2021-2022

ONLINE FLIGHT BOOKING



**DONE BY:**

**AYUSHI AGARWAL**





(Affiliated to Central Board of Secondary Education, New Delhi)

(Chettinad House, R.A.Puram, Chennai – 600 028)

# COMPUTER SCIENCE

Certified to be the Bonafide Record of work done by

Ayushi Agarwal\_\_\_\_\_\_\_\_\_ of Std XII Sec \_\_\_\_\_\_\_E1\_\_\_\_\_\_

in the Computer Science Lab of the CHETTINAD VIDYASHRAM,

CHENNAI, during the year 2021 – 2022.

Date: Teacher-in-charge

REGISTER NO. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### **Submitted for All India Senior Secondary Practical Examination in**

#### **Computer Science held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_at**

#### **Chettinad Vidyashram, Chennai – 600 028.**

#### **Principal Internal Examiner External Examiner**

**ACKNOWLEDGEMENT**

I would like to express my sincere thanks to Meena Aunty, Principal Mrs. S.Amudhalakshmi for their encouragement and support to work on this Project. I am grateful to my computer science teacher Mrs. V.Mala and to the computer science department for the constant guidance and support to complete the project.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| TOPIC | PAGE NO. |
| Overview of Python | 4 |
| Project Description | 7 |
| Functions Used | 8 |
| Database & Files Used | 10 |
| Files Used | 11 |
| Source Code | 12 |
| Sample Outputs | 36 |
| Conclusion | 49 |
| Bibliography | 50 |

**OVERVIEW OF PYTHON**

Python is a high-level, Interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords whereas other languages use punctuation and it has fewer syntactical construction than other languages.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language −** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**HISTORY OF PYTHON**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**FEATURES OF PYTHON**

Features of Python are listed as below.

* **Easy-to-learn −** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read −** Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain −** Python's source code is fairly easy to maintain.
* **A broad standard library −** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode −** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable −** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable −** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases −** Python provides interfaces to all major commercial databases.
* **GUI Programming −** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable −** Python provides a better structure and support for large programs than shell scripting.

Apart from the general features, Python has a big list of other good features that makes it a special language, as listed below –

* It supports functional and structured programming methods as well as Object Oriented Programming (OOP).
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**PROJECT DESCRIPTION**

This project is based on Online flight reservation system. User need to create his account on our website “MakeMyAirTrip” and can avail a wide variety of feature offerings by the website. User account is fully secured and protected by password for any unauthorised access. Once account is created, it can be accessed from anywhere, anytime.

Online flight reservation system offers – display of different airline schedules & timings for different destinations, tickets availability status, online reservation of flight tickets (max 4 passengers on single ticket), booking cancellations, processing of refunds, selection of travel class (Economy, Premium, Business), selection of preferences (Regular, Armed forces, Senior citizen, Infant) for availing the discounts in fare, pre-book food facility, Web check-in, seat selection, a wide variety of payment options, e-tickets etc.

A wide variety of vegetarian and non-vegetarian food options are available for pre-booking. As per government regulations Web check-in is mandatory prior to travel, this system offers to do Web check-in at the time of ticket booking itself & allow users to select from available seats based on his selection of travel class.

User will get consolidated fare summary based on number of passengers, applicable fare discounts, GST, online booking charges, pre-booked food prices etc. He will be able to make online payments using Cards, Google Pay and Paytm.

The aim of this project is to offer a user-friendly website to the air travellers that they can use to do hassle-free and secure travel bookings from the comfort of their home.

Refer to our Terms and conditions page before booking tickets.

To know more about us please visit About Us Page.

**FUNCTIONS USED**

1. **ENTER\_USER\_NAME():** Function take input as “User name” for Account creation & Login, Empty string not allowed.
2. **CREATE\_ACCOUNT():** Function enable users to create their account on website by entering “User name” & “Password”. Details are stored in a table “LoginDetails” for validation of user identity for next time Login. “User name” already exit in table are not allowed.
3. **LOGIN\_ACCOUNT():** Function enables users to Login their account by entering correct “User name” & “Password”. For wrong entries, it will not allow Login & will display appropriate messages.
4. **TRAVELLER\_DETAILS():** Function captures the traveler details (Name & Gender). Details are stored in a table “PassengerDetails”.
5. **SHOW\_TRAVELLER\_DETAILS():** Function displays traveler details ( Name & Gender) reading from “PassengerDetails” table.
6. **TRAVELLER\_DETAILS2():** Function captures traveler details (mobile number & email id) in global variables.
7. **TRAVEL\_FROM\_TO():** Functions captures travel details (From, To, Date, Class, Category).
8. **read():** Function displays travel details reading from file “Booking”
9. **CALCULATE\_FARE(Travel\_class,Travel\_category):** Function calculates fare based on Travel class & Category.
10. **FLIGHT\_SELECTION():** Function captures user inputs for flight selection (Flight Name, Flight No, Dept Time) from “Availability” table.
11. **Eticket(BookingStatus):** Function displays E-Ticket for Booking & Cancellation.
12. **FOOD():** Function offers Pre-book food selections to user.
13. **about\_us():** Function writes text file (about\_us.txt), read back and display general information about website.
14. **travel\_rules():** Function writes text file (travel\_rules.txt), read back & display travel Rules & Regulatiions.
15. **refund\_policy():** Function writes text file (refund.txt), read back & display refund related policies in the case of cancellation.
16. **INPUTING\_FLIGHT\_DETAILS():** Function loads data in “Availability” table.
17. **SHOW\_AVAILABLE\_FLIGHT():** Function displays flight availability details reading from “Availability” table.
18. **PAYMENT(Total\_amt):** Function offer different mode of payment options to user.
19. **WEBCHK():** Function offer Web check-in options to user. Including seat selections based on travel Class opted by user.
20. **CANCELATION():** Function offers booking cancellation facility to user.

**DATABASE USED**

Database is collection of interrelated data stored together to serve multiple applications. DBMS or RDBMS e.g. MySQL is a software responsible for storing, maintaining & utilizing database. Operations with DBMS are create, save, display, search, update, insert, delete etc. Data model used is relational database model, in which data is organised in tables i.e. rows & columns.

Name of the database used in Project for SQL connectivity & storing tables is “**Reservation**”.

**RELATIONS USED**

Relation / Table is a collection of tuples (row).

1. LoginDetails : Table stores User name & Password to validate user

identity for future Login

1. PassengerDetails : Table stores Passenger Name & Gender
2. Availability : Table stores Flight related details (Name, Number,

Departure time, Ticket availability)

**FILES USED**

A text file stores information in the form of a stream of ASCII or Unicode characters. In text files, each line of text is terminated (delimited) with a special character (as per the Operating System) known as EOL (End of Line) character. In text files, some internal translations take place when this EOL character is read or written. In Python by default this EOL character is the newline character (‘\n’) or carriage-return.

Text Files used in the Project are as listed below.

1. Booking.txt : It is a file used to store & display booking details
2. Intro.txt : It is a file stores introductory details of our website
3. about\_us.txt : It is a file stores general detail about us
4. travel\_rules.txt : It is a file stores travel related rules & regulations
5. refund.txt : It is a file stores refund related policies

**SOURCE CODE**

#Capture User Name for Account creation, Empty string not allowed

def ENTER\_USER\_NAME():

U=input("Enter Username: ") #User name

if U=='':

print("User name can not be empty...Retry")

U=input("\nEnter Username: ") #User name

if U=='':

print("Number of retry exceeded...")

exit()

return(U)

#Create User Account, Store Name & Password in table

#User Name already existing is Not allowed

def CREATE\_ACCOUNT():

cur.execute("create table if not exists LoginDetails(UserName char(25),Account\_password char(20))")

U=ENTER\_USER\_NAME()

cur.execute("select Account\_password from LoginDetails where UserName like '%s'"%(U))

data=cur.fetchall() #Fetch data from table to check if User exits

if any(data):

print("Username already exist... Try another name !!!\n")

U=ENTER\_USER\_NAME()

cur.execute("select Account\_password from LoginDetails where UserName like '%s'"%(U))

data=cur.fetchall() #Fetch data from table to check if User exits

if any(data):

print("Username already exist... Try another name !!!")

exit()

Pa=input("Enter Password: ") #Password

if Pa=='':

print("Password can not be empty...Retry")

Pa=input("\nEnter Password: ") #Password

if Pa=='':

print("Number of retry exceeded...")

exit()

E=input("Enter Email ID: ") #Email

print("Congratulations!!! Your Account has been successfully created.\n")

cur.execute("insert ignore into LoginDetails values(%s,%s)",(U,Pa))

con.commit()

#Login to User Account using Name & Password

def LOGIN\_ACCOUNT():

print("\nEnter Login details...")

L=ENTER\_USER\_NAME()

P=input("Enter Password: ")

cur.execute("select Account\_password from LoginDetails where UserName like '%s'"%(L))

data=cur.fetchall() #Fetch data from table to check if User exits

if any(data):

cur.execute("select Account\_password from LoginDetails where UserName like '%s'"%(L))

data1=cur.fetchone() #Fetch password from table & compare

if data1[0]==P:

print("\t!!! Logged in successfully !!!")

else:

print("\n\t!!! Wrong Password !!!")

exit()

else:

print("\n\t!!! User doesnot exist !!!\n\t!!! Create Account!!!")

exit()

#Capture Traveller details..No of Adults, Kids, Gender...

def TRAVELLER\_DETAILS():

print("\nTRAVELLER DETAILS")

Adult=int(input("NO. OF ADULTS (12+yrs): "))

if No\_passenger>=Adult:

Kid=No\_passenger-Adult

print("NO. OF CHILD (2-12yrs): ",Kid)

for i in range(Adult):

print("\nENTER ADULT",i+1,"DETAILS")

Fname=input("FIRST NAME: ")

Lname=input("LAST NAME : ")

NAME=Fname+' '+Lname

print("GENDER - PRESS: 1.MALE 2.FEMALE")

Gender=int(input("ENTER GENDER: "))

if Gender==1:

GENDER='MALE'

elif Gender==2:

GENDER='FEMALE'

else:

GENDER='MALE-default'

cur.execute("insert ignore into PassengerDetails values(%s,%s)",(NAME,GENDER))

con.commit()

for i in range(Kid):

print("\nENTER KID",i+1,"DETAILS")

Fname=input("FIRST NAME: ")

Lname=input("LAST NAME : ")

NAME=Fname+' '+Lname

print("GENDER - PRESS: 1.MALE 2.FEMALE")

Gender=int(input("ENTER GENDER: "))

if Gender==1:

GENDER='MALE'

elif Gender==2:

GENDER='FEMALE'

else:

GENDER='MALE-default'

cur.execute("insert ignore into PassengerDetails values(%s,%s)",(NAME,GENDER))

con.commit()

return 1

else:

return 0

#Display Traveller details from Table

def SHOW\_TRAVELLER\_DETAILS():

print("Name\tGender")

cur.execute("select \* from PassengerDetails")

for info in cur:

print(info)

#Capture Traveller details to send booking details

def TRAVELLER\_DETAILS2():

global no #global variable protecting Mobile no

global Email #global variable protecting Email Isd

no=int(input("ENTER MOBILE NUMBER: "))

Email=input("ENTER EMAIL ID: ")

#Capture Travel details - From, To, Date, Class, Category.....

def TRAVEL\_FROM\_TO():

global Ticket\_fare

global No\_passenger

global TClass

global Date

f=open('Booking','w')

Departure=input("Enter From : ")

if Departure=='':

Departure='Chennai-default'

Destination=input("Enter To : ")

if Destination=='':

Destination='Delhi-default'

Date=input("Departure Date(DD/MM/YYYY): ")

No\_passenger=input("\nEnter no. of Passengers Travelling (max 4 booking at a Time): ")

if No\_passenger=='':

print("Wrong entry!!! default selection - 1")

No\_passenger=1 #default

No\_passenger=int(No\_passenger)

if No\_passenger<=4:

pass

elif No\_passenger>4:

print("\n!!! Max number of passanger allowed in single booking is 4 !!! Try again !!!")

