

Table of Contents

<u>S. No.</u>	<u>Content</u>	<u>Page no.</u>
1.	Certificate	3
2.	Acknowledgment	4
3.	Introduction (Definition ,Objectives)	5
4.	Features of Python & MySQL	6-7
5.	System Requirements	8
6.	Database Design/ Files for the Project	9
7.	Source Code	10
8.	Screenshot of outputs	11-12
9.	Impact of the Project	13
10.	Bibliography/References	14

Introduction

This project is developed to automate the functionalities of a The purpose of the software project is to develop the Management Information System (MIS) to automate the record keeping in digital form with a view to enhance the decision making of the functionaries.

A MIS mainly consists of a computerized database, a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application program or front-end, we can store, retrieve and manage all information in proper way.

This software, being simple in design and working, does not require much of training to users, and can be used as a powerful tool for automating a **HOTEL MANAGEMENT SYSTEM**.

During coding and design of the software Project, Python, a powerful tool is used apart from this a powerful, open source RDBMS, My SQL is used as per requirement of the CBSE curriculum of COMPUTER SCIENCE Course.

Feature of Python & MySQL

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming.

There are many features in Python, some of which are discussed below –

1. **Easy to code:** Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.
2. **Free and Open Source:** Python language is freely available at official website and you can download it from the given download link below click on the Download Python keyword.
3. **Object-Oriented Language:** One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.
4. **GUI Programming Support:** Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python.
5. **High-Level Language:** Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.
6. **Extensible feature:** Python is a Extensible language. we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.
7. **Python is Portable language:** Python language is also a portable language. for example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.
8. **Python is Integrated language:** Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.
9. **Interpreted Language:** Python is an Interpreted Language. because python code is executed line by line at a time. like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

10. Large Standard Library: Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

11. Dynamically Typed Language: Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance. because of this feature we don't need to specify the type of variable.

The features of MySQL are as follows:

1. Ease of Management – The software very easily gets downloaded and also uses an event scheduler to schedule the tasks automatically.
2. Robust Transactional Support – Holds the ACID (Atomicity, Consistency, Isolation, Durability) property, and also allows distributed multi-version support.
3. Comprehensive Application Development – MySQL has plugin libraries to embed the database into any application. It also supports stored procedures, triggers, functions, views and many more for application development.
4. High Performance – Provides fast load utilities with distinct memory caches and table index partitioning.
5. Low Total Cost Of Ownership – This reduces licensing costs and hardware expenditures.
6. Open Source & 24 * 7 Support – This RDBMS can be used on any platform and offers 24*7 supports for open source and enterprise edition.
7. Secure Data Protection – MySQL supports powerful mechanisms to ensure that only authorized users have access to the databases.
8. High Availability – MySQL can run high-speed master/slave replication configurations and it offers cluster servers.
9. Scalability & Flexibility – With MySQL you can run deeply embedded applications and create data warehouses holding a humongous amount of data.

System Requirement

Recommended System Requirements

Processors: Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAM Intel® Xeon® processor E5-2698 v3 at 2.30 GHz (2 sockets, 16 cores each, 1 thread per core), 64 GB of DRAM Intel® Xeon Phi™ processor 7210 at 1.30 GHz (1 socket, 64 cores, 4 threads per core), 32 GB of DRAM, 16 GB of MCDRAM (flat mode enabled)

Disk space: 2 to 3 GB

Operating systems: Windows® 10, macOS*, and Linux*

Minimum System Requirements

Processors: Intel Atom® processor or Intel® Core™ i3 processor

Disk space: 1 GB

Operating systems: Windows* 7 or later, macOS, and Linux

Python* versions: 2.7.X, 3.6.X

Recommended System Requirements (My SQL)

CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMDCPU.

Cores: Single (Dual/Quad Core is recommended)

RAM: 4 GB (6 GB recommended)

Graphic Accelerators: nVidia or ATI with support of OpenGL 1.5 or higher.

Display Resolution: 1280×1024 is recommended, 1024×768 is minimum.

