COMPUTER SCIENCE PROJECT

**AIRLINES MANAGEMENT**

**SYSTEM**





Presented By:

Name : **Manika Gupta**

Class : **XII-A**

Table of Contents

1. COVER PAGE…………………………………………………………………………...(i)

2. TABLE OF CONTENTS………………………………………………………………..(ii)

3. CERTIFICATE…………………………………........................................................(ii)

4. ACKNOWLEDGEMENT……………………………………………………………….(iii)

5. SYSTEM REQUIREMENTS………………...........................................................(iv)

6. PROJECT FRONT END AND BACK END…………………………………………..(v)

7. INTRODUCTION ……………………………………………………………………….(6)

8. OBJECTIVE……………………………………………………………………………..(7)

9. SCOPE OF THE PROJECT…………………………………………………………...(8)

10. SOURCE CODE……………………………………………………………………...(9)

11. PYTHON OUTPUTS………………………………………………………………...(16)

12. SQL OUTPUTS………………………………………………………………………(21)

13. BIBLIOGRAPHY……………………………………………………………………..(23)

AIRLINES

CERTIFICATE

This is to certify that *Manika Gupta* of class XII-A have worked under my supervision on project Airlines Management System and completed it to my full satisfaction.

I wish him success in his life.

Examiner Name : *Siddharth Bhardwaj*

ACKNOWLEDGEMENT

I am thankful to my computer teacher Mr Siddharth Bhardwaj who helped and guided me while making this project without whose guidance this project would have been incomplete and imperfect.

I would also like to thank computer department of my school who helped and gave me time for the completion of my project.

Student Details : *Vansh Gupta XII-A 16*

Student Sign :

Teacher Name : *Mr Siddharth Bhardwaj*

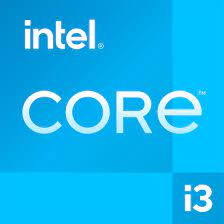
Teacher Sign. :

System Requirements

Hardware Specification

Processor : Intel Core i3 processor

Disk Space : 1 GB

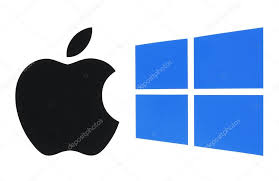


Software Specification

Operating System : Windows 10, MAC O/S

Python Version : 3.8 or higher

MYSQL Version : 8.0



Project Front End & Back End

Front End : Python



Back End : MYSQL

:



**Introduction**

This program is very useful in real life situation for providing instant information of employees working under an airline company. It contains information about their monthly salary, employee ID, designation etc.

Apart from employees information it can also display several details about the customers travelling with the airlines which includes their travel class, arrival & departure timings, complaints (if any) etc. It also lets customers book flights according to their choice.

In this python program we can modify, add, delete, recall and list the records.

OBJECTIVE

The objectives of the system are :-

1. Reduced operational time.

2. Increased accuracy and reliability.

3. Increased operational efficiency.

This software package can be readily used by non-programming personal avoiding human handled chance of error. This project is used by two types of users :-

1) Customer

2) Service Administrator

Customer can check their flight details and give their feedback.

Service Administrator can keep a record and modify customer details and employee details.

New features can be added as per requirements.

SCOPE OF THE PROJECT

* It keeps a record of all passengers and their flight details.
* Service administrators can keep organised details about every passenger & employee and access them easily.
* The software eliminates the need of paper records and is user-friendly as well.
* It helps to retrieve data in very short time and works efficiently.
* It also provides a platform to customers where they can book flights and share their grievances (if any).

SOURCE CODE

1) Creating database “Airlines\_Management\_System”

import mysql.connector as sql  
  
conn = sql.connect(host='localhost', password='vanshsql', user='root')  
cur = conn.cursor()  
  
*# ========== DATABASE CREATED ==========*cur.execute('create database Airlines\_Management\_System')  
print('Database created !')  
cur.execute('show databases')  
for x in cur:  
 print(x)

2) Creating tables “Employees\_Info” & “Customers\_Info”

import mysql.connector as sql  
  
conn = sql.connect(host='localhost', password='vanshsql', user='root', database='Airlines\_Management\_System')  
cur = conn.cursor()  
  
*# ========== TABLES CREATED ==========*cur.execute('create table Employees\_Info(Employee\_ID varchar(5) primary key, Employee\_Name varchar(20), Designation varchar(20), Monthly\_Salary int(10), Gender varchar(10), Age int(3))')  
cur.execute('create table Customers\_Info(Ticket\_ID varchar(5) primary key, Name varchar(20), Arrival\_Departure varchar(20), Travel\_Class varchar(20), Complaint varchar(300), Complaint\_Status varchar(20))')  
print('Tables created !')