No\_passenger=int(input("\nEnter no. of Passengers Travelling: "))

if No\_passenger>4:

print("\n!!! Exceed nummber of re-tries !!!")

exit()

#Choose Travel class - Economy, Pre. economy, Business

print("\nChoose Travel Class")

print("Press:\t1.Economy\t2.Premium Economy\t3.Business")

Travel\_class=input("Enter Class: ")

if Travel\_class=='':

print("Wrong entry!!! default selection - 1")

Travel\_class=1 #default

Travel\_class=int(Travel\_class)

if Travel\_class==1:

TClass='Economy'

elif Travel\_class==2:

TClass='Premium Economy'

elif Travel\_class==3:

TClass='Business'

else:

Travel\_class==1

TClass='Economy'

#Choose Category - Regular,Armed forces,Sr citizen, Infant

print("\nChoose Fare Category")

print("Press: 1.Regular 2.Armed forces 3.Senior citizen 4.Infant")

Travel\_category=input("Enter category: ")

if Travel\_category=='':

print("Wrong entry!!! default selection - 1")

Travel\_category=1 #default

Travel\_category=int(Travel\_category)

if Travel\_category==1:

TCategory='Regular'

elif Travel\_category==2:

TCategory='Armed Forces'

elif Travel\_category==3:

TCategory='Sr Citizen'

elif Travel\_category==4:

TCategory='Infant'

else:

Travel\_category=1 #default

TCategory='Regular-default'

Ticket\_fare=CALCULATE\_FARE(Travel\_class,Travel\_category)

rec=Departure+','+Destination+','+Date+','+str(No\_passenger)+','+TClass+','+TCategory+','+'\n'

f.write(rec)

f.close()

#Read contents of file - Booking

def read():

f=open('Booking')

for i in f:

l=i.split(',')

print('Departure : ',l[0])

print('Destination : ',l[1])

print('Date : ',l[2])

print('No. of passenger: ',l[3])

print('Travel Class : ',l[4])

print('Category : ',l[5])

f.close()

#Calculate fare based on Travel class & Category

def CALCULATE\_FARE(Travel\_class,Travel\_category):

Final\_fare=0

if Travel\_class==1: #Economy

Base\_fare=6000

elif Travel\_class==2: #Premium

Base\_fare=8000

elif Travel\_class==3: #Business

Base\_fare=10000

else:

Base\_fare=6000 #default-Economy

if Travel\_category ==1: #Regular

Final\_fare=Base\_fare

elif Travel\_category ==2: #Armed forces, 20%

Final\_fare=Base\_fare-Base\_fare/5

elif Travel\_category ==3: #Sr citizen, 10%

Final\_fare=Base\_fare-Base\_fare/10

elif Travel\_category ==4: #Infant, 50%

Final\_fare=Base\_fare/2

else:

Final\_fare=Base\_fare #default-Regular

return Final\_fare

#Capture flight details

def FLIGHT\_SELECTION():

global FlightName

global FlightNo

global DeptTime

Fs=input("Enter SNo for Flight selection: ")

if Fs=="":

Fs="1" #default

print("\nWrong Selection...default - 1")

Fs=int(Fs)

if Fs>=10:

Fs=1 #default

print("\nWrong Selection...default - 1")

cur.execute("select Flight\_Name from Availability where S\_No like '%s'"%(Fs))

FlightName=cur.fetchone()

cur.execute("select Flight\_No from Availability where S\_No like '%s'"%(Fs))

FlightNo=cur.fetchone()

cur.execute("select Departure\_Time from Availability where S\_No like '%s'"%(Fs))

DeptTime=cur.fetchone()

cur.execute("select \* from Availability where S\_No like '%s'"%(Fs))

data2=cur.fetchall()

if any(data2):

print("\nYou have selected: ")

print(data2,"\n")

return 1

else:

return 0

#Print E-Ticket

def Eticket(BookingStatus):

print("E-TICKET")

print("Booking ID: MMAT34578")

print("\nBooking Status: ",BookingStatus)

cur.execute("select Name from PassengerDetails")

PasName=cur.fetchall()

l=list(PasName)

for i in range(No\_passenger):

print("Passenger",i+1,": ",l[i],"Seat No: ",seat\_list[i])

print("\n-----------------------------------")

print("Flight Name : ",FlightName[0])

print("Flight Number : ",FlightNo[0])

read()

print("Departure Time : ",DeptTime[0])

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

#Choose - food options

def FOOD():

FoodTotal=0

print("\nPre Book snacks now before you fly....")

flag=1

while flag!=0:

print("\nPRESS 1.Veg 2.Non veg 3.Beverages")

food=int(input("Choice: "))

if food==1:

ch=int(input("\nMenu:\nPress1: Sandwich\nPress2: Ready to Eat\n\nSelect the opton you prefer: "))

if ch==1:

print("Press 1 for Tomato, Cucmber lettuce Sandwich")

print("Press 2 for Panner Tikka Sandwich")

n=int(input("Choice: "))

if n==1:

print("Tomato, Cucmber lettuce Sandwich")

FoodTotal+=350

print("Price=350")

elif n==2:

print("Panner Tikka Sandwich")

FoodTotal+=300

print("Price=300")

print("\nDo you want to continue food selection ....")

print("Press0: No\nPress1: Yes")

flag=int(input("Choice: "))

elif ch==2:

print("Press1: Vada Pao")

print("Press2: Veg Biryani")

print("Press3: Poha")

print("Press4: Cup noodles")

n=int(input("Choice: "))

if n==1:

print("Vada Pao")

FoodTotal+=100

print("Price=100")

elif n==2:

print("Veg Biryani")

FoodTotal+=350

print("Price=350")

elif n==3:

print("Poha")

FoodTotal+=250

print("Price=250")

elif n==4:

print("Cup noodles")

FoodTotal+=200

print("Price=200")

print("\nDo you want to continue food selection ....")

print("Press0: No\nPress1: Yes")

flag=int(input("Choice: "))

elif food==2:

ch=int(input("\nPress: 1.For Meals 2.For Cup Noodles\n\nEnter your choice:"))

if ch==1:

print("Press1 Chicken Tikka Sandwich")

print("Press2 Chicken Nuggets")

print("Press3 Tikka Chilli Chicken")

print("Press4 Chapati with Fish currry")

n=int(input("Choice: "))

if n==1:

print("Chicken Tikka Sandwich")

FoodTotal+=150

print("Price=150")

elif n==2:

print("Chicken Nuggets")

FoodTotal+=350

print("Price=350")

elif n==3:

print("Tikka Chilli Chicken")

FoodTotal+=250

print("Price=250")

elif n==4:

print("Chapati with Fish curry")

FoodTotal+=200

print("Price=200")

print("\nDo you want to continue food selection ....")

print("Press0: No\nPress1: Yes")

flag=int(input("Choice: "))

elif ch==2:

print("Press 1:Chicken noodles")

print("Press 2:Fish and meat noodles")

n=int(input("Choice: "))

if n==1:

print("Chicken noodles")

FoodTotal+=150

print("Price=150")

elif n==2:

print("Fish and meat noodles")

FoodTotal+=200

print("Price=200")

print("\nDo you want to continue food selection ....")

print("Press0: No\nPress1: Yes")

flag=int(input("Choice: "))

elif food==3:

print("Press1 for Tea")

print("Press2 for Coffee")

print("Press3 for Paper boat mango")

n=int(input("Choice: "))

if n==1:

print("Tea")

FoodTotal+=100

print("Price=100")

elif n==2:

print("Coffee")

FoodTotal+=100

print("Price=100")

elif n==3:

print("Paper boat mango")

FoodTotal+=150

print("Price=150")

print("\nDo you want to continue food selection ....")

print("Press0: No\nPress1: Yes")

flag=int(input("Choice: "))

else:

print("\nDo you want to continue food selection ....")

