PYTHON FULL STACK DEVELOPER COURSE

PYTHON PROJECT

HOTEL MANAGEMENT SYSTEM

NAME: SINI. M

PUMO TECHNOVATION INDIA PRIVATE LIMITED

PROJECT TITLE SHEET

PYTHON Hotel Management System

Project Report Submitted

In partial fulfillment of the requirement for the

proficientcertificate course

Done By

SINI .M

Under the guidance of

SHAKTHI

Approved by

BALAMURUGAN T S

PUMO TECHNOVATION
INDIA PRIVATE LIMITED

ABSTRACT

PYTHON PROJECT

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

OOPS:

In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming. The main concept of OOPs is to bind the data and the functions that work on that together as a single unit so that no other part of the code can access this data.

OOPs Concepts in Python:

Class

Objects

Polymorphism

Encapsulation

Inheritance

Data Abstraction

CLASS:

A class is a collection of objects. A class contains the blueprints or the prototype from which the objects are being created. It is a logical entity that contains some attributes and methods.

SYNTAX:

```
class ClassName:
  # Statement-1
  .
  .
  .
  # Statement-N
```

OBJECT:

Objects are instances of a class created with specifically defined data. Objects can correspond to real-world objects or an abstract entity.

EXCEPTION HANDLING:

An exception in Python is an incident that happens while executing a program that causes the regular course of the program's commands to be disrupted. When a Python code comes across a condition it can't handle, it raises an exception. An object in Python that describes an error is called an exception.

SYNTAX:

```
try:

   # statement(s)

except IndexError:

   # statement(s)

except ValueError:

   # statement(s)
```

MYSQL DATABASE:

Python MySQL Connector is a Python driver that helps to integrate Python and MySQL. This Python MySQL library allows the conversion between Python and MySQL data types. MySQL Connector API is implemented using pure Python and does not require any third-party library.

SYNTAX:

```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="yourusername",
  password="yourpassword"
)

print(mydb)
```

WHILE LOOP WITH BREAK STATEMENT:

Python while loop executes a set of statements in a loop based on a condition. But, in addition to the standard breaking of loop when this while condition evaluates to false, you can also break the while loop using builtin Python break statement.

SYNTAX:

```
#statement(s)
while condition :
   #statement(s)
   if break_condition :
      break
   #statement(s)
```

SOURCE CODE:

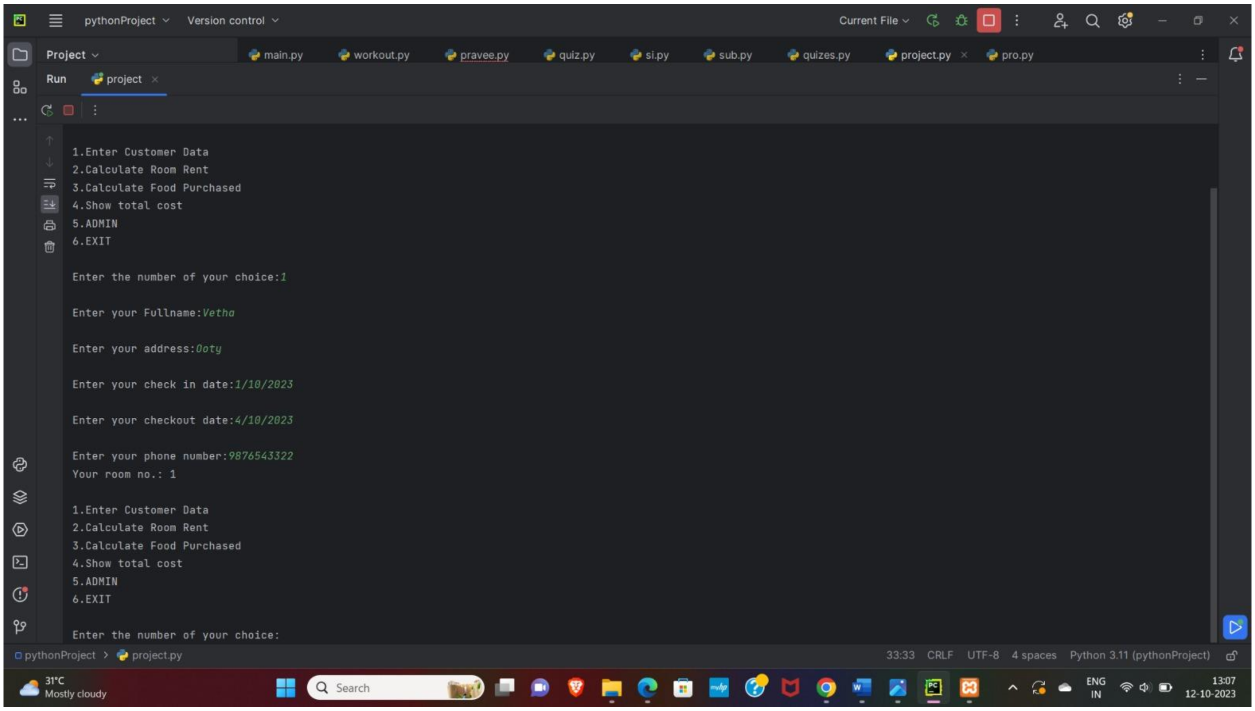
SOURCE CODE

```
import mysql.connector
def establish_database_connection():
   try:
      connection = mysql.connector.connect(
         host="localhost",
         user="root",
         password=" ",
         database="project"
      )
      return connection
```

