     | NULL    |       |
| date_checkout     | date        | YES  |     | NULL    |       |
| days_booked       | int         | YES  |     | NULL    |       |
| early_checkout    | varchar(30) | YES  |     | NULL    |       |
| status            | varchar(30) | YES  |     | NULL    |       |
| late_checkout     | varchar(10) | YES  |     | NULL    |       |
| a_date_checkout   | date        | YES  |     | NULL    |       |
| booking_completion | date       | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
10 rows in set (0.02 sec)
```

Consists of booking information the lodging firm’s Mogappair branch. Used as a list of all previous bookings for use of statistics, investigations if need be.

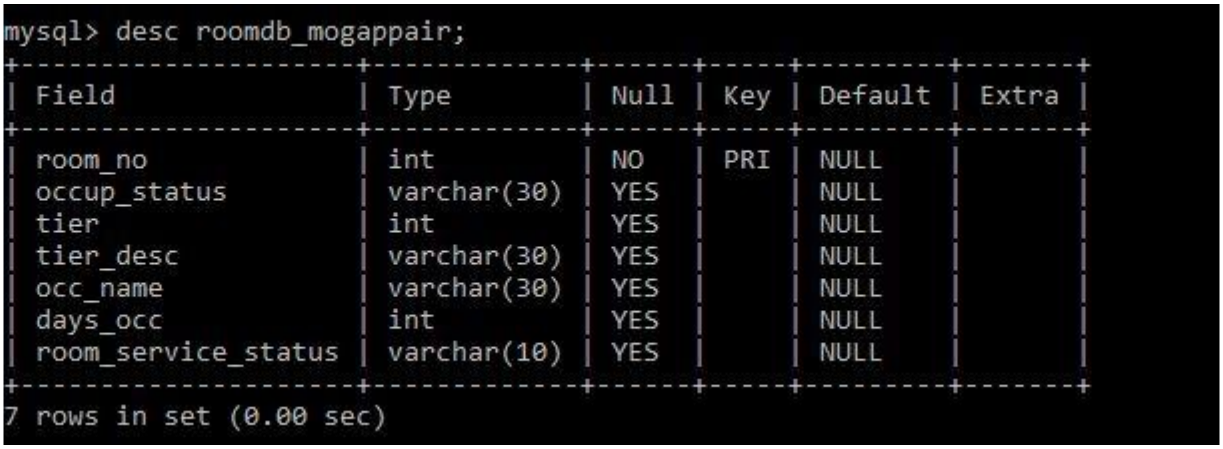
roomDB_annanagar:



```
mysql> desc roomdb_annanagar;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| room_no             | int         | NO   | PRI | NULL    |       |
| occup_status        | varchar(20) | YES  |     | N       |       |
| tier                | int         | YES  |     | NULL    |       |
| tier_desc           | varchar(30) | YES  |     | NULL    |       |
| occ_name            | varchar(30) | YES  |     | None    |       |
| days_occ            | int         | YES  |     | 0       |       |
| room_service_status | varchar(10) | YES  |     | N       |       |
+---------------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Used as data set/list of rooms in the Anna Nagar branch that is shown as a UI to the user in order pick rooms. Defines tiers, occupation status and handles requests for room service.

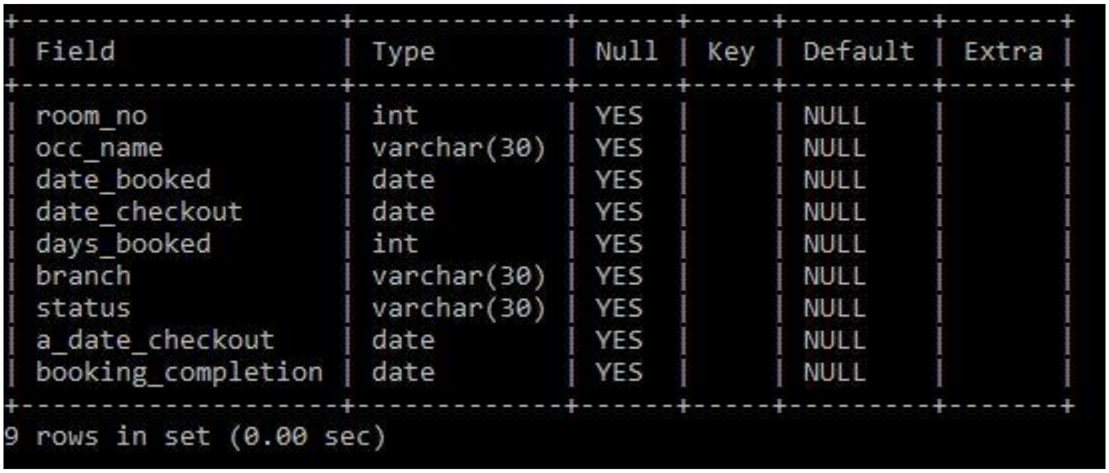
roomDB_mogappair:



```
mysql> desc roomdb_mogappair;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| room_no             | int         | NO   | PRI | NULL    |       |
| occup_status        | varchar(30) | YES  |     | NULL    |       |
| tier                | int         | YES  |     | NULL    |       |
| tier_desc           | varchar(30) | YES  |     | NULL    |       |
| occ_name            | varchar(30) | YES  |     | NULL    |       |
| days_occ            | int         | YES  |     | NULL    |       |
| room_service_status | varchar(10) | YES  |     | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Used as data set/list of rooms in the Mogappair branch that is shown as a UI to the user in order pick rooms. Defines tiers, occupation status and handles requests for room service.

bookings_allbranches



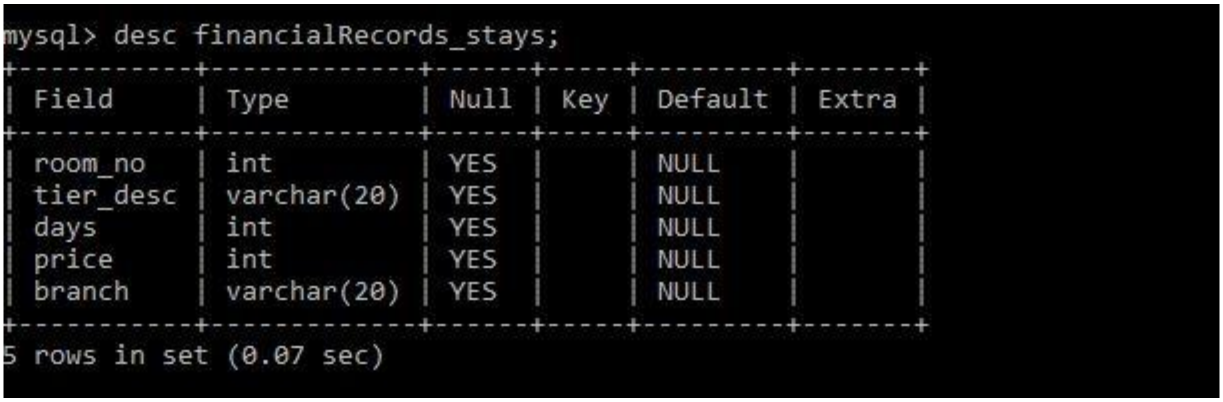
```
+-------------------+------------+------+-----+---------+-------+
| Field             | Type       | Null | Key | Default | Extra |
+-------------------+------------+------+-----+---------+-------+
| room_no           | int        | YES  |     | NULL    |       |
| occ_name          | varchar(30)| YES  |     | NULL    |       |
| date_booked       | date       | YES  |     | NULL    |       |
| date_checkout     | date       | YES  |     | NULL    |       |
| days_booked       | int        | YES  |     | NULL    |       |
| branch            | varchar(30)| YES  |     | NULL    |       |
| status            | varchar(30)| YES  |     | NULL    |       |
| a_date_checkout   | date       | YES  |     | NULL    |       |
| booking_completion| date       | YES  |     | NULL    |       |
+-------------------+------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