print("Press0: No\nPress1: Yes")

flag=int(input("Choice: "))

return FoodTotal

#Load text file (about\_us.txt) & read back with information about website

def about\_us():

text\_file=open("about\_us.txt","w+")

text\_file.write("It's all about easy booking to fly in India....\n")

text\_file.write("MakeMyAirTrip is a pioneer in India’s online travel services.\n\n")

text\_file.write("We started in 2000, with the target to offer easy, at your fingertips booking\n")

text\_file.write("services to the flyers in Indian continent.\n")

text\_file.write("We have completed 10 years and become the most recognizable mascot all over.\n")

text\_file.write("Our air service has won numerous awards for humour, originality & reliability. \n")

text\_file.write("Our service is more like a friend who reaches out with warmth and hospitality,\n")

text\_file.write("even to the farthest corners of the country.\n\n")

text\_file.write("Some Innovation in our flights to redefine the total passenger experience - \n")

text\_file.write("\t 1.Wider seats and aisles\n")

text\_file.write("\t 2.Bigger windows, 65% larger than other aircraft in the same category\n")

text\_file.write("\t 3.Improved cabin environment (lower altitude, cleaner air, higher humidity)\n")

text\_file.write("\t 4.Versatile range – from efficient short, medium and long-range routes \n")

text\_file.write("\t 5.15% lower fuel consumption compared other same-sized airplanes\n")

text\_file.write("\t 6.Lower maintenance costs than peer airplanes \n")

text\_file.write("\t 7.Cleaner throughout its lifecycle (designed for the environment) \n")

text\_file.write("\t 8.Versatile family – big jet ranges, small jet trip costs \n")

text\_file.write("\t 9. 85 dbA noise stays within airport property\n")

text\_file.write("\t 10.Less hazardous waste and overall waste in production\n\n")

text\_file.write("To know more about us, refer the following details -\n")

text\_file.write("\tContact us on: 93xxxxxx36\n")

text\_file.write("\tEmail us at: makemyairtrip@gmail.com\n")

text\_file.write("\tTweet at: www.makemyairtriptwitter.com \n ")

text\_file.write("\tTo know more log on to: www.makemyairtrip.org.in \n")

text\_file.write("\tGeneral Queries: makemyairtrip.base@trip.in \n")

text\_file.write("\tMissing Miles/Retro Credit on AI: makemytrip.retros@trip.in\n\n")

text\_file.write("\t\tHope you enjoyed with us.. \n")

text\_file.write("\t\tKeep enjoying your jouney with us.. \n")

text\_file.write("\t\tStay safe and stay healthy!!! \n")

text\_file.seek(0)

for Lines in text\_file.readlines(): #readline return line with \n

print(Lines.rstrip()) #rstrip() remove \n

#Load text file (travel\_rules.txt) & read back - Rules & Regulatiions

def travel\_rules():

travel\_rules=open("travel\_rules.txt","w+")

travel\_rules.write("We seek your kind assistance and cooperation for your own safety as well as safety\n")

travel\_rules.write("of fellow passengers………Please follow below guidelines: \n\n")

travel\_rules.write("1) All flights shall provide a complimentary safety kit(three layered surgical masks,\n")

travel\_rules.write("a face shield, a sanitizer, a pair of gloves) to all passengers at the boarding gate.\n\n")

travel\_rules.write("2) Passengers seated in middle seats will also be provided with wrap-around gown.\n\n")

travel\_rules.write("3) Passengers must wear a face mask covering their nose and mouth, throughout the\n")

travel\_rules.write("journey. The mask may be removed only while eating and drinking. \n\n")

travel\_rules.write("4) Please maintain appropriate social distancing while boarding and de-boarding the\n")

travel\_rules.write("aircraft.\n\n")

travel\_rules.write("5) Passengers are requested to familiarise themselves with the guidelines published\n")

travel\_rules.write("by the Indian Ministry of Civil Aviation for air passengers.\n\n")

travel\_rules.write("6) Caution: Passengers are advised to strictly follow COVID-19 protocols. Failure\n")

travel\_rules.write("to comply with protocols may attract penal action against the concerned individual.\n\n")

travel\_rules.write("7) As per the current government guidelines, it is mandatory to do web check-in for\n")

travel\_rules.write("all domestic flight. Complete it for free 48 hrs - 60 min before flight.")

travel\_rules.seek(0)

print(travel\_rules.read())

travel\_rules.close()

#Load text file (refund\_text.txt) & read back - Refund policies

def refund\_policy():

refund\_text=open("refund.txt",'w+')

read=["1) Cancellation within 24 hours of booking will subject to zero cancellation\n charges and a full Refund will be provided.\n\n2) Cancellation can be done online and refund will be given after deducting\n cancellation charges as applicable.\n\n3) Cancellation or changes must be done at least 2 hrs prior to departure.\n\n4) Processing of Refund may take upto 7 working days.\n\n5) Refund will be credited to the same account from which the payment was made.\n\n6) In the case of refund, upfront discount and promo code discount availed at\n the time of booking would be deducted from the refund amount.\n\n7) Passengers can also claim refund for an unused tickets, however for such\n tickets only a small amount depending on the flight and fare rules will \n be refunded, along with the additional fee for cancellation.\n\n8) Please contact our executives for any query from our contact details."]

refund\_text.writelines(read)

refund\_text.seek(0)

ref=refund\_text.readlines()

for lines in read:

print(lines.rstrip('\n'))

#Load Availability table statically

def INPUTING\_FLIGHT\_DETAILS():

cur.execute("delete from Availability")

cur.execute("insert into Availability values(1,'INDIGO','IDI678A','04:00:00','TICKETS AVAILABLE')")

cur.execute("insert into Availability values(2,'INDIGO','IDI676B','07:00:00','TICKETS AVAILABLE')")

cur.execute("insert into Availability values(3,'INDIGO','IDI650C','11:00:00','TICKETS AVAILABLE')")

cur.execute("insert into Availability values(4,'GO AIR','AIR221C','14:00:00','TICKETS AVAILABLE')")

cur.execute("insert into Availability values(5,'GO AIR','AIR221B','17:00:00','TICKETS AVAILABLE')")

cur.execute("insert into Availability values(6,'GO AIR','AIR123A','18:00:00','TICKETS AVAILABLE')")

cur.execute("insert into Availability values(7,'SPIJET','SPI878C','20:00:00','TICKETS AVAILABLE')")

cur.execute("insert into Availability values(8,'SPIJET','SPI999B','22:00:00','TICKETS AVAILABLE')")

cur.execute("insert into Availability values(9,'SPIJET','SPI987A','00:00:00','TICKETS AVAILABLE')")

def SHOW\_AVAILABLE\_FLIGHT():

cur.execute("select \* from Availability")

for info in cur:

print(info)

#Paymets selection - Credit card, Debit card, Google Pay, PayTM

#Capture Payment details

def PAYMENT(Total\_amt):

Pay=Total\_amt

print("\nPayment Mode - 1.Credit card 2.Debit card 3.Google Pay 4.PayTM")

n=int(input("Choose mode of payment from the above choices: "))

fl=1

while fl!=0:

if n==1:

cn=int(input("\nCard number: "))

nc=input("Name on card: ")

cv=int(input("CVV: "))

ed=input("Expiry date on card (MM/YYYY): ")

fl=0

elif n==2:

cn=int(input("Card number: "))

nc=input("Name on card: ")

cv=int(input("CVV: "))

ed=input("Expiry date on card (MM/YYYY): ")

fl=0

elif n==3:

print("\nKindly Google Pay on 96xxxxxx74 ")

nu=int(input("Enter Gpay Number, in case of Refund: "))

fl=0

elif n==4:

print("\nKindly PayTM on 96xxxxxx84 ")

nu=int(input("Enter Paytm Number, in case of Refund: "))

fl=0

else:

print("INVALID MODE OF PAYMENT")

fl=0

exit()

print("\nBOOKINGS CONFIRMED\nThank You for Booking with us!! We would like to Help You Again!!")

print("E-tickets will be sent to registered number")

#Web check-in seats display & selection for differnt travel classes

def WEBCHK():

global seat\_list

seat\_list=[]

if TClass.lower()=="business":#Seat selection Business class

print ("Available seats are")

l=['1','2','3','4','5']

print(l)

for i in range(No\_passenger):

n=input("\nChoose the seat number: ")

if n not in l:

n='1' #default wrong entry by user

seat\_list.insert(i,n)

m=l.index(n)

l.pop(m)

if No\_passenger==1:

break

if No\_passenger!=i+1:

print("Remaining seats",l)

elif TClass.lower()=="premium economy": #Seat selection Premium class

print ("Available seats are")

l=['6a','6b','7a','7b','8a','8b','9a','9b','10a','10b']

print(l)

for i in range(No\_passenger):

n=input("Choose the seat number: ")

if n not in l:

n='6a' #default wrong entry by user

seat\_list.insert(i,n)

m=l.index(n)

l.pop(m)

if No\_passenger==1:

break

if No\_passenger!=i+1:

print("Remaining seats ",l)

elif TClass.lower()=="economy": #Seat selection Economy class

la=['11a','12a','13a','14a','15a']

lm=['11b','12b','13b','14b','15b']

lw=['11c','12c','13c','14c','15c']

for i in range(No\_passenger):

s=int(input("Choose the kind of seat you prefer\n1.Aisle 2.Middle 3.Window\nChoose the option: "))

if s==1:

print ("Available Aisle seats are")

print(la)

n=input("\nType the seat number: ")

if n not in la:

n='11a' #default wrong entry by user

seat\_list.insert(i,n)

m=la.index(n)

la.pop(m)

elif s==2:

print ("Available Middle seats are")

print(lm)

n=input("\nType the seat number: ")

if n not in lm:

n='11b' #default wrong entry by user

seat\_list.insert(i,n)

m=lm.index(n)

lm.pop(m)

elif s==3:

print ("Available Window seats are")

print(lw)

n=input("\nType the seat number: ")

if n not in lw:

n='11c' #default wrong entry by user

seat\_list.insert(i,n)

m=lw.index(n)

lw.pop(m)

def CANCELATION():

print("Do you want to cancel booking (Y(yes) or N(no))")

Del=input("Choice: ")

if Del in'Yy':

bookingid=input("Enter Booking ID: ")

if bookingid=='MMAT34578':

Eticket(BookingStatus)

else:

print("Invalid Booking ID")

else:

print("Booking Not Cancelled")

################################################################

#Main program

################################################################

import mysql.connector

import datetime

con=mysql.connector.connect(host='localhost',user='root',password='root',charset='utf8')

cur=con.cursor()

#Introduction text display

print("\t\t\t\tWELCOME TO MakeMyAirTrip")

print("\t\t\t\t-------------------------")

file=open("Intro.txt","w+")

file.write("MakeMyAirTrip is an online air ticket booking website that provides services like\nflight schedules, ticket reservations, cancellations, web check-in, pre book food.\n\nWhy choose MakeMyAirTrip?\nOffers : Lowest fares, Multiple mode of payments, Safe refunds.")

file.seek(0)

print(file.read())

file.close()

print()

#Create Database

cur.execute("create database if not exists Reservation")

cur.execute("use Reservation")

print("---------------------------------------------------------------------------------------------------------------------\n")

print("Kindly proceed with the account creation...")

cur.execute("delete from PassengerDetails")

cur.execute("create table if not exists PassengerDetails(Name char(20),Gender char(10))")

cur.execute("create table if not exists Availability(S\_No int primary key,Flight\_Name char(20),Flight\_No char(20),Departure\_Time char(20),Tickect\_Booking char(20))")

INPUTING\_FLIGHT\_DETAILS() #Load Availabilty table statically

#Clear all records from LoginDetails table

#cur.execute("delete from LoginDetails")

#cur.execute("drop table LoginDetails")

choice=input("1.Create account\t2.Login\nEnter your Choice: ")

if choice=='':

print("\nWrong entry!!! default selection - 1")

print("Proceed to Create account...\n")

choice=1 #default

choice=int(choice)

if choice==1:

CREATE\_ACCOUNT() #Account creation by user

LOGIN\_ACCOUNT() #Login by user

elif choice==2:

LOGIN\_ACCOUNT() #Login by user

else:

print("\nWrong entry!!! default selection - 1")

print("Proceed to Create account...\n")