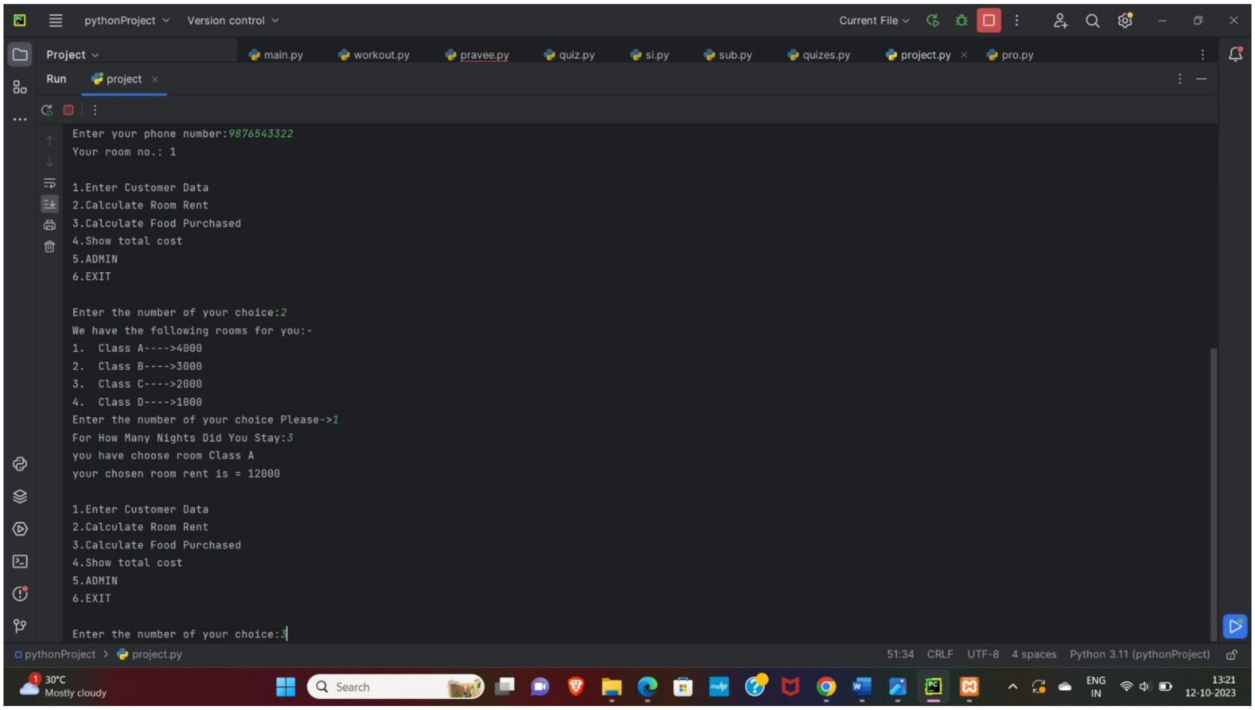
```
    except mysql.connector.Error as err:
        print(f"Error: {err}")
        return None

class hotelmanage:

    def _init_(self, rt=", s=0, p=0, r=0, t=0, a=1000, name=", address=",
cindate=", coutdate=",phoneno=", rno=1):

        print("\n\n**WELCOME TO HOTEl DIANA**\n")

        self.rt = rt

        self.r = r

        self.t = t

        self.p = p

        self.s = s
        self.a = a
        self.name = name
        self.address = address
        self.cindate = cindate
        self.coutdate = coutdate
        self.phoneno= phoneno
        self.rno = rno

    def inputdata(self):
        connection = establish_database_connection()
        if connection is None:
            return
        cursor=connection.cursor()
        self.name = input("\nEnter your Fullname:")
        self.address = input("\nEnter your address:")
        self.cindate = input("\nEnter your check in date:")
        self.coutdate = input("\nEnter your checkout date:")
        self.phoneno= input("\nEnter your phone number:")
        print("Your room no.:", self.rno, "\n")
        sql="insert into
hotelmanagement(name,address,cindate,coutdate,phoneno)
values(%s,%s,%s,%s,%s)"
        data=(self.name,self.address,self.cindate,self.coutdate,self.phoneno)
        cursor.execute(sql,data)
        connection.commit()

    def roomrent(self):  # sel1353

        print("We have the following rooms for you:-")

        print("1.  Class A---->4000")

        print("2.  Class B---->3000")

        print("3.  Class C---->2000")

        print("4.  Class D---->1000")

        x = int(input("Enter the number of your choice Please->"))

        n = int(input("For How Many Nights Did You Stay:"))

        if (x == 1):

            print("you have choose room Class A")

            self.s = 4000 * n

        elif (x == 2):
            print("you have choose room Class B")

            self.s = 3000 * n

        elif (x == 3):

            print("you have choose room Class C")

            self.s = 2000 * n

        elif (x == 4):
            print("you have choose room Class D")

            self.s = 1000 * n

        else:

            print("please choose a room")

        print("your chosen room rent is =", self.s, "\n")

    def foodpurchased(self):
        print("**RESTAURANT MENU**")

        connection = establish_database_connection()
        if connection is None:
            return
        cursor = connection.cursor()

        menu = ["Dessert", "Cool Drinks", "Breakfast", "Lunch", "Dinner"]

        print("1. Dessert -----> 100", "2. Cool Drinks -----> 50", "3. Breakfast
-----> 90", "4. Lunch -----> 110",
              "5. Dinner -----> 150", "6. Exit")
        self.r = 0

        while True:
