INTRODUCTION

This system is developed to automate day to day activity of a restaurant. Restaurant is a kind of business that serves people all over world with ready-made food. This system is developed to provide service facility to restaurant and also to the customer. This restaurant management system can be used by employees in a restaurant to handle the clients, their orders and can help them easily find free tables or place orders. The services that are provided is food ordering and reservation table management by the customer through the system online, customer information management and waiter information management, menu information management and report

Restaurant management system is the system for managing the restaurant business. After successful login the customer can access the menu page with the items listed according to the desired time. The main point of developing this system is to help restaurant administrator manage the restaurant business and help customer for online ordering and reserve table. In proposed system user can search for a menu according to his choice i.e. according to price range and category of food and later he can order a meal.

REQUIREMENTS

➢ HARDWARE

1. System model HP 202 G1 MT.
2. System typex64-based PC.
3. Processor Intel(R) Pentium(R) CPU G2030 @ 3.00GHz,3000 MHz, 2 Core(S), 2 Logical Processor(s).
4. SMBIOS Version 2.7.
5. Installed physical memory (RAM) 2.00GB.
6. Total physical memory 1.89GB.
7. Available physical memory 1.21GB.
8. Total virtual memory 3.78GB.
9. Available virtual memory 2.83GB.

➢ SOFTWARE

1. Python 3.7
2. My Sql 8.0
3. My Sql command line client 8.0
4. Database
5. Database Connector

PROJECT DESCRIPTION

A project, RESTAURANT MANAGEMENT SYSTEM, is quite a very basic program of functioning of a restaurant management .But it displays fundamental programing operations in restaurant management. The programming has been laid down in Python Programming Language

The program has been constructed using class concept, extracting inputs using input functions like int(input()) for numeric inputs and input() for character (string) inputs.

The program asks for the details of the customer (customer name, address, check in date, check out date, restaurant bill, laundry bill, games bill)

If the customer selects food option then it gets input from the user on what type of food he likes to have and asks for the quantity and finally displays the total cost of the food along with the added service charges

If the customer selects laundry option then it gets input from user on what type of cloths he would like to be washed and asks for the quantity and finally displays the total cost along with added service charges

If the customer selects games option then it gets input from user about what games he would like to pa and for how

many hours he would like to play and finally displays the total cost along with added service charges

Then the data is recorded in the restaurant database and The total cost of the restaurant is displayed

This is the actual fundamental process carried out in restaurant management system

MODULE FUNCTION

- Import MYSQL.CONNECTOR: It is used to connect database with python
- From: It is used to get a specific function in the code instead of complete file.

LOOPS

- For loop: It is used to run the program again and again.
- While loop: It is a conditional loop which is used to run the program for particular number of times.

MODULES AND LIBRARIES USED

1. CLASS CONCEPTS

A Class is a logical grouping of data and functions. It gives Freedom to create data structures that contains arbitrary content and hence easily accessible .For example, for any restaurant manager who want to fetch the customer details online would go to customer class, where all its attributes like purchasing details, food and games details, concessions, etc. would be listed out.

2. MYSQL CONNECTOR

MySQL Connector/ODBC (sometimes called just Connector/ODBC or My ODBC) is a driver for connecting to a MySQL database server through the Open Database Connectivity (ODBC) application program interface (API), which is the standard means of connecting to any database. Users can connect from within common applications and programming environments, such as Microsoft Access or Excel My ODBC is available for most major operating systems, including Windows, Unix, Linux, Solaris, AIX.