CREATE\_ACCOUNT() #Account creation by user

LOGIN\_ACCOUNT() #Login by user

start=input("\nEnter any key to start booking...")

if len(start)!=0:

pass

while True:

print("\nSelect Service -\n\t1.Book Tickects \n\t2.Display Booking \n\t3.Search Booking \n\t4.Update Ticket \n\t5.Cancellation \n\t6.Sort Booking\n\t7.About us\n\t8.Rules and Regulations\n\t9.Refund Policy\n\t10.Exit")

choice=input("\nEnter Your Choice: ")

if choice=='':

print("Wrong entry!!! default selection - 1.Book Tickets\n")

choice=1 #default

choice=int(choice)

if choice==1: #Booking ticket

TRAVEL\_FROM\_TO()

print("\nDetails Entered ")

read()

print("\nFlights Availability")

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print("SNo Airline Number Timing Availability")

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

SHOW\_AVAILABLE\_FLIGHT() #display Availability table

print("\nFare of each flight per head: Rs",Ticket\_fare)

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print()

fselect=FLIGHT\_SELECTION() #Flight selection by user

if fselect==1:

pass

elif fselect==0:

print("\nWrong Flight selection........Retry")

fselect=FLIGHT\_SELECTION()

if fselect==1:

pass

else:

print("\nCrossed Max attempts\n")

exit()

Tdetails=TRAVELLER\_DETAILS()

if Tdetails==1:

pass

elif Tdetails==0:

print("Invalid Passenger Count...Pls Retry\n")

a=TRAVELLER\_DETAILS()

if a==1:

pass

else:

print("Invalid")

exit()

print("\nBooking Details will be sent to...\n")

TRAVELLER\_DETAILS2()

print("\nFetching Details for Available Seat .....")

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print("\nWEB CHECK-IN")

WEBCHK() #Web check-in

print("\nTOTAL AMOUNT:Rs.",Ticket\_fare\*No\_passenger+Ticket\_fare\*No\_passenger\*0.18+500,"(inclusive of GST and sub charges)")

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

Food=input("\nPre-book your Foods now (Y:yes or N:no): ")

if Food.lower() in 'yY':

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

Food\_Tot=FOOD()

print("\nFood Total= Rs",Food\_Tot)

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

else:

Food\_Tot=0

print("\nTo confirm the Booking, Kindly Proceed with the Payment \n ")

print("PAYMENT ")

Total\_amt=((Ticket\_fare\*No\_passenger)+(Ticket\_fare\*No\_passenger\*0.18)+500)+Food\_Tot

print("GRAND TOTAL: ",Total\_amt)

PAYMENT(Total\_amt)

BookingStatus='CONFIRMED'

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

Eticket(BookingStatus)

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

elif choice==2: #Display Booking & Passenger details

print("Press 1:To display Booking Details\t2:To display Passenger Details")

Display\_Ticket=int(input("Enter Choice: "))

if Display\_Ticket==1:

print("\n-----------------------------------------------------------------------------------")

print("TRAVEL DETAILS: ")

read()

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

elif Display\_Ticket==2:

print("\n-----------------------------------------------------------------------------------")

print("PASSENGER DETAILS: ")

SHOW\_TRAVELLER\_DETAILS()

print("Mobile no.: ",no)

print("Email Id: ",Email)

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

elif choice==3: #Search

sn=input("\nEnter Full Name: ")

cur.execute("select \* from PassengerDetails where NAME like '%s'"%(sn))

data3=cur.fetchall()

if any(data3):

print("BOOKING FOUND")

print(data3)

else:

print("BOOKING NOT FOUND")

elif choice==4: #Update ticket details

print("Press: 1.Update by Name\t 2.Update Mobile number\t3.Update Email id")

n=int(input("Choose: "))

if n==1:

oldname=input("Enter Old Name to be updated: ")

cur.execute("select \* from PassengerDetails")

row=cur.fetchall()

for i in range(0,cur.rowcount):

if (row[i][0]==oldname):

newname=input("Enter New Name : ")

gender=input("Enter New Gender: ")

cur.execute("update PassengerDetails set Name=%s,Gender=%s where Name=%s",(newname,gender,oldname))

con.commit()

print("\t!!!Updated Successsfully!!! ")

SHOW\_TRAVELLER\_DETAILS()

break

else:

print("Name to be updated not found")

elif n==2:

newno=int(input("ENTER MOBILE NUMBER: "))

no=newno

print("\t!!!Updated Successsfully!!! ")

print("Updated Mobile Number: ",no)

else:

email\_id=input("ENTER NEW EMAIL ID: ")

Email=email\_id

print("\t!!!Updated Successsfully!!! ")

print("Updated Email Id: ",Email)

elif choice==5: #Cancel Booking

BookingStatus='Cancelled'

CANCELATION()

break

elif choice==6: #Sorting

cur.execute("select \* from PassengerDetailS order by NAME asc")

data4=cur.fetchall()

for x in data4:

print(x)

elif choice==7: #Display about Website

print("\n\t\t\t\tABOUT US\n\t\t\t\t--------")

about\_us()

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

elif choice==8: #Display Rules and regulations

print("\nRules and regulations")

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

travel\_rules()

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

elif choice==9: #Display Refund Policy

print("\nMakeMyAirTrip Refund Policy")

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

refund\_policy()

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

else: #Exit

print("Exit\n")

break

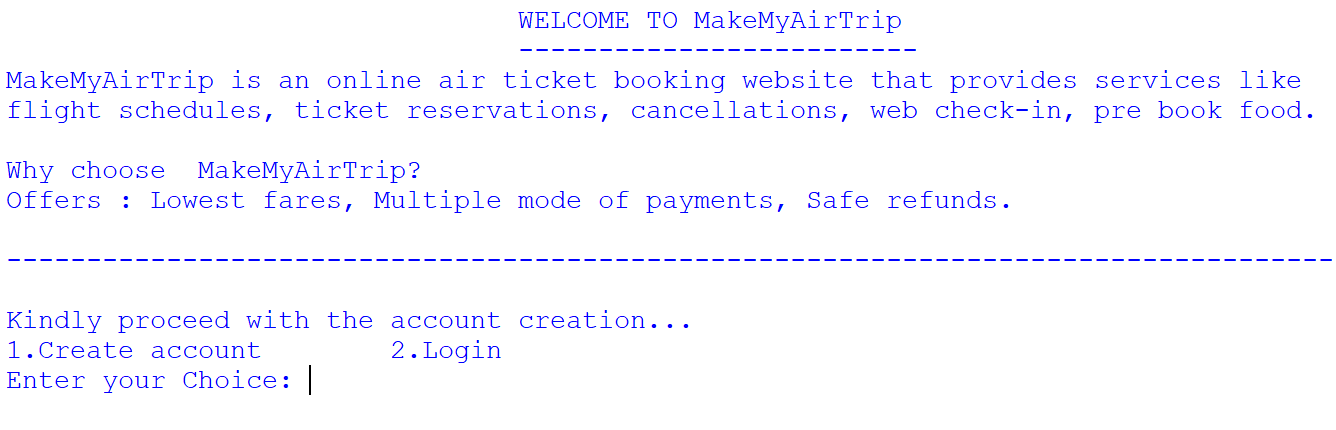
con.close()

cur.close()

**SAMPLE OUTPUT**

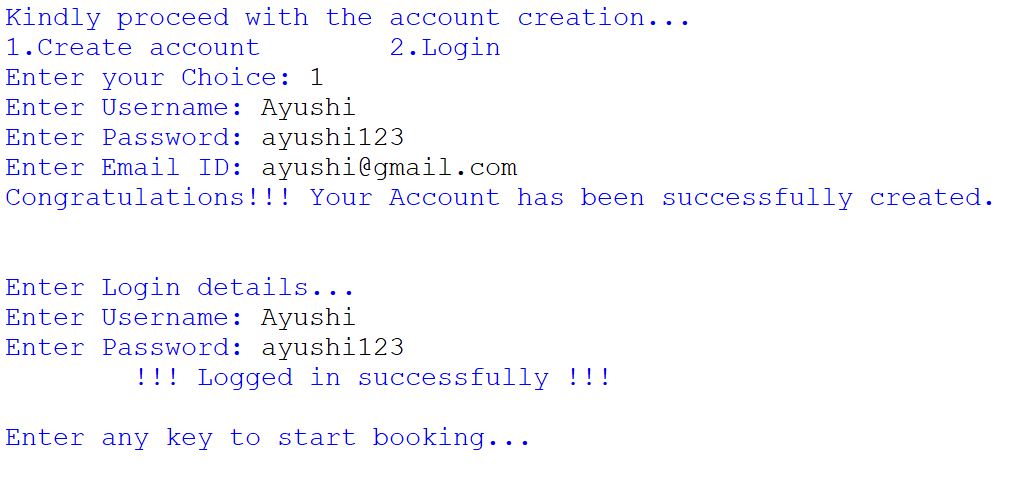
1. Welcome screen for MakeMyAirTrip

* User can “Create New Account” or “Login” to existing Account

****

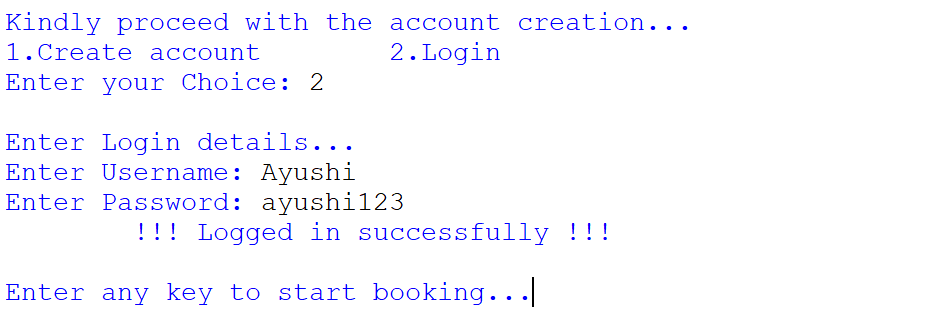
1. User create new account & logged in with this account

* Details saved in a table for next time login
* Empty User name, not allowed
* User name that already exists, not allowed

****

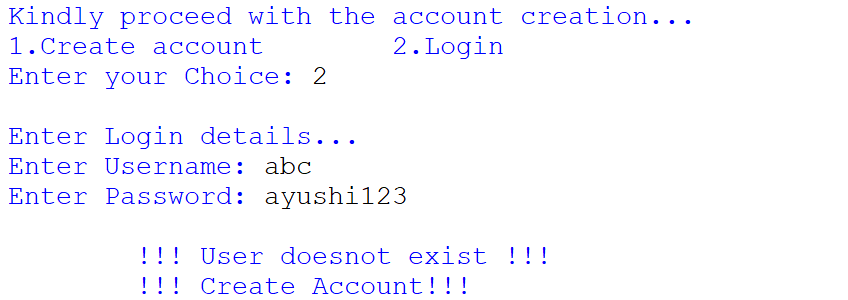
1. User logged in with already existing account with website