Used as data set/list of rooms for all branchesthat is shown as a UI to

the user in order pick rooms. Defines tiers, occupation status and handles requests

for room service. Accessible only by higher management

financialRecords_stays:



```
mysql> desc financialRecords_stays;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| room_no   | int         | YES  |     | NULL    |       |
| tier_desc | varchar(20) | YES  |     | NULL    |       |
| days      | int         | YES  |     | NULL    |       |
| price     | int         | YES  |     | NULL    |       |
| branch    | varchar(20) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.07 sec)
```

Used as ledger for financial proceedings. Also serves as a data set for profit calculations, max prices etc.

SOURCE CODE:

```
import mysql.connector as mc
import datetime
from datetime import date
from datetime import timedelta
import csv
from tabulate import tabulate
conn=mc.connect(host="localhost",user="root",passwd="vish100%",database="lodging")
cur=conn.cursor()

def LODGING_MAIN_UI():
    def bookRoom():
        branch=int(input("""
                                    Branch
                                   ---------\n
                               [1]:Anna Nagar
                               [2]:Mogappair
                               [3]:Exit
                               """))
        if branch==3:
            LODGING_MAIN_UI()
        if branch==1:
            cur.execute("select room_no,occup_status,tier,tier_desc from roomdb_annanagar;")
            rooms=cur.fetchall()

            for i in rooms:
                table=tabulate([i],headers=["Room No","Occupied","Tier","Tier Description"])
                print(table)
                print("\n")
            room_no=int(input("Book Room No.[1,2,3,4]: "))
            occ_name=input("Name of occupant: ")
            date_booked_day=int(input("Enter booking day: "))
            date_booked_month=int(input("Enter booking month: "))
            date_booked_year=int(input("Enter booking year: " ))
            date_booked=datetime.date(date_booked_year,date_booked_month,date_booked_day)
            days_booked=int(input("No. of days for stay: "))
            date_checkout=date_booked+datetime.timedelta(days=days_booked)
            current_datetime=date.today()

            cur.execute("select*from roomdb_annanagar")
            roomdb_show=cur.fetchall()
            print
            cur.execute(f"select(occup_status)from roomdb_annanagar where room_no={room_no};")
            occup_status_check=cur.fetchall()
            occup_status_check=occup_status_check[0][0]
            if occup_status_check=="staying":
                print("Room already booked")



            else:
                addtoallbranch=f"insert into bookings_allbranches values({room_no},'{occ_name}','{date_booked}',NULL,{days_booked},'{branch}','staying','{date_checkout}', NULL);"
                cur.execute(addtoallbranch)
                set_current_date=f"update bookings_allbranches set booking_completion='{current_datetime}' where room_no={room_no};"
                cur.execute(set_current_date)

                addtospecificbranch=f"insert into bookings_annanagar values({room_no},'{occ_name}','{date_booked}',NULL,{days_booked},'N','staying','N','{date_checkout}',NULL);"
                cur.execute(addtospecificbranch)
                set_current_date_S=f"update bookings_annanagar set booking_completion='{current_datetime}'
        where room_no={room_no};"
                cur.execute(set_current_date_S)




                update_room=f"update roomdb_annanagar set
        occup_status='Y',occ_name='{occ_name}',days_occ='{days_booked}'where room_no={room_no};"
                cur.execute(update_room)

                tier_check_query=f"select(tier)from roomdb_annanagar where room_no={room_no};"
                cur.execute(tier_check_query)
                tier_res=cur.fetchall()
                if tier_res[0][0]==1:
                  price=250*days_booked
                elif tier_res[0][0]==2:
                  price=500*days_booked
               elif tier_res[0][0]==3:
                  price=1000*days_booked
                cur.execute(f"select tier_desc from roomdb_annanagar where room_no={room_no};")
                tier_desc=cur.fetchall()
                tier_description=tier_desc[0][0]



                cur.execute(f"insert into financialRecords_stays
        values({room_no},'{tier_description}',{days_booked},{price},'anna nagar');")




            print(f"""                          BILLING:
                  Date:{current_datetime}
```

```
                                            Name:{occ_name}
                                            Room No:{room_no}
                                            Booked from:{date_booked}|To:{date_checkout}|For:{days_booked} days

                                            Price:
                                           {price}""")
    conn.commit()



if branch==2:
    cur.execute("select room_no,occup_status,tier,tier_desc from roomdb_mogappair;")
    rooms=cur.fetchall()

    for i in rooms:
        table=tabulate([i],headers=["Room No","Occupied","Tier","Tier Description"])
        print(table)
        print("\n")

    room_no=int(input("Book Room No.[1,2,3,4]: "))
    occ_name=input("Name of occupant: ")
    date_booked_day=int(input("Enter booking day: "))
    date_booked_month=int(input("Enter booking month: "))
    date_booked_year=int(input("Enter booking year: " ))
    date_booked=datetime.date(date_booked_year,date_booked_month,date_booked_day)
    days_booked=int(input("No. of days for stay: "))
    date_checkout=date_booked+datetime.timedelta(days=days_booked)
    current_datetime=date.today()
    cur.execute(f"select(occup_status)from roomdb_mogappair where room_no={room_no};")
    occup_status_check=cur.fetchall()
```

```
        occup_status_check=occup_status_check[0][0]



        if occup_status_check=="staying":
            print("Room already booked")
        else:
            addtoallbranch=addtoallbranch=f"insert into bookings_allbranches values({room_no},'{occ_name}','{date_booked}',NULL,{days_booked},'{branch}','staying','{date_checkout}', NULL);"
            cur.execute(addtoallbranch)
            set_current_date=f"update bookings_allbranches set booking_completion='{current_datetime}' where room_no={room_no};"

            cur.execute(set_current_date)
            addtospecificbranch=f"insert into bookings_mogappair values({room_no},'{occ_name}','{date_booked}',NULL,{days_booked},'N','staying','N','{date_checkout}',NULL);"
            cur.execute(addtospecificbranch)
            set_current_date_S=f"update bookings_mogappair set booking_completion='{current_datetime}' where room_no={room_no};"
            cur.execute(set_current_date_S)

            update_room=f"update roomdb_mogappair set occup_status='Y',occ_name='{occ_name}',days_occ='{days_booked}'where room_no={room_no};"
            cur.execute(update_room)



            tier_check_query=f"select(tier)from roomdb_annanagar where room_no={room_no}"
            cur.execute(tier_check_query)
            tier_res=cur.fetchall()
            if tier_res[0][0]==1:
                price=250*days_booked
                elif tier_res[0][0]==2:
                    price=500*days_booked
                elif tier_res[0][0]==3:
                    price=1000*days_booked

                cur.execute(f"select tier_desc from roomdb_annanagar where room_no={room_no};")
                tier_desc=cur.fetchall()
                tier_description=tier_desc[0][0]



                cur.execute(f"insert into financialRecords_stays
values({room_no},'{tier_description}',{days_booked},{price},'mogappair');")

                print(f"""                              BILLING:
                Date:{current_datetime}

                            Name:{occ_name}
                            Room No:{room_no}
                            Booked from:{date_booked} |To:{date_checkout}|For:{days_booked}

                            Price{price}""")

    def checkout():
        branch=int(input("Enter the branch[1]Anna Nagar [2]Mogappair [3]Exit: "))
        current_datetime=date.today()
        if branch==1:
            cur.execute("select room_no,occup_status,tier,tier_desc from roomdb_annanagar;")
            rooms=cur.fetchall()
            for i in rooms:
                table=tabulate([i],headers=["Room No","Occupied","Tier","Tier Description"])
```

```
           print(table)
           print("\n")
        room_no=int(input("Enter room no: "))
        cur.execute(f"select(occup_status)from roomdb_annanagar where room_no={room_no};")
        occup_status_check=cur.fetchall()
        occup_status_check=occup_status_check[0][0]

        if occup_status_check=="Y":
            cur.execute("select(a_date_checkout)from bookings_annanagar;")
            r=cur.fetchall()
            r=r[0][0]

            if current_datetime<r:
               update_checkedoutEARLY=f"update bookings_allbranches set status='Checked Out',
date_checkout='{current_datetime}' where room_no={room_no};"
               cur.execute(update_checkedoutEARLY)


               b_update_checkedoutEARLY=f"update bookings_annanagar set status='Checked Out',
early_checkout='Y', date_checkout='{current_datetime}' where room_no={room_no};"
               cur.execute(b_update_checkedoutEARLY)
               update_room_checkedoutEARLY=f"update roomdb_annanagar set occup_status='N',
occ_name='None',days_occ=0 where room_no={room_no};"
               cur.execute(update_room_checkedoutEARLY)
               print("Check Out Complte, Thanks for staying")
            if current_datetime>r:
               print("EXTENDED CHECKOUT")
               update_checkedoutLATE=f"update bookings_allbranches set status='Checked
Out',date_checkout='{current_datetime}' where room_no={room_no};"
               curexecute(update_checkedoutLATE)


               b_update_checkedoutLATE=f"update bookings_annanagar set status='Checked Out',
late_checkout='Y',date_checkout='{current_datetime}' where room_no={room_no};"
```

```
            cur.execute(b_update_checkedoutLATE)


            update_room_checkedoutLATE=f"update roomdb_annanagar set occup_status='N',
occ_name='None',days_occ=0 where room_no={room_no};"
            cur.execute(update_room_checkedoutLATE)
            print("Check Out Complte, Thanks for staying")
            print(f"""                  EXTENDED CHECKOUT NOTICE
                     The Client has overstayed/ extended their checkout for|{current_datetime-r}
days|
                     A penalty of{(current_datetime-r)*50} is to be paid""")


          else:
            update_checkedout=f"update bookings_allbranches set status='Checked
Out',date_checkout='{current_datetime}' where room_no={room_no};"
            cur.execute(update_checkedout)


            b_update_checkedout=f"update bookings_annanagar set status='Checked
Out',date_checkout='{current_datetime}' where room_no={room_no};"
            cur.execute(b_update_checkedout)
            update_room_checkedout=f"update roomdb_annanagar set occup_status='N',
occ_name='None',days_occ=0 where room_no={room_no};"
            cur.execute(update_room_checkedout)
            print("Check Out Complete, Thanks for staying")
      else:
        print("Room Not Booked")
    if branch==2:
      cur.execute("select room_no,occup_status,tier,tier_desc from roomdb_mogappair;")
      rooms=cur.fetchall()
      for i in rooms:
        table=tabulate([i],headers=["Room No","Occupied","Tier","Tier Description"])
        print(table)
        print("\n")
