KENDRIYA VIDYALAYA

SOUTHERN COMMAND PUNE - 411001.

INFORMATICS PRACTICES (065)

SESSION 2019-20

NAME: SHRAWANI SHRIKANT THOMBRE

CLASS: XII-B[SCIENCE]

ROLL NO: 15617099

PROJECT TITLE: AIRLINE TICKET RESERVATION

TEACHER GUIDE: RICHA CHAUDHARY

INTRODUCTION

PROJECT ON

AIRLINE TICKET RESERVATION

USING

PYTHON &MYSQL CONNECTIVITY

UNDER THE GUIDANCE OF

MRS. RICHA CHAUDHARY

CERTIFICATE

THIS IS TO CERTIFY THAT MS. SHRAWANI SHRIKANT THOMBRE STUDENT OF CLASS XII-B HAS SUCCESSFULLY COMPLETED HER INFORMATICS PRACTICES PROJECT ENTITLED "AIRLINE TICKET RESERVATION" UNDER MY GUIDANCE. THIS PROJECT REPORT IS SUBMITTED FOR EVALUATION AS A PART OF CBSE CURRICULUM FOR AISSE 2019-20.

_____	_____	_____
INFORMATICS PRACTICES TEACHER [MRS. RICHA CHAUDHARY]	EXAMINER SIGNATURE	PRINCIPAL SIGNATURE [MRS. SNEHAL MARATHE]

ACKNOWLEDGEMENT

I WOULD LIKE TO EXPRESS MY SINCERE THANKS TO THE TEACHER GUIDE MRS. RICHA CHAUDHARY WHO GAVE ME THE OPPORTUNITY TO DO THIS PROJECT SUCCESSFULLY AND ALSO GUIDED ME THROUGHOUT THE PROJECT WITH THEIR VALUABLE SUGGESTIONS. I WOULD LIKE TO EXPRESS MY GRATITUDE TOWARDS MY

PARENTS, FAMILY AND MY FRIENDS FOR GUIDANCE AND ADVICE AT EVERY STEP OF MY PROJECT.

13.	BIBLOGRAPHY

INDEX

REQUIREMENTS AND ANALYSIS

THE APPLICATION 'AIRLINE TICKET RESERVATION' BEING A SMALL, PORTABLE AND HAVING GOOD GUI, A COMPUTER SYSTEM SHOULD HAVE THE COMBINATION OF FOLLOWING HARDWARE AND SOFTWARE-

1. **HARDWARE REQUIREMENT**:
   - INTEL 13 PROCESSOR
   - MINIMUM 2 GB RAM
   - ATLEAST 500GB HDD

2. **SOFTWARE REQUIREMENT**:
   - WINDOWS 7/8/10 OS
   - PYTHON 3.X IDLE

- MYSQL

A BRIEF INTRODUCTION OF 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 LAID ITS FOUNDATION IN THE LATE 1980S.
- THE IMPLEMENTATION OF PYTHON WAS STARTED IN THE DECEMBER 1989 BY GUIDO VAN ROSSUM AT CWI IN NETHERLAND.

➢ In 1994, Python 1.0 was released with new features like:lambda, map, filter, and reduce.

➢ Python 2.0 added new features like: list comprehensions, garbage collection system.

➢ On December 3, 2008, Python 3.0 (also called "Py3k") was released. It was designed to rectify fundamental flaw of the language.

➢ ABC programming language is said to be the predecessor of Python language which was capable of exception handling and interfacing with Amoeba operating system.

A brief introduction of MySQL

MySQL was created by a Swedish company, MySQL AB, founded by David Axmark, Allan Larsson and Michael "Monty" Widenius in 1995.

MySQL features:

Relational Database Management System (RDBMS): MySQL is a relational database management system.

Easy to use: MySQL is easy to use. You have to get only the basic knowledge of SQL. You can

build and interact with MySQL with only a few simple SQL statements.

Security: MySQL consist of a solid data security layer that protects sensitive data from intruders. Passwords are encrypted in MySQL.

Client/ server architecture: MySQL follows a client /server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc.

Free to download: MySQL is free to use and you can download it from MySQL official website.

Scalable: MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.

Compatible on many operating systems: MySQL is compatible to run on many operating systems, like Novell NetWare, Windows* Linux*, many varieties of UNIX* (such as Sun* Solaris*,

AIX, and DEC* UNIX), OS/2, FREEBSD*, and others. MYSQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network or the internet).

ALLOWS ROLL-BACK: MYSQL allows transactions to be rollback, commit and crash recovery.

PERFORMANCE: MYSQL is faster, more reliable and cheaper because of its unique storage engine architecture.

FLEXIBILITY: MYSQL supports a large number of embedded applications which makes MYSQL very flexible.

PRODUCTIVITY: MYSQL uses triggers, stored procedures and views which allows the developer to give a higher productivity.

OBJECTIVE OF PROJECT REPORT

THE MAIN OBJECTIVE OF THIS PROJECT IS TO DESIGN AND DEVELOP AN AIRLINE TICKET RESERVATION .

