

AIRLINE MANAGEMENT SYSTEM

Computer Science
Project

Harshita Nagpaul

XII-H

2020-21





CONTENTS



IN THIS REPORT:

1. Certificate
2. Acknowledgement
3. Synopsis
4. Code
5. Outputs
6. Scope of improvement
7. Bibliography



Certificate

This is to certify that Harshita Nagpaul of class XII-H of Delhi Public School, Sector-45, Gurgaon has satisfactorily completed her project on the topic:

Airline Management System

Under the guidance of Ms. Ruchi Baweja in the academic year 2020-21.

Signature of
Internal teacher

Signature of
External Examiner

ACKNOWLEDGEMENT

I would like to express a deep sense of thanks and gratitude to my Computer Science teacher Ms. Ruchi Baweja for guiding me immensely through the course of the project. She always expressed keen interest in my work. Her constructive advice and constant motivation have been responsible for the successful completion of the project.

I would like to thank my project partner Megha Agarwal for her timely help and support for compilation of this project.

I would also like to thank my parents for their motivation and support for the completion of this project.

Last but not the least, I would like to thank all those who had helped me directly or indirectly towards the completion of this project.

SYNOPSIS

The main aim of this project was to manage the booking end-to-end travel booking packages with main focus on Airlines (includes Hotel, Taxi) and also keeping in mind the recent COVID-19 Standard Operating Procedures (SOP's) 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 Connector

The project on Airline Management System (AMS) is based on a menu driven code written in programming language 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 a wide variety of options such as viewing the available flights, booking a flight, rescheduling flights, cancelling flights, booking hotel/taxi, rating the overall experience and being provided with proper contact information.

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

CODE

```
## Airline Management System

## Class XII Computer Science project

## Prepared by Harshita Nagpaul & Megha Agarwal


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(displflt_sql);

myresult = mycursor.fetchall();

if myresult==[]:

    print("no flights available")

else:

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

    print("-----")
    print("-----");

    for x in myresult:

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

        print("-----")
    print("-----");

#DisplayAllCustomers

def AMS_Display_AllCustomers(adf_conn):

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

    displcust_sql ="select first_name,last_name,phone_num,email_id from
ams_customer";

    print(" Executing <",displcust_sql,>");

    mycursor = adf_conn.cursor()

    mycursor.execute(displcust_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");

```

```

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          ams_customer
(first_name,last_name,phone_num,email_id)          values
('{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();

```

```

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(disb_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_flg_id):
    avl_seats=AMS_Check_Flg_Avail(main_conn,new_flg_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_flg_id},Booking_seat_num={avl_seats}      where
BookingID = {book_id} and  ams_booking_cust_id={cust_id}"

            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_flg_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_flggt_id=input();

result=AMS_Change_booking(main_conn,int(book_id),int(cust_id),int(new
_flggt_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 ");
    print(" Prepared by Harshita Nagpaul & Megha Agarwal");
    print(" Computer Science School Project");
    print(" Class XII H");
    print(" DPS Sector-45 Gurgaon ");
    print(" Session: 2020-21");

print("*****");

print("*****");

```

OUTPUTS

Main Menu

```
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Dell-pc\OneDrive\Desktop\AMS for CS project(Harshita and Megha).py
Connected to MySQL Server version 8.0.3-rc-log
You're connected to database ('ams',)
-----
*                                                                    *
*                                                                    *
*              AIRLINE MANAGEMENT SYSTEM                            *
*                                                                    *
*                                                                    *
-----
Main Menu
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 test before flight booking
12. Exit
13. Contact Information
14. Share Feedback

Please enter your Choice (1-14)
-----
```

1. Create a new customer

```
Please enter your Choice (1-14)
-----
1
You have selected option 1
Create a new customer
You are going to create a new customer
Enter your First name
Harshita
Enter your Last name
Nagpaul
Enter your phone Number
9999912345
Enter your email id
harshita.nagpaul03@gmail.com
SQL= insert into ams_customer (first_name,last_name,phone_num,email_id) values ('Harshita','Nagpaul','9999912345','harshita.nagpaul03@gmail.com')
Customer created
-----
```