File Requirement:

PATH: - C:\Users\USER\Desktop\ip project report\

Database Design/ Files for the Project

MODULE USED:

1. random module:

- Purpose: The random module in Python provides a suite of functions that allow for generating random numbers and performing random operations. In this code, it is used to generate unique random values for:
 - Room Numbers (rn): This ensures that each room assigned to a customer has a unique room number.
 - Customer IDs (cid): Similar to room numbers, a random customer ID is generated to uniquely identify each customer.
- Functions used in the code:
 - `random.randint(a, b)`:
 - This function returns a random integer between a (inclusive) and b (inclusive).
 - In the code:
 - `rn = random.randint(100, 999)` generates a random integer between 100 and 999 for the room number.
 - `cid = random.randint(10, 99)` generates a random integer between 10 and 99 for the customer ID.
 - The while loop ensures that the generated room number and customer ID are unique by checking if the values already exist in the respective lists (roomno and custid). If they do exist, it keeps generating new random values until it finds a unique one.

2. datetime module:

- Purpose: The datetime module in Python is used for manipulating dates and times. It allows you to create datetime objects and perform operations such as comparisons, date arithmetic, formatting, and validation. In this code, it serves two primary purposes:
 - Validating Dates: It is used to check if the user inputs for the check-in and check-out dates are valid dates.

- Calculating the Number of Days: It is used to calculate the duration of a customer's stay by finding the difference between check-in and check-out dates.
- Functions used in the code:
 - `datetime.datetime(year, month, day)`:
 - This function creates a datetime object based on the provided year, month, and day values. It is used to represent a specific point in time.
 - In the code:
 - `datetime.datetime(c[2], c[1], c[0])` is used to create a datetime object from the list of integers representing the date (in the format [dd, mm, yyyy]).
 - This helps in verifying if a date is valid or not (in the `date()` function) and is also used for comparing check-in and check-out dates to ensure the check-out date is later than the check-in date.
 - `datetime.timedelta(days)`:
 - This class is used to represent the difference between two dates or times.
 - In the code:
 - `d1 = datetime.datetime(ci[2], ci[1], ci[0])` and `d2 = datetime.datetime(co[2], co[1], co[0])` create datetime objects for the check-in and check-out dates.
 - `d = (d2 - d1).days` calculates the difference between these two datetime objects and gives the number of days the customer will stay at the hotel.
- Date Validation:
 - In the `date()` function, the `datetime.datetime(c[2], c[1], c[0])` is used within a try-except block to attempt creating a datetime object for the user-entered date. If the date is invalid (for example, if the day is out of range or the month is invalid), the `ValueError` will be raised, and an error message is displayed.
- Date Comparison:

- The code also ensures that the check-out date is later than the check-in date by comparing two datetime objects (d1 for check-in and d2 for check-out). If d2 (check-out) is not later than d1 (check-in), an error message is displayed, prompting the user to enter a valid check-out date.

Example Walkthrough:

1. Room Number and Customer ID Generation:

- When a new customer books a room, the code generates a random room number and customer ID.

python

Copy code

```
rn = random.randint(100, 999)
```

```
cid = random.randint(10, 99)
```

This generates a random number for the room and customer, ensuring they are unique by checking if the generated number is already in use.

2. Date Validation and Comparison:

- A user enters the check-in date as 01/02/2025. The code attempts to convert it into a datetime object:

python

Copy code

```
ci = list(map(int, ci.split('/')))
```

```
d1 = datetime.datetime(ci[2], ci[1], ci[0])
```

This will result in a datetime object representing 01/02/2025. If the user enters an invalid date, such as 31/02/2025, the ValueError will be raised, prompting the user to re-enter the date.

- For the check-out date, the code ensures it is after the check-in date:

python

Copy code

```
if datetime.datetime(co[2], co[1], co[0]) > datetime.datetime(ci[2], ci[1], ci[0]):
```

This condition ensures that the check-out date (e.g., 05/02/2025) is after the check-in date (e.g., 01/02/2025). If the check-out date is invalid (earlier than the check-in), the user is asked to re-enter a valid date.

3. Stay Duration Calculation:

- The number of days the customer will stay at the hotel is calculated using:

python

Copy code

```
d = (d2 - d1).days
```

If d1 is 01/02/2025 and d2 is 05/02/2025, the result will be 4 days. This value is stored in the day list to calculate the total room charge based on the number of days.