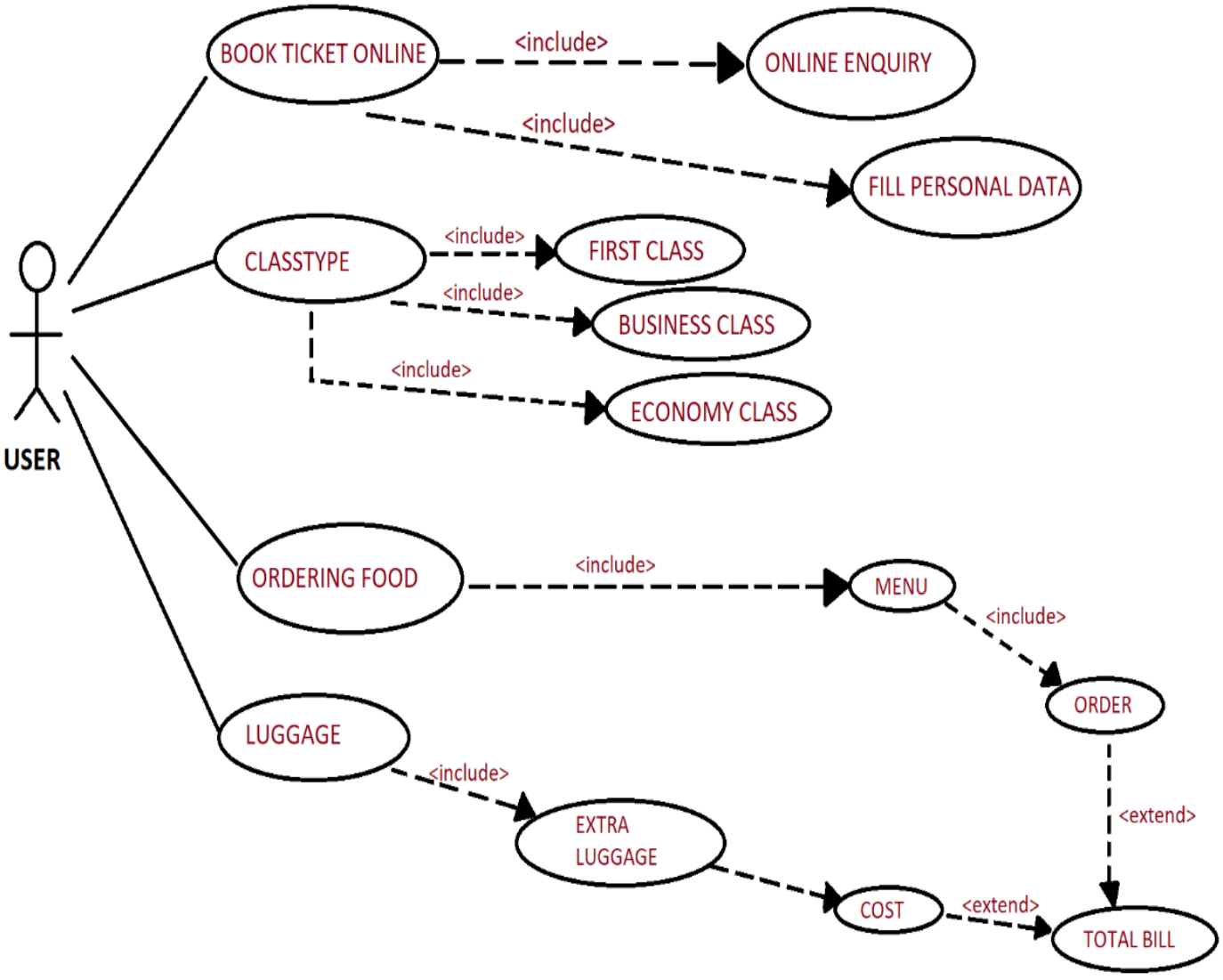
This project is mainly intended for the customers, who use the airline websites to make reservations on flights.

Development of this application includes a number of fields such that user feels comfortable and the system appears as dynamic to them.

The project "Airline Ticket Reservation" includes the following functionalities:

- Keeping records of reservation of customers.
- Facilities provided by airline are fully utilized in effective and efficient manner.
- Keeping records of different bills and expenses of customer.
- Keeping the total cost and printing the bills accordingly.
- Easy to use for better execution of the project.

