A

Project Report

On

BUS RESERVATION SYSTEM

Submitted by-

Mrinmoy Mahato

ACKNOWLEDGEMENT

The objective of this project is to let the students apply the programming knowledge into a real-world situation/problem and exposed the students how programming skills helps in developing a good software.

• Write programs utilizing modern software tools.

• Apply object-oriented programming principles effectively when developing small to medium sized projects.

• Write effective procedural code to solve small to medium sized problems.

• Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.

• Students will demonstrate ability to conduct research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

CONTENTS

INTRODUCTION

This project is about Bus Ticket Reservation System. This is the system for ticket booking in online remotely any type of location in India and also helps to the bus owner to manage their business by using this system it's also helps to owner record from bus business and also helps to reduced paper to track any type of bus information such as ticket information passenger information, seat booking information any type of user can see the details of seat which are booked and which is available it's an automated process if any user book ticket in confirmation of ticket other user can book ticket no need to refresh page when user confirm ticket in other hand it is totally automated system not only admin see and operator but also seat details of bus passengers.

The main purpose of our system is the business owner not only benefit general user but also maintain their business and reduced paper to save information any type of bus seats, departure date, seat booking. passenger every record saved in database and also collated money from bus ticket from user they can know the update of transaction this project done ourself we are trying our best to save record money but in future we will try to implement this feature.

HARDWARE AND SOFTWARE REQUIREMENTS

- Proposed System Environment
- Hardware Specifications

  RAM.....8GB

  Hard disk..... 500 GB

  CPU Type.....12thGeneration

- Software Specifications

  Operating system.....Windows11

  Software .....Python 3.6

  Microsoft Word 2021

  Printer

SOFTWARE DETAILS

PYTHON: -

Python is a high-level, general-purpose programming language its design philosophy emphasizes code readability with the use of significant indentation. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

FEATURES IN PYTHON

- Easy to code: Python is a high-level programming language....

- Free and Open Source: Python is developed under an OSI, making it freely usable and distributable.
- Object-Oriented Language: Python is having the exception of control flow, everything in python is an object.
- GUI Programming Support: Python supports GUI frameworks or toolkits, from Tkinter which is traditionally bundled.
- High-Level Language: Python is easy and free to use.
- Extensible feature: Python is also written in C++ language.
- Python is Integrated language: Python interpreter executes codes one line at a time.

FILE HANDLING IN PYTHON: -

Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files. The concept of file handling has stretched over various other languages, but the implementation is either complicated or lengthy, but like other concepts of Python. this concept here is also easy and short. Python

treats file differently as text or binary and this is important. Each line of code includes a sequence of characters and they form text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma (.) or newline character. It ends the current line and tells the interpreter a new one has begun. Let's start with Reading and Writing files.

WORKING OF OPEN() FUNCTION

Before performing any operation on the file like read or write, first we have to open that file. For this, we should use Python's inbuilt function open()

But at the time of opening, we have to specify the mode, which represents the purpose of the opening file.

f = open(filename, mode)

Where the following mode is supported:

1. r: open an existing file for a read operation

2. w: open an existing file for a write operation. If the file already contains some data, then it will be overridden.

3. a: open an existing file for append operation. It won't override existing data.

4. r+: To read and write data into the file. The previous data in the file will not be deleted.

5. w+: To write and read data. It will override existing data.

6. a+: To append and read data from the file. It won't override existing data.

INTRODUCTION

BUS RESERVATION SYSTEM

Bus Ticket Reservation system gives the opportunity to the general users/visitors to book the ticket remotely anytime in anywhere. By using this application, users will be able to select the bus reserves the seats for going from staring place to destination place at their fixed time and date. For booking ticket, users confirm their personal information such as name; email number and mobile number for confirmation of ticket. If any user thinks about to go anywhere by using bus. They can get whole features in our application. If any owner thinks about to maintain his/her business and they can get whole privilege in our application.

Bus Ticket Booking System is aimed at reducing paper for the bus industry and hence improving its efficiency and speeding up of all processes.

This project is aimed at developing an automation application that will apply a system approach to depict the functional architecture. The Purpose of our project

is to develop " Bus Ticket Booking system" for owner to maintain their business and total details of bus and departure bus and also know ticket information which is booked and which is not booked and details of total collection of money for specific days and instant details of collection of money from bus ticket. Our systems innovative facilities are added for general users to book ticket anytime and anywhere remotely and general users can be able to know bus details and fare and seats combination is easy to track the information of all and departure destination General users can be informed the seat which is booked and which is not booked. This feature is added for general users. Every business owner can take the opportunity to use our system. The main goal of our system is to build a good management tool for all customers. The main purpose of this software is to reduce the time taken through manual system in order to maintain all the records and operation virtually. Those activities are longer in physically in our present life are easy to access virtually and save the customer’s valuable time. All aged customers can use our software to do the completion booking ticket online at their house and their working environment.