VARIABLE USED:

- **Global Lists:**

- name: Stores customer names.
- phno: Stores customer phone numbers.
- add: Stores customer addresses.
- checkin: Stores check-in dates.
- checkout: Stores check-out dates.
- room: Stores the type of room booked by the customer.
- price: Stores the price of the room.
- rc: Stores restaurant charges for each customer.
- p: Stores payment status (0 for pending, 1 for paid).
- roomno: Stores the room numbers assigned to customers.
- custid: Stores customer IDs.
- day: Stores the number of days of stay.

- **Global Variables:**

- i: A counter variable, initialized to 0, but not used explicitly in the code.

USER DEFINED FUNCTION USED:

User-Defined Functions:

1. **Home()**: The home function that displays the main menu for the hotel and directs the user to various functionalities based on the user's input.
2. **date(c)**: Validates the date by checking if the date is correct or not, and handles any errors in date format.
3. **Booking()**: Manages the booking process, including entering customer details, selecting check-in and check-out dates, room type, and generating a unique room number and customer ID.
4. **Rooms_Info()**: Displays the room types and their prices.
5. **restaurant()**: Handles restaurant orders and charges for a customer.
6. **Payment()**: Handles payment processing for a customer and updates the payment status.
7. **Record()**: Displays customer records, including their names, room numbers, and payment status.

LIBRARY FUNCTION USED:

Library Functions Used:

1. `input()`: A built-in function used to take user input from the console.
2. `print()`: A built-in function used to display output on the console.
3. `int()`: A built-in function used to convert a string input to an integer.
4. `list()`: A built-in function used to convert a string (date input) to a list of integers.
5. `map()`: A built-in function used to apply the `int()` function to each element of the input string (date input).
6. `random.randint()`: A function from the `random` module that generates a random integer between two specified numbers (used for room number and customer ID generation).
7. `datetime.datetime()`: A function from the `datetime` module that creates a `datetime` object from year, month, and day values, used for date validation and calculations.
8. `str.lower()`: A string method used to convert the user's input to lowercase (for confirming payment).

OUTPUT SCREENS

Main Page :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

      1 Booking
      2 Rooms Info
      3 Room Service(Menu Card)
      4 Payment
      5 Record
      0 Exit

->1

BOOKING ROOMS

Name: Aditya
Phone No.: 123456789
Address: Sitamarhi
```

Booking Info :

```
----SELECT ROOM TYPE----
1. Standard Non-AC
2. Standard AC
3. 3-Bed Non-AC
4. 3-Bed AC