      room_no=int(input("Enter room no: "))
      cur.execute(f"select(occup_status)from roomdb_mogappair where room_no={room_no};")
      occup_status_check=cur.fetchall()
      occup_status_check=occup_status_check[0][0]



      if occup_status_check=="Y":
          cur.execute("select(a_date_checkout)from bookings_mogappair;")
          r=cur.fetchall()
          r=r[0][0]

          if current_datetime<r:
            update_checkedoutEARLY=f"update bookings_allbranches set status='Checked Out',
date_checkout='{current_datetime}' where room_no={room_no};"
            cur.execute(update_checkedoutEARLY)

            b_update_checkedoutEARLY=f"update bookings_mogappair set status='Checked Out',
early_checkout='Y', date_checkout='{current_datetime}' where room_no={room_no};"
            cur.execute(b_update_checkedoutEARLY)
            update_room_checkedoutEARLY=f"update roomdb_mogappair set occup_status='N',
occ_name='None',days_occ=0 where room_no={room_no};"
            cur.execute(update_room_checkedoutEARLY)
            print("Check Out Complte, Thanks for staying")
          if current_datetime>r:
            print("EXTENDED CHECKOUT")
            update_checkedoutLATE=f"update bookings_allbranches set status='Checked
Out',date_checkout='{current_datetime}' where room_no={room_no};"
            curexecute(update_checkedoutLATE)

            b_update_checkedoutLATE=f"update bookings_mogappair set status='Checked Out',
late_checkout='Y',cdate_checkout='{current_datetime}' where room_no={room_no};"
            cur.execute(b_update_checkedoutLATE)
```

```
           update_room_checkedoutLATE=f"update roomdb_mogappair  set occup_status='N',
occ_name='None',days_occ=0 where room_no={room_no};"
           cur.execute(update_room_checkedoutLATE)
           print("Check Out Complte, Thanks for staying")
           print(f"""                    EXTENDED CHECKOUT NOTICE
                        The Client has overstayed/ extended their checkout for|{current_datetime-r}
days|
                        A penalty of{(current_datetime-r)*50} is to be paid""")
         else:
           update_checkedout=f"update bookings_allbranches set status='Checked
Out',date_checkout='{current_datetime}' where room_no={room_no};"
           cur.execute(update_checkedout)


           b_update_checkedout=f"update bookings_mogappair set status='Checked Out',
date_checkout='{current_datetime}' where room_no={room_no};"
           cur.execute(b_update_checkedout)
           update_room_checkedout=f"update roomdb_mogappair  set occup_status='N',
occ_name='None',days_occ=0 where room_no={room_no};"
           cur.execute(update_room_checkedout)
           print("Check Out Complte, Thanks for staying")
      else:
        print("Room not booked")
    def checkRoomOccupancy_RecpAction():
    print("""
                              Room Occupancy
                            ----------------------""")


    branch=int(input("Enter branch:[1]Anna nagar [2]Mogappair\n"))


    if branch==1:
            check_room_occupancy=f"select room_no,tier_desc from roomdb_annanagar where
occup_status='N';"
            cur.execute(check_room_occupancy)
            empty_rooms=cur.fetchall()
            print("""                        EMPTY ROOMS\n""")
            for i in empty_rooms:
                table=tabulate([i],headers=["Room.No","Tier Desc"])
                print(table,"\n")

        if branch==2:
            check_room_occupancy=f"select room_no,tier_desc from roomdb_mogappair where
occup_status='N';"
            cur.execute(check_room_occupancy)
            empty_rooms=cur.fetchall()
            print("""                        EMPTY ROOMS\n""")
            for i in empty_rooms:
                table=tabulate([i],headers=["Room.No","Tier Desc"])
                print(table,"\n")

    def bookingDetails_branch():
        branch=int(input("Enter branch:[1]Anna Nagar [2]Mogappair\n "))
        if branch==1:
            showDetails=f"select*from bookings_annanagar;"
            cur.execute(showDetails)
            branchBookingDetails=cur.fetchall()
            print("""
                    BOOKING DETAILS
                ------------------------""")
            for i in branchBookingDetails:
                print("Room No:",i[0],"\nOccupant Name:",i[1],"\nDate Booked",i[2],"\nDate checked
Out",i[3],"\nDays Booked",i[4],"\nEarly Checkout?:",i[5],"\nStatus:",i[6],"\nLate
Checkout:",i[7],"\nChecked Out on:",i[8],"\nDate of booking Completion:",i[9],"\n----------------")
        if branch==2:
            showDetails=f"select*from bookings_mogappair;"
            cur.execute(showDetails)
            branchBookingDetails=cur.fetchall()
            for i in branchBookingDetails:
                print("Room No:",i[0],"\nOccupant Name:",i[1],"\nDate Booked",i[2],"\nDate checked
Out",i[3],"\nDays Booked",i[4],"\nEarly Checkout?:",i[5],"\nStatus:",i[6],"\nLate
Checkout:",i[7],"\nChecked Out on:",i[8],"\nDate of booking Completion:",i[9],"\n----------------")




    def bookingDetails_ForPrint(x):
        if x=="Mgmt":
            cur.execute("select*from bookings_allbranches;")
            rs=cur.fetchall()
            with open("BookingsDetails_All.csv","w",newline='')as file:
                writer=csv.writer(file)
                writer.writerow(["Room No","Occ.Name","Check In Date","Checked Out on","Days
Booked","Branch","Status","Date to CheckOut","Booking Completion"])
                writer.writerows(rs)
            print("CSV File created: BookingsDetails_All.csv")
        else:
            branch=int(input("Enter branch [1]Anna Nagar [2]Mogappair"))
            if branch==1:
                cur.execute("select*from bookings_annanagar;")
                rs=cur.fetchall()
                with open("BookingsDetails_AnnaNagar.csv","w",newline='')as file:
                    writer=csv.writer(file)
```

```
          writer.writerow(["Room No","Occ.Name","Check In Date","Checket out on","Days
Booked","Early Checkout","Status","Late Checkout","Date to checkout","Booking Completion"])
          writer.writerows(rs)
        print("CSV File created: BookingsDetails_AnnaNagar.csv")
      if branch==2:
        cur.execute("select*from bookings_annanagar;")
        rs=cur.fetchall()
        with open("BookingsDetails_Mogappair.csv","w",newline='')as file:
          writer=csv.writer(file)
          writer.writerow((["Room No","Occ.Name","Check In Date","Checket out on","Days
Booked","Early Checkout","Status","Late Checkout","Date to checkout","Booking Completion"]))
          writer.writerows(rs)
        print("CSV File created: BookingsDetails_Mogappair.csv")
  def roomServices():
        print("""
                          ROOM SERVICES
                        ----------------------""")
        branchRoomServ=int(input("                    Enter branch:[1]Anna nagar [2] Mogappair
[3]Back\n\n"))
        if branchRoomServ==1:
          queryServ="select room_no from roomdb_annanagar where room_service_status='Y'"
          cur.execute(queryServ)
          ServReq=cur.fetchall()

          if len(ServReq)==0:
            print("None Requiring service at the moment.")
          print("                              Requiring Service: \n")
          for i in ServReq:
            print("                              Room No: ",i[0],"\n")
          served_q=input("                      Press Y once served")
          if served_q.lower()=='y':
```

```
                    print("Served")
                    served_query=f"update roomdb_annanagar set room_service_status='N';"
                    cur.execute(served_query)

            if branchRoomServ==2:
                queryServ="select room_no from roomdb_annanagar where room_service_status='Y'"
                cur.execute(queryServ)
                ServReq=cur.fetchall()

                if len(ServReq)==0:
                    print("None Requiring service at the moment.")
                print("Requiring Service: ")
                for i in ServReq:
                    print("Room No: ",i[0])
                served_q=input("Press Y once served")
                if served_q.lower()=='y':
                    print("Served")
                    served_query=f"update roomdb_annagar set room_service_status='N';"
                    cur.execute(served_query)
            if branchRoomServ==3:
                EmployeePortalFunc()
def Financial_Records_UI():
    mgmt_action_two=int(input("""
                                            FINANCIAL RECORDS UI
                                            ---------------------\n
                                            [1]All Branches
                                            [2]Anna Nagar
                                            [3]Moggapair\n
                                            """))
    if mgmt_action_two==1:
        mgmt_action_three=int(input("""
                                            [1]CUMULATIVE RECORDS[ALL BRANCHES]
                                            [2]COMPLETE DATA SET[ALL BRANCHES]\n"""))
        if mgmt_action_three==1:
            execQuery=f"select count(room_no),sum(price)from financialRecords_stays;"
            cur.execute(execQuery)
            cumulativeRecords=cur.fetchall()
            for i in cumulativeRecords:
                table=tabulate([i],headers=["No.Of Bookings","Total[in Rs.]"])
                print(table)
        if mgmt_action_three==2:
            execQuery=f"select*from financialRecords_stays;"
            cur.execute(execQuery)
            dataset=cur.fetchall()
            for i in dataset:
                print("Room No: ",i[0]," |Tier:",i[1]," |Days Occupied:",i[2], " |Price: ",i[3]," |Branch:",i[4])
    if mgmt_action_two==2:
        mgmt_action_three=int(input("""
                                            [1]CUMULATIVE RECORDS[ANNA NAGAR]
                                            [2]COMPLETE DATA SET[ANNA NAGAR]\n"""))
        if mgmt_action_three==1:
            execQuery=f"select count(room_no),sum(price)from financialRecords_stays where branch='anna nagar';"
            cur.execute(execQuery)
            cumulativeRecords=cur.fetchall()
            for i in cumulativeRecords:
                table=tabulate([i],headers=["No.Of Bookings","Total[in Rs.]"])
                print(table)
        if mgmt_action_three==2:
            execQuery=f"select*from financialRecords_stays where branch='anna nagar';"
```

```
                cur.execute(execQuery)

                dataset=cur.fetchall()

                for i in dataset:

                    print("Room No: ",i[0]," |Tier:",i[1]," |Days Occupied:",i[2], " |Price: ",i[3]," |Branch:",i[4])


        if mgmt_action_two==3:

            mgmt_action_three=int(input("""

                                        [1]CUMULATIVE RECORDS[MOGAPPAIR]

                                        [2]COMPLETE DATA SET[MOGAPPAIR]\n"""))

            if mgmt_action_three==1:

                execQuery=f"select count(room_no),sum(price)from financialRecords_stays where 
branch='mogappair';"

                cur.execute(execQuery)

                cumulativeRecords=cur.fetchall()

                for i in cumulativeRecords:

                    table=tabulate([i],headers=["No.Of Bookings","Total[in Rs.]"])

                    print(table)

            if mgmt_action_three==2:

                execQuery=f"select*from financialRecords_stays where branch='mogappair';"

                cur.execute(execQuery)

                dataset=cur.fetchall()

                for i in dataset:

                    print("Room No: ",i[0]," |Tier:",i[1]," |Days Occupied:",i[2], " |Price: ",i[3]," |Branch:",i[4])










    def EmployeePortalFunc():
```