3) Main Source Code

f\_list = ['1) New Delhi to New York', '2) Beijing to London', '3) Johannesburg to Buenos Aires', '4) Colombo to Bombay', '5) Paras to Bali']  
t\_list = ['23:45 to 14:55', '02:30 to 18:30', '07:40 to 06:30', '08:00 to 10:35', '01:25 to 12:30']  
c\_list = ['1) BUSINESS CLASS', '2) PREMIUM ECONOMY', '3) ECONOMY CLASS']  
  
def check\_customer():  
 global cur  
 t\_ID = input('Enter the Ticket ID :\t')  
 task = "select \* from Customers\_Info where Ticket\_ID='{}'".format(t\_ID)  
 cur.execute(task)   
 record = cur.fetchall()  
 if len(record) == 0:  
 return [False]  
 else:  
 return [True, t\_ID]  
  
def check\_employee():  
 global cur  
 e\_ID = input('Enter the Employee ID :\t')  
 task = "select \* from Employees\_Info where Employee\_ID='{}'".format(e\_ID)  
 cur.execute(task)  
 record = cur.fetchall()  
 if len(record) == 0:  
 return [False]  
 else:  
 return [True, e\_ID]  
  
def id\_generator():  
 id\_gen = ''  
 for i in range(4):  
 x = str(randint(0, 9))  
 id\_gen += x  
 return id\_gen  
  
def side\_func(category):  
 if category == 'E':  
 print("\nFollowing data is in the order: (Employee ID, Employee Name, Designation, Monthly Salary, Gender, Age)\n")  
 elif category == 'C':  
 print("\nFollowing data is in the order: (Ticket ID, Customer Name, Arrival/Departure, Travel Class, Complaint, Complaint Status)\n")  
  
print('\n==============================WELCOME TO AIRLINES MANAGEMENT SYSTEM==============================\n')  
  
