**Airline management system**

**Using python &MySQL** 

**Prepared by:**

**Megha Agarwal**

**IN THIS**

**R**

**EPORT**

**:**

1.

Certificate

2.

Acknowledgement

3.

Synopsis

4.

Code

5.

Outputs

6.

Scope of

improvement

7.

Bibliography

CONTENTS

SYNOPSIS

The main aim of this project was to manage the booking end-to-end travel booking packages with the main focus including (includes Hotel, Taxi) and also keeping in mind the recent COVID-19 Standard OperatingSOPsedures (SOPs) by including salient safety features in the form of a questionnaire.

To develop and run this project we will be using:

1. Python
2. MySQL
3. MySQL ConnecAirline Management System (AMS) project is based on a menu drivenmenu-drivenen in the programming language Python,uage Python which accepts data from the user and stores it in a relational database written in MySQL. It is a Python-MySQL interface program.

The source code contains a total of 16 functions in which one function connects the source code to the MySQL database and one function displays the main menu which consists of fourteen options. The options in the main menu each have a function of their own and are executed when the user enters the number of the option they wish to execute when asked by the program which later gets stored in a variable. The main menu is repeatedly displayed using a while loop, one can exit from the loop by using the exit option in the main menu.

The main menu consists of the following options:

* 1. Create a new customer
  2. Display list of flights with destinations (from/to)
  3. Check availability of flights
  4. Book a flight
  5. Cancel a flight
  6. Reschedule a flight
  7. Add hotel/taxi
  8. Display all current customers
  9. Display all current Flights
  10. Create a new flight
  11. Check and Update if customer has done a COVID-19 test before flight booking
  12. Exit
  13. Contact Information
  14. Share Feedback

The database in MySQL contains 3 tables in which data entered by the user (customer name and details, booking details) is stored and some pre-existing data (in this case flight names and information) are stored. The 3 tables comprise of ams\_flight (stores flight names and information), ams\_customer (stores customer name and details) and ams\_booking (stores booking details generated by the program).

The AMS provides the customer with a wide variety of options such as viewing the available flights, booking a flight, rescheduling flights, canceling flights, booking a hotel/taxi, rating the overall experience, and being provided with proper contact information.

The system makes overall project management much easier and more flexible. It is designed to create a professional and friendly environment for the user to perform multiple operations.

CODE

## Airline Management System

import mysql. connector from mysql.connector import Error

#Connection def AMS\_Connect\_to\_db():

connection = mysql.connector.connect(host='localhost',database='ams', user='root',password='harshi')

if connection.is\_connected():

db\_Info = connection.get\_server\_info() print("Connected to MySQL Server version", db\_Info) cursor = connection.cursor() cursor.execute("select database();") record = cursor.fetchone() print("You're connected to database",record) return connection; else :

print("Error while connecting to MySQL")

#Display Main Menu def AMS\_Display\_Menu():

global choice

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

-----------------");

print("\*

\*");

print("\*

\*");

print("\* AIRLINE MANAGEMENT SYSTEM

\*");

print("\*

\*");

print("\*

\*");

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

-----------------"); print("Main Menu") print("1. Create a new customer") print("2. Display list of flights with destinations (from/to)") print("3. Check availability of flights") print("4. Book a flight ") print("5. Cancel a flight") print("6. Reschedule a flight") print("7. Add hotel/taxi") print("8. Display all current customers") print("9. Display all current Flights") print("10. Create a new flight")

print("11. Check and Update if customer has done a covid test before flight booking") print("12. Exit") print("13. Contact Information") print("14. Share Feedback"); print("\n Please enter your Choice (1-14)");

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

---------------");

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

---------------"); choice=input(); print(" You have selected option " ,choice) if choice.isdigit():

return choice; else :

return 0;

#AMS\_Get\_feedback def AMS\_Get\_feedback():

print(" We would like to have your feedback on your experience with the AMS Booking system") print(" Please answer the following five questions")

print(" Answer Scale 1..5 where 1 - Bad , 2 Average , 3 Good,4 V.

Good, 5 Excellent")

ch=1 while (ch==1) :

print(" Q1 : How was your overall experience with AMS ") choice=input(); if choice.isdigit(): if (int(choice)>0 and int(choice)<6): print (" Your entered",choice); ch=0; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average ,

3 Good,4 V. Good, 5 Excellent") ch=1; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average , 3 Good,4 V. Good, 5 Excellent") ch=1; ch=1 while (ch==1) :

print(" Q2 : How friendly was the AMS Menu") choice=input(); if choice.isdigit(): if (int(choice)>0 and int(choice)<6): print (" Your entered",choice);

ch=0; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average ,

3 Good,4 V. Good, 5 Excellent") ch=1; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average , 3 Good,4 V. Good, 5 Excellent") ch=1;

#Q3 ch=1 while (ch==1) :

print(" Q3 : How would you rate the Booking option ") choice=input(); if choice.isdigit(): if (int(choice)>0 and int(choice)<6): print (" Your entered",choice); ch=0; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average ,

3 Good,4 V. Good, 5 Excellent") ch=1;

else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average , 3 Good,4 V. Good, 5 Excellent") ch=1;

#Q4 ch=1 while (ch==1) :

print(" Q4 : How would you rate the Cancel option ") choice=input(); if choice.isdigit(): if (int(choice)>0 and int(choice)<6): print (" Your entered",choice); ch=0; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average ,

3 Good,4 V. Good, 5 Excellent") ch=1; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average , 3

Good,4 V. Good, 5 Excellent")

ch=1;

#Q5 ch=1 while (ch==1) :

print(" Q5 : How would you rate the Display Flight option") choice=input(); if choice.isdigit():

if (int(choice)>0 and int(choice)<6): print (" Your entered",choice); ch=0; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average ,

3 Good,4 V. Good, 5 Excellent") ch=1; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average , 3 Good,4 V. Good, 5 Excellent") ch=1;

#Q6 ch=1 while (ch==1) : print(" Q6 : How would you rate the Reschdule Flight option") choice=input(); if choice.isdigit(): if (int(choice)>0 and int(choice)<6): print (" Your entered",choice); ch=0; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average ,

3 Good,4 V. Good, 5 Excellent") ch=1; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average , 3 Good,4 V. Good, 5 Excellent") ch=1;