      Press 0 for Room Prices

->1
Room Type- Standard Non-AC
Price- 3500

      ***ROOM BOOKED SUCCESSFULLY***

Room No. - 336
Customer Id - 23
0-BACK
->
```

Room Info :

----- HOTEL ROOMS INFO -----	
STANDARD NON-AC	

Room amenities include: 1 Double Bed, Television, Telephone, Double-Door Cupboard, 1 Coffee table with 2 sofa, Balcony and an attached washroom with hot/cold water.	
STANDARD NON-AC	

Room amenities include: 1 Double Bed, Television, Telephone, Double-Door Cupboard, 1 Coffee table with 2 sofa, Balcony and an attached washroom with hot/cold water + Window/Split AC.	
3-Bed NON-AC	

Room amenities include: 1 Double Bed + 1 Single Bed, Television, Telephone, a Triple-Door Cupboard, 1 Coffee table with 2 sofa, 1 side table, Balcony with an Accent table with 2 Chair and an attached washroom with hot/cold water.	
3-Bed AC	

Room amenities include: 1 Double Bed + 1 Single Bed, Television, Telephone, a Triple-Door Cupboard, 1 Coffee table with 2 sofa, 1 Side table, Balcony with an Accent table with 2 Chair and an attached washroom with hot/cold water + Window/Split AC.	

Room Services (Menu Card):

Customer Id: 23	

Hotel AnCasa	

Menu Card	

BEVARAGES	

1 Regular Tea..... 20.00	26 Dal Fry..... 140.00
2 Masala Tea..... 25.00	27 Dal Makhani..... 150.00
3 Coffee..... 25.00	28 Dal Tadka..... 150.00
4 Cold Drink..... 25.00	
5 Bread Butter..... 30.00	ROTI
6 Bread Jam..... 30.00	-----
7 Veg. Sandwich..... 50.00	29 Plain Roti..... 15.00
8 Veg. Toast Sandwich..... 50.00	30 Butter Roti..... 15.00
9 Cheese Toast Sandwich... 70.00	31 Tandoori Roti..... 20.00
10 Grilled Sandwich..... 70.00	32 Butter Naan..... 20.00
	RICE

SOUPS	33 Plain Rice..... 90.00

11 Tomato Soup..... 110.00	34 Jeera Rice..... 90.00
12 Hot & Sour..... 110.00	35 Veg Pulao..... 110.00
13 Veg. Noodle Soup..... 110.00	36 Peas Pulao..... 110.00
14 Sweet Corn..... 110.00	
15 Veg. Munchow..... 110.00	SOUTH INDIAN

MAIN COURSE	37 Plain Dosa..... 100.00

16 Shahi Paneer..... 110.00	38 Onion Dosa..... 110.00
17 Kadai Paneer..... 110.00	
18 Handi Paneer..... 120.00	39 Masala Dosa..... 130.00
19 Palak Paneer..... 120.00	40 Paneer Dosa..... 130.00
20 Chilli Paneer..... 140.00	41 Rice Idli..... 130.00
	42 Sambhar Vada..... 140.00
	ICE CREAM

Payment Method :

```
MODE OF PAYMENT
1- Credit/Debit Card
2- Paytm/PhonePe
3- Using UPI
4- Cash
-> 3

Amount: 10500

Pay For AnCasa
(y/n)
->y

-----
Hotel AnCasa
-----
Bill
-----
Name: Aditya
Phone No.: 123456789
Address: Sitamarhi

Check-In: 12/03/2020
Check-Out: 15/03/2020

Room Type: Standard Non-AC
Room Charges: 10500
Restaurant Charges: 0
-----

Total Amount: 10500
-----

Thank You
Visit Again :)
```

Records :

```
->5

*** HOTEL RECORD ***

| Name | Phone No. | Address | Check-In | Check-Out | Room Type | Price |
-----
| Aditya | 123456789 | Sitamarhi | 12/03/2020 | 15/03/2020 | Standard Non-AC | 3500 |
-----

0-BACK
->|
```

Impact of the Project

This Hotel Management System can have several important impacts:

1. **Improved Efficiency:** Automates room booking, payment tracking, and restaurant billing, reducing human errors and operational workload.
2. **Enhanced Customer Experience:** Customers can easily book rooms, pay, and track their charges, improving satisfaction and reducing confusion.
3. **Financial Management:** Ensures accurate billing and tracking of payments, helping the hotel manage revenue and reduce losses.
4. **Scalability:** The system can be expanded with new features like room availability, loyalty programs, or integration with online platforms.
5. **Cost Reduction:** Automating key tasks reduces manual work, saving time and operational costs.
6. **Data Insights:** Collects data on customer preferences and trends, which can be used for marketing and service improvements.
7. **Time-Saving:** Automation of tasks like room bookings, check-in/check-out, and payment tracking saves time for both hotel staff and customers.
8. **Error Reduction:** With automatic room allocation and payment status tracking, the chances of errors are minimized.
9. **Data Security:** Customer data and payment details can be securely stored and managed, preventing mishandling of sensitive information.
10. **Better Decision Making:** By tracking customer preferences and stay patterns, the hotel can make informed decisions on pricing and services.
11. **Customer Retention:** A smooth booking and payment process increases customer satisfaction, potentially leading to repeat visits.
12. **Business Growth:** With easy scalability, the system can accommodate growing customer numbers, larger inventories, or even multiple hotel branches in the future.

SOURCE CODE

```
import random
import datetime

# Global List Declaration
name = []
phno = []
add = []
checkin = []
checkout = []
room = []
price = []
rc = []
p = []
roomno = []
custid = []
day = []