            c = int(input("Enter the number of your choice:"))

            if c == 6:
                break

            if c < 1 or c > 5:
                print("You've entered an invalid key")
                continue

            d = int(input("Enter the quantity:"))

            if (c == 1):
                cost = 100 * d
            elif (c == 2):
                cost = 50 * d
            elif (c == 3):
                cost = 90 * d
            elif (c == 4):
                cost = 110 * d
            elif (c == 5):
                cost = 150 * d

            self.r += cost

            cursor.execute("INSERT INTO fooditems
(customer_name,item_name, quantity, total_cost) VALUES (%s, %s, %s,
%s)",
                           (self.name,menu[c - 1], d, cost))
            connection.commit()

        print("Total food Cost = Rs", self.r, "\n")
```

```
    cursor.close()
    connection.close()

def display(self):
    print("***HOTEL BILL***")
    print("Customer details:")
    print("Customer name:", self.name)
    print("Customer address:", self.address)
    print("Check in date:", self.cindate)
    print("Check out date", self.coutdate)
    print("Customer phone number",self.phoneno)
    print("Room no.", self.rno)
    print("Your Room rent is:", self.s)
    print("Your Food bill is:", self.r)

    self.rt = self.s + self.t + self.p + self.r

    print("Your sub total Purchased is:", self.rt)
    print("Additional Service Charges is", self.a)
    print("Your Grand total Purchased is:", self.rt + self.a, "\n")
    self.rno += 1

def isadmin(self):
    connection = establish_database_connection()
    if connection is None:
        return
    cursor = connection.cursor()

    user = ["admin","admin1","admin2"]
    user_id=input("\nEnter the userid:")

    if (user):
        print(f"{user} is an admin.")
    else:
        print(f"{user} is not an admin.")
        sql = "select * from hotelmanagement"
        cursor.execute(sql)
        myresult = cursor.fetchall()
        for row in myresult:
            print(row)

        connection.close()

def main():
    a = hotelmanage()

    while (1):
        print("1.Enter Customer Data")

        print("2.Calculate Room Rent")

        print("3.Calculate Food Purchased")

        print("4.Show total cost")

        print("5.ADMIN")

        print("6.EXIT")

        b = int(input("\nEnter the number of your choice:"))
        if (b == 1):
            a.inputdata()

        if (b == 2):
            a.roomrent()

        if (b == 3):
            a.foodpurchased()

        if (b == 4):
            a.display()

        if (b == 5):
            a.isadmin()

        if (b == 6):
            quit()
if _name_ == "_main_":
    menu = ["Dessert", "Cool Drinks", "Breakfast", "Lunch", "Dinner"]
    r = 0

main()
```