USE CASE DIAGRAM



IMPORTS USED

1. import os

2. import platform

3. import pymysql as py

4. import pandas as pd

DATABASE/TABLES USED

- **DATABASE:-**

1. AIRINDIA

- **TABLES:-**

1. PDATA
2. FOOD
3. CLASSTYPE
4. LUGGAGE

PROJECT CODE

```
GLOBAL Z
MYDB = PY.CONNECT(USER='ROOT', PASSWORD='',
HOST='LOCALHOST',
DATABASE='AIRINDIA')
MYCURSOR=MYDB.CURSOR()

DEF REGISTERCUST():
L=[]
CUSTNAME=RAW_INPUT("ENTER NAME:")
L.APPEND(CUSTNAME)
ADDR=RAW_INPUT("ENTER ADDRESS:")
L.APPEND(ADDR)
JRDATE=RAW_INPUT("ENTER DATE OF JOURNEY:")
L.APPEND(JRDATE)
SOURCE=RAW_INPUT("ENTER SOURCE:")
```

```
L.APPEND(SOURCE)
DESTINATION=RAW_INPUT("ENTER DESTINATION:")
L.APPEND(DESTINATION)
CUST=(L)
SQL="INSERT INTO
PDATA(CUSTNAME,ADDR,JRDATE,SOURCE,DESTINATI
ON)VALUES(%S,%S,%S,%S,%S)"
MYCURSOR.EXECUTE(SQL,CUST)
MYDB.COMMIT()

DEF CLASSTYPEVIEW():
PRINT("DO YOU WANT TO SEE CLASS TYPE AVAILABLE
: ENTER 1 FOR YES :")
CH=INT(INPUT("ENTER YOUR CHOICE:"))
IF CH==1:
SQL="SELECT * FROM CLASSTYPE"
MYCURSOR.EXECUTE(SQL)
ROWS=MYCURSOR.FETCHALL()
FOR X IN ROWS:
PRINT(X)

DEF TICKETPRICE():
PRINT ("WE HAVE THE FOLLOWING ROOMS FOR YOU:-")
PRINT ("1. TYPE FIRST CLASS---->RS 6000 PN\-")
PRINT ("2. TYPE BUSINESS CLASS---->RS 4000 PN\-")
PRINT ("3. TYPE ECONOMY CLASS---->RS 2000 PN\-")
X=INT(INPUT("ENTER YOUR CHOICE PLEASE->"))
N=INT(INPUT("NO OF PASSENGER:"))
IF(X==1):
```

```
PRINT ("YOU HAVE OPTED FIRST CLASS")
S=6000*N
ELIF(X==2):
PRINT ("YOU HAVE OPTED BUSINESS CLASS")
S=4000*N
ELIF (X==3):
PRINT ("YOU HAVE OPTED ECONOMY CLASS")
S=2000*N
ELSE:
PRINT ("PLEASE CHOOSE A CLASS TYPE")
PRINT ("YOUR ROOM RENT IS =",S,"\N")

DEF MENUVIEW():
PRINT("DO YOU WANT TO SEE MENU AVAILABLE :
ENTER 1 FOR YES :")
CH=INT(INPUT("ENTER YOUR CHOICE:"))
IF CH==1:
SQL="SELECT * FROM FOOD"
MYCURSOR.EXECUTE(SQL)
ROWS=MYCURSOR.FETCHALL()
FOR X IN ROWS:
PRINT(X)

DEF ORDERITEM():
GLOBAL S
PRINT ("DO YOU WANT TO SEE MENU AVAILABLE:
ENTER 1 FOR YES :")
CH=INT (INPUT ("ENTER YOUR CHOICE :"))
IF CH==1:
```

```
SQL="SELECT * FROM FOOD"
MYCURSOR.EXECUTE (SQL)
ROWS=MYCURSOR.FETCHALL ()
FOR X IN ROWS:
PRINT(X)
PRINT ("DO YOU WANT TO PURCHASE FROM ABOVE
LIST: ENTER YOUR CHOICE :")
D=INT (INPUT ("ENTER YOUR CHOICE 1-10 :"))
IF(D==1):
PRINT("YOU HAVE ORDERED TEA")
A=INT(INPUT("ENTER QUANTITY"))
S=10*A
PRINT("YOUR AMOUNT FOR TEA IS :",S,"\N")
ELIF (D==2):
PRINT("YOU HAVE ORDERED COFFEE")
A=INT(INPUT("ENTER QUANTITY"))
S=10*A
PRINT("YOUR AMOUNT FOR COFFEE IS :",S,"\N")
ELIF(D==3):
PRINT("YOU HAVE ORDERED COLDDRINK")
A=INT(INPUT("ENTER QUANTITY"))
S=20*A
PRINT("YOUR AMOUNT FOR COLDDRINK IS :",S,"\N")
ELIF(D==4):
PRINT("YOU HAVE ORDERED SAMOSA")
A=INT(INPUT("ENTER QUANTITY"))
S=10*A
PRINT("YOUR AMOUNT FOR SAMOSA IS :",S,"\N")
ELIF(D==5):
```

```
print("you have ordered sandwich")
a=int(input("enter quantity"))
s=50*a
print("your amount for sandwich is :",s,"\n")
elif(d==6):
print("you have ordered dhokla")
a=int(input("enter quantity"))
s=30*a
print("your amount for dhokla is :",s,"\n")
elif(d==7):
print("you have ordered kachori")
a=int(input("enter quantity"))
s=10*a
print("your amount for kachori is :",s,"\n")
elif(d==8):
print("you have ordered milk")
a=int(input("enter quantity"))
s=20*a
print("your amount for kachori is :",s,"\n")
elif(d==9):
print("you have ordered noodles")
a=int(input("enter quantity"))
s=50*a
print("your amount for noodles is :",s,"\n")
elif(d==10):
print("you have ordered pasta")
a=int(input("enter quantity"))
s=50*a
print("your amount for pasta is :",s,"\n")
```

```
ELSE:
PRINT("PLEASE ENTER YOUR CHOICE FROM THE
MENU")

DEF LUGGAGEBILL():
GLOBAL Z
PRINT ("DO YOU WANT TO SEE RATE FOR LUGGAGE :
ENTER 1 FOR YES :")
CH=INT(INPUT("ENTER YOUR CHOICE:"))
IF CH==1:
SQL="SELECT * FROM LUGGAGE"
MYCURSOR.EXECUTE(SQL)
ROWS=MYCURSOR.FETCHALL()
FOR X IN ROWS:
PRINT(X)
Y=INT(INPUT("ENTER YOUR WEIGHT OF EXTRA
LUGGAGE->"))
Z=Y*1000
PRINT("YOUR LUGGAGE BILL :",Z,"\N")
RETURN Z

DEF LB():
PRINT("LUGGAGEBILL",Z)

DEF RES():
PRINT('FOOD BILL',S)

DEF TICKETAMOUNT():
A=RAW_INPUT("ENTER CUSTOMER NAME:")
```

```
PRINT("CUSTOMER NAME :",A,"\N")
PRINT("LUGGAGE BILL:")
PRINT(LB)
PRINT("FOOD BILL:")
PRINT(RES)

DEF QUIT():
PRINT ("THANKYOU! FLY WITH US AGAIN")

DEF MENUSET():
PRINT("ENTER 1: TO ENTER CUSTOMER DATA")
PRINT("ENTER 2 : TO VIEW CLASS")
PRINT("ENTER 3 : FOR TICKETAMOUNT")
PRINT("ENTER 4 : FOR VIEWING FOOD MENU")
PRINT("ENTER 5 : FOR ORDERING AND FOOD BILL")
PRINT("ENTER 6 :FOR LUGGAGE BILL")
PRINT("ENTER 7 : FOR COMPLETE AMOUNT")
PRINT("ENTER 8 : FOR EXIT:")
USERINPUT=INT(INPUT("ENTER YOUR CHOICE"))
IF(USERINPUT==1):
REGISTERCUST()
ELIF(USERINPUT==2):
CLASSTYPEVIEW()
ELIF(USERINPUT==3):
TICKETPRICE()
ELIF(USERINPUT==4):
MENUVIEW()
ELIF(USERINPUT==5):
ORDERITEM()
```

```
elif(userinput==6):
luggagebill()
elif(userinput==7):
ticketa mount()
elif(userinput==8):
quit()
else:
print("enter correct choice")
menuset()
def runagain():
runagn=raw_input("\n want to run again y/n:")
while(runagn.lower()=='y'):
if(platform.system()=="windows"):
print(os.system('cls'))
else:
print(os.system('clear'))
menuset()
runagn=raw_input("\n want to run again y/n:")
runagain()
```