# Global Variable Declaration

i = 0

# Home Function
def Home():

    print("\t\t\t\t\t WELCOME TO HOTEL ANCASA\n")
    print("\t\t\t 1 Booking\n")
    print("\t\t\t 2 Rooms Info\n")
    print("\t\t\t 3 Room Service(Menu Card)\n")
    print("\t\t\t 4 Payment\n")
    print("\t\t\t 5 Record\n")
    print("\t\t\t 0 Exit\n")

    ch=int(input("->"))

    if ch == 1:
        print(" ")
        Booking()

    elif ch == 2:
        print(" ")
        Rooms_Info()

    elif ch == 3:
```

```
print(" ")
restaurant()
```

```
elif ch == 4:
    print(" ")
    Payment()
```

```
elif ch == 5:
    print(" ")
    Record()
```

```
else:
    exit()
```

Function used in booking

```
def date(c):
```

```
    if c[2] >= 2019 and c[2] <= 2020:
```

```
        if c[1] != 0 and c[1] <= 12:
```

```
            if c[1] == 2 and c[0] != 0 and c[0] <= 31:
```

```
                if c[2]%4 == 0 and c[0] <= 29:
                    pass
```

```
                elif c[0]<29:
                    pass
```

```
            else:
                print("Invalid date\n")
                name.pop(i)
                phno.pop(i)
                add.pop(i)
                checkin.pop(i)
                checkout.pop(i)
                Booking()
```

```
        # if month is odd & less than equal
        # to 7th month
```

```
        elif c[1] <= 7 and c[1]%2 != 0 and c[0] <= 31:
            pass
```

```
        # if month is even & less than equal to 7th
        # month and not 2nd month
```

```
elif c[1] <= 7 and c[1]%2 == 0 and c[0] <= 30 and c[1] != 2:  
    pass
```

```
# if month is even & greater than equal  
# to 8th month  
elif c[1] >= 8 and c[1]%2 == 0 and c[0] <= 31:  
    pass
```

```
# if month is odd & greater than equal  
# to 8th month  
elif c[1]>=8 and c[1]%2!=0 and c[0]<=30:  
    pass
```

```
else:  
    print("Invalid date\n")  
    name.pop(i)  
    phno.pop(i)  
    add.pop(i)  
    checkin.pop(i)  
    checkout.pop(i)  
    Booking()
```

```
else:  
    print("Invalid date\n")  
    name.pop(i)  
    phno.pop(i)  
    add.pop(i)  
    checkin.pop(i)  
    checkout.pop(i)  
    Booking()
```

```
else:  
    print("Invalid date\n")  
    name.pop(i)  
    phno.pop(i)  
    add.pop(i)  
    checkin.pop(i)  
    checkout.pop(i)  
    Booking()
```

```
# Booking function  
def Booking():
```

```
    # used global keyword to  
    # use global variable 'i'  
    global i  
    print(" BOOKING ROOMS")
```

```

print(" ")

while 1:
    n = str(input("Name: "))
    p1 = str(input("Phone No.: "))
    a = str(input("Address: "))

    # checks if any field is not empty
    if n!=" " and p1!=" " and a!=" ":
        name.append(n)
        add.append(a)
        break

    else:
        print("\tName, Phone no. & Address cannot be empty...!!")

cii=str(input("Check-In: "))
checkin.append(cii)
cii=cii.split('/')
ci=cii
ci[0]=int(ci[0])
ci[1]=int(ci[1])
ci[2]=int(ci[2])
date(ci)

coo=str(input("Check-Out: "))
checkout.append(coo)
coo=coo.split('/')
co=coo
co[0]=int(co[0])
co[1]=int(co[1])
co[2]=int(co[2])

# checks if check-out date falls after
# check-in date
if co[1]<ci[1] and co[2]<ci[2]:

    print("\n\tErr...!!\n\tCheck-Out date must fall after Check-In\n")
    name.pop(i)
    add.pop(i)
    checkin.pop(i)
    checkout.pop(i)
    Booking()
elif co[1]==ci[1] and co[2]>=ci[2] and co[0]<=ci[0]:

    print("\n\tErr...!!\n\tCheck-Out date must fall after Check-In\n")
    name.pop(i)

```