TABLE DISCRIPTION

Table : LOGIN

Field	Type	Size	Constraints
UserName	Varchar2	20	Primary Key
Password	Varchar2	20	Not null

LOGIN data-

User Name	Password
Mrinmoy	qwerty12
Ishita	Asdfg34
Priyanshu	Mann07
Dharmendar	Dhar56

Table : DISTANCE

Field	Type	Size	Constraints
FROM	Varchar2	15	Not null
TO	Varchar2	15	Not null
DISTANCE	Number	4	Not null

DISTANCE data-

FROM	TO	DISTANCE
Jamshedpur	Narwa	20
Jamshedpur	Chandil	30
Jamshedpur	Ghatshila	48
Jamshedpur	Chaibasa	67
Jamshedpur	Ranchi	126
Jamshedpur	Bokaro	133
Jamshedpur	Dhandbad	140
Narwa	Jamshedpur	20
Hata	Jamshedpur	26
Chandil	Jamshedpur	30
Ghatshila	Jamshedpur	48
Hazaribagh	Jamshedpur	77
Ranchi	Jamshedpur	126
Bokaro	Jamshedpur	133
Dhandbad	Jamshedpur	140

Table : BUS_DETAILS

Field	Type	Size	Constraints
BUS_ID	Varchar2	3	Primary Key
BUS_NAME	Varchar2	15	Not null
FROM	Varchar2	15	Not null
TO	Varchar2	15	Not null
TOTAL_SEATS	Number	3	---
DAY	Varchar2	50	---
DISTANCE	Number	4	Not null
TIME	Varchar2	5	Not null
TERRIF	Number	3	---
FARE	Number	5	Not null

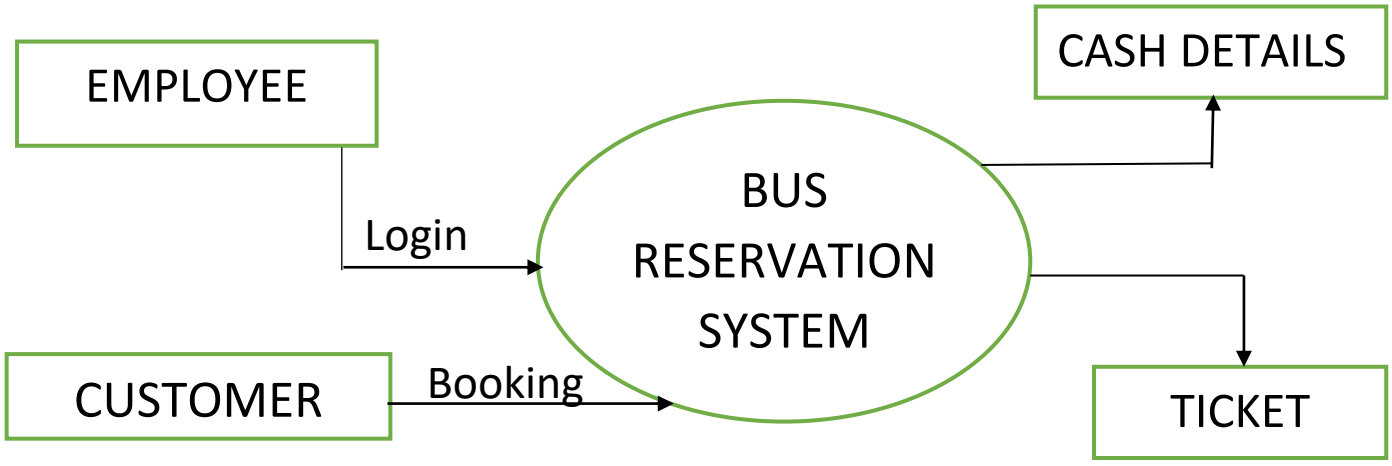
Table : PASSENGER DETAILS

Field	Type	Size	Constraints
BOOKING_ID	Varchar2	20	Primary Key
DATE	Varchar2	20	Not null
SEAT_NO.	Number	3	Not null
BUS_ID	Varchar2	4	Foreign key references BUS_Details
BUS_NAME	Varchar2	15	
FROM	Varchar2	15	