#Q7 ch=1 while (ch==1) :

print(" Q7 : How would you rate the responsiveness of the Menu") choice=input(); if choice.isdigit(): if (int(choice)>0 and int(choice)<6): print (" Your entered",choice);

ch=0; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average ,

3 Good,4 V. Good, 5 Excellent") ch=1; else:

print("Invalid Input");

print(" Answer Scale 1..5 where 1 - Bad , 2 Average , 3 Good,4 V. Good, 5 Excellent") ch=1; print(" Thank you for sharing your feedback ");

print(" We value your feedback ans look forward to serving you again")

#AMS\_Display\_Flights def AMS\_Display\_Flights(adf\_conn):

print(" You have selected the Display Flights option"); print(" Please enter your City of Departure"); From\_city=input(); print(" Please enter your city of Arrival"); To\_city=input(); print("Checking Flights From ",From\_city," To ",To\_city);

disp\_flt\_sql ="select

ams\_flt\_id,TIME\_FORMAT(ams\_flt\_departure\_time,'%T'),TIME\_FORMAT(ams\_f lt\_arrival\_time,'%T'),flt\_price from ams\_flight where ams\_flt\_from\_location='"+From\_city+"' and ams\_flt\_to\_location='"+To\_city+"'" ; mycursor = adf\_conn.cursor() mycursor.execute(disp\_flt\_sql); myresult = mycursor.fetchall(); if myresult==[]:

print("no flights available") else:

print("Flt ID\t\tDeparture time\tArrival Time\t Price");

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

-------"); for x in myresult:

print(x[0],'\t\t',x[1],'\t',x[2],'\t',x[3]);

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

---------");

#DisplayAllCustomers def AMS\_Display\_AllCustomers(adf\_conn):

print(" You have selected the Display All Flights option");

disp\_cust\_sql ="select first\_name,last\_name,phone\_num,email\_id from ams\_customer"; print(" Executing <",disp\_cust\_sql,">"); mycursor = adf\_conn.cursor() mycursor.execute(disp\_cust\_sql); myresult = mycursor.fetchall(); if myresult==[]:

print("no flights available") else:

print("Firstname".ljust(20,' '),"Lastname".ljust(20,' '),"Phone Num".ljust(20,' '),"Email Id".ljust(20,' '))

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

-------------------------------------"); for x in myresult:

print(x[0].ljust(20,' '),x[1].ljust(20,' '),x[2].ljust(20,'

'),x[3].ljust(20,' '));

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

--------------------------------------");

#DisplayAllFlights def AMS\_Display\_AllFlights(adf\_conn):

print(" You have selected the Display All Flights option");

disp\_flt\_sql ="select

ams\_flt\_id,ams\_flt\_from\_location,ams\_flt\_to\_location,TIME\_FORMAT(ams\_ flt\_departure\_time,'%T'),TIME\_FORMAT(ams\_flt\_arrival\_time,'%T'),flt\_p rice from ams\_flight"; print(" Executing <",disp\_flt\_sql,">"); mycursor = adf\_conn.cursor() mycursor.execute(disp\_flt\_sql); myresult = mycursor.fetchall(); if myresult==[]:

print("no flights available") else:

print("Flt ID\t","From".ljust(20,' '),"To".ljust(20,' '),"Departure Time".ljust(20,' '),"Arrival Time".ljust(20,'

'),"Price".ljust(20,' ')); print("----------------------------------------------------------

--------------------------------------"); for x in myresult:

print(x[0],'\t',x[1].ljust(20,' '),x[2].ljust(20,'

'),x[3].ljust(20,' '),x[4].ljust(20,' '),x[5]);

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

--------------------------------------");

#check flight availability def AMS\_Check\_Flgt\_Avail(adf\_conn,flt\_id):

chk\_flt\_avl\_sql="Select avail\_seats from ams\_flight where ams\_flt\_id=%s"; Available\_seats=0 mycursor = adf\_conn.cursor() mycursor.execute(chk\_flt\_avl\_sql,(flt\_id,)); myresult = mycursor.fetchall(); for x in myresult:

print(x)

Available\_seats=int(x[0]); print(' Available seats =',Available\_seats); return Available\_seats

#create customer def AMS\_Create\_Cust(adf\_conn):

print(" You are going to create a new customer");

ams\_customer values

print(" Enter your First name"); cfname=input(); print("Enter your Last name"); clname=input(); print("Enter your phone Number"); cphone=input(); print("Enter your email id"); cemail=input();

create\_cust\_sql= f"insert into

(first\_name,last\_name,phone\_num,email\_id) ('{cfname}','{clname}','{cphone}','{cemail}')"; print(" SQL=",create\_cust\_sql); mycursor = adf\_conn.cursor() mycursor.execute(create\_cust\_sql); mycursor.execute("commit;"); print("Customer created");

#CreateFlight def AMS\_Create\_flt(adf\_conn):

print(" You are going to create a new Flight"); print(" enter Airline name"); airline\_name=input(); print(" Enter your Source city"); from\_city=input(); print("Enter your Destination city"); to\_city=input();

your Destination city"); to\_city=input();

print("Enter departure time (HH:MM:SS)"); dep\_time=input(); print("Enter arrival time HH:MM:SS"); arr\_time=input(); print("Enter total seats in flight"); tot\_seats=input(); print("Enter Flight Price"); flight\_price=float(input());

## INSERT INTO AMS\_flight

create\_flt\_sql= f"INSERT INTO ams\_flight

(ams\_flt\_name,AMS\_FLT\_FROM\_LOCATION,AMS\_FLT\_TO\_LOCATION,AMS\_FLT\_DEPAR

TURE\_TIME,AMS\_FLT\_ARRIVAL\_TIME,AMS\_FLT\_SEATS,Avail\_seats,flt\_price)

VALUES

('{airline\_name}','{from\_city}','{to\_city}','{dep\_time}','{arr\_time}' ,'{tot\_seats}','{tot\_seats}','{flight\_price}')"; print(" SQL=",create\_flt\_sql); mycursor = adf\_conn.cursor() mycursor.execute(create\_flt\_sql); mycursor.execute("commit;"); print("New Flight created");