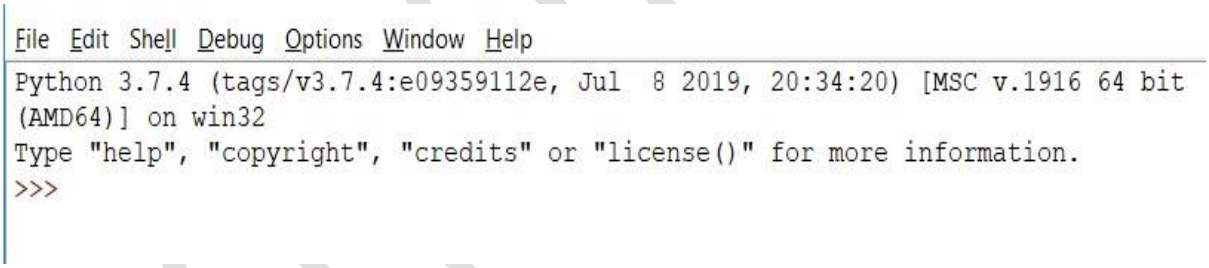
3. DATABASE CONNECTIVITY

A Database connection is a facility in computer science that allows client software to talk to database server software, whether on the same machine or not. A connection is required to send commands and receive answers, usually in the front of a result set. Connections are a key concept in data-centric programming. Since some DBMS engines require considerable time to connect, connection pooling was invented to improve performance. No command can be performed against a database without an "open and available" connection to it. Connections are built by supplying an underlying driver or provider with a connection string, which is a way of addressing a specific database or server and instance as well as user authentication credentials (for example, Server=sql_box; Database=Common; User ID=uid; Pwd=password). Once a connection has been built it can be opened and closed at will, and properties (such as the command time-out-length, or transaction, if one exists) can be set. The Connection String is composed of a set of key/value pairs as dictated by the data acess interface and data provider being used.

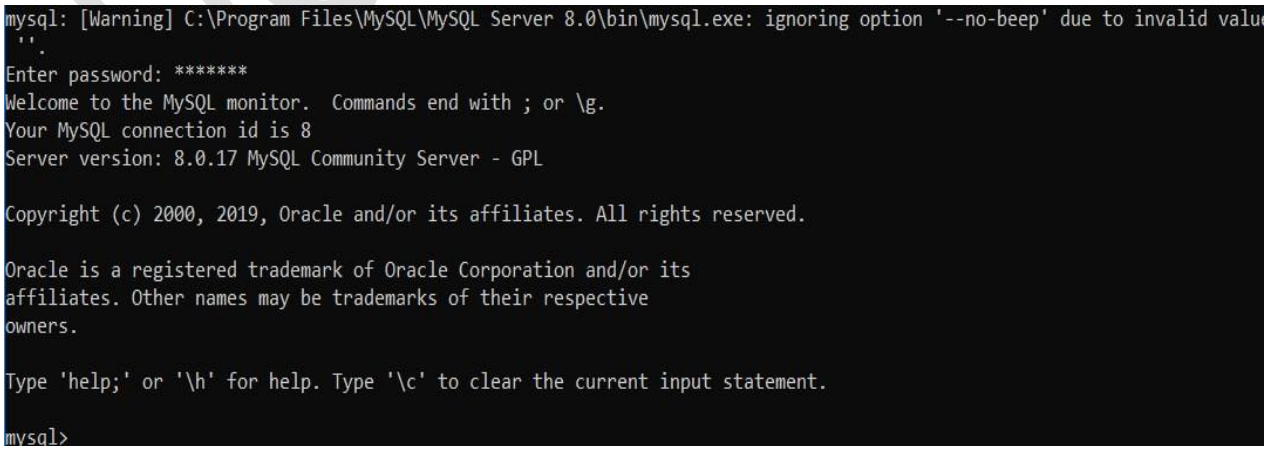
WORKING DESCRIPTIONS

1. Enter Customer Data
2. Calculate room rent
3. Calculate restaurant bill
4. Calculate laundry bill
5. Calculate game bill
6. Show total cost
7. Enter into database
8. Exit