OUTPUT

-

```
mysql> desc pdata;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| custname    | varchar(20) | YES  |     | NULL    |       |
| addr        | varchar(30) | YES  |     | NULL    |       |
| jrdate      | varchar(10) | YES  |     | NULL    |       |
| source      | varchar(10) | YES  |     | NULL    |       |
| destination | varchar(10) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.03 sec)
```

```
mysql> desc food;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| s_no     | int(10)     | YES  |     | NULL    |       |
| itemname | varchar(10) | YES  |     | NULL    |       |
| rate     | int(10)     | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.11 sec)
```

```
mysql> desc classtype;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| s_no      | varchar(5)  | YES  |     | NULL    |       |
| classtype | varchar(30) | YES  |     | NULL    |       |
| price     | int(10)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.04 sec)
```

```
mysql> desc luggage;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| s_no   | int(10)     | YES  |     | NULL    |       |
| weight | varchar(10) | YES  |     | NULL    |       |
| rate   | int(10)     | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.03 sec)
```

```
mysql> select * from pdata;
+----------+-----------+------------+----------+-------------+
| custname | addr      | jrdate     | source   | destination |
+----------+-----------+------------+----------+-------------+
| ayushi   | camp pune | 2020-01-12 | hadapsar | kerela      |
| shrawani | delhi     | 2020-05-10 | pune     | surat       |
| soham    | swargate  | 2020-02-12 | dehradun | hawai       |
+----------+-----------+------------+----------+-------------+
3 rows in set (0.00 sec)
```

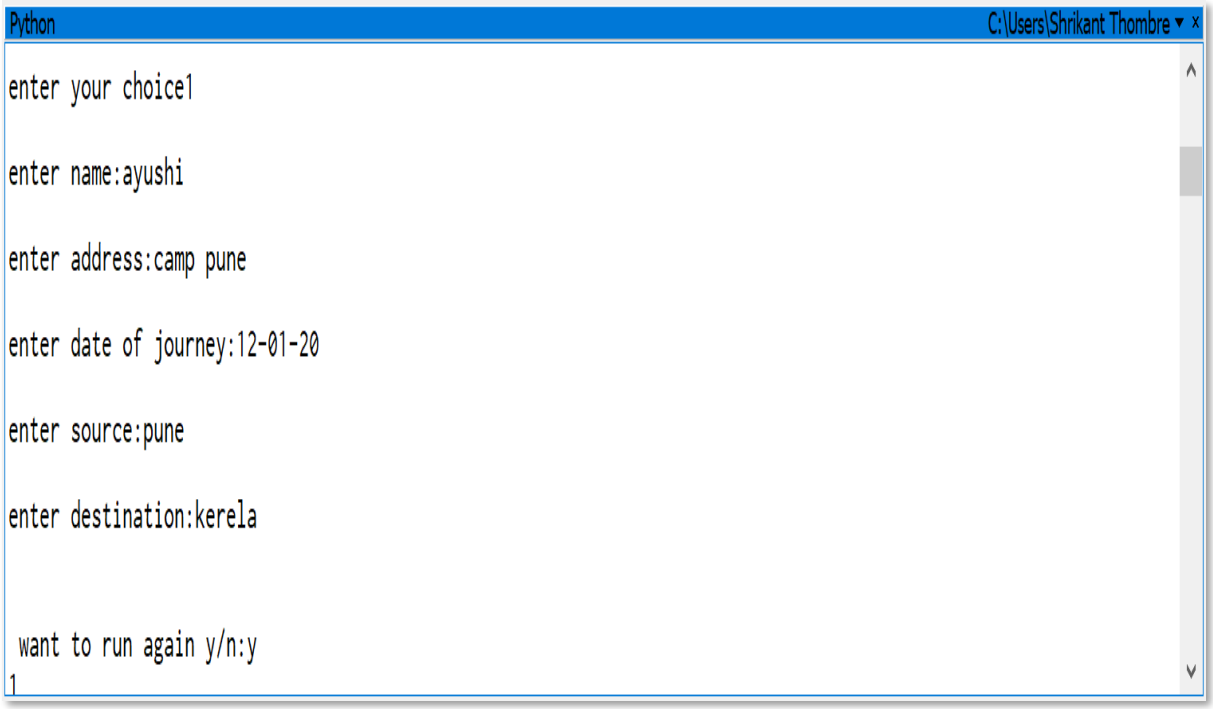
```
mysql> select * from food;
+------+-----------+------+
| s_no | itemname  | rate |
+------+-----------+------+
|    1 | Tea       |   10 |
|    2 | Coffee    |   10 |
|    3 | Colddrink |   20 |
|    4 | Samosa    |   10 |
|    5 | Sandwich  |   50 |
|    6 | Dhokla    |   30 |
|    7 | Kachori   |   10 |
|    8 | Milk      |   20 |
|    9 | Noodles   |   50 |
|   10 | Pasta     |   50 |
+------+-----------+------+
10 rows in set (0.00 sec)
```