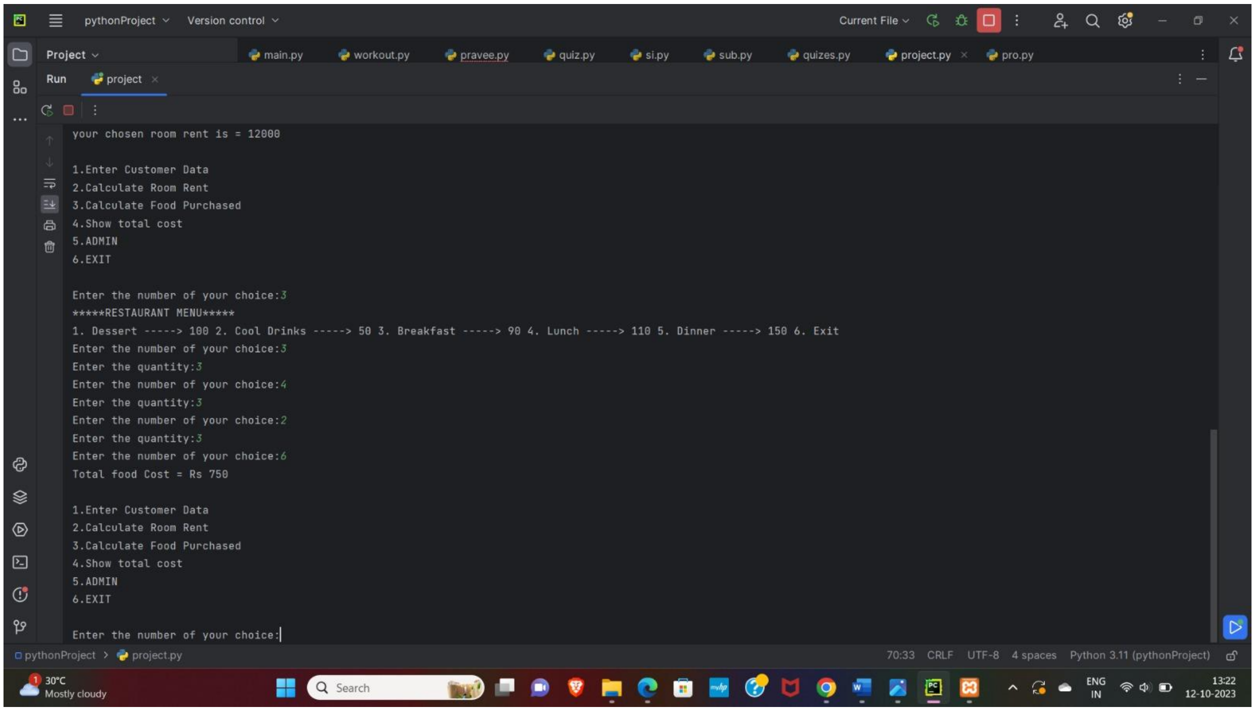
OUTPUT:



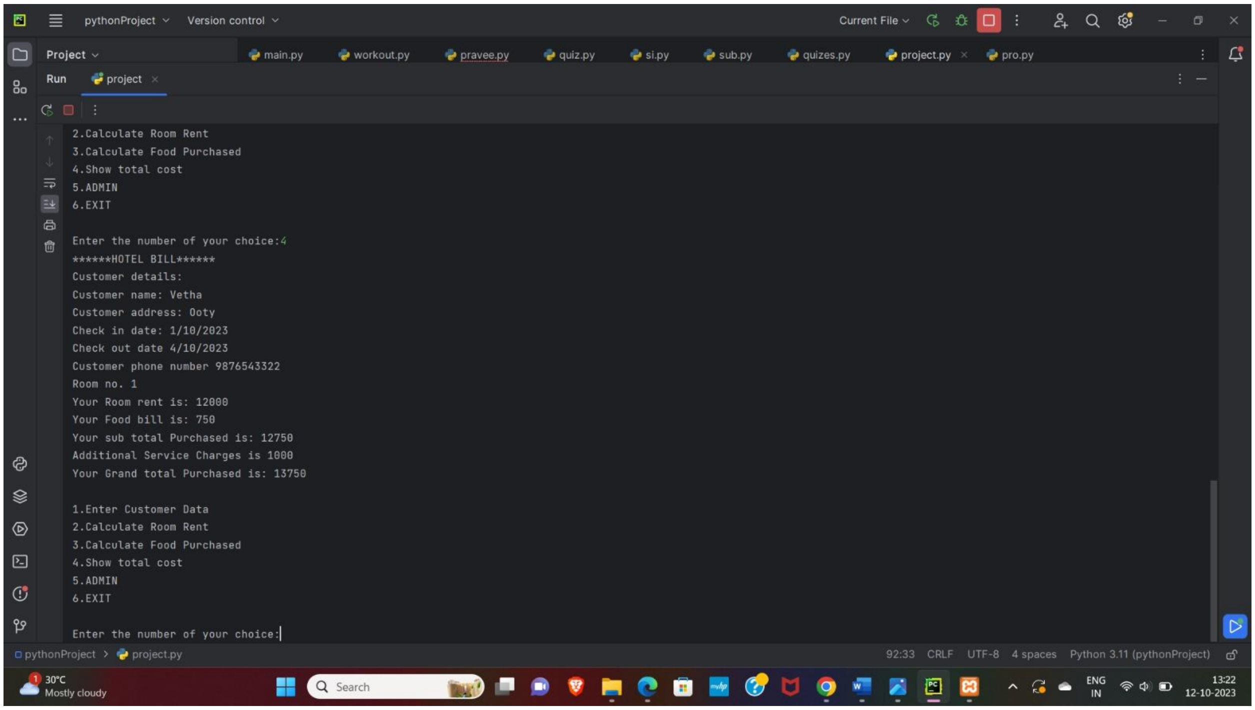
```
1.Enter Customer Data
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:1

Enter your Fullname:Vetha

Enter your address:Ooty

Enter your check in date:1/10/2023

Enter your checkout date:4/10/2023

Enter your phone number:9876543322
Your room no.: 1

1.Enter Customer Data
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:
```