PYTHON SHELL



MYSQL COMMAND LINE CLIENT



SOURCE CODE

```
class hotelprestige:
    def
__init__(self,rt='',s=0,p=0,r=0,t=0,a=1800,name='',address='',cindate='',c
outdate='',rno=50):
        print ("\n\n*****WELCOME TO BHAVISH GROUP OF
RESTAURANTS*****\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.rno=rno
    def inputdata(self):
        self.name=input("\nEnter your name:")
        self.address=input("\nEnter your address:")
        self.cindate=input("\nEnter your check in date:")
        self.coutdate=input("\nEnter your checkout date:")
        print("Your room no.:",self.rno,"\n")
```

```
def roomrent(self):
    print ("We have the following rooms for you:-")
    print ("1.  type A---->rs 6000 PN\-")
    print ("2.  type B---->rs 5000 PN\-")
    print ("3.  type C---->rs 4000 PN\-")
    print ("4.  type D---->rs 3000 PN\-")
    x=int(input("Enter Your Choice Please->"))
    n=int(input("For How Many Nights Did You Stay:"))
    if(x==1):
        print ("you have opted room type A")
        self.s=6000*n
    elif (x==2):
        print ("you have opted room type B")
        self.s=5000*n
    elif (x==3):
        print ("you have opted room type C")
        self.s=4000*n
    elif (x==4):
        print ("you have opted room type D")
        self.s=3000*n
    else:
        print ("please choose a room")
    print ("your room rent is =",self.s,"\n")
 def restaurentbill(self):
    print("*****RESTAURANT MENU*****")
```

```
print("1.water----->Rs20")
print("2.tea----->Rs10")
print("3.breakfast combo--->Rs90")
print("4.lunch---->Rs110")
print("5.dinner--->Rs150")
print("6.Exit")
while (1):
    c=int(input("Enter your choice:"))
    if (c==1):
        d=int(input("Enter the quantity:"))
        self.r=self.r+20*d
    elif (c==2):
         d=int(input("Enter the quantity:"))
         self.r=self.r+10*d
    elif (c==3):
         d=int(input("Enter the quantity:"))
         self.r=self.r+90*d
    elif (c==4):
         d=int(input("Enter the quantity:"))
         self.r=self.r+110*d
    elif (c==5):
         d=int(input("Enter the quantity:"))
         self.r=self.r+150*d
    elif (c==6):
        break;
    else:
```

```
            print("Invalid option")
        print ("Total food Cost=Rs",self.r,"\n")
    def         laundrybill(self):
        print ("******LAUNDRY MENU*******")
        print ("1.Shorts----->Rs3")
        print("2.Trousers----->Rs4")
        print("3.Shirt--->Rs5")
        print("4.Jeans---->Rs6")
        print("5.Girlsuit--->Rs8")
        print("6.Exit")
        while (1):
            e=int(input("Enter your choice:"))
            if (e==1):
                f=int(input("Enter the quantity:"))
                self.t=self.t+3*f
            elif (e==2):
                f=int(input("Enter the quantity:"))
                self.t=self.t+4*f
            elif (e==3):
                f=int(input("Enter the quantity:"))
                self.t=self.t+5*f
            elif (e==4):
                f=int(input("Enter the quantity:"))
                self.t=self.t+6*f
            elif (e==5):
                f=int(input("Enter the quantity:"))
```

```
            self.t=self.t+8*f
        elif (e==6):
            break;
        else:
            print ("Invalid option")
    print ("Total Laundary Cost=Rs",self.t,"\n")
def gamebill(self):
    print ("******GAME MENU*******")
    print ("1.Table tennis----->Rs60")
    print("2.Bowling----->Rs80")
    print("3.Snooker--->Rs70")
    print("4.Video games---->Rs90")
    print("5.Pool--->Rs50")
    print("6.Exit")
    while (1):
        g=int(input("Enter your choice:"))
        if (g==1):
            h=int(input("No. of hours:"))
            self.p=self.p+60*h
        elif (g==2):
            h=int(input("No. of hours:"))
            self.p=self.p+80*h
        elif (g==3):
            h=int(input("No. of hours:"))
            self.p=self.p+70*h
        elif (g==4):
```

```
            h=int(input("No. of hours:"))
            self.p=self.p+90*h
        elif (g==5):
            h=int(input("No. of hours:"))
            self.p=self.p+50*h
        elif (g==6):
            break;
        else:
            print ("Invalid option")
    print ("Total Game Bill=Rs",self.p,"\n")
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 ("Room no.",self.rno)
    print ("Your Room rent is:",self.s)
    print ("Your Food bill is:",self.r)
    print ("Your laundary bill is:",self.t)
    print ("Your Game bill is:",self.p)
    self.rt=self.s+self.t+self.p+self.r
    print ("Your sub total bill is:",self.rt)
    print ("Additional Service Charges is",self.a)
```