```

    add.pop(i)
    checkin.pop(i)
    checkout.pop(i)
    Booking()
else:
    pass

date(co)
d1 = datetime.datetime(ci[2],ci[1],ci[0])
d2 = datetime.datetime(co[2],co[1],co[0])
d = (d2-d1).days
day.append(d)

print("----SELECT ROOM TYPE----")
print(" 1. Standard Non-AC")
print(" 2. Standard AC")
print(" 3. 3-Bed Non-AC")
print(" 4. 3-Bed AC")
print(("\\t\\tPress 0 for Room Prices"))

ch=int(input("->"))

# if-conditions to display allotted room
# type and it's price
if ch==0:
    print(" 1. Standard Non-AC - Rs. 3500")
    print(" 2. Standard AC - Rs. 4000")
    print(" 3. 3-Bed Non-AC - Rs. 4500")
    print(" 4. 3-Bed AC - Rs. 5000")
    ch=int(input("->"))
if ch==1:
    room.append('Standard Non-AC')
    print("Room Type- Standard Non-AC")
    price.append(3500)
    print("Price- 3500")
elif ch==2:
    room.append('Standard AC')
    print("Room Type- Standard AC")
    price.append(4000)
    print("Price- 4000")
elif ch==3:
    room.append('3-Bed Non-AC')
    print("Room Type- 3-Bed Non-AC")
    price.append(4500)
    print("Price- 4500")
elif ch==4:
    room.append('3-Bed AC')

```

```

    print("Room Type- 3-Bed AC")
    price.append(5000)
    print("Price- 5000")
else:
    print(" Wrong choice..!!")

# randomly generating room no. and customer
# id for customer
rn = random.randrange(40)+300
cid = random.randrange(40)+10

# checks if allotted room no. & customer
# id already not allotted
while rn in roomno or cid in custid:
    rn = random.randrange(60)+300
    cid = random.randrange(60)+10

rc.append(0)
p.append(0)

if p1 not in phno:
    phno.append(p1)
elif p1 in phno:
    for n in range(0,i):
        if p1== phno[n]:
            if p[n]==1:
                phno.append(p1)
elif p1 in phno:
    for n in range(0,i):
        if p1== phno[n]:
            if p[n]==0:
                print("\tPhone no. already exists and payment yet not done..!!")
                name.pop(i)
                add.pop(i)
                checkin.pop(i)
                checkout.pop(i)
                Booking()

print("")
print("\t\t\t\t***ROOM BOOKED SUCCESSFULLY***\n")
print("Room No. - ",rn)
print("Customer Id - ",cid)
roomno.append(rn)
custid.append(cid)
i=i+1
n=int(input("0-BACK\n->"))
if n==0:

```



```
    Home()
```

```
else:
```

```
    exit()
```

ROOMS INFO

```
def Rooms_Info():
```

```
    print("    ----- HOTEL ROOMS INFO -----")
```

```
    print("")
```

```
    print("STANDARD NON-AC")
```

```
    print("-----")
```

```
    print("Room amenities include: 1 Double Bed, Television, Telephone,")
```

```
    print("Double-Door Cupboard, 1 Coffee table with 2 sofa, Balcony and")
```

```
    print("an attached washroom with hot/cold water.\n")
```

```
    print("STANDARD NON-AC")
```

```
    print("-----")
```

```
    print("Room amenities include: 1 Double Bed, Television, Telephone,")
```

```
    print("Double-Door Cupboard, 1 Coffee table with 2 sofa, Balcony and")
```

```
    print("an attached washroom with hot/cold water + Window/Split AC.\n")
```

```
    print("3-Bed NON-AC")
```

```
    print("-----")
```

```
    print("Room amenities include: 1 Double Bed + 1 Single Bed, Television,")
```

```
    print("Telephone, a Triple-Door Cupboard, 1 Coffee table with 2 sofa, 1")
```

```
    print("Side table, Balcony with an Accent table with 2 Chair and an")
```

```
    print("attached washroom with hot/cold water.\n")
```

```
    print("3-Bed AC")
```

```
    print("-----")
```

```
    print("Room amenities include: 1 Double Bed + 1 Single Bed, Television,")
```