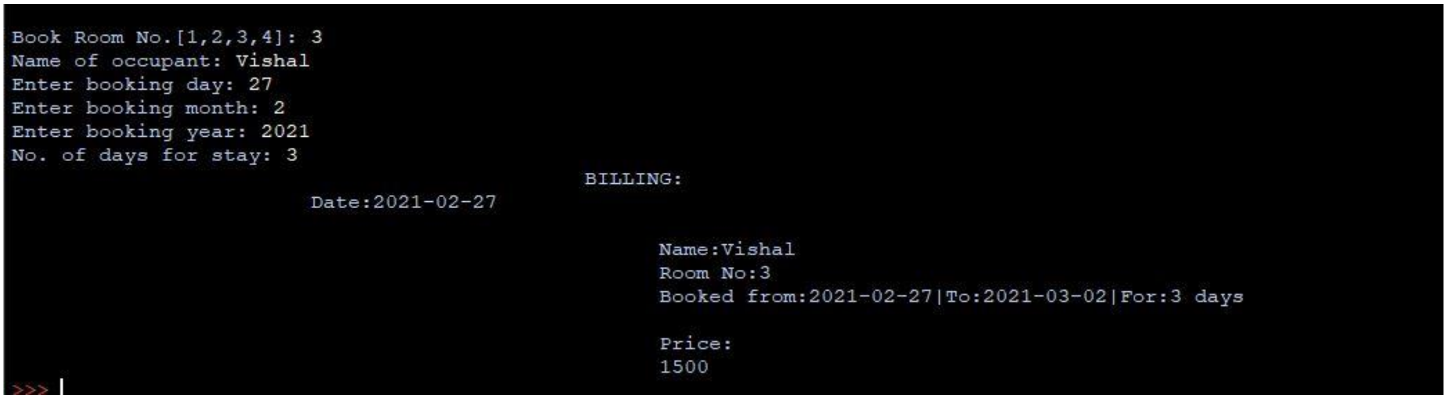
```
ch=int(input("""
            [1]Room Services:
            [2]Reception:
            [3]Higher Mgmt:
            [4]Back"""))

if ch==1:
    roomServices()



if ch==2:
    reception_action=int(input("""
                                  [1]Check room occupancy
                                  [2]Booking Details[Offical Purposes]
                                  [3]Check In
                                  [4]Checkout
                                  [5]Booking Details[For Print][Offical Purposes]
                                  [6]Back"""))
    if reception_action==1:
        checkRoomOccupancy_RecpAction()
    if reception_action==2:
        bookingDetails_branch()
    if reception_action==3:
        bookRoom()
    if reception_action==4:
        checkout()
    if reception_action==5:
        bookingDetails_ForPrint("emp")
    if reception_action==6:
```

```
                EmployeePortalFunc()




        if ch==3:
            mgmtPassword=input("INPUT PASSWORD[MGMT CLEARANCE]")
            if mgmtPassword=="mgmt123":
                mgmt_action=int(input("""

                                    [1]Booking Details All branches[For Print]
                                    [2]Open Financial Records
                                    [3]Back
                                    """))
                if mgmt_action==1:
                    bookingDetails_ForPrint("Mgmt")
                if mgmt_action==2:
                    Financial_Records_UI()



                if mgmt_action==3:
                    EmployeePortalFunc()
        if ch==4:
            LODGING_MAIN_UI()






option=int(input('''
        -----------------------------
```

```
             ||LODGING-MANAGEMENT SYSTEM||
             ----------------------------\n
             [1]Book Room:



             [2]Employee Portal[Employee Access]:



             [3]EXIT'''))
    if option==1:
        bookRoom()
    if option==2:
          EmpPassword=input("Enter password[Emp. Access: ")
          if EmpPassword=="emp123":
              EmployeePortalFunc()
    if option==3:
        exit()




    conn.commit()
    cur.close()
    conn.close()

LODGING_MAIN_UI()
```