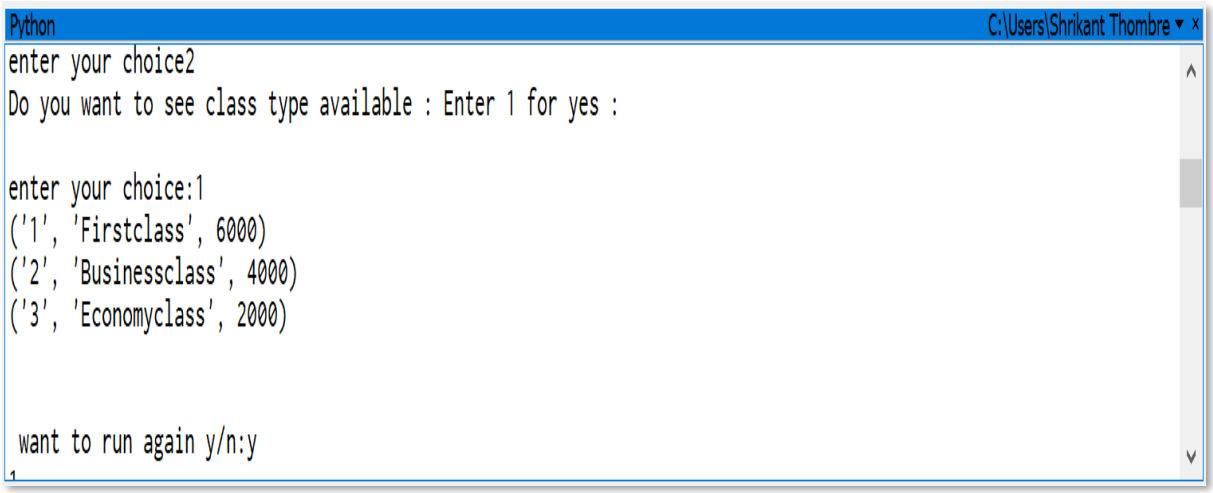
```
mysql> select * from classtype;
+------+--------------+-------+
| s_no | classtype    | price |
+------+--------------+-------+
| 1    | Firstclass   |  6000 |
| 2    | Businessclass|  4000 |
| 3    | Economyclass |  2000 |
+------+--------------+-------+
3 rows in set (0.00 sec)
```