flag = True  
while True:  
 choice = input('Are you an admin or a user ?\n').lower()  
 temp\_flag = True  
 while True:  
 if choice == 'admin':  
 if temp\_flag is True:  
 user = input('Enter username :\t')  
 pswd = input('Enter password :\t')  
 print()  
 elif temp\_flag is False:  
 user, pswd = 'admin', '@dmin$77'  
 if user == 'admin' and pswd == '@dmin$77':  
 temp\_flag = False  
 print('1) ADD EMPLOYEE INFORMATION')  
 print('2) DELETE EMPLOYEE INFORMATION')  
 print('3) UPDATE EMPLOYEE INFORMATION')  
 print("4) VIEW CERTAIN EMPLOYEE'S INFORMATION")  
 print("5) VIEW ALL EMPLOYEES' INFORMATION")  
 print("6) VIEW CERTAIN CUSTOMER'S INFORMATION / COMPLAINT")  
 print("7) VIEW ALL CUSTOMERS' INFORMATION")  
 print('8) UPDATE CUSTOMER COMPLAINT STATUS')  
 print('9) DELETE CUSTOMER INFORMATION')  
 print('10) QUIT\n')  
 a\_ans = input('Enter your choice :\t')  
 if a\_ans == '1':  
 emp\_ID = 'E' + id\_generator()  
 f\_n = input("Enter employee's first name :\t").lower().capitalize()  
 l\_n = input("Enter employee's last name :\t").lower().capitalize()  
 name = f\_n + ' ' + l\_n  
 designation = input("Enter employee's designation :\t").lower().capitalize()  
 m\_salary = int(input("Enter employee's monthly salary :\t"))  
 gender = input("Enter employee's gender :\t").lower().capitalize()  
 age = int(input("Enter employee's age :\t"))  
 task = "insert into Employees\_Info values ('{}', '{}', '{}', {}, '{}', {})".format(emp\_ID, name, designation, m\_salary, gender, age)  
 cur.execute(task)  
 conn.commit()  
 print('Process successfully completed.\n')  
 elif a\_ans == '2':  
 check\_flag = check\_employee()  
 if check\_flag[0] is True:  
 task = "delete from Employees\_Info where Employee\_ID='{}'".format(check\_flag[1])  
 cur.execute(task)  
 conn.commit()  
 print('Process successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif a\_ans == '3':  
 check\_flag = check\_employee()  
 if check\_flag[0] is True:  
 print('\n1) EMPLOYEE ID\n2) NAME\n3) DESIGNATION\n4) MONTHLY SALARY\n5) GENDER\n6) AGE\n')  
 u\_option = input('Select an option that you would like to update :\t')  
 if u\_option == '1':  
 u\_value = input('Enter the updated ID here :\t')  
 task = "update Employees\_Info set Employee\_ID='{}' where Employee\_ID='{}'".format(u\_value, check\_flag[1])  
 elif u\_option == '2':  
 u\_value = input('Enter the updated name here :\t').lower().capitalize()  
 task = "update Employees\_Info set Employee\_Name='{}' where Employee\_ID='{}'".format(u\_value, check\_flag[1])  
 elif u\_option == '3':  
 u\_value = input('Enter the updated designation here :\t').lower().capitalize()  
 task = "update Employees\_Info set Designation='{}' where Employee\_ID='{}'".format(u\_value, check\_flag[1])  
 elif u\_option == '4':  
 u\_value = int(input('Enter the updated monthly salary here :\t'))  
 task = "update Employees\_Info set Monthly\_Salary={} where Employee\_ID='{}'".format(u\_value, check\_flag[1])  
 elif u\_option == '5':  
 u\_value = input('Enter the updated gender here :\t').lower().capitalize()  
 task = "update Employees\_Info set Gender='{}' where Employee\_ID='{}'".format(u\_value, check\_flag[1])  
 elif u\_option == '6':  
 u\_value = int(input('Enter the updated age here :\t'))  
 task = "update Employees\_Info set Age={} where Employee\_ID='{}'".format(u\_value, check\_flag[1])  
 else:  
 print('Invalid numerical choice! Session terminated.\n')  
 break  
 cur.execute(task)  
 conn.commit()  
 print('Process successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif a\_ans == '4':  
 check\_flag = check\_employee()  
 if check\_flag[0] is True:  
 task = "select \* from Employees\_Info where Employee\_ID='{}'".format(check\_flag[1])  
 cur.execute(task)  
 record = cur.fetchall()  
 side\_func('E')  
 for r in record:  
 print(r)  
 print('Process successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif a\_ans == '5':  
 cur.execute('select \* from Employees\_Info')  
 record = cur.fetchall()  
 if len(record) == 0:  
 print('No Data Exists.\nProcess successfully completed.\n')  
 else:  
 side\_func('E')  
 for r in record:  
 print(r)  
 print('\nProcess successfully completed.\n')  
 elif a\_ans == '6':  
 check\_flag = check\_customer()  
 if check\_flag[0] is True:  
 task = "select \* from Customers\_Info where Ticket\_ID='{}'".format(check\_flag[1])  
 cur.execute(task)  
 record = cur.fetchall()  
 side\_func('C')  
 for r in record:  
 print(r)  
 print('Process successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif a\_ans == '7':  
 cur.execute('select \* from Customers\_Info')  
 record = cur.fetchall()  
 if len(record) == 0:  
 print('No Data Exists.\nProcess successfully completed.\n')  
 else:  
 side\_func('C')  
 for r in record:  
 print(r)  
 print('\nProcess successfully completed.\n')  
 elif a\_ans == '8':  
 check\_flag = check\_customer()  
 if check\_flag[0] is True:  
 u\_status = input('Enter the updated complaint status here :\n')  