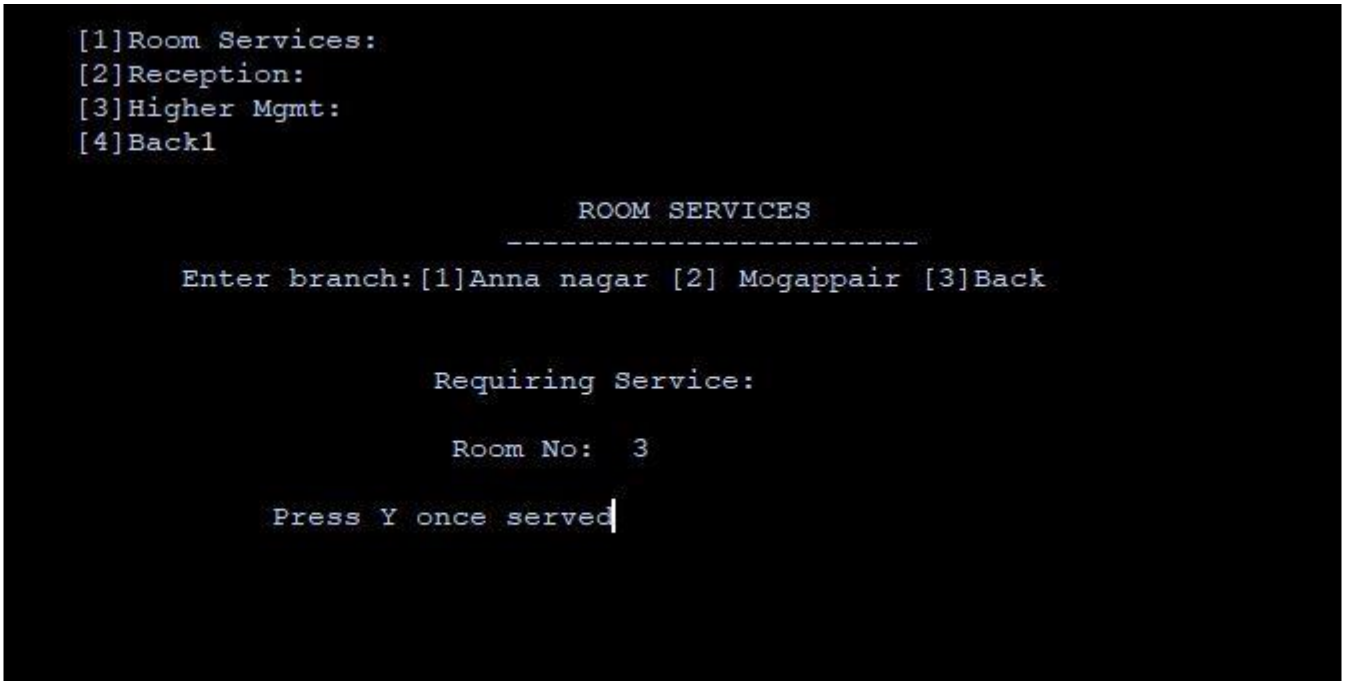
OUTPUT:

Room Booking:



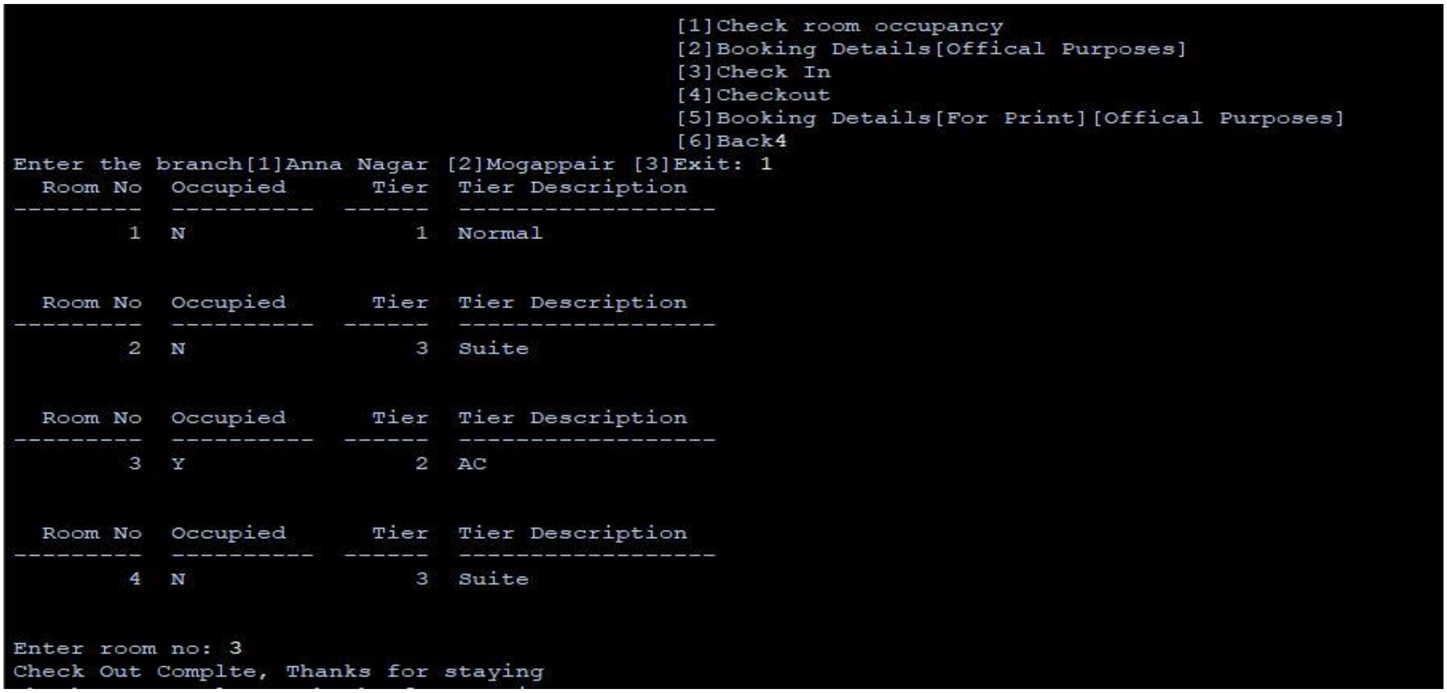
```
Book Room No.[1,2,3,4]: 3
Name of occupant: Vishal
Enter booking day: 27
Enter booking month: 2
Enter booking year: 2021
No. of days for stay: 3
                                              BILLING:
                        Date:2021-02-27

                                                    Name:Vishal
                                                    Room No:3
                                                    Booked from:2021-02-27|To:2021-03-02|For:3 days

                                                    Price:
                                                    1500
>>> 
```

Room Services:



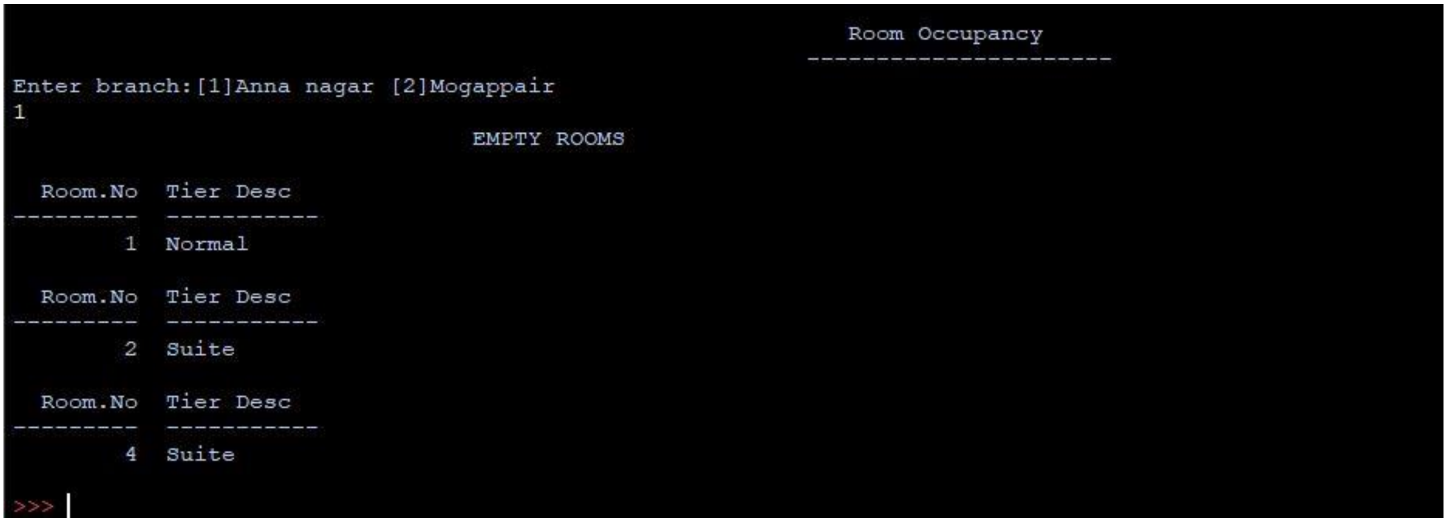
```
    [1]Room Services:
    [2]Reception:
    [3]Higher Mgmt:
    [4]Back1

                                ROOM SERVICES
                            -----------------------
          Enter branch:[1]Anna nagar [2] Mogappair [3]Back


                        Requiring Service:

                         Room No:  3

               Press Y once served
```