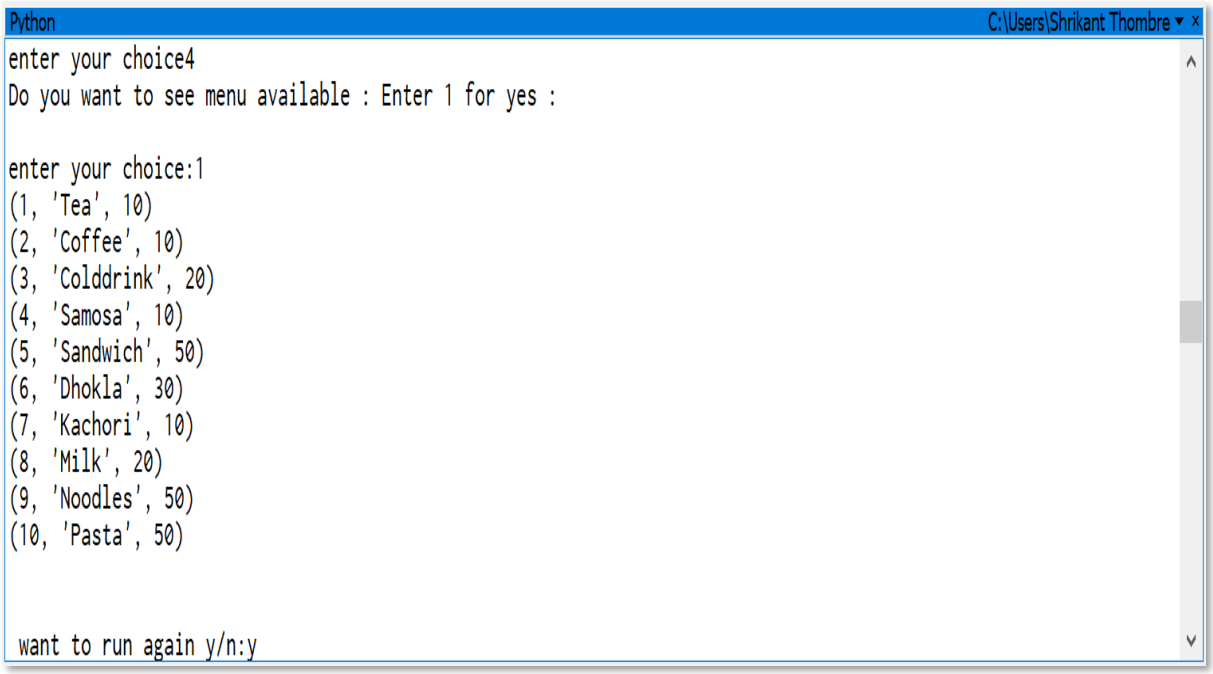
+

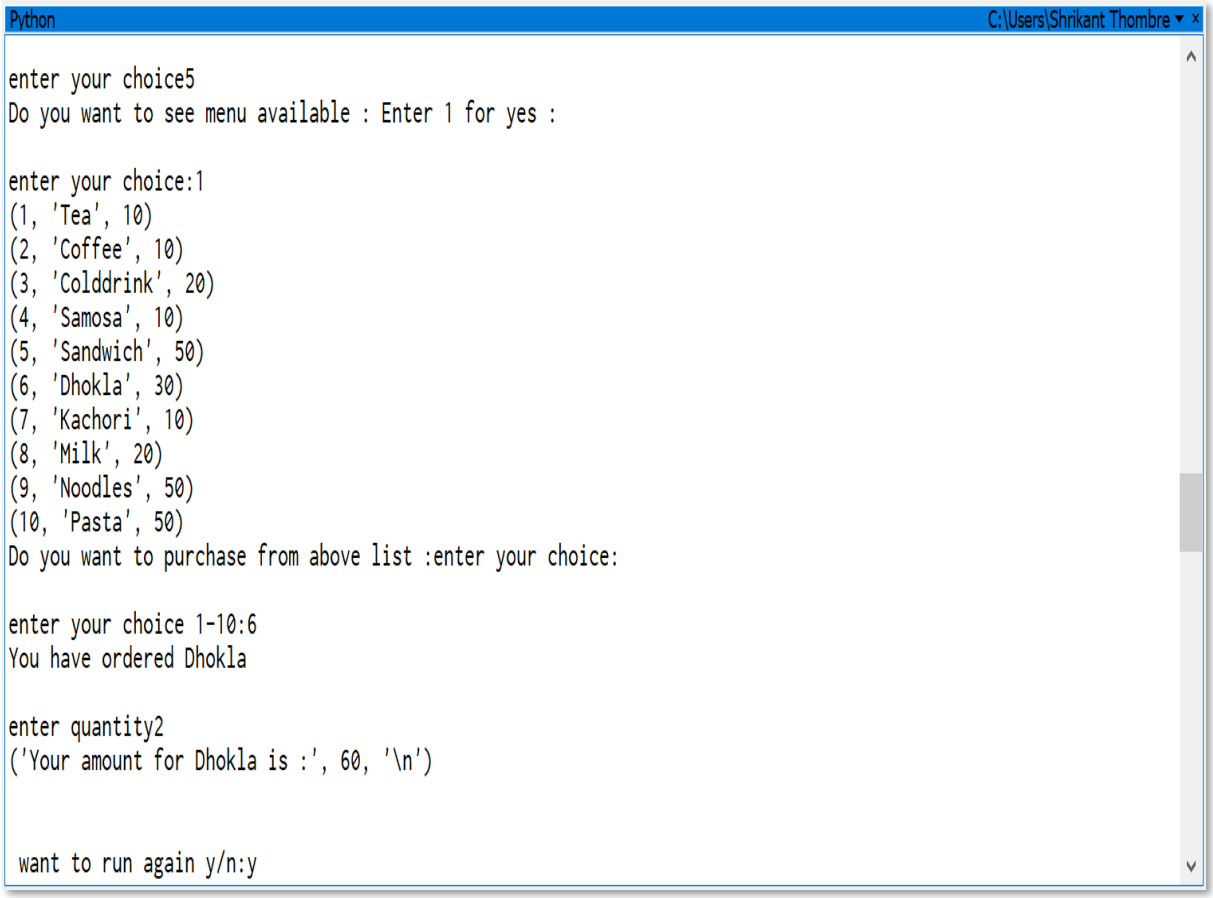
```
mysql> select * from luggage;
+------+--------+------+
| s_no | weight | rate |
+------+--------+------+
|    1 | 20kg   | 1000 |
|    2 | 25kg   | 1500 |
|    3 | 30kg   | 2000 |
|    4 | 50kg   | 3000 |
+------+--------+------+
```