 task = "update Customers\_Info set Complaint\_Status='{}' where Ticket\_ID='{}'".format(u\_status, check\_flag[1])  
 cur.execute(task)  
 conn.commit()  
 print('Process successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif a\_ans == '9':  
 check\_flag = check\_customer()  
 if check\_flag[0] is True:  
 task = "delete from Customers\_Info where Ticket\_ID='{}'".format(check\_flag[1])  
 cur.execute(task)  
 conn.commit()  
 print('Process successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif a\_ans == '10':  
 print('Exiting system...')  
 flag = False  
 break  
 else:  
 print('Invalid numerical choice! Session terminated.\n')  
 break  
 else:  
 print('Invalid credentials! Session terminated.\n')  
 break  
 elif choice == 'user':  
 print('1) BOOK A FLIGHT')  
 print('2) ADD COMPLAINT')  
 print('3) DELETE COMPLAINT')  
 print('4) UPDATE COMPLAINT')  
 print('5) VIEW YOUR INFORMATION / COMPLAINT STATUS')  
 print('6) CANCEL A FLIGHT')  
 print('7) QUIT\n')  
 u\_ans = input('Enter your choice :\t')  
 if u\_ans == '1':  
 ticket\_ID = 'C' + id\_generator()  
 f\_n = input('Enter first name :\t').lower().capitalize()  
 l\_n = input('Enter last name :\t').lower().capitalize()  
 name = f\_n + ' ' + l\_n  
 print('\nCurrently scheduled flights :')  
 for i in f\_list:  
 print(i)  
 option = input('Enter your choice :\t')  
 print()  
 if option == '1':  
 time = t\_list[0]  
 elif option == '2':  
 time = t\_list[1]  
 elif option == '3':  
 time = t\_list[2]  
 elif option == '4':  
 time = t\_list[3]  
 elif option == '5':  
 time = t\_list[4]  
 else:  
 print('Invalid numerical choice! Session terminated.\n')  
 break  
 for i in c\_list:  
 print(i)  
 class\_ans = input('Enter your desired travel class :\t')  
 if class\_ans == '1':  
 travel\_class = c\_list[0]  
 elif class\_ans == '2':  
 travel\_class = c\_list[1]  
 elif class\_ans == '3':  
 travel\_class = c\_list[2]  
 else:  
 print('Invalid numerical choice! Session terminated.\n')  
 break  
 c\_description = c\_status = 'N/A'  
 task = "insert into Customers\_Info values ('{}', '{}', '{}', '{}', '{}', '{}')".format(ticket\_ID, name, time, travel\_class, c\_description, c\_status)  
 cur.execute(task)  
 conn.commit()  
 print('\nFor your reference, your ticket ID is "{}".'.format(ticket\_ID))  
 print('Process successfully completed.\n')  
 elif u\_ans == '2':  
 check\_flag = check\_customer()  
 if check\_flag[0] is True:  
 new\_complaint = input('Enter the complaint below :\n').lower().capitalize()  
 task = "update Customers\_Info set Complaint='{}' where Ticket\_ID='{}'".format(new\_complaint, check\_flag[1])  
 cur.execute(task)  
 conn.commit()  
 side\_task = "update Customers\_Info set Complaint\_Status='{}' where Ticket\_ID='{}'".format('Pending', check\_flag[1])  
 cur.execute(side\_task)  
 conn.commit()  
 print('\nProcess successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif u\_ans == '3':  
 check\_flag = check\_customer()  
 if check\_flag[0] is True:  
 task = "update Customers\_Info set Complaint='{}' where Ticket\_ID='{}'".format('N/A', check\_flag[1])  
 cur.execute(task)  
 conn.commit()  
 side\_task = "update Customers\_Info set Complaint\_Status='{}' where Ticket\_ID='{}'".format('N/A', check\_flag[1])  
 cur.execute(side\_task)  
 conn.commit()  
 print('Process successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif u\_ans == '4':  
 check\_flag = check\_customer()  
 if check\_flag[0] is True:  
 u\_complaint = input('Enter the updated complaint below :\n').lower().capitalize()  
 task = "update Customers\_Info set Complaint='{}' where Ticket\_ID='{}'".format(u\_complaint, check\_flag[1])  
 cur.execute(task)  
 conn.commit()  
 side\_task = "update Customers\_Info set Complaint\_Status='{}' where Ticket\_ID='{}'".format('Pending', check\_flag[1])  
 cur.execute(side\_task)  
 conn.commit()  
 print('\nProcess successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif u\_ans == '5':  
 check\_flag = check\_customer()  
 if check\_flag[0] is True:  
 task = "select \* from Customers\_Info where Ticket\_ID='{}'".format(check\_flag[1])  
 cur.execute(task)  
 record = cur.fetchall()  
 side\_func('C')  
 for r in record:  
 print(r)  
 print('Process successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif u\_ans == '6':  
 check\_flag = check\_customer()  
 if check\_flag[0] is True:  
 task = "delete from Customers\_Info where Ticket\_ID='{}'".format(check\_flag[1])  
 cur.execute(task)  
 conn.commit()  
 print('Process successfully completed.\n')  
 else:  
 print('No Data Exists. Session terminated.\n')  
 break  
 elif u\_ans == '7':  
 print('Exiting system...')  
 flag = False  
 break  
 else:  
 print('Invalid numerical choice! Session terminated.\n')  
 break  
 elif choice == 'exit':  
 print('Exiting system...')  
 flag = False  
 break  
 else:  
 print('Please enter a valid response! (admin/user/exit)\n')  
 break  
 if flag is False:  
 break