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

```
Please enter your Choice (1-14)
-----
2
You have selected option 2
Display Flights Selection
You have selected the Display Flights option
Please enter your City of Departure
MUMBAI
Please enter your city of Arrival
GOA
Checking Flights From MUMBAI To GOA
Flt ID      Departure time  Arrival Time    Price
-----
5           13:00:00      15:00:00        5000.00
-----
7           18:00:00      20:00:00        7000.00
-----
```

3. Check availability of flights

```
Please enter your Choice (1-14)
-----
3
You have selected option 3
Check Availability Selection
please enter the Flight ID
7
(99,)
Available seats = 99
```

4. Book a flight

```
Please enter your Choice (1-14)
-----
-----
4
You have selected option  4
Book a flight Selection
please enter the Flight ID
7
Have you done a covid antigen test which was negative in last 5 days
Y
You can proceed with booking
Your Covid Risk is Low
You can now proceed with your booking
(99,)
Available seats = 99
Are you an existing customer
Y
Please enter your customer id
7
SQL= insert into ams_booking (ams_booking_cust_id,booking_flight_id,booking_seat_num) values (7,7,99)
Row inserted
SQL= update ams_flight set avail_seats = avail_seats -1 where ams_flt_id=7
SQL= select bookingid from ams_booking where ams_booking_cust_id = 7
Booking ID = 13
```

5. Cancel a flight

```
Please enter your Choice (1-14)
-----
-----
5
You have selected option  5
Cancel a flight Selection
please enter the Booking ID
13
please enter your customer ID
7
SQL= Delete from ams_booking where bookingid = 13 and ams_booking_cust_id = 7
```

6. Reschedule a flight

```
Please enter your Choice (1-14)
-----
6
You have selected option 6
Reschedule a flight Selection
please enter your Booking ID
13
please enter your customer ID
7
please enter the revised flight id you are looking for
5
(99,)
Available seats = 99
SQL= Select booking_flight_id from ams_booking where BookingID =13 and ams_booking_cust_id =7
Old Flt ID = 7
SQL= Update ams_booking set booking_flight_id=5,Booking_seat_num=99 where BookingID = 13 and ams_booking_cust_id =7
SQL= update ams_flight set avail_seats = avail_seats -1 where ams_flt_id=5
SQL= update ams_flight set avail_seats = avail_seats + 1 where ams_flt_id=7
```

7. Add hotel/taxi

```
Please enter your Choice (1-14)
-----
7
You have selected option 7
Add Hotel/taxi Selection
please enter your Booking ID
13
please enter your customer ID
7
Do you want to include Taxi for Airport pickup/drop? Type Y or N
Y
Do you want to include Hotel in your booking? Type Y or N
Y
SQL= update ams_booking set taxi_reqd='Y' , hotel_reqd='Y' where BookingID = 13 and ams_booking_cust_id =7
```

8. Display all current customers

```
Please enter your Choice (1-14)
-----
8
You have selected option 8
Display all current customers
You have selected the Display All Flights option
Executing < select first_name,last_name,phone_num,email_id from ams_customer >
Firstname      Lastname      Phone Num      Email Id
-----
Tom            Hanks         9988776655     Tom.hanks@email.com
-----
Bruce          Willis        9988776644     Bruce.willis@email.com
-----
Charles        Babbage       0070070071     charlesb@hotmail.com
-----
Clark          Kent          2356889833     superman@krypton.com
-----
Megha          Agarwal       3456789321     megha2345@gmail.com
-----
RUCHI          B             123467845      RUCHI123@GMAIL.COM
-----
Harshita       Nagpaul       9999912345     harshita.nagpaul03@gmail.com
-----
```