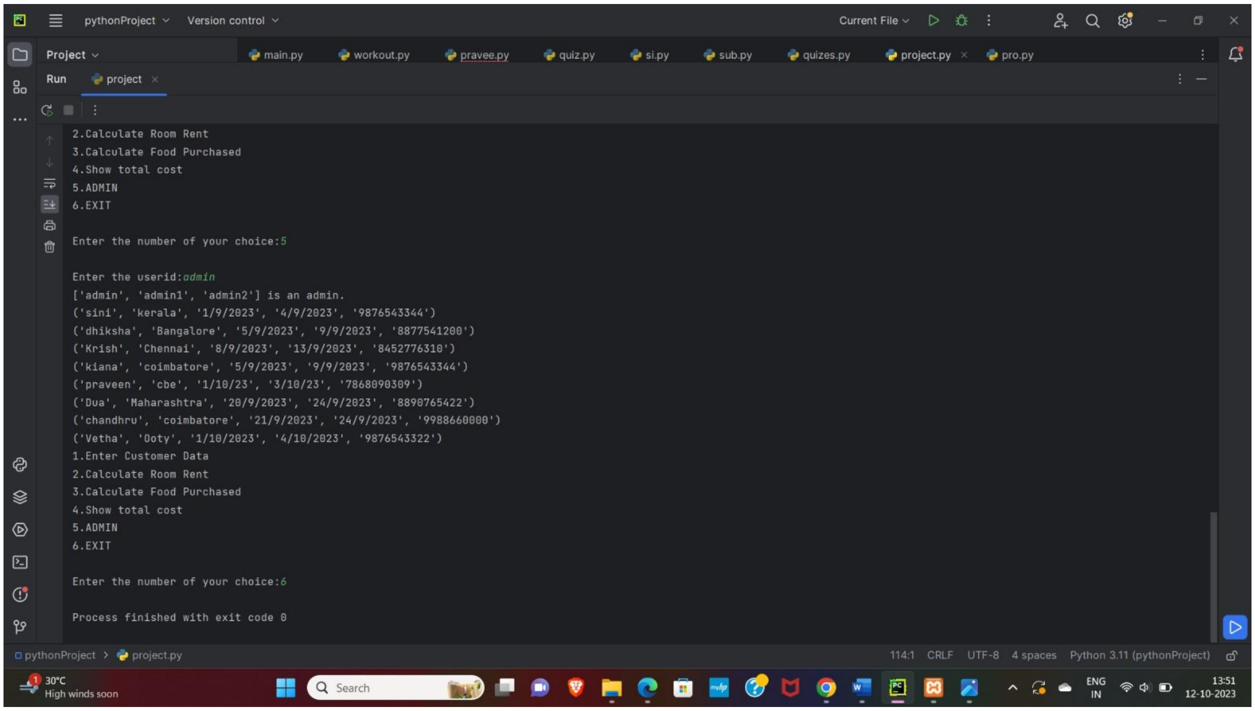
```
Enter your phone number:9876543322
Your room no.: 1

1.Enter Customer Data
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:2
We have the following rooms for you:-
1.  Class A---->4000
2.  Class B---->3000
3.  Class C---->2000
4.  Class D---->1000
Enter the number of your choice Please->1
For How Many Nights Did You Stay:3
you have choose room Class A
your chosen room rent is = 12000

1.Enter Customer Data
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:3
```



```
your chosen room rent is = 12000

1.Enter Customer Data
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:3
*****RESTAURANT MENU*****
1. Dessert -----> 100 2. Cool Drinks -----> 50 3. Breakfast -----> 90 4. Lunch -----> 110 5. Dinner -----> 150 6. Exit
Enter the number of your choice:3
Enter the quantity:3
Enter the number of your choice:4
Enter the quantity:3
Enter the number of your choice:2
Enter the quantity:3
Enter the number of your choice:6
Total food Cost = Rs 750

1.Enter Customer Data
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:
```



```
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:4
******HOTEL BILL******
Customer details:
Customer name: Vetha
Customer address: Ooty
Check in date: 1/10/2023
Check out date 4/10/2023
Customer phone number 9876543322
Room no. 1
Your Room rent is: 12000
Your Food bill is: 750
Your sub total Purchased is: 12750
Additional Service Charges is 1000
Your Grand total Purchased is: 13750

1.Enter Customer Data
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:
```



```
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:5

Enter the userid:admin
['admin', 'admin1', 'admin2'] is an admin.
('sini', 'kerala', '1/9/2023', '4/9/2023', '9876543344')
('dhiksha', 'Bangalore', '5/9/2023', '9/9/2023', '8877541200')
('Krish', 'Chennai', '8/9/2023', '13/9/2023', '8452776310')
('kiana', 'coimbatore', '5/9/2023', '9/9/2023', '9876543344')
('praveen', 'cbe', '1/10/23', '3/10/23', '7868090309')
('Dua', 'Maharashtra', '20/9/2023', '24/9/2023', '8890765422')
('chandhru', 'coimbatore', '21/9/2023', '24/9/2023', '9988660000')
('Vetha', 'Ooty', '1/10/2023', '4/10/2023', '9876543322')
1.Enter Customer Data
2.Calculate Room Rent
3.Calculate Food Purchased
4.Show total cost
5.ADMIN
6.EXIT

Enter the number of your choice:6

Process finished with exit code 0
```

CONCLUSION:

In conclusion, the "Hotel Diana Management System" project has successfully met its primary objectives of efficiently managing guest information, room reservations, and food orders. The project's implementation of object-oriented programming (OOP) concepts, integration with a MySQL database, and robust exception handling has yielded a powerful and reliable system for our hotel.