#Chk\_Covid\_Risk def Chk\_Covid\_Risk():

print("Have you done a covid antigen test which was negative in last 5 days"); test\_ok=input(); if (test\_ok=='Y') or (test\_ok=='y') :

print(" You can proceed with booking"); return 1;

else :

print(" Please answer a few questions to understand your covid risk profile ");

print(" Have you travelled internationally in the last 14 days"); ch=input(); if(ch=='Y') or (ch=='y'):

return 0;

print(" Have you come in contact with anyone who has travelled internationally during last 14 days"); ch=input(); if(ch=='Y') or (ch=='y'):

return 0; print(" Do you have any of the following symptions "); print("• Fever"); print("• Cough"); print("• Sore throat"); print("• Shortness of breath"); print("• Difficulty breathing"); print("• Chills"); print("• Muscle pain"); print("• Headache"); print("• GI symptoms"); print("• New loss of taste or smell");

ch=input(); if(ch=='Y') or (ch=='y'):

return 0;

#Book A FLIGHT def AMS\_Book\_Flgt(adf\_conn,flt\_id):

## first check the customer Covid Risk profile

cvd\_rsk=Chk\_Covid\_Risk(); if (cvd\_rsk==0):

print("We are unable to accept your booking as your Covid Risk Profile is High");

print(" Please try after 2 weeks and it is recommended to get your Covid test done"); return 0; else :

print(" Your Covid Risk is Low"); print(" You can now proceed with your booking");

## First Check Seat Availability avl\_seats=AMS\_Check\_Flgt\_Avail(adf\_conn,flt\_id); if (avl\_seats == 0):

print(" Seats not available on this flight"); return 0; else:

print("Are you an existing customer"); ch=input();

if(ch=='Y') or (ch=='y'):

print(" Please enter your customer id"); cid=input();

## INSERT INTO AMS\_BOOKING

create\_bk\_sql= f"insert into ams\_booking (ams\_booking\_cust\_id,booking\_flight\_id,booking\_seat\_num) values

({cid},{flt\_id},{avl\_seats})"; print("SQL=",create\_bk\_sql); mycursor = adf\_conn.cursor() mycursor.execute(create\_bk\_sql); mycursor.execute("commit;"); print("Row inserted"); ## UPDATE AMS\_FLIGHT

upd\_flt\_sql=f"update ams\_flight set avail\_seats = avail\_seats -1 where ams\_flt\_id={flt\_id}"; print(" SQL=",upd\_flt\_sql); mycursor = adf\_conn.cursor() mycursor.execute(upd\_flt\_sql); mycursor.execute("commit;"); ## DISPLAY BOOKING ID CREATED

disp\_book\_sql=f"select bookingid from ams\_booking where ams\_booking\_cust\_id = {cid}"; print(" SQL=",disp\_book\_sql); mycursor = adf\_conn.cursor() mycursor.execute(disp\_book\_sql); myresult = mycursor.fetchall(); for x in myresult:

print( "Booking ID =",x[0]) else:

print(" We need to first create your customer id"); return 1;

# AMS\_Cancel\_booking def AMS\_Cancel\_booking(adf\_conn,book\_id,cust\_id):

delete\_bk\_sql= f"Delete from ams\_booking where bookingid =

{book\_id} and ams\_booking\_cust\_id = {cust\_id}"; print(" SQL=",delete\_bk\_sql); mycursor = adf\_conn.cursor() mycursor.execute(delete\_bk\_sql); mycursor.execute("commit;"); return 1;

#AMS\_Display\_Contact\_info def AMS\_Display\_Contact\_info():

print("---------------------------------------------") print(" Airline Management System Pvt Ltd") print("---------------------------------------------") print(" Head Office : AMS-1 , Sector 45") print(" Gurgaon ,122001 "); print(" Haryana ,India");

print(" Contact number : +91-124-222210000 ") print("---------------------------------------------") print(" Branch Office : ZZ007, Z Block") print(" MG Marg"); print(" New Delhi, 110001") print(" India") print("---------------------------------------------") print(" International Office") print(" 100th Floor , Burj Khalifa") print(" Mall of Dubai street") print(" Dubai ,23156") print(" UAE") print("---------------------------------------------")

# AMS\_Add\_Taxi\_Hotel def AMS\_Add\_Taxi\_Hotel(main\_conn,book\_id,cust\_id,taxi\_r,hotel\_r):

add\_taxi\_hotel\_sql=f"update ams\_booking set taxi\_reqd='{taxi\_r}' , hotel\_reqd='{hotel\_r}' where BookingID =

{book\_id} and ams\_booking\_cust\_id ={cust\_id}" print(" SQL=",add\_taxi\_hotel\_sql); mycursor = main\_conn.cursor() mycursor.execute(add\_taxi\_hotel\_sql); mycursor.execute("commit;") return 1;

#AMS\_Change\_booking

def AMS\_Change\_booking(main\_conn,book\_id,cust\_id,new\_flgt\_id):

avl\_seats=AMS\_Check\_Flgt\_Avail(main\_conn,new\_flgt\_id); if (avl\_seats == 0):

print(" Seats not available on this flight"); return 0; else:

current\_flt\_sql = f"Select booking\_flight\_id from ams\_booking where BookingID ={book\_id} and ams\_booking\_cust\_id ={cust\_id}"; print(" SQL=",current\_flt\_sql); mycursor = main\_conn.cursor() mycursor.execute(current\_flt\_sql); myresult = mycursor.fetchall(); for x in myresult:

print( "Old Flt ID =",x[0]) old\_flt\_id=int(x[0]);

|  |  |
| --- | --- |
| change\_bk\_sql = f"Update ams\_booking | set |
| booking\_flight\_id={new\_flgt\_id},Booking\_seat\_num={avl\_seats}  BookingID = {book\_id} and ams\_booking\_cust\_id ={cust\_id}" | where |

print(" SQL=",change\_bk\_sql); mycursor = main\_conn.cursor() mycursor.execute(change\_bk\_sql); mycursor.execute("commit;");