```
    print("Telephone, a Triple-Door Cupboard, 1 Coffee table with 2 sofa, ")
```

```
    print("1 Side table, Balcony with an Accent table with 2 Chair and an")
```

```
    print("attached washroom with hot/cold water + Window/Split AC.\n\n")
```

```
    print()
```

```
    n=int(input("0-BACK\n->"))
```

```
    if n==0:
```

```
        Home()
```

```
    else:
```

```
        exit()
```

RESTAURANT FUNCTION

```
def restaurant():
```

```
    ph=int(input("Customer Id: "))
```

```
    global i
```

```
    f=0
```

```
    r=0
```

```
    for n in range(0,i):
```

```
        if custid[n]==ph and p[n]==0:
```

```
            f=1
```

```
            print("-----")
```

```

print("          Hotel AnCasa")
print("-----")
print("          Menu Card")
print("-----")
print("\n BEVARAGES          26 Dal Fry..... 140.00")
print("----- 27 Dal Makhani..... 150.00")
print(" 1 Regular Tea..... 20.00 28 Dal Tadka..... 150.00")
print(" 2 Masala Tea..... 25.00")
print(" 3 Coffee..... 25.00  ROTI")
print(" 4 Cold Drink..... 25.00  -----")
print(" 5 Bread Butter..... 30.00 29 Plain Roti..... 15.00")
print(" 6 Bread Jam..... 30.00 30 Butter Roti..... 15.00")
print(" 7 Veg. Sandwich..... 50.00 31 Tandoori Roti..... 20.00")
print(" 8 Veg. Toast Sandwich..... 50.00 32 Butter Naan..... 20.00")
print(" 9 Cheese Toast Sandwich... 70.00")
print(" 10 Grilled Sandwich..... 70.00  RICE")
print("          -----")
print(" SOUPS          33 Plain Rice..... 90.00")
print("----- 34 Jeera Rice..... 90.00")
print(" 11 Tomato Soup..... 110.00 35 Veg Pulao..... 110.00")
print(" 12 Hot & Sour..... 110.00 36 Peas Pulao..... 110.00")
print(" 13 Veg. Noodle Soup..... 110.00")
print(" 14 Sweet Corn..... 110.00  SOUTH INDIAN")
print(" 15 Veg. Munchow..... 110.00  -----")
print("          37 Plain Dosa..... 100.00")
print(" MAIN COURSE          38 Onion Dosa..... 110.00")
print("----- 39 Masala Dosa..... 130.00")
print(" 16 Shahi Paneer..... 110.00 40 Paneer Dosa..... 130.00")
print(" 17 Kadai Paneer..... 110.00 41 Rice Idli..... 130.00")
print(" 18 Handi Paneer..... 120.00 42 Sambhar Vada..... 140.00")
print(" 19 Palak Paneer..... 120.00")
print(" 20 Chilli Paneer..... 140.00  ICE CREAM")
print(" 21 Matar Mushroom..... 140.00  -----")
print(" 22 Mix Veg..... 140.00 43 Vanilla..... 60.00")
print(" 23 Jeera Aloo..... 140.00 44 Strawberry..... 60.00")
print(" 24 Malai Kofta..... 140.00 45 Pineapple..... 60.00")
print(" 25 Aloo Matar..... 140.00 46 Butter Scotch..... 60.00")
print("Press 0 -to end ")
ch=1
while(ch!=0):

    ch=int(input(" -> "))

    # if-elif-conditions to assign item
    # prices listed in menu card
    if ch==1 or ch==31 or ch==32:
        rs=20

```

```

    r=r+rs
elif ch<=4 and ch>=2:
    rs=25
    r=r+rs
elif ch<=6 and ch>=5:
    rs=30
    r=r+rs
elif ch<=8 and ch>=7:
    rs=50
    r=r+rs
elif ch<=10 and ch>=9:
    rs=70
    r=r+rs
elif (ch<=17 and ch>=11) or ch==35 or ch==36 or ch==38:
    rs=110
    r=r+rs
elif ch<=19 and ch>=18:
    rs=120
    r=r+rs
elif (ch<=26 and ch>=20) or ch==42:
    rs=140
    r=r+rs
elif ch<=28 and ch>=27:
    rs=150
    r=r+rs
elif ch<=30 and ch>=29:
    rs=15
    r=r+rs
elif ch==33 or ch==34:
    rs=90
    r=r+rs
elif ch==37:
    rs=100
    r=r+rs
elif ch<=41 and ch>=39:
    rs=130
    r=r+rs
elif ch<=46 and ch>=43:
    rs=60
    r=r+rs
elif ch==0:
    pass
else:
    print("Wrong Choice..!!")
print("Total Bill: ",r)

```

updates restaurant charges and then