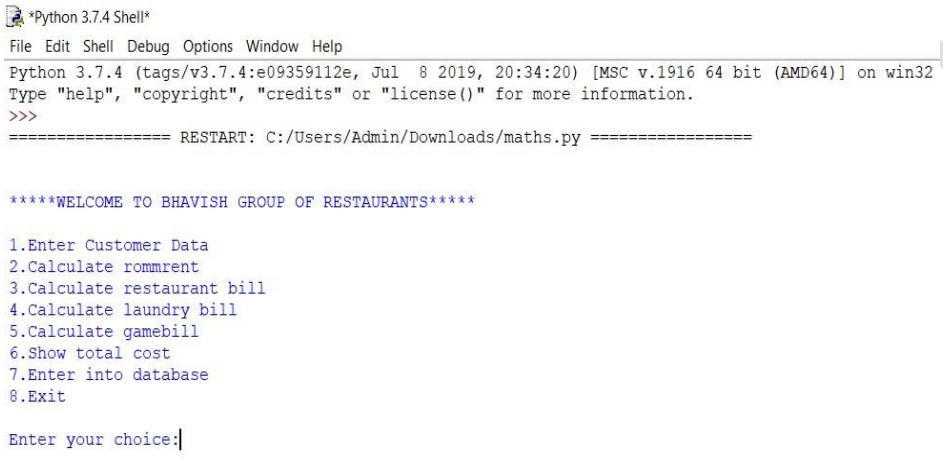
```
        print ("Your grandtotal bill is:",self.rt+self.a,"\n")
    def database(self):
        print("database selected")
        import mysql.connector
        from mysql.connector import Error
        def
insertVariblesIntoTable(name,adress,check_in_date,check_out_date,ro
om_number,rent,restaurant_bill,laundary_bill,games_bill,service_bill,to
tal_bill):
            try:
                connection = mysql.connector.connect(host='localhost',
                                        database='school',
                                        user='root',
                                        password='bhavish')
                cursor = connection.cursor()
                mySql_insert_query = """INSERT INTO
restaurant(name,adress,check_in_date,check_out_date,room_number,
rent,restaurant_bill,laundary_bill,games_bill,service_bill,total_bill)
                        VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
%s) """
                recordTuple =
(name,adress,check_in_date,check_out_date,room_number,rent,restau
rant_bill,laundary_bill,games_bill,service_bill,total_bill)
                cursor.execute(mySql_insert_query, recordTuple)
                connection.commit()
                print("Record inserted successfully into restaurant table")
```

```
            except mysql.connector.Error as error:
                print("Failed to insert into MySQL table {}".format(error))


insertVariblesIntoTable(self.name,self.address,self.cindate,self.coutdate
,self.rno,self.s,self.r,self.t,self.p,self.a,self.rt)
def main():
    a=hotelprestige()
    while (1):
        print("1.Enter Customer Data")
        print("2.Calculate rommrent")
        print("3.Calculate restaurant bill")
        print("4.Calculate laundry bill")
        print("5.Calculate gamebill")
        print("6.Show total cost")
        print("7.Enter into database")
        print("8.Exit")
        b=int(input("\nEnter your choice:"))
        if (b==1):
            a.inputdata()
        if (b==2):
            a.roomrent()
        if (b==3):
            a.restaurentbill()
        if (b==4):
            a.laundrybill()
        if (b==5):
```

```
        a.gamebill()
    if (b==6):
        a.display()
    if (b==7):
        a.database()
    if (b==8):
        quit()
main()
```

OUTPUT SCREENS



CHOICE 1

```
1.Enter Customer Data
2.Calculate rommrent
3.Calculate restaurant bill
4.Calculate laundry bill
5.Calculate gamebill
6.Show total cost
7.Enter into database
8.Exit

Enter your choice:1

Enter your name:book

Enter your address:bangalore

Enter your check in date:10-10-2019

Enter your checkout date:15-10-2019
Your room no.: 50
```

CHOICE 2

```
1.Enter Customer Data
2.Calculate rommrent
3.Calculate restaurant bill
4.Calculate laundry bill
5.Calculate gamebill
6.Show total cost
7.Enter into database
8.Exit

Enter your choice:2
We have the following rooms for you:-
1.  type A---->rs 6000 PN\-
2.  type B---->rs 5000 PN\-
3.  type C---->rs 4000 PN\-
4.  type D---->rs 3000 PN\-
Enter Your Choice Please->1
For How Many Nights Did You Stay:5
you have opted room type A
your room rent is = 30000
```

CHOICE 3

```
1.Enter Customer Data
2.Calculate rommrent
3.Calculate restaurant bill
4.Calculate laundry bill
5.Calculate gamebill
6.Show total cost
7.Enter into database
8.Exit

Enter your choice:3
*****RESTAURANT MENU*****
1.water----->Rs20
2.tea----->Rs10
3.breakfast combo--->Rs90
4.lunch---->Rs110
5.dinner--->Rs150
6.Exit
Enter your choice:1
Enter the quantity:3
Enter your choice:2
Enter the quantity:3
Enter your choice:3
Enter the quantity:3
Enter your choice:4
Enter the quantity:3
Enter your choice:5
Enter the quantity:3
Enter your choice:6
Total food Cost=Rs 1140
```