PYTHON OUTPUTS

1) For Admin Interface

(Add employee data)

==============================WELCOME TO AIRLINES MANAGEMENT SYSTEM==============================

Are you an admin or a user ?

admin

Enter username : admin

Enter password : @dmin$77

1) ADD EMPLOYEE INFORMATION

2) DELETE EMPLOYEE INFORMATION

3) UPDATE EMPLOYEE INFORMATION

4) VIEW CERTAIN EMPLOYEE'S INFORMATION

5) VIEW ALL EMPLOYEES' INFORMATION

6) VIEW CERTAIN CUSTOMER'S INFORMATION / COMPLAINT

7) VIEW ALL CUSTOMERS' INFORMATION

8) UPDATE CUSTOMER COMPLAINT STATUS

9) DELETE CUSTOMER INFORMATION

10) QUIT

Enter your choice : 1

Enter employee's first name : Rohan

Enter employee's last name : Sharma

Enter employee's designation : Passenger Assistant

Enter employee's monthly salary : 90000

Enter employee's gender : Male

Enter employee's age : 28

Process successfully completed.

(View all employees’ information)

1) ADD EMPLOYEE INFORMATION

2) DELETE EMPLOYEE INFORMATION

3) UPDATE EMPLOYEE INFORMATION

4) VIEW CERTAIN EMPLOYEE'S INFORMATION

5) VIEW ALL EMPLOYEES' INFORMATION

6) VIEW CERTAIN CUSTOMER'S INFORMATION / COMPLAINT

7) VIEW ALL CUSTOMERS' INFORMATION

8) UPDATE CUSTOMER COMPLAINT STATUS

9) DELETE CUSTOMER INFORMATION

10) QUIT

Enter your choice : 5

Following data is in the order: (Employee ID, Employee Name, Designation, Monthly Salary, Gender, Age)

('E1394', 'Hritesh Yadav', 'Passenger assistant', 90000, 'Male', 30)

('E1589', 'Rohan Sharma', 'Passenger assistant', 90000, 'Male', 28)

('E2903', 'Rahul Gupta', 'Assistant supervisor', 120000, 'Male', 34)

('E3934', 'Neha Goyal', 'Airport manager', 150000, 'Female', 40)

('E9822', 'Rishita Jain', 'Assistant manager', 125000, 'Female', 36)

Process successfully completed.

2) For User Interface

(Book a flight)

==============================WELCOME TO AIRLINES MANAGEMENT SYSTEM==============================

Are you an admin or a user ?

user

1) BOOK A FLIGHT

2) ADD COMPLAINT

3) DELETE COMPLAINT

4) UPDATE COMPLAINT

5) VIEW YOUR INFORMATION / COMPLAINT STATUS

6) CANCEL A FLIGHT

7) QUIT

Enter your choice : 1

Enter first name : Lavanya

Enter last name : Saraf

Currently scheduled flights :

1) New Delhi to New York

2) Beijing to London

3) Johannesburg to Buenos Aires

4) Colombo to Bombay

5) Paras to Bali

Enter your choice : 4

1) BUSINESS CLASS

2) PREMIUM ECONOMY

3) ECONOMY CLASS

Enter your desired travel class : 2

For your reference, your ticket ID is "C9518".