## UPDATE AMS\_FLIGHT

upd\_flt\_sql=f"update ams\_flight set avail\_seats = avail\_seats

-1 where ams\_flt\_id={new\_flgt\_id}"; print(" SQL=",upd\_flt\_sql); mycursor = main\_conn.cursor() mycursor.execute(upd\_flt\_sql); mycursor.execute("commit;");

upd\_flt\_sql2=f"update ams\_flight set avail\_seats = avail\_seats

+ 1 where ams\_flt\_id={old\_flt\_id}"; print(" SQL=",upd\_flt\_sql2); mycursor = main\_conn.cursor() mycursor.execute(upd\_flt\_sql2); mycursor.execute("commit;"); return 1; try:

## present the Initial menu ch=1 otherch=1 main\_conn=AMS\_Connect\_to\_db(); while (ch<15):

ch=int(AMS\_Display\_Menu()); if (ch==1):

print(" Create a new customer")

AMS\_Create\_Cust(main\_conn);

elif (ch==2):

print(" Display Flights Selection");

AMS\_Display\_Flights(main\_conn);

elif (ch==3):

print(" Check Availability Selection"); print("please enter the Flight ID"); flt\_id=input(); result=AMS\_Check\_Flgt\_Avail(main\_conn,int(flt\_id)); elif (ch==4):

print(" Book a flight Selection"); print("please enter the Flight ID"); flt\_id=input(); result=AMS\_Book\_Flgt(main\_conn,int(flt\_id)); if ( result==0):

print(" We were unable to accept your Flight booking Request"); elif (ch==5):

print(" Cancel a flight Selection"); print("please enter the Booking ID"); book\_id=input(); print("please enter your customer ID"); cust\_id=input();

result=AMS\_Cancel\_booking(main\_conn,int(book\_id),int(cust\_id)); elif (ch==6):

print(" Reschedule a flight Selection");

print("please enter your Booking ID"); book\_id=input(); print("please enter your customer ID"); cust\_id=input();

print("please enter the revised flight id you are looking for "); new\_flgt\_id=input();

result=AMS\_Change\_booking(main\_conn,int(book\_id),int(cust\_id),int(new

\_flgt\_id)); elif (ch==7):

print(" Add Hotel/taxi Selection"); print("please enter your Booking ID"); book\_id=input(); print("please enter your customer ID"); cust\_id=input();

print("Do you want to include Taxi for Airport pickup/drop? Type Y or N "); taxi\_reqd=input();

print("Do you want to include Hotel in your booking? Type Y or N"); hotel\_reqd=input();

result=AMS\_Add\_Taxi\_Hotel(main\_conn,int(book\_id),int(cust\_id),taxi\_re qd,hotel\_reqd);

elif (ch==8): print(" Display all current customers")

AMS\_Display\_AllCustomers(main\_conn);

elif (ch==9):

print(" Display all current Flights")

AMS\_Display\_AllFlights(main\_conn);

elif (ch==10):

print(" Create a new flight")

AMS\_Create\_flt(main\_conn);

elif (ch==11):

print(" Check and Update if customer has done a covid test before flight booking") cvd\_rsk=Chk\_Covid\_Risk(); if (cvd\_rsk==0):

print(" Please try after 2 weeks and it is recommended to get your Covid test done"); else :

print(" Your Covid Risk is Low"); print(" You can now proceed with your booking");

elif (ch==12):

print("Exit Option Selection"); break;

elif (ch==13):

print("Contact Information Selection");

AMS\_Display\_Contact\_info();

elif (ch==14):

print("Share Feedback Option");

AMS\_Get\_feedback();

elif (ch==15):

print("You have selected Others Option which is presently disabled in Main menu"); otherch=0; while (otherch<6):

otherch=int(AMS\_Display\_OthersMenu()); print("Other Ch=",otherch) if (otherch==1):

print("1. Display all current Flight Bookings") AMS\_Display\_AllFlights(main\_conn); elif (otherch==2):

print("2. Display all current customers") AMS\_Display\_AllCustomers(main\_conn); elif (otherch==3):

print("3. Create a new customer") AMS\_Create\_Cust(main\_conn); elif (otherch==4):

print("4. Create a new flight") AMS\_Create\_flt(main\_conn); elif (otherch==5):

print("5. Check and Update if customer has done a covid test before flight booking") cvd\_rsk=Chk\_Covid\_Risk(); if (cvd\_rsk==0):

print(" Please try after 2 weeks and it is recommended to get your Covid test done"); else :

print(" Your Covid Risk is Low");

print(" You can now proceed with your booking"); elif (otherch==6):

print("6. Back to Main Menu") break; else:

print("Invalid Selection in others menu"); otherch=1;

else:

print("Invalid Selection"); ch=1; except Error as e:

print("Error ", e);

finally: if (main\_conn.is\_connected()):

main\_conn.close() print("MySQL connection is closed") print("Exiting AMS")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"); print(" Thank you for using Airline Management System ");