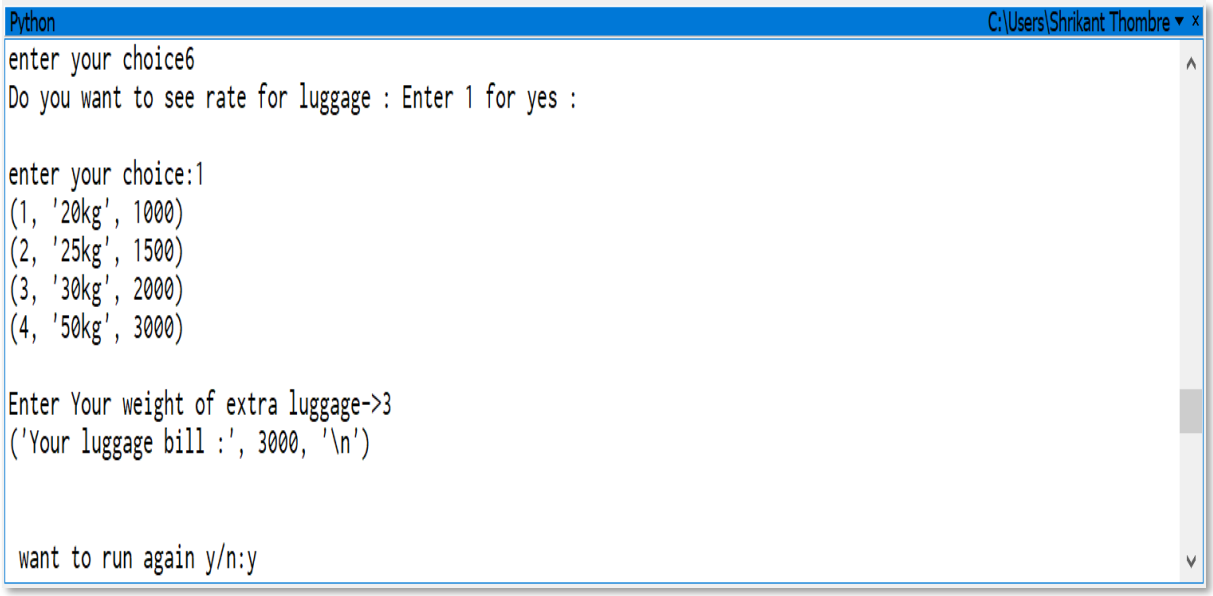
```
Python                                                    C:\Users\Shrikant Thombre
In [1]: %run "C:\Users\Shrikant Thombre\Desktop\IP class 12\project 2020\ip project.py"
enter 1: To enter customer data
enter 2 : To view class
enter 3 : for ticketamount
enter 4 : for viewing food menu
enter 5 : for ordering and food bill
enter 6 :for luggage bill
enter 7 : for complete amount
enter 8 : for exit:
```

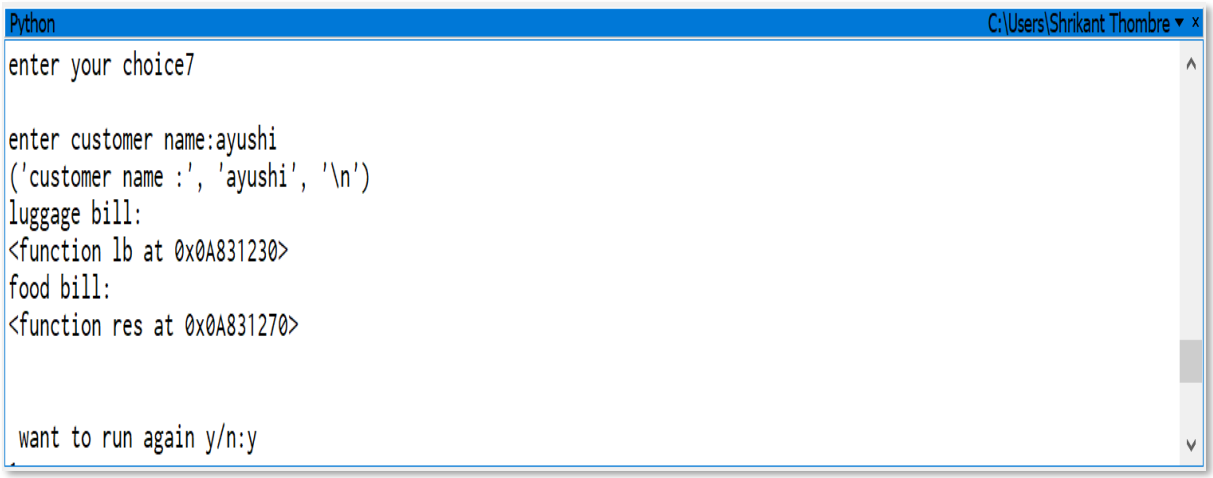

Python
C:\Users\Shrikant Thombre
enter your choice1
enter name:ayushi
enter address:camp pune
enter date of journey:12-01-20
enter source:pune
enter destination:kerela
want to run again y/n:y
1