* User name & Password verified from table to allow login – **correct “User name” & “Password” entered by User**



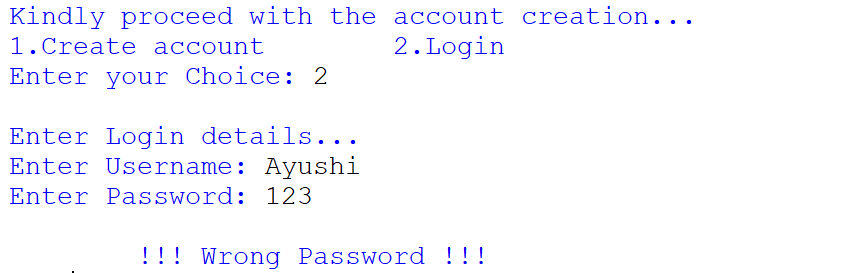
1. User logged in with already existing account with website

* User name & Password verified from table to allow login – **wrong “User name” entered by User**

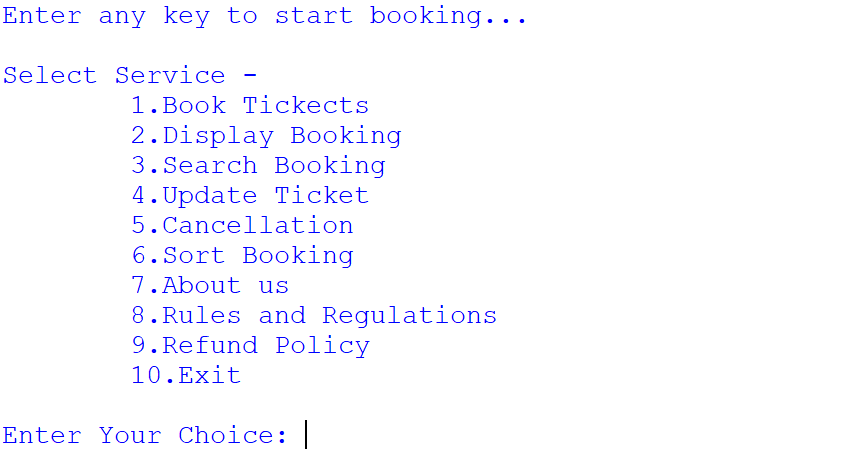


1. User logged in with already existing account with website

* User name & Password verified from table to allow login – **wrong “Password” entered by User**

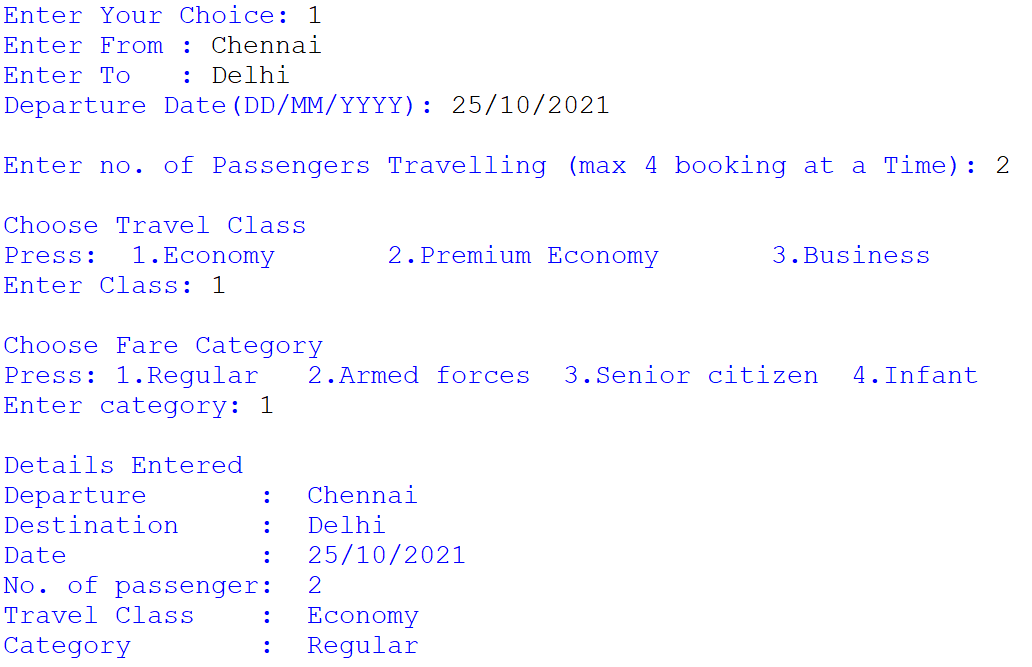


1. User logged in successfully. Various selections available to user.

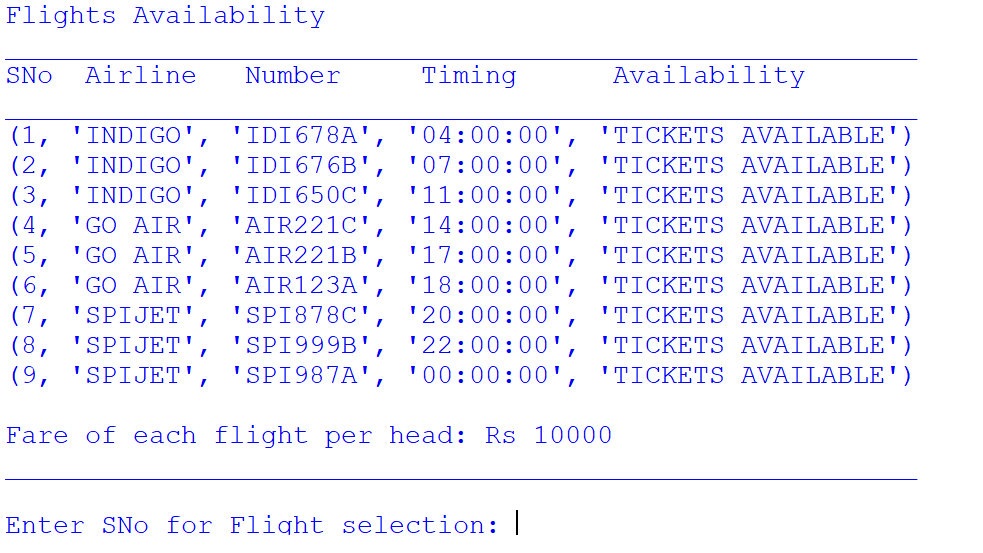


1. User Selects option 1 for “Booking Tickets”

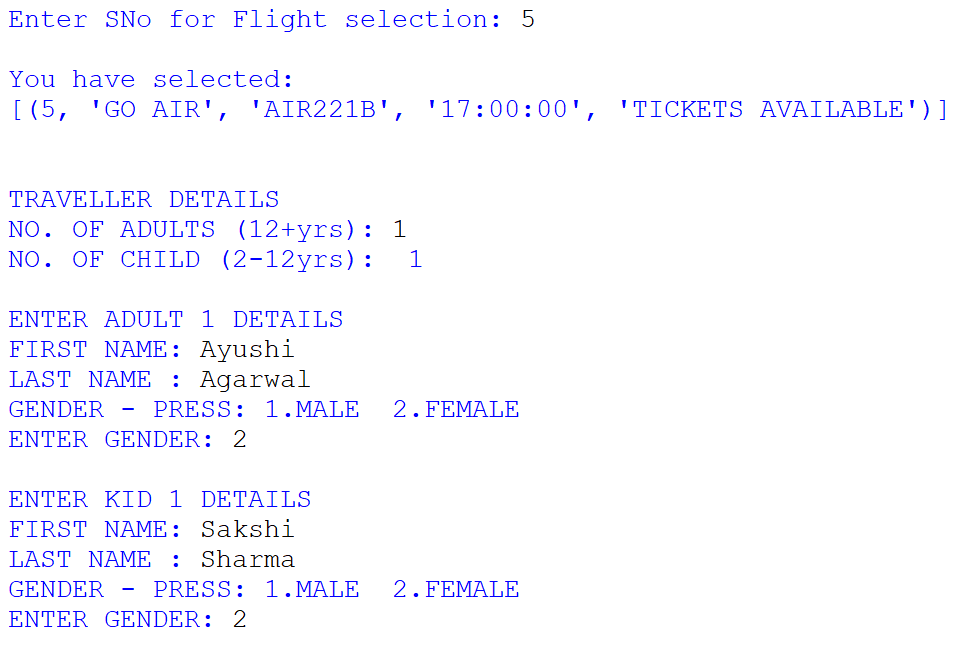
* User need to provide travel details like – travel From, travel To, travel Date, Number of Passengers (max 4 allowed per ticket), travel Class, travel Category.



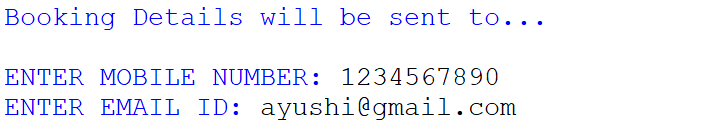
1. Based on travel details entered by User, he will be shown with list of available flights & User need to select his Flight.



1. After Flight selection, User need to enter the details (Name / Gender) of all the passengers to be booked. e.g. If he choose for 2 passengers, he will be asked to enter the details for 2 passengers.