Checkout Window:



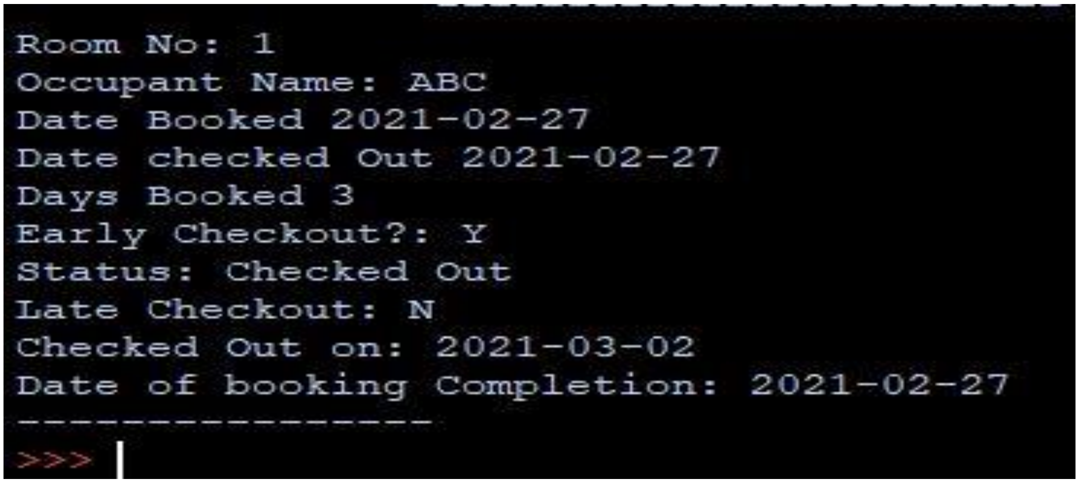
```
                                                             [1]Check room occupancy
                                                             [2]Booking Details[Offical Purposes]
                                                             [3]Check In
                                                             [4]Checkout
                                                             [5]Booking Details[For Print][Offical Purposes]
                                                             [6]Back4
Enter the branch[1]Anna Nagar [2]Mogappair [3]Exit: 1
  Room No  Occupied      Tier  Tier Description
---------  ----------  ------  ------------------
        1  N                1  Normal


  Room No  Occupied      Tier  Tier Description
---------  ----------  ------  ------------------
        2  N                3  Suite


  Room No  Occupied      Tier  Tier Description
---------  ----------  ------  ------------------
        3  Y                2  AC


  Room No  Occupied      Tier  Tier Description
---------  ----------  ------  ------------------
        4  N                3  Suite


Enter room no: 3
Check Out Complte, Thanks for staying
```

Room Occupancy Check:



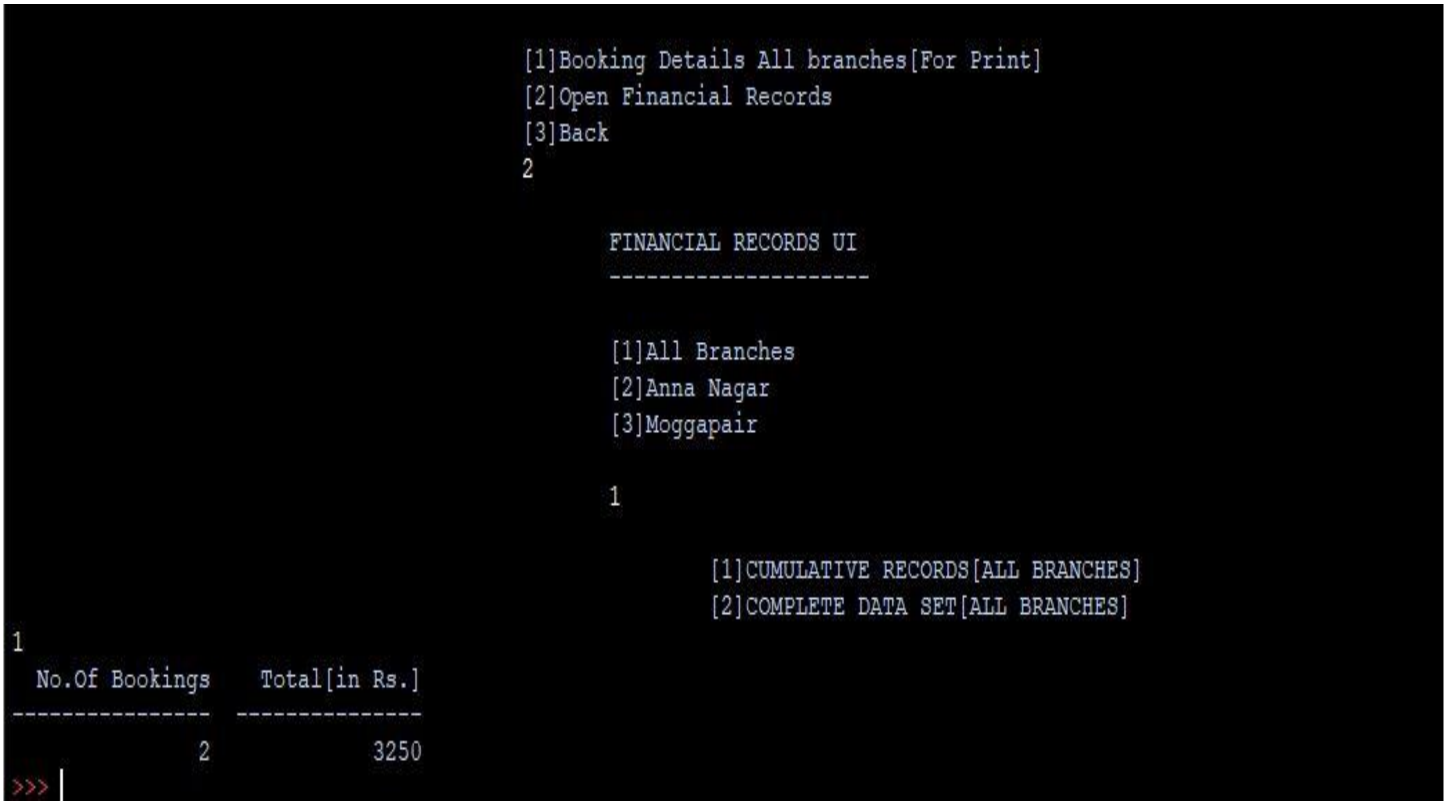
```
                                                                  Room Occupancy
                                                               ----------------------
Enter branch:[1]Anna nagar [2]Mogappair
1
                                EMPTY ROOMS

  Room.No  Tier Desc
---------  -----------
        1  Normal

  Room.No  Tier Desc
---------  -----------
        2  Suite

  Room.No  Tier Desc
---------  -----------
        4  Suite

>>> 
```