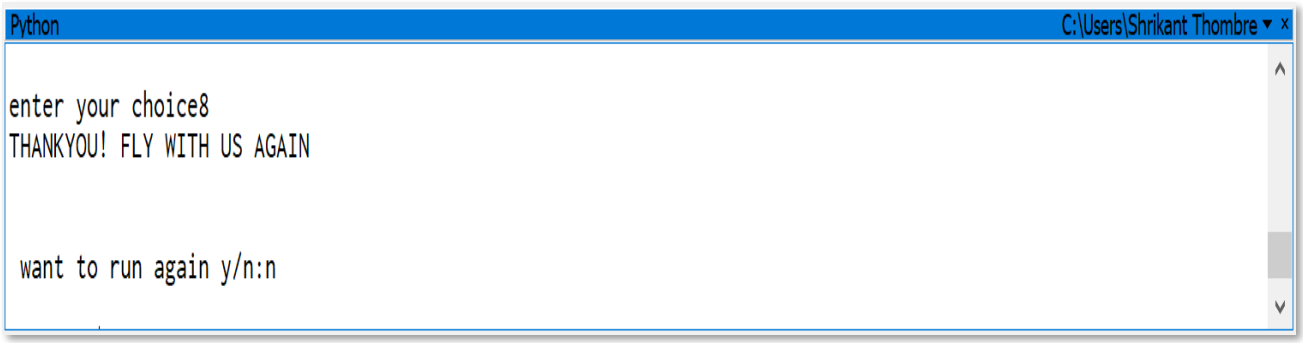

Python
C:\Users\Shrikant Thombre
enter your choice2
Do you want to see class type available : Enter 1 for yes :
enter your choice:1
('1', 'Firstclass', 6000)
('2', 'Businessclass', 4000)
('3', 'Economyclass', 2000)
want to run again y/n:y
1


Python
C:\Users\Shrikant Thombre
enter your choice4
Do you want to see menu available : Enter 1 for yes :
enter your choice:1
(1, 'Tea', 10)
(2, 'Coffee', 10)
(3, 'Colddrink', 20)
(4, 'Samosa', 10)
(5, 'Sandwich', 50)
(6, 'Dhokla', 30)
(7, 'Kachori', 10)
(8, 'Milk', 20)
(9, 'Noodles', 50)
(10, 'Pasta', 50)
want to run again y/n:y


Python
C:\Users\Shrikant Thombre
enter your choice5
Do you want to see menu available : Enter 1 for yes :
enter your choice:1
(1, 'Tea', 10)
(2, 'Coffee', 10)
(3, 'Colddrink', 20)
(4, 'Samosa', 10)
(5, 'Sandwich', 50)
(6, 'Dhokla', 30)
(7, 'Kachori', 10)
(8, 'Milk', 20)
(9, 'Noodles', 50)
(10, 'Pasta', 50)
Do you want to purchase from above list :enter your choice:
enter your choice 1-10:6
You have ordered Dhokla
enter quantity2
('Your amount for Dhokla is :', 60, '\n')
want to run again y/n:y


Python
C:\Users\Shrikant Thombre
enter your choice6
Do you want to see rate for luggage : Enter 1 for yes :
enter your choice:1
(1, '20kg', 1000)
(2, '25kg', 1500)
(3, '30kg', 2000)
(4, '50kg', 3000)
Enter Your weight of extra luggage->3
('Your luggage bill :', 3000, '\n')
want to run again y/n:y


Python
C:\Users\Shrikant Thombre
enter your choice7
enter customer name:ayushi
('customer name :', 'ayushi', '\n')
luggage bill:
<function lb at 0x0A831230>
food bill:
<function res at 0x0A831270>
want to run again y/n:y


Python
C:\Users\Shrikant Thombre
enter your choice8
THANKYOU! FLY WITH US AGAIN
want to run again y/n:n

CONCLUSION

THE CONCLUSION OF THIS PROJECT IS AN AIRLINE RESERVATION WHICH IS A COMPUTERIZED MANAGEMENT SYSTEM. THE PROPOSED SYSTEM WILL KEEP A TRACK OF GENERATION OF REPORT REGARDING THE PRESENT STATUS. THIS PROJECT HAS GUI BASED SOFTWARE THAT WILL HELP IN STORING, UPDATING AND RETRIEVING THE INFORMATION THROUGH VARIOUS USER-FRIENDLY MENU-DRIVEN MODULES. THE PROJECT "AIRLINE TICKET RESERVATION" IS AIMED TO

DEVELOP TO MAINTAIN THE DAY-TO-DAY STATE OF ADMISSION OF CUSTOMERS, PAYMENT DETAILS ETC. MAIN OBJECTIVE OF THIS PROJECT IS TO PROVIDE SOLUTION FOR AIRLINES TO MANAGE MOST OF THEIR WORK USING COMPUTERIZED PROCESS. THIS SOFTWARE APPLICATION WILL HELP ADMIN TO HANDLE CUSTOMER'S INFORMATION, SEAT ALLOCATION, PAYMENT DETAILS, BILLING INFORMATION, ETC. DETAILED EXPLANATION ABOUT MODULES AND DESIGN ARE PROVIDED IN PROJECT DOCUMENTATION. THE EXISTING SYSTEM IS A MANUALLY MAINTAINED SYSTEM. ALL THE AIRLINE RECORDS ARE TO BE MAINTAINED FOR THE DETAILS OF EACH CUSTOMER, FEE DETAILS, CLASS ALLOCATION, ETC. ALL THESE DETAILS ARE ENTERED AND RETRIEVED MANUALLY, BECAUSE OF THIS THERE ARE MANY DISADVANTAGES LIKE TIME CONSUMING, UPDATING PROCESS, INACCURACY OF DATA. FOR AVOIDING THIS WE INTRODUCED OR PROPOSED A NEW SYSTEM IN PROPOSED SYSTEM THE COMPUTERIZED VERSION OF THE EXISTING

system. provides easy and quick access over the data.

BIBLOGRAPHY

THE FOLLOWING BOOKS AND WEBSITES HAVE BEEN REFERRED DURING THE PROJECT DEVELOPMENT-

TEXT BOOK: INFORMATICS PRACTICES WITH PYTHON BY SUMITA ARORA.

WEBSITES:

- http://www.w3schools.com
- www.pythontutorial.com