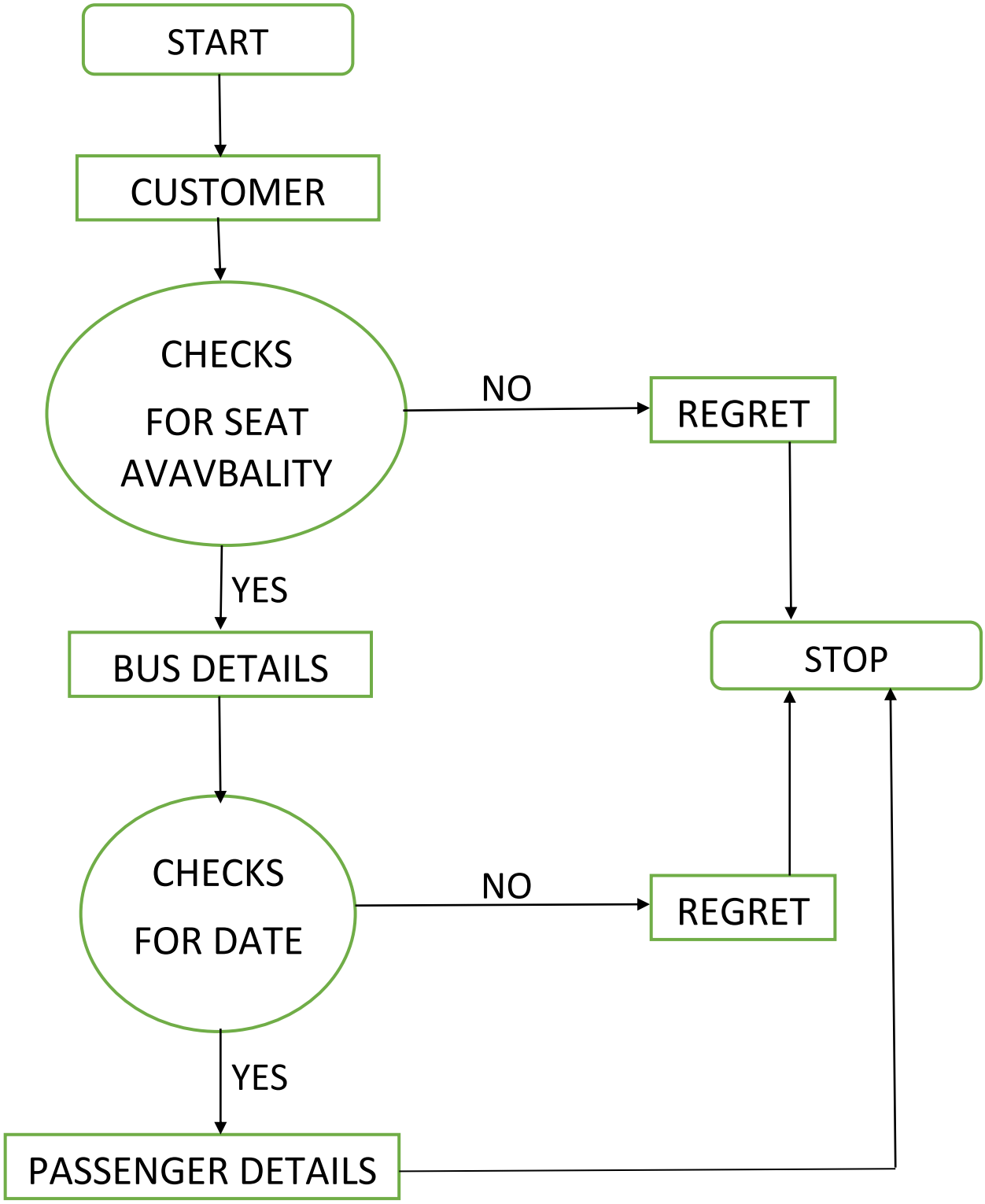
TO	Varchar2	15	
TIME	Varchar2	5	
FARE	Number	5	
NAME	Varchar2	20	Not null
AGE	Number	3	---
PHONE	Number	10	Not null
EMAIL	Varchar2	25	Not null

DATA FLOW DIAGRAM

1. Bus Reservation System,0 LEVEL DFD:



2.Bus Reservation System,1 LEVEL DFD:



PYTHON CODING

Login code:

```
import pickle
f=open('LoginDetails.obj','wb')
L=[['Mrinmoy','qwerty12'],
   ['Ishita','Asdfg34'],
   ['Priyanshu','Mann07'],
   ['Dharmendar','Dhar56']]
pickle.dump(L,f)
f.close()
```