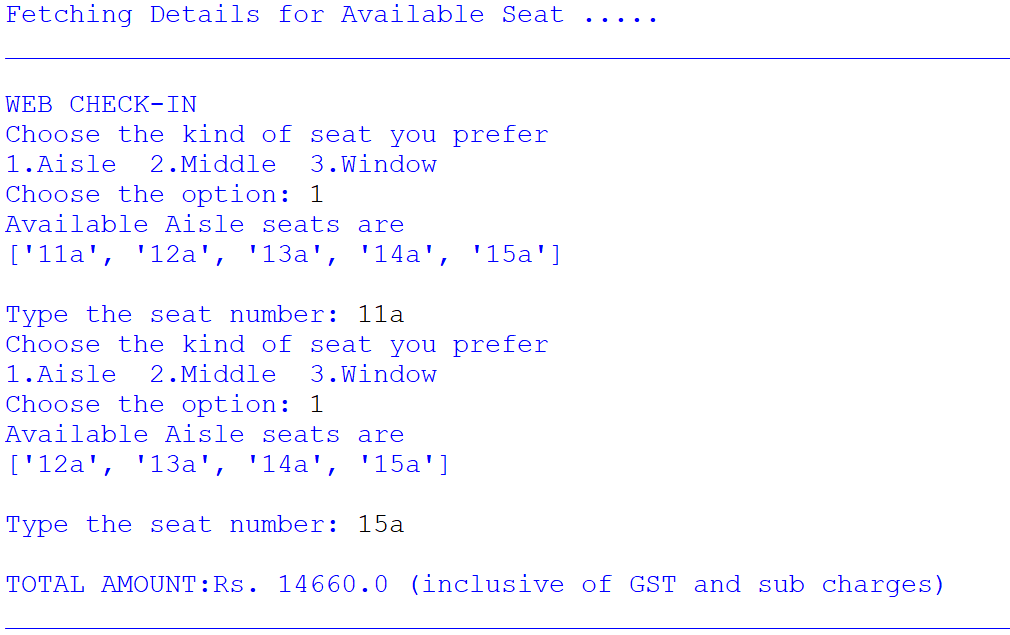


1. User need to enter Mobile number & Email Id, where he want to receive the booking details.

****

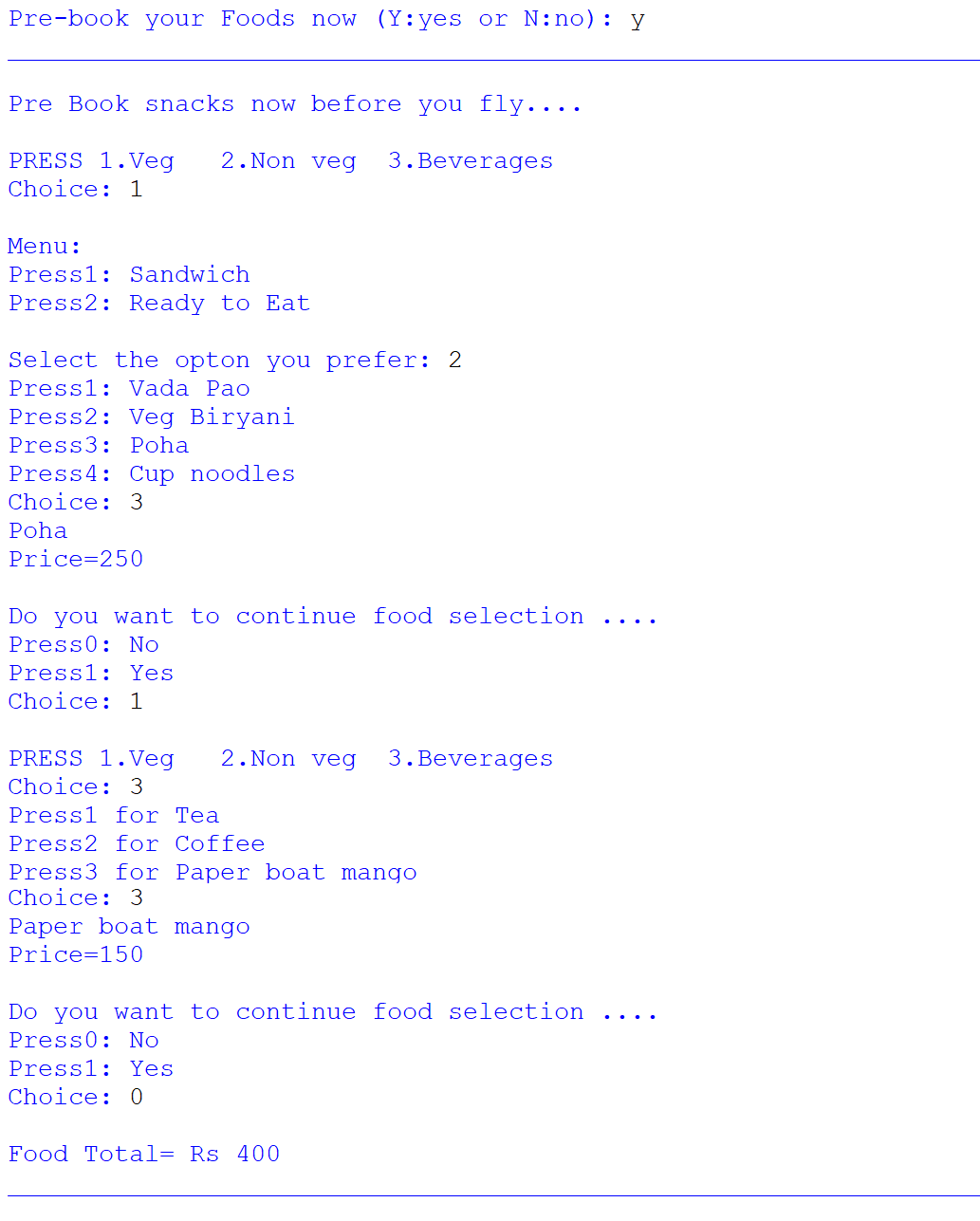
1. User will be asked to do Web check-in. Available seat options will be displayed based on chosen travel Class by User.

* User need to provide seat preferences (Aisle, Middle, Window). Accordingly list of available seats will be displayed.
* Once seat is selected for 1st passenger, it will not be available for next passenger.
* Total fare will be displayed for all passengers, including GST & Sub charges.

****

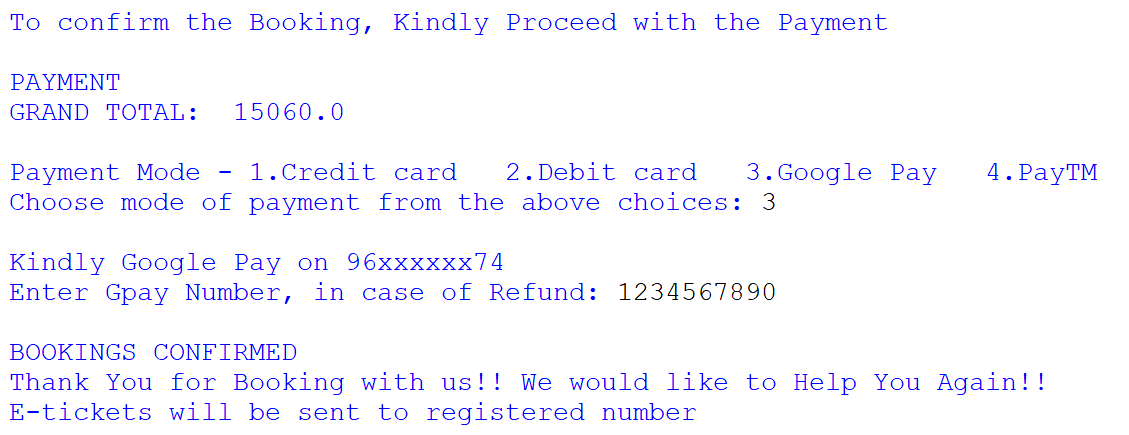
1. User will be given choice to Pre-book food or not.

* Total Price for all the food items selected will be displayed.

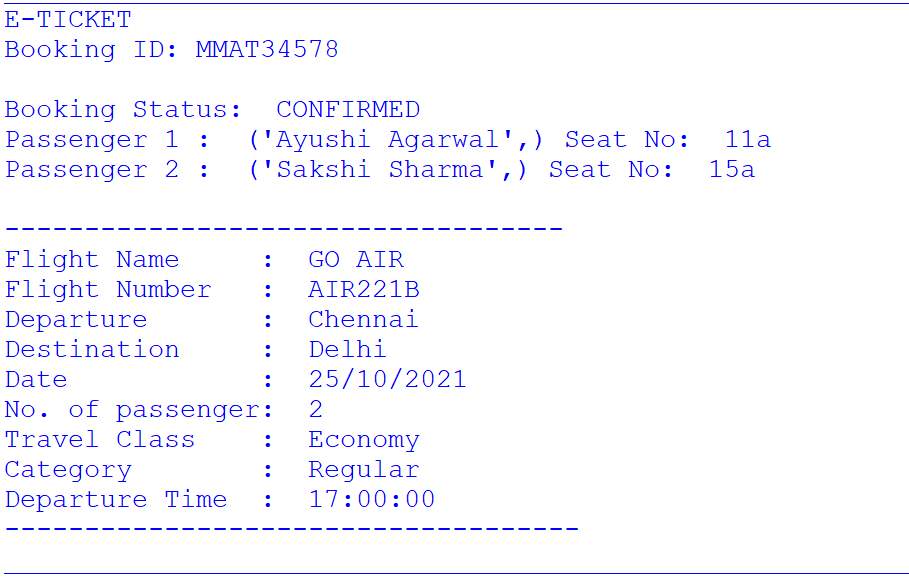


1. Total amount to be paid (inclusive of food) will be displayed & User will be given different payment options.

* User need to enter additional details in case of Refund
* Finally booking confirmation message will be displayed

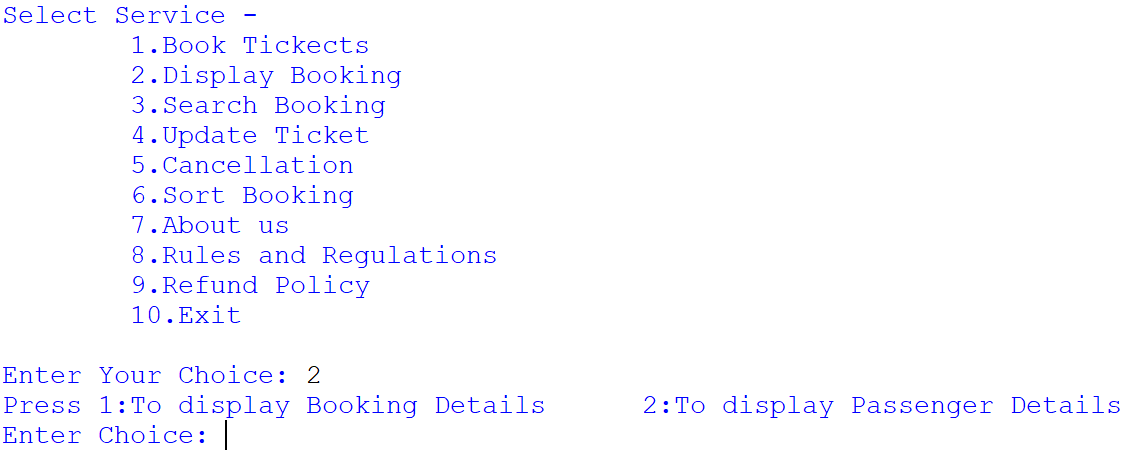
****

1. E-ticket will be displayed

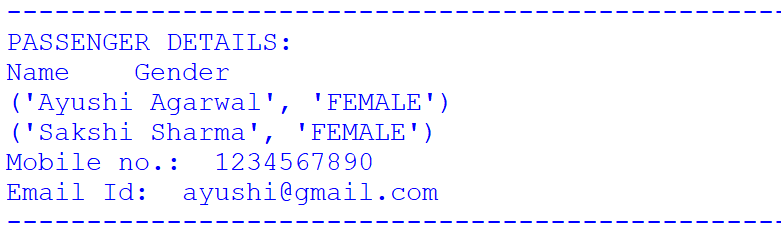


1. User selects 2 “Display Booking” option from list of Services.

* Can see Booking details by pressing 1
* Can see Passenger details by pressing 2