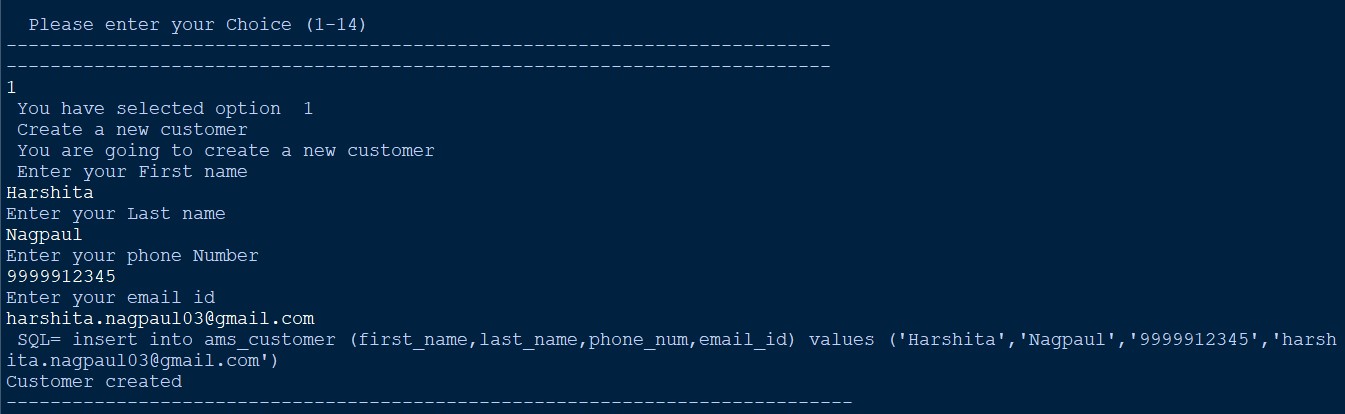
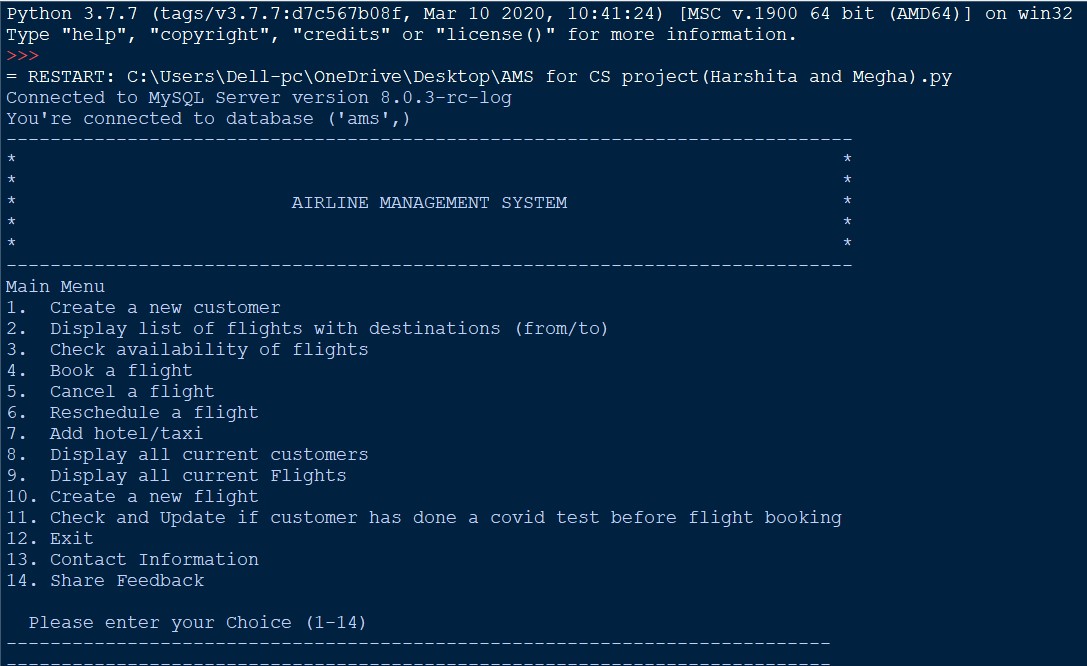
OUTPUTS

Main

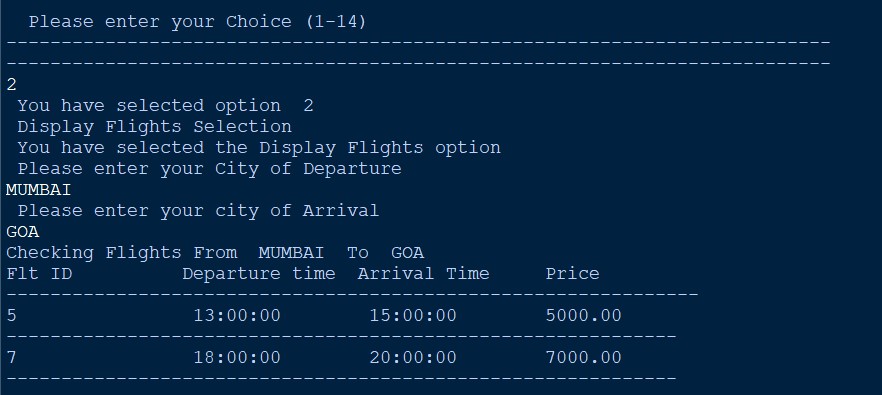
Menu

1.

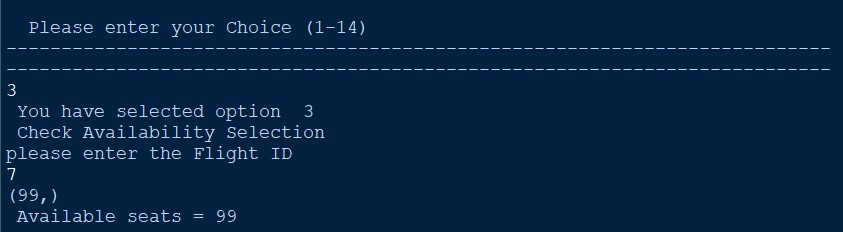
Create a new customer



1. Display list of flights with destinations (from/to)



1. Check availability of flights

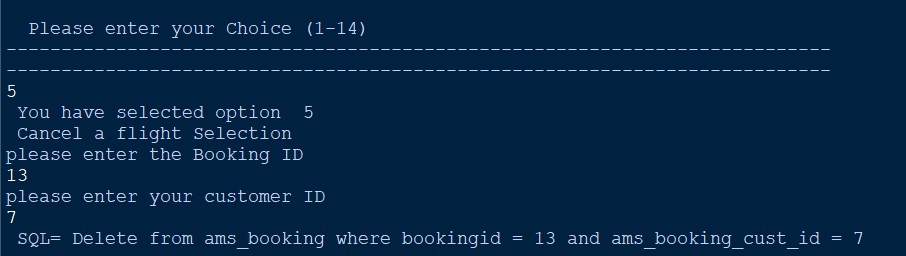
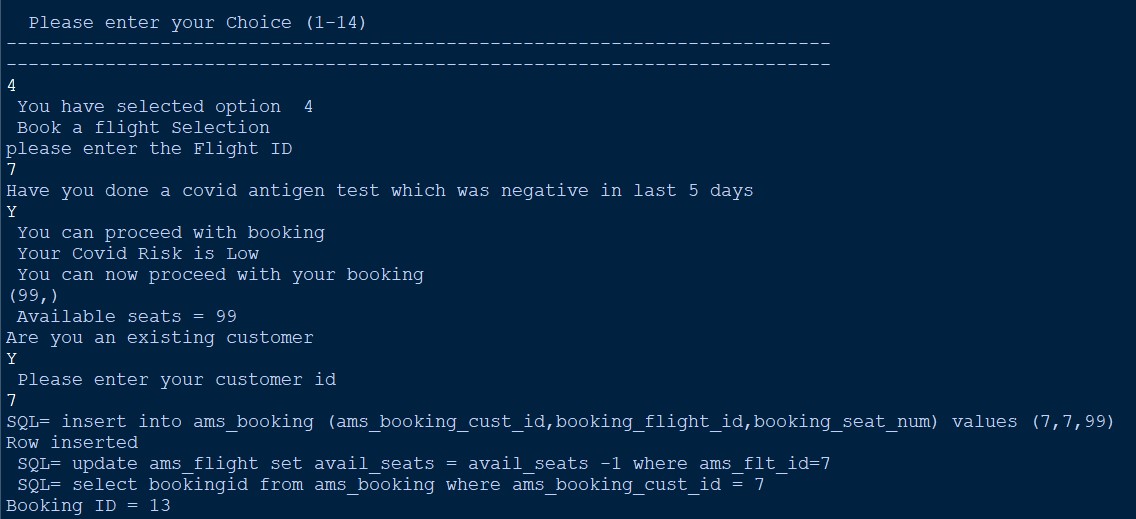