9. Display all current Flights

```
Please enter your Choice (1-14)
-----
9
You have selected option 9
Display all current Flights
You have selected the Display All Flights option
Executing < select ams_flight_id,ams_flight_from_location,ams_flight_to_location,TIME_FORMAT(ams_flight_departure_time,'%T'),TIME_FORMAT(ams_flight_arrival_time,'%T'),flight_price from ams_flight >
Flt ID  From      To      Departure Time      Arrival Time      Price
-----
1       DELHI     MUMBAI   08:00:00            11:00:00          1000.00
2       DELHI     MUMBAI   10:00:00            13:00:00          2000.00
3       DELHI     MUMBAI   13:00:00            15:00:00          3000.00
4       DELHI     CHENNAI  13:00:00            15:00:00          4000.00
5       MUMBAI    GOA      13:00:00            15:00:00          5000.00
6       MUMBAI    PATNA    13:00:00            15:00:00          6000.00
7       MUMBAI    GOA      18:00:00            20:00:00          7000.00
8       DELHI     CHENNAI  08:00:00            11:00:00          8000.00
9       Chennai  Goa      12:00:00            14:00:00          9000.00
10      NEW DELHI GOA      12:00:00            02:00:00          10000.00
11      DELHI     MUMBAI   12:30:00            16:40:00          5500.00
-----
```

10. Create a new flight

```
Please enter your Choice (1-14)
-----
-----
10
You have selected option 10
Create a new flight
You are going to create a new Flight
enter Airline name
AIR INDIA
Enter your Source city
DELHI
Enter your Destination city
CHENNAI
Enter departure time (HH:MM:SS)
13:00:00
Enter arrival time HH:MM:SS
15:30:00
Enter total seats in flight
98
Enter Flight Price
5500
SQL= INSERT INTO ams_flight (ams_flight_name,AMS_FLT_FROM_LOCATION,AMS_FLT_TO_LOCATION,AMS_FLT_DEPARTURE_TIME,AMS_FLT_ARRIVAL_TIME,AMS_FLT_SEATS,Avail_seats,flt_price) VALUES ('AIR INDIA','DELHI','CHENNAI','13:00:00','15:30:00','98','98','5500.0')
New Flight created
```

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

```
Please enter your Choice (1-14)
-----
-----
11
You have selected option 11
Check and Update if customer has done a covid test before flight booking
Have you done a covid antigen test which was negative in last 5 days
Y
You can proceed with booking
Your Covid Risk is Low
You can now proceed with your booking
```

12. Exit

```
Please enter your Choice (1-14)
-----
12
You have selected option 12
Exit Option Selection
MySQL connection is closed
Exiting AMS
*****
*****
Thank you for using Airline Management System
Prepared by Harshita Nagpaul & Megha Agarwal
Computer Science School Project
Class XII H
DPS Sector-45 Gurgaon
Session: 2020-21
*****
*****
>>>
```

13. Contact Information

```
Please enter your Choice (1-14)
-----
13
You have selected option 13
Contact Information Selection
-----
Airline Management System Pvt Ltd
-----
Head Office : AMS-1 , Sector 45
Gurgaon ,122001
Haryana ,India
Contact number : +91-124-222210000
-----
Branch Office : ZZ007, Z Block
MG Marg
New Delhi, 110001
India
-----
International Office
100th Floor , Burj Khalifa
Mall of Dubai street
Dubai ,23156
UAE
-----
```


14. Share Feedback

```
-----
14
You have selected option 14
Share Feedback Option
We would like to have your feedback on your experience with the AMS Booking system
Please answer the following five questions
Answer Scale 1..5 where 1 - Bad , 2 Average , 3 Good, 4 V. Good, 5 Excellent
Q1 : How was your overall experience with AMS
3
Your entered 3
Q2 : How friendly was the AMS Menu
3
Your entered 3
Q3 : How would you rate the Booking option
4
Your entered 4
Q4 : How would you rate the Cancel option
4
Your entered 4
Q5 : How would you rate the Display Flight option
3
Your entered 3
Q6 : How would you rate the Reschedule Flight option
2
Your entered 2
Q7 : How would you rate the responsiveness of the Menu
5
Your entered 5
Thank you for sharing your feedback
We value your feedback and look forward to serving you again
-----
```