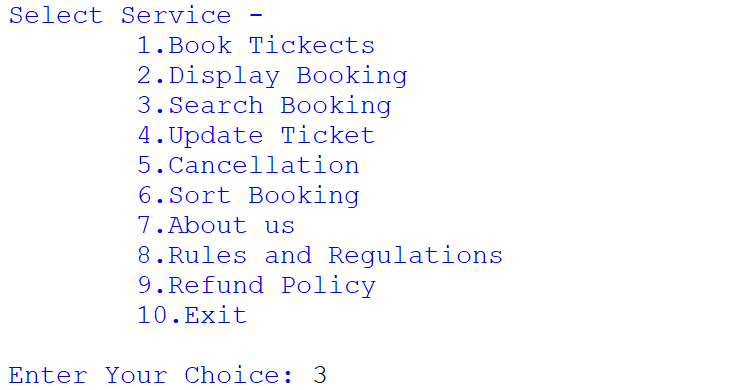
****

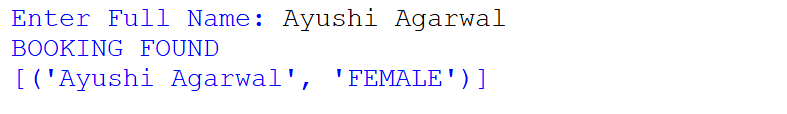
****

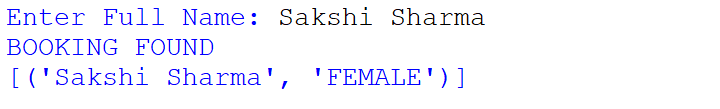
****

1. User selects 3 “Search Booking” option from list of Services.

* Can search his booking by entering his full name



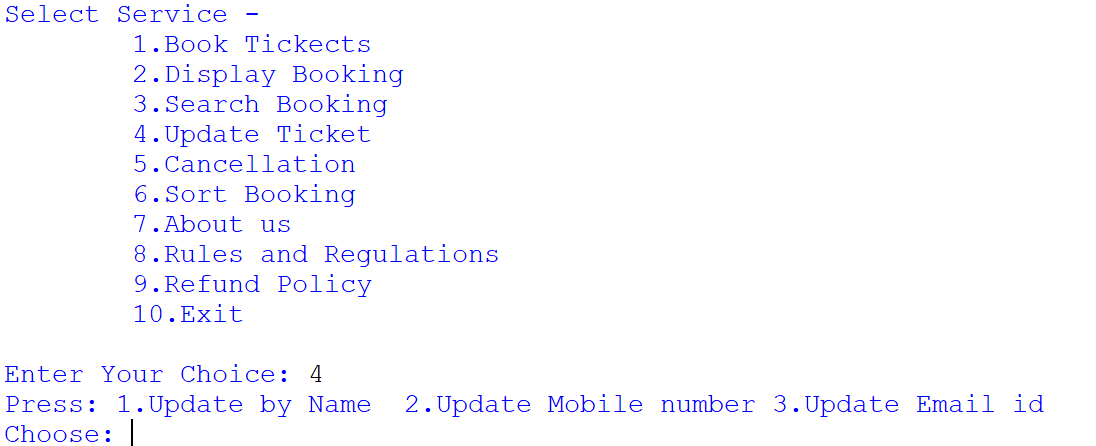
****

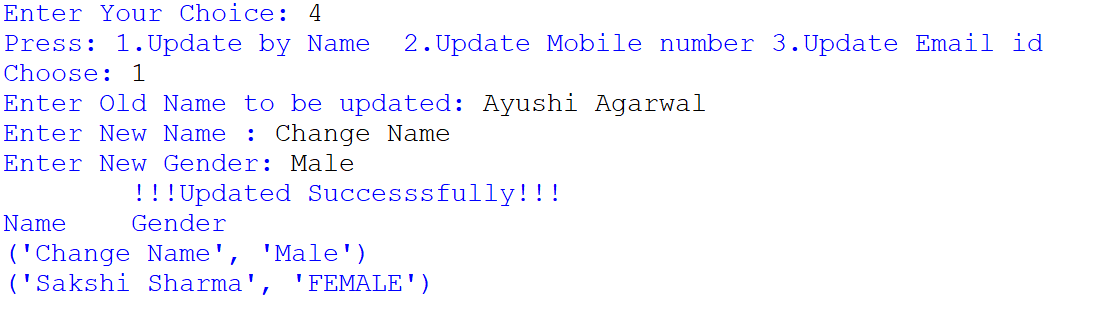
****

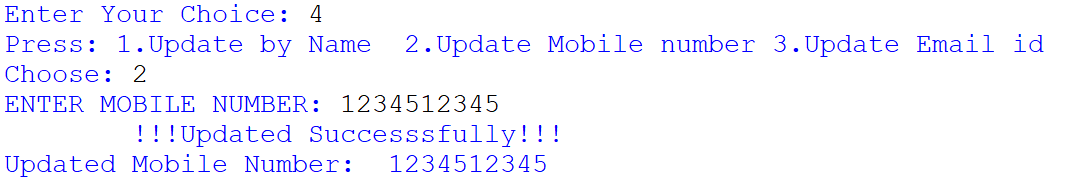
****

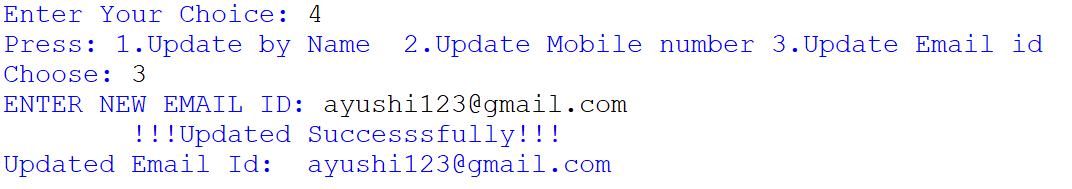
1. User selects 4 “Update Ticket” option from list of Services.

* Can search his booking by entering his full name & update
* Can update mobile number
* Can update email id

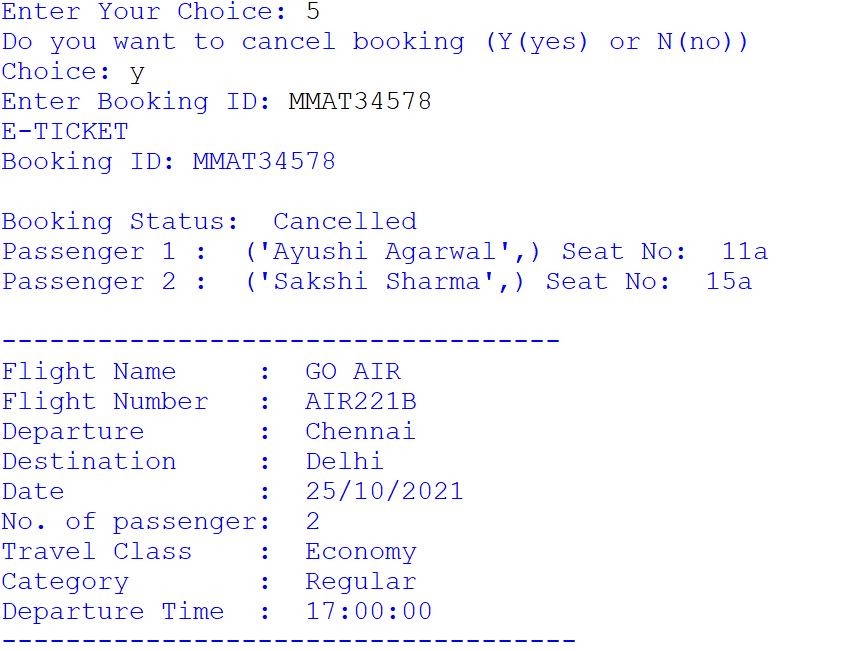
****

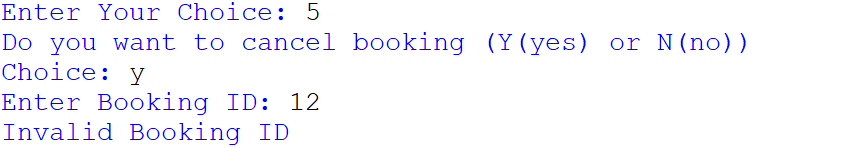
****

****

****

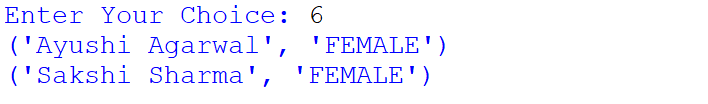
1. User selects 5 “Cancellation” option from list of Services.