```

        # appends in 'rc' list
        r=r+rc.pop(n)
        rc.append(r)
    else:
        pass
if f == 0:
    print("Invalid Customer Id")
n=int(input("0-BACK\n->"))
if n==0:
    Home()
else:
    exit()

```

PAYMENT FUNCTION

```

def Payment():

    ph=str(input("Phone Number: "))
    global i
    f=0

    for n in range(0,i):
        if ph==phno[n] :

            # checks if payment is
            # not already done
            if p[n]==0:
                f=1
                print(" Payment")
                print(" -----")
                print(" MODE OF PAYMENT")

                print(" 1- Credit/Debit Card")
                print(" 2- Paytm/PhonePe")
                print(" 3- Using UPI")
                print(" 4- Cash")
                x=int(input("-> "))
                print("\n Amount: ",(price[n]*day[n])+rc[n])
                print("\n    Pay For AnCasa")
                print(" (y/n)")
                ch=str(input("->"))

                if ch=='y' or ch=='Y':
                    print("\n\n -----")
                    print("    Hotel AnCasa")
                    print(" -----")
                    print("        Bill")

```

```

print(" -----")
print(" Name: ",name[n],"\\t\\n Phone No.: ",phno[n],"\\t\\n Address: ",add[n],"\\t")
print("\\n Check-In: ",checkin[n],"\\t\\n Check-Out: ",checkout[n],"\\t")
print("\\n Room Type: ",room[n],"\\t\\n Room Charges: ",price[n]*day[n],"\\t")
print(" Restaurant Charges: \\t",rc[n])
print(" -----")
print("\\n Total Amount: ",(price[n]*day[n])+rc[n],"\\t")
print(" -----")
print("    Thank You")
print("    Visit Again :)")
print(" -----\\n")
p.pop(n)
p.insert(n,1)

```

```

# pops room no. and customer id from list and
# later assigns zero at same position
roomno.pop(n)
custid.pop(n)
roomno.insert(n,0)
custid.insert(n,0)

```

else:

```

for j in range(n+1,i):
    if ph==phno[j] :
        if p[j]==0:
            pass

        else:
            f=1
            print("\\n\\tPayment has been Made :)\\n\\n")

```

```

if f==0:
    print("Invalid Customer Id")

```

```

n = int(input("0-BACK\\n ->"))
if n == 0:
    Home()
else:
    exit()

```

RECORD FUNCTION

```
def Record():
```

```

# checks if any record exists or not
if phno!=[]:
    print(" *** HOTEL RECORD ***\\n")
    print("| Name   | Phone No. | Address   | Check-In | Check-Out | Room Type   | Price   |")

```

```

        print("-----")
    -")

    for n in range(0,i):
        print("|",name[n],"\t
|",phno[n],"\t",add[n],"\t",checkin[n],"\t",checkout[n],"\t",room[n],"\t",price[n])

    print("-----")
    -")

    else:
        print("No Records Found")
        n = int(input("0-BACK\n->"))
        if n == 0:
            Home()

    else:
        exit()

# Driver Code
Home()

```

Bibliography/Reference

In order to work on this project, the following books and literature are referred by me during the various phases of development of the project:

Reference Book:

- Computer Science by CBSE & NCERT,
- SUMITA ARORA,
- PREETI ARORA
- "Python for Data Analysis" by Wes McKinney
- "Learning Python" by Mark Lutz

Reference Websites:

- www.python.org
- www.w3schools.com
- www.mysql.com
- www.mysqltutorial.org
- [GreeksforGreeks](http://www.greekforGreeks.com)