Distance code:

```
import pickle
f=open("DISTANCE.obj",'wb')
L=[['Jamshedpur','Narwa',20],
   ['Jamshedpur','Chandil',30],
   ['Jamshedpur','Ghatshila',48],
   ['Jamshedpur','Ranchi',126],
```

```
    ['Jamshedpur','Bokaro',133],
    ['Jamshedpur','Dhandbad',140],
    ['Narwa','Jamshedpur',20],
    ['Chandil','Jamshedpur',30],
    ['Ghatshila','Jamshedpur',48],
    ['Hazaribagh','Jamshedpur',77],
    ['Ranchi','Jamshedpur',126],
    ['Bokaro','Jamshedpur',133],
    ['Dhandbad','Jamshedpur',140]]
pickle.dump(L,f)
f.close()
```

Bus Detail code:

```
def Distance(From,To):
    import pickle
    f1=open("DISTANCE.obj",'rb')
    R=pickle.load(f1)
    F=0
    for i in R:
```

```
            if(i[0]==From and i[1]==To):
                D=i[2]
                F=1
                break
    if F==0:
        D=int(input('Enter Distance Manually(in Km):'))
    f1.close()
    return(D)

def NewBus():
    print('ADDING A NEW BUS')
    import pickle
    f1=open('Bus_Details.obj','rb')
    L=pickle.load(f1)
    f1.close()
    f=open('Bus_Details.obj','wb')
    Busid=input('Enter Bus id:')
    Bname=input('Enter Bus Name:')
    From=input('Enter pickup Location:')
```

```
To=input('Enter Destination:')
d='y'
Day=[]
while d in 'yY':
    D=input('Enter Day of Bus:')
    Day.append(D)
    d=input('Want to enter more day(Y/N):')
Dist=Distance(From,To)
Time=input('Enter Time:')
Seat=int(input('Enter Total Available Seats:'))
T=20
TF=T*Dist
Rec=[Busid,Bname,From,To,Seat,Day,Dist,Time,T,TF]
L.append(Rec)
pickle.dump(L,f)
f.close()
print('NEW BUS ADDED SUCESSFULLY')
print('.'*187)
```

```
def DeleteBus():
    print('REMOVING A BUS')
    import pickle
    f=open('Bus_Details.obj','rb')
    R=pickle.load(f)
    lst=[]
    Bid=input('Enter Bus id:')
    for i in R:
        if(i[0]!=Bid):
            lst.append(i)
    f.close()
    f=open('Bus_Details.obj','wb')
    pickle.dump(lst,f)
    f.close()
    print('BUS REMOVED SUCESSFULLY')
    print('.'*187)
def DisplayBus():
    import pickle
    f=open('Bus_Details.obj','rb')
```

```
    R=pickle.load(f)
print('\nBusid\tBusname\tFrom\tTo\tTotal_Seats\tDay
\tDistance\tTime\tTeriff\tFare')
    print('--------     -------------     -------     ----        ----------
----             -----    ------------      -------     -------      --------
')
    for Rec in R:
        for i in Rec:
            print(i,end='\t')
        print()
    f.close()
    print('.'*187)
def AllBooking():
    import pickle
    f=open('Passenger_Details.obj','rb')
    R=pickle.load(f)
    print('\nBookingID | Date         Seat_No.
Busname\tTime\tFare\tName\t\tPhone\t\tE-mail')
    print('---------------  ------      ------------    -------------     ---
----        -------       --------          ---------          ---------')
```

```
    for i in R:
        print(f'{i[0]}\t  {i[1]}  {i[2]}\t   {i[4]}
            {i[7]}\t{i[8]}\t{i[9]}\t{i[11]}\t{i[12]}\t')
    f.close()
    print('.'*187)

def Cost():
    import pickle
    f=open('Passenger_Details.obj','rb')
    R=pickle.load(f)
    D=[]
    d=R[0][1]
    for Rec in R:
        if d!=Rec[1]:
            d=Rec[1]
        if d not in D:
            D.append(d)
    for S in D:
        c=0
```

```
        for Rec in R:
            if S==Rec[1]:
                c+=Rec[8]
        print(f'Total PROFIT on {S} is ₹{c}')
```