****

****

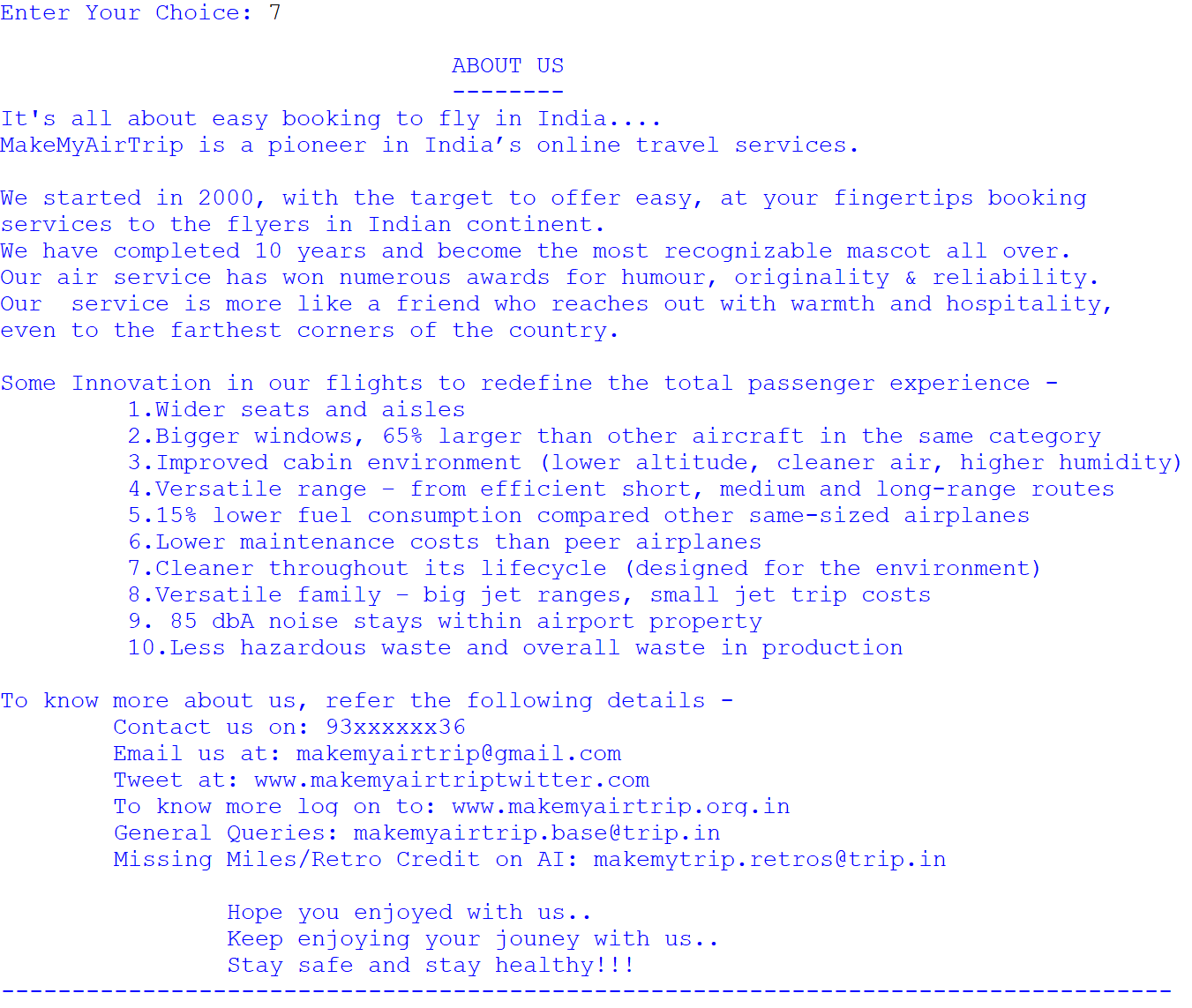
1. User selects 6 “Sort Booking” option from list of Services.

* Passengers on a ticket will be displayed in alphabetical sorted order



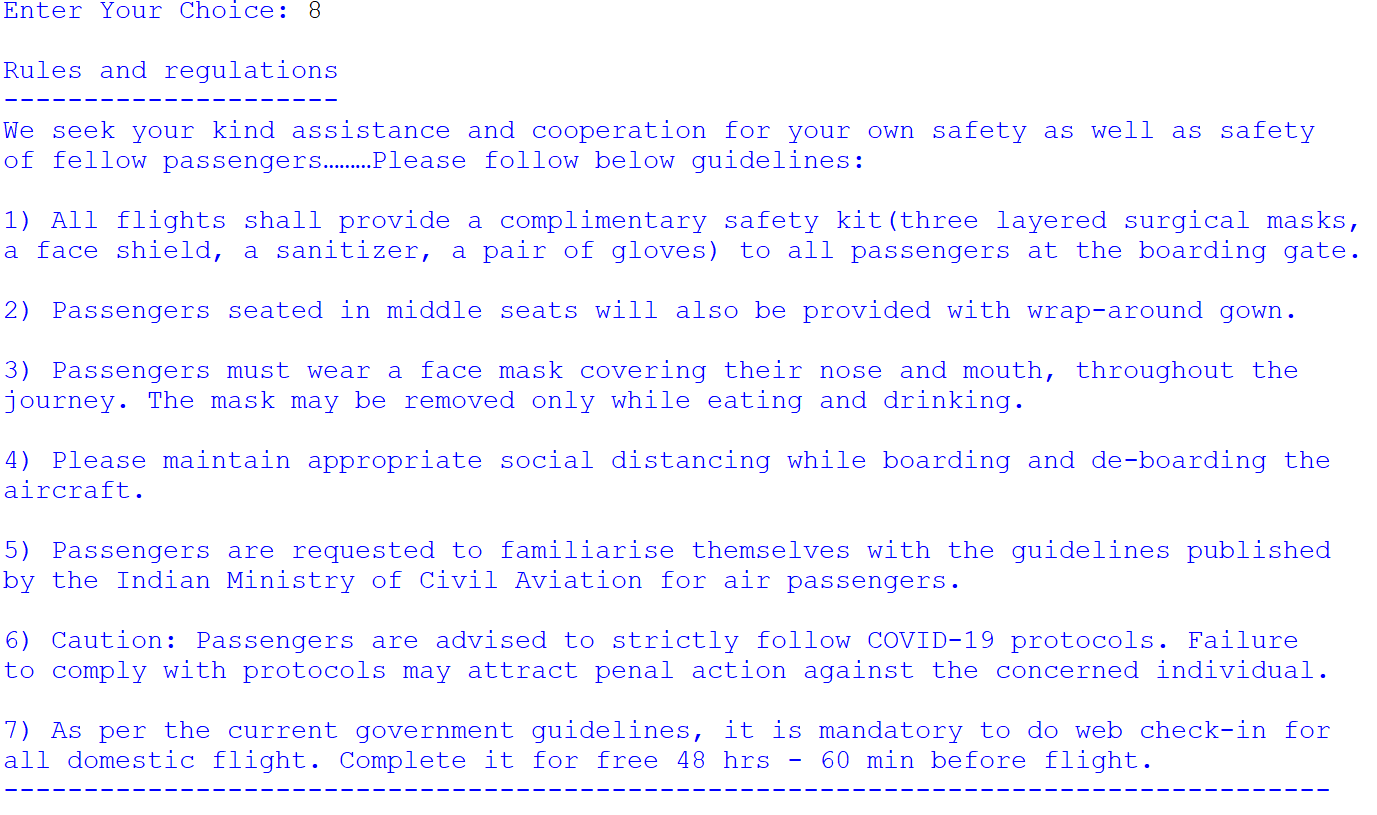
1. User selects 7 “About us” option from list of Services.

* Details about “MakeMyAirTrip” will display

****

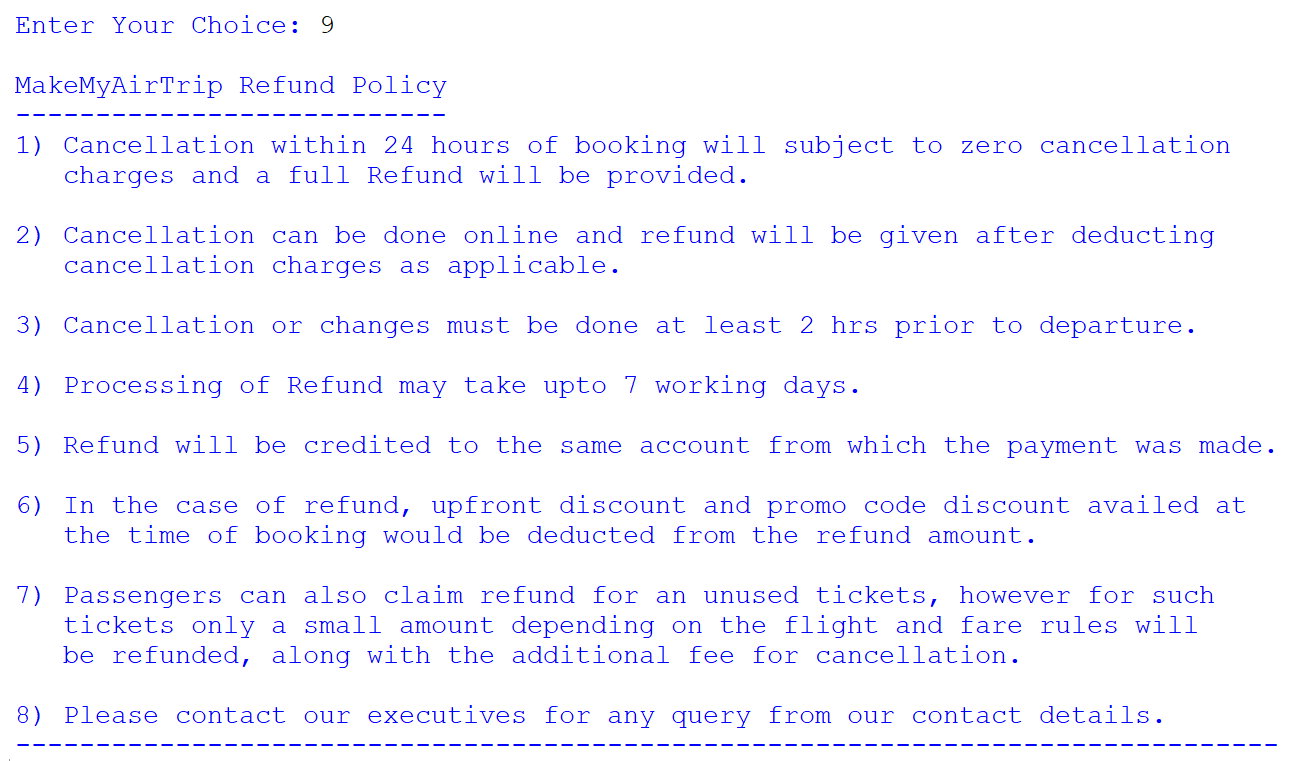
1. User selects 8 “Rules and Regulations” option from list of Services.

* Government air travel “Rules and Regulations” will display



1. User selects 9 “Refund Policy” option from list of Services.

* Refund related information will display



**CONCLUSION**

First of all, we are very thankful to our computer teacher and our school for giving us this wonderful opportunity to take up this project. It was not possible to complete this project without their extensive help and immense support.

Doing this project was lot of fun and full of learnings. We explored and learned many things like python programming, SQL connectivity with python, creation of tables & their handing, creation of files & their handing, basics of database creation and handling, air ticket booking process by a website etc.

We have not used GUI for this project, which would have made this program more interactive, user-friendly and attractive. The scope of extension of this project is to provide connectivity with actual flight databases and extract real life data for using this website for air travel booking services.

In the end, we are proud of our project as this is the outcome of our immense hard work, determination, team work and support.

**BIBLIOGRAPHY**

**Text book used**

Sumita Arora python textbook Class 12

**Websites used**

[Python (programming language) - Wikipedia](https://en.wikipedia.org/wiki/Python_(programming_language))

[www.thispointer.com](http://www.thispointer.com)