4.

Book a flight

5.

Cancel a flight

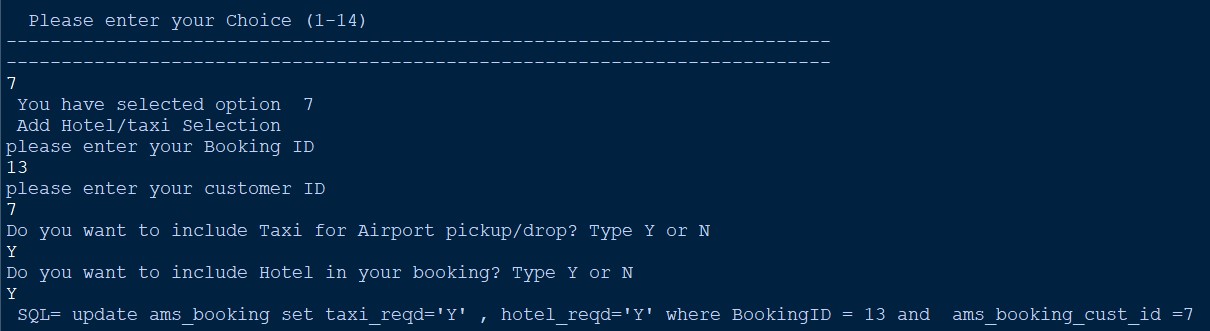


6.

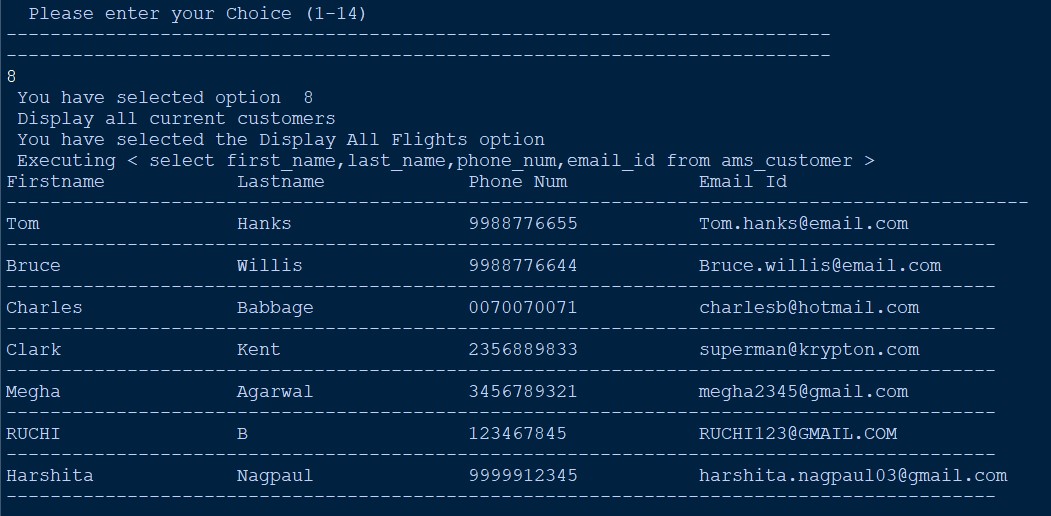
Reschedule a flight

7.