ams_customer

```
mysql> select*from ams_customer;
+-----+-----+-----+-----+-----+
| ams_customer_id | First_name | Last_name | Phone_num | email_id |
+-----+-----+-----+-----+-----+
| 1 | Tom | Hanks | 9988776655 | Tom.hanks@email.com |
| 2 | Bruce | Willis | 9988776644 | Bruce.willis@email.com |
| 3 | Charles | Babbage | 0070070071 | charlesb@hotmail.com |
| 4 | Clark | Kent | 2356889833 | superman@krypton.com |
| 5 | Megha | Agarwal | 3456789321 | megha2345@gmail.com |
| 6 | RUCHI | B | 123467845 | RUCHI123@GMAIL.COM |
| 7 | Harshita | Nagpaul | 9999912345 | harshita.nagpaul03@gmail.com |
+-----+-----+-----+-----+-----+
7 rows in set (0.17 sec)
```

ams_flight

```
mysql> select*from ams_flight;
```

AMS_FLT_ID	AMS_FLT_NAME	AMS_FLT_FROM_LOCATION	AMS_FLT_TO_LOCATION	AMS_FLT_DEPARTURE_TIME	AMS_FLT_ARRIVAL_TIME	AMS_FLT_SEATS	Avail_seats	flt_price
1	INDIGO	DELHI	MUMBAI	08:00:00	11:00:00	100	97	1000.00
2	INDIGO	DELHI	MUMBAI	10:00:00	13:00:00	100	94	2000.00
3	INDIGO	DELHI	MUMBAI	13:00:00	15:00:00	100	0	3000.00
4	SPICEJET	DELHI	CHENNAI	13:00:00	15:00:00	100	99	4000.00
5	VISTARA	MUMBAI	GOA	13:00:00	15:00:00	100	98	5000.00
6	VISTARA	MUMBAI	PATNA	13:00:00	15:00:00	100	100	6000.00
7	VISTARA	MUMBAI	GOA	18:00:00	20:00:00	100	99	7000.00
8	AIR INDIA	DELHI	CHENNAI	08:00:00	11:00:00	100	99	8000.00
9	Spicejet	Chennai	Goa	12:00:00	14:00:00	100	100	9000.00
10	AIR INDIA	NEW DELHI	GOA	12:00:00	02:00:00	47	46	10000.00
11	SPICEJET	DELHI	MUMBAI	12:30:00	16:40:00	100	100	5500.00
12	AIR INDIA	DELHI	CHENNAI	13:00:00	15:30:00	98	98	5500.00

```
12 rows in set (0.14 sec)
```

ams_booking

```
mysql> select*from ams_booking;
```

BookingID	ams_booking_cust_id	booking_flight_id	Booking_seat_num	Taxi_reqd	Hotel_reqd
1	2	2	100	Y	N
10	4	7	100	NULL	NULL
11	5	5	100	NULL	NULL

```
3 rows in set (0.10 sec)
```



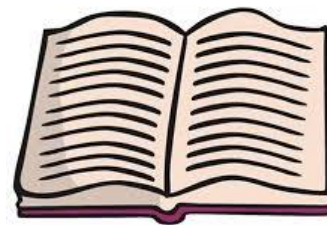
SCOPE OF IMPROVEMENT

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.
2. The overall error handling can be improved further to ensure invalid inputs are rejected and displaying prompts/messages to user when he enters invalid inputs so that next time, he can enter correct inputs.
3. The formatting of the output for the program can be enhanced using table-based formatting for better presentation.
4. 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.
5. More functions can be added to the code to make it more modular.

Keeping in mind the changing choices of the 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.stackoverflow.com>
- <https://docs.python.org/3/>