Booking Details:



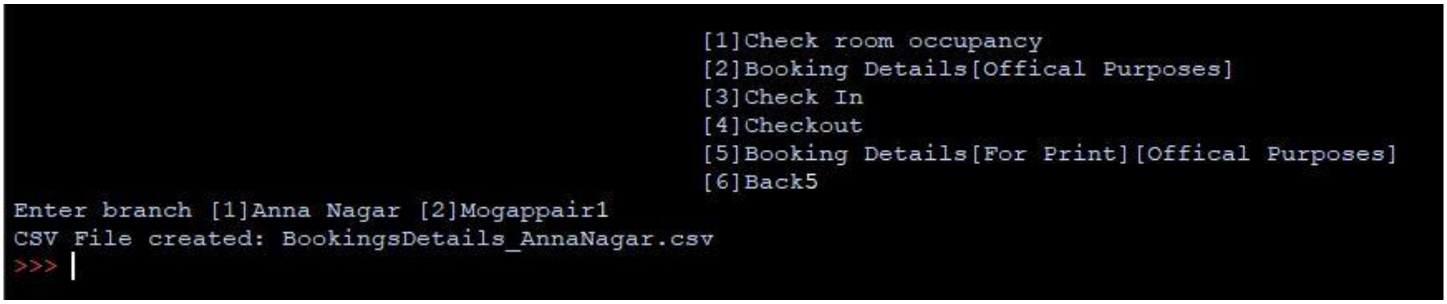
```
Room No: 1
Occupant Name: ABC
Date Booked 2021-02-27
Date checked Out 2021-02-27
Days Booked 3
Early Checkout?: Y
Status: Checked Out
Late Checkout: N
Checked Out on: 2021-03-02
Date of booking Completion: 2021-02-27
----------------
>>> 
```

Financial Records:



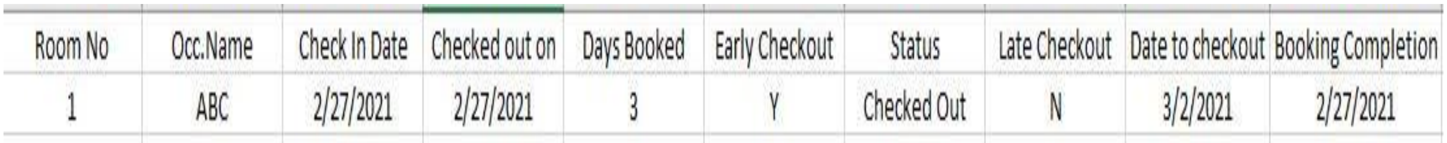
```
                                        [1]Booking Details All branches[For Print]
                                        [2]Open Financial Records
                                        [3]Back
                                        2

                                               FINANCIAL RECORDS UI
                                               --------------------

                                               [1]All Branches
                                               [2]Anna Nagar
                                               [3]Moggapair

                                               1

                                                       [1]CUMULATIVE RECORDS[ALL BRANCHES]
                                                       [2]COMPLETE DATA SET[ALL BRANCHES]
1
  No.Of Bookings   Total[in Rs.]
---------------- --------------
               2           3250
>>> 
```

CSV Files OUTPUT:



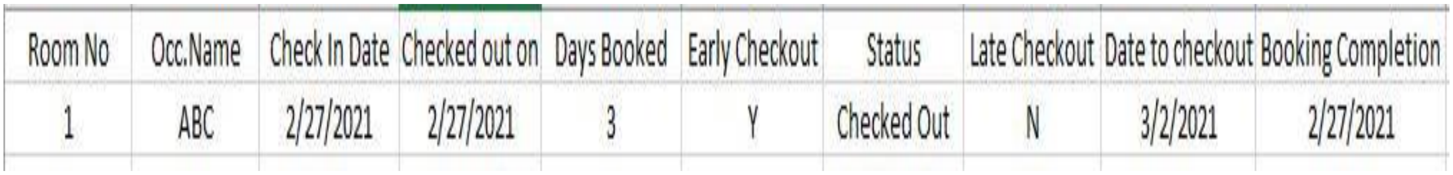
```
                                              [1]Check room occupancy
                                              [2]Booking Details[Offical Purposes]
                                              [3]Check In
                                              [4]Checkout
                                              [5]Booking Details[For Print][Offical Purposes]
                                              [6]Back5
Enter branch [1]Anna Nagar [2]Mogappair1
CSV File created: BookingsDetails_AnnaNagar.csv
>>> |
```

ANNA NAGAR(Above example):



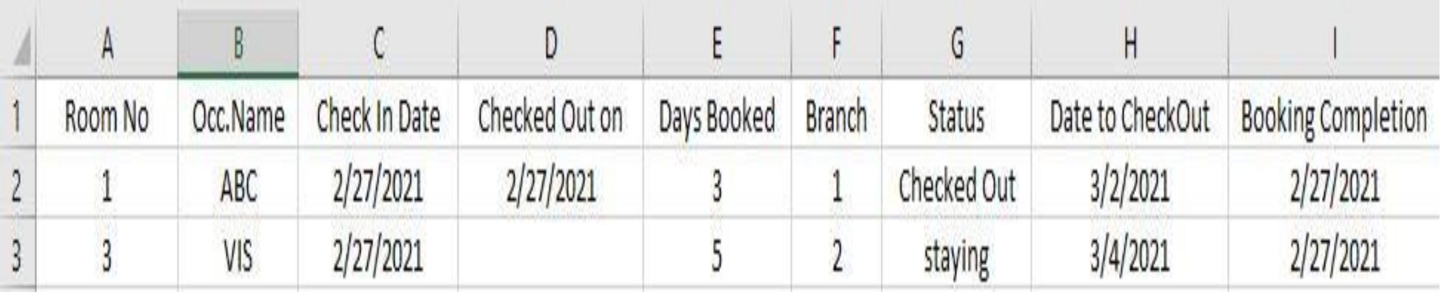
Room No	Occ.Name	Check In Date	Checked out on	Days Booked	Early Checkout	Status	Late Checkout	Date to checkout	Booking Completion
1	ABC	2/27/2021	2/27/2021	3	Y	Checked Out	N	3/2/2021	2/27/2021

MOGAPPAIR:



Room No	Occ.Name	Check In Date	Checked out on	Days Booked	Early Checkout	Status	Late Checkout	Date to checkout	Booking Completion
1	ABC	2/27/2021	2/27/2021	3	Y	Checked Out	N	3/2/2021	2/27/2021

ALL BRANCHES:



	A	B	C	D	E	F	G	H	I
1	Room No	Occ.Name	Check In Date	Checked Out on	Days Booked	Branch	Status	Date to CheckOut	Booking Completion
2	1	ABC	2/27/2021	2/27/2021	3	1	Checked Out	3/2/2021	2/27/2021
3	3	VIS	2/27/2021		5	2	staying	3/4/2021	2/27/2021