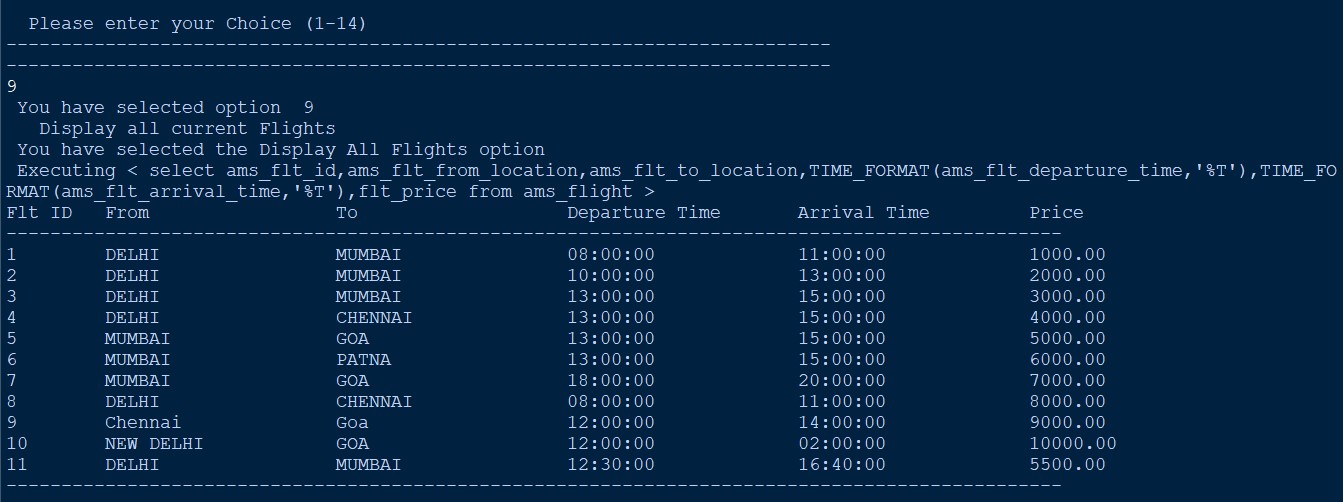
Add hotel/taxi



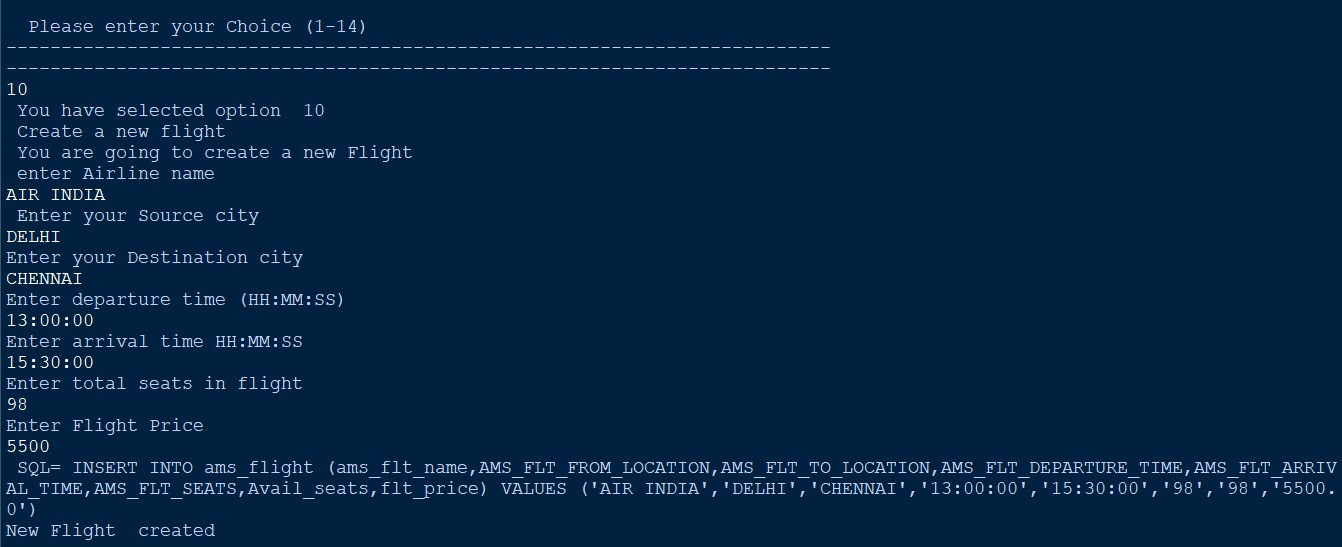
1. Display all current customers



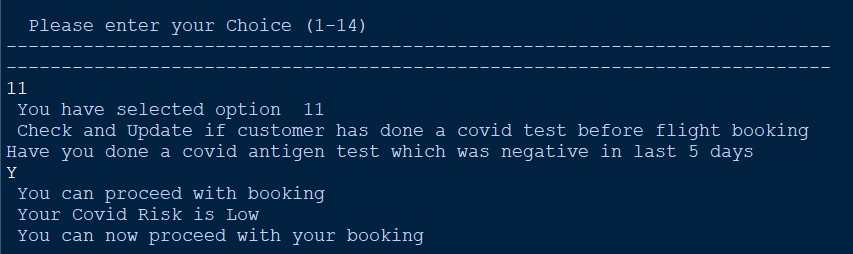
1. Display all current Flights



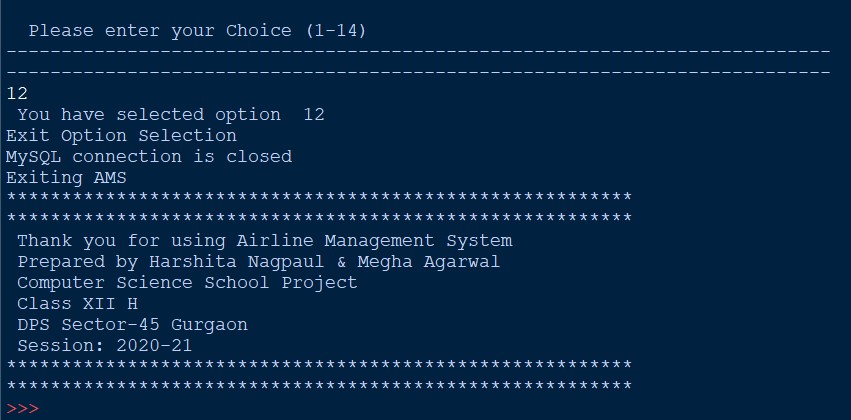
1. Create a new flight



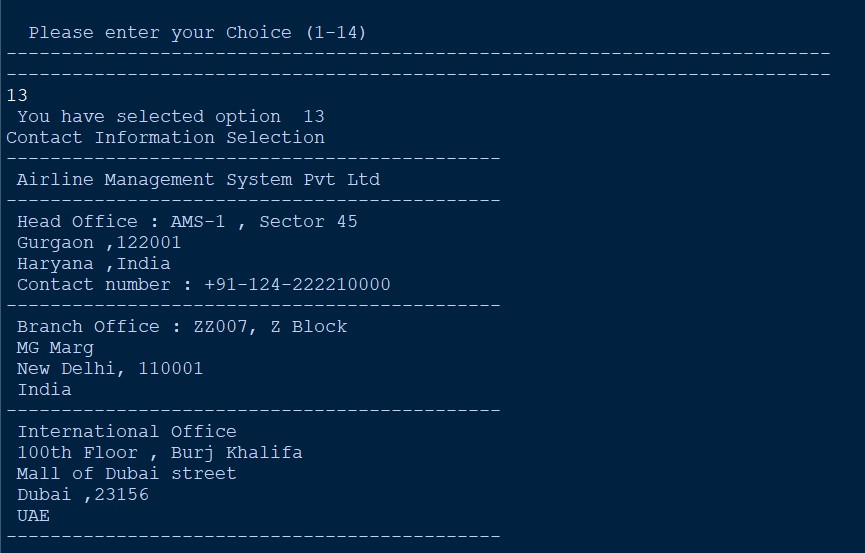
1. Check and Update if customer has done a COVID-19 test before flight booking