Process successfully completed.

(Add complaint)

1) BOOK A FLIGHT

2) ADD COMPLAINT

3) DELETE COMPLAINT

4) UPDATE COMPLAINT

5) VIEW YOUR INFORMATION / COMPLAINT STATUS

6) CANCEL A FLIGHT

7) QUIT

Enter your choice : 2

Enter the Ticket ID : C9518

Enter the complaint below :

The food came later than the scheduled time and felt like it was stale.

Process successfully completed.

(View your information / complaint status)

1) BOOK A FLIGHT

2) ADD COMPLAINT

3) DELETE COMPLAINT

4) UPDATE COMPLAINT

5) VIEW YOUR INFORMATION / COMPLAINT STATUS

6) CANCEL A FLIGHT

7) QUIT

Enter your choice : 5

Enter the Ticket ID : C9518

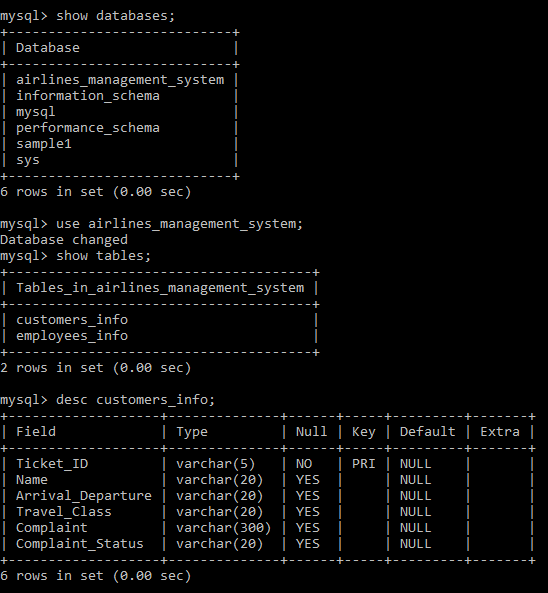
Following data is in the order: (Ticket ID, Customer Name, Arrival/Departure, Travel Class, Complaint, Complaint Status)

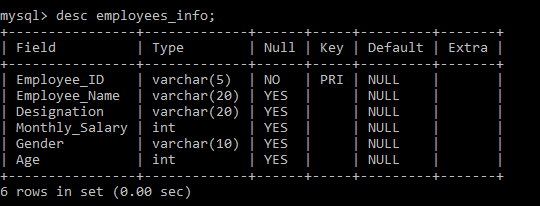
('C9518', 'Lavanya Saraf', '08:00 to 10:35', '2) PREMIUM ECONOMY', 'The food came later than the scheduled time and felt like it was stale.', 'Pending')

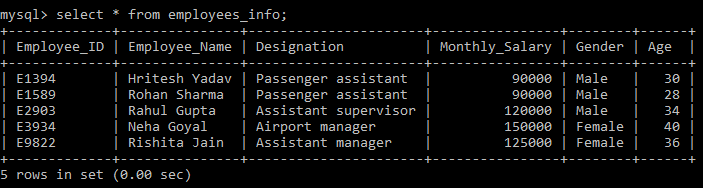
Process successfully completed.

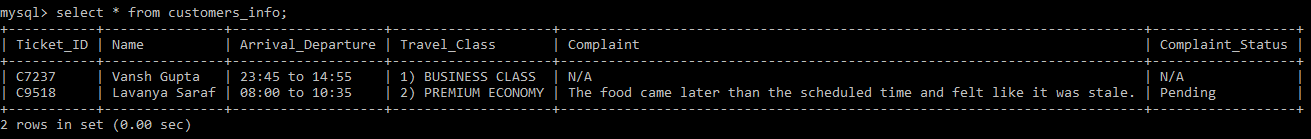
SQL OUTPUTS

(Database & tables created)





(Information inserted into “Employees\_Info” table)

(Information inserted into “Customers\_Info” table)

BIBLIOGRAPHY

* Valuable inputs from Mr. Siddharth Bhardwaj.
* *“Computer Science with Python”* by Sumita Arora.