CHOICE 4

```
1.Enter Customer Data
2.Calculate rommrent
3.Calculate restaurant bill
4.Calculate laundry bill
5.Calculate gamebill
6.Show total cost
7.Enter into database
8.Exit

Enter your choice:4
******LAUNDRY MENU*******
1.Shorts----->Rs3
2.Trousers----->Rs4
3.Shirt--->Rs5
4.Jeans---->Rs6
5.Girlsuit--->Rs8
6.Exit
Enter your choice:1
Enter the quantity:3
Enter your choice:2
Enter the quantity:3
Enter your choice:3
Enter the quantity:3
Enter your choice:4
Enter the quantity:3
Enter your choice:5
Enter the quantity:3
Enter your choice:6
Total Laundary Cost=Rs 78
```

CHOICE *5*

```
1.Enter Customer Data
2.Calculate rommrent
3.Calculate restaurant bill
4.Calculate laundry bill
5.Calculate gamebill
6.Show total cost
7.Enter into database
8.Exit

Enter your choice:5
******GAME MENU*******
1.Table tennis----->Rs60
2.Bowling----->Rs80
3.Snooker--->Rs70
4.Video games---->Rs90
5.Pool--->Rs50
6.Exit
Enter your choice:1
No. of hours:3
Enter your choice:2
No. of hours:3
Enter your choice:3
No. of hours:3
Enter your choice:4
No. of hours:3
Enter your choice:5
No. of hours:3
Enter your choice:6
Total Game Bill=Rs 1050
```

CHOICE 6

```
1.Enter Customer Data
2.Calculate rommrent
3.Calculate restaurant bill
4.Calculate laundry bill
5.Calculate gamebill
6.Show total cost
7.Enter into database
8.Exit

Enter your choice:6
******HOTEL BILL******
Customer details:
Customer name: book
Customer address: bangalore
Check in date: 10-10-2019
Check out date 15-10-2019
Room no. 50
Your Room rent is: 30000
Your Food bill is: 1140
Your laundary bill is: 78
Your Game bill is: 1050
Your sub total bill is: 32268
Additional Service Charges is 1800
Your grandtotal bill is: 34068
```

CHOICE 7

```
1.Enter Customer Data
2.Calculate rommrent
3.Calculate restaurant bill
4.Calculate laundry bill
5.Calculate gamebill
6.Show total cost
7.Enter into database
8.Exit

Enter your choice:7
database selected
Record inserted successfully into restaurant table
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 176
Server version: 8.0.17 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

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

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

mysql> create database school;
Query OK, 1 row affected (0.57 sec)

mysql> use school;
Database changed
mysql> create table restaurant(name varchar(100),adress varchar(100),check_in_date varchar(100),check_out_date varchar(100),room_number varchar(100),rent varchar(100),restaurant_bill varchar(100),laundary_bill v
archar(100),games_bill varchar(100),service_bill varchar(100),total_bill varchar(100));
Query OK, 0 rows affected (2.15 sec)

mysql> select * from restaurant;
+------+----------+--------------+----------------+-------------+------+----------------+--------------+------------+--------------+------------+
| name | adress   | check_in_date | check_out_date | room_number | rent | restaurant_bill | laundary_bill | games_bill | service_bill | total_bill |
+------+----------+--------------+----------------+-------------+------+----------------+--------------+------------+--------------+------------+
| book | bangalore | 10-10-2019   | 15-10-2019     | 50          | 30000 | 1140           | 78           | 1050       | 1800         | 32268      |
+------+----------+--------------+----------------+-------------+------+----------------+--------------+------------+--------------+------------+
1 row in set (0.17 sec)
```

CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of restaurant management planning is to provide a framework that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as a project progress.

BIBILIOGRAPHY

SUMIT ARORA PYTHON CLASS 12

SUMIT ARORA PYTHON CLASS 11

COMPUTER NETWORKS BY Andrew S. Tanenbaum.

GEEKS FOR GEEKS (https://www.geeksforgeeks.org)

PYTHON COURSE (https://www.udemy.com/PythonVideos/Online-Course)

DATABASE CONNECTIVITY COURSE (https://pynative.com)

SQL COURSE (http://learndatabasereporting.com/