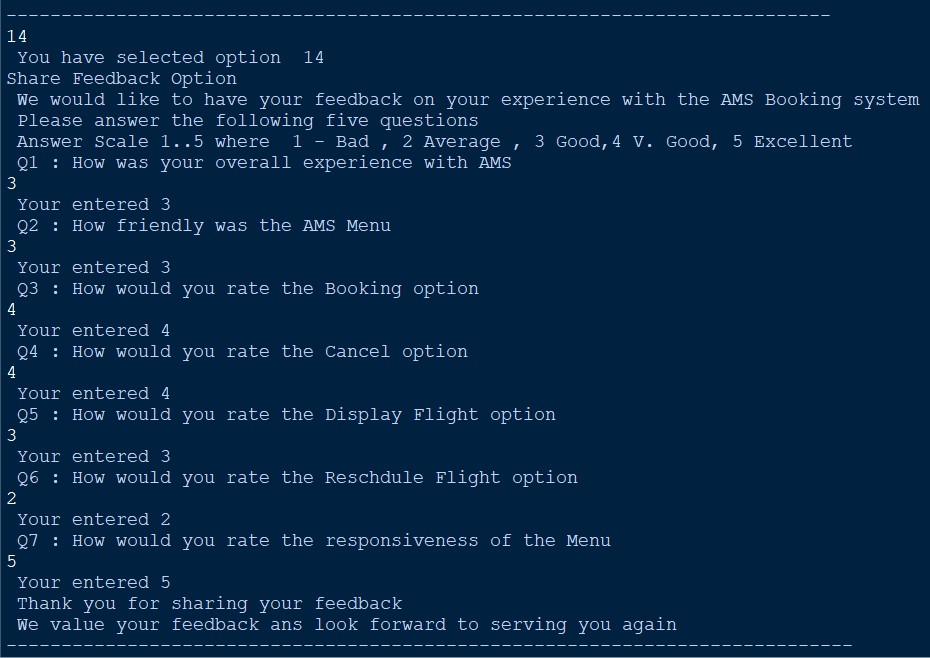
1. Exit



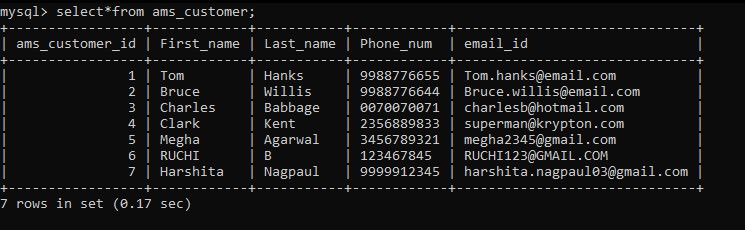
1. Contact Information



1. Share Feedback

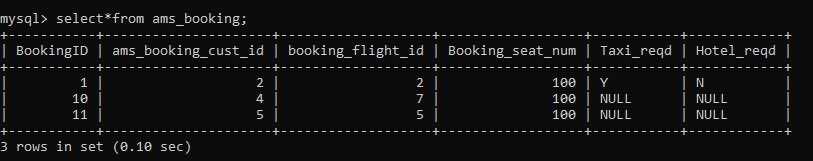
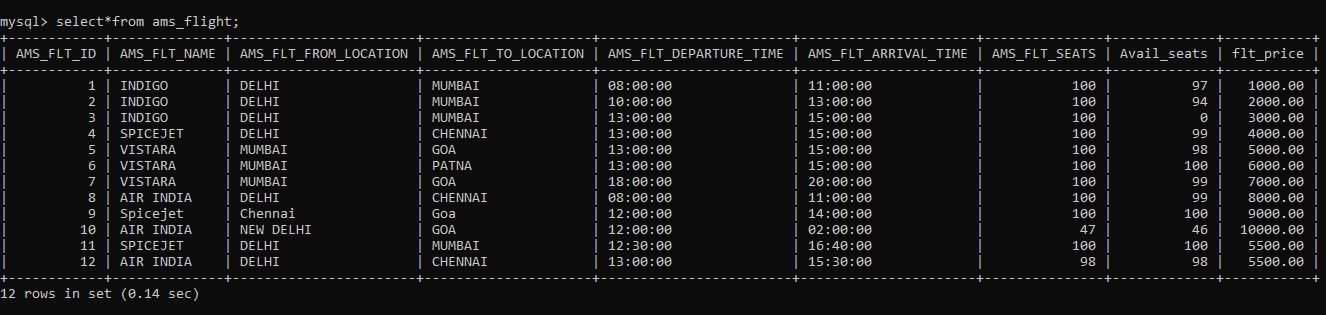


ams\_customer



ams\_flight

ams\_booking



# SCOPE OF IMPROVEMENT

Following are some of the possible areas of improvement in the project:

1. At present the user interface is Menu driven. A more friendly user interface which is Graphical in nature can be considered for the future.

1. The overall error handling can be improved further to ensure invalid inputs are rejected and displays prompts/messages to the user when he enters invalid inputs so that he can next time enter correct inputs.

1. The program’s output formatting can be enhanced using table-based formatting for better presentation.

1. The database can be enhanced to store more information about customer like his address and common customer preferences. More information about the flight can be stored in the database and displayed like split of first class and economy class.

1. More functions can be added to the code to make it more modular.

Keeping in mind the changing choices of modern-day customers, valid improvements can be incorporated from time to time to increase the user catering capability of the Airline Management System.

# BIBLIOGRAPHY

## TEXTBOOKS

* Saraswati Computer Science with Python for Class 12th

By Reeta Sahoo and Gagan Sahoo

* Computer Science with Python for Class 12th By Sumita Arora
* Computer Science with Python for Class 12th By Preeti Arora

## WEBSITES

* [https://www.google.com](https://www.google.com/)
* [https://www.stackoverflow.com](https://www.stackoverflow.com/)
* <https://docs.python.org/3/>