Booking Code:

```
def FindDay(date):
    import calendar
    d=date.split('/')
    D,M,Y=int(d[0]),int(d[1]),int(d[2])
    wno=calendar.weekday(Y,M,D)
    w=['Mon','Tue','Wed','Thr','Fri','Sat','Sun']
    return(w[wno])


def login():
    import pickle
    f=open('LoginDetails.obj','rb')
    Uid=input('Enter UserName:')
```

```
    Pwd=input('Enter Password:')
    R=pickle.load(f)
    for i in R:
        if (i[0]==Uid and i[1]==Pwd):
            return(1)
    return(0)

def SeatCheck(L):
    import pickle
    f=open('Passenger_Details.obj','rb')
    try:
        R=pickle.load(f)
        c=0
        for i in R:
            if i[3]==L[0]:
                c+=1
        if c<L[4]:
            return(1,c+1)
        else:
```

```
            return(0,c+1)
    except:
        return(1,1)


def FindBus(From,To):
    global NB
    NB=[]
    import pickle
    f=open('Bus_Details.obj','rb')
    R=pickle.load(f)
    for Rec in R:
        if Rec[2]==From and Rec[3]==To:
            print(f'\nBusid-{Rec[0]}\nBus Name-
{Rec[1]}\nFrom-{Rec[2]}\nTo-{Rec[3]}\nDistance-
{Rec[6]}\nTime-{Rec[7]}\nTeriff-{Rec[8]}\nTotalFare-
{Rec[9]}')
            q,Sno=SeatCheck(Rec)
            print('AVAILABLE DAYS:')
            for i in Rec[5]:
                print(i,end='\n')
```

```
        w=input('Want to Book this Bus(Y/N):')
        if w not in 'yY':
            break
        if q==1:
            Date=input('Enter Your applicable
Date(D/M/Y):')
            while True:
                Day=FindDay(Date)
                if Day in Rec[5]:
                    print(f'The Date {Date} is available for this
bus. ')
                    break
                else:
                    Date=input('This Date is not availabe \n
Please write another Date(D/M/Y):')

NB=[Date,Sno,Rec[0],Rec[1],From,To,Rec[7],Rec[9]]
        break
    else:
        print('##WRONG BUS DESTINATION##')
```

```
    f.close()


def Bookingid(B):
    import pickle
    f=open('Passenger_Details.obj','rb')
    R=pickle.load(f)
    c=1
    for i in R:
        if i[3]==B:
            c+=1
    e=B+'_'+str(c)
    return(e)


def NewBooking():
    print('\nBOOKING A BUS:-')
    import pickle
    f=open('Passenger_Details.obj','rb')
    N=pickle.load(f)
    From=input('Enter pickup Location:')
```

```
To=input('Enter Destination:')
FindBus(From,To)
if len(NB)!=0:
    Bookid=Bookingid(NB[2])
    Name=input('Enter Passenger\'s Name:')
    Age=int(input('Enter Passenger\'s Age:'))
    Phone=int(input('Enter Passenger\'s Mobile no.:'))
    Mail=input('Enter Passenger\'s Mail-id: ')
    print('\nYour Booking Id is:',Bookid)
    Rec=[Bookid]+NB+[Name,Age,Phone,Mail]
    f=open('Passenger_Details.obj','wb')
    N.append(Rec)
    pickle.dump(N,f)
    f.close()
    print('\nBOOKING PLACED SUCESSFULY\n')
else:
    print('---NO SEAT AVAILABLE---')
print('.'*187)
```

```
def DisplayBooking():
    print('SHOWING YOUR BOOKING:-')
    Id=input('Enter your Booking Id:')
    m=input('Enter your Mail Id:')
    import pickle
    f=open('Passenger_Details.obj','rb')
    R=pickle.load(f)
    for Rec in R:
        if Rec[0]==Id and Rec[12]==m:
            print(f'\nBooking id-{Rec[0]}\nBus Date-
{Rec[1]}\nSeat No.:{Rec[2]}\nBus Id-
{Rec[3]}\nBusname-{Rec[4]}\nFrom-{Rec[5]}\nTo-
{Rec[6]}\nTime-{Rec[7]}\nTotalFare-{Rec[8]}\nName-
{Rec[9]}\nAge-{Rec[10]}\nMobile-{Rec[11]}\nMail Id-
{Rec[12]}\n')
            break
    else:
        print('**WRONG BOOKING ID OR E-MAIL**')
    f.close()
    print('.'*187)
```

```
def CancelBooking():
    print('CANCELING YOUR BOOKING')
    import pickle
    f=open('Passenger_Details.obj','rb')
    R=pickle.load(f)
    lst=[]
    Bid=input('Enter Booking id:')
    m=input('Enter your Mail Id:')
    for Rec in R:
        if Rec[0]==Bid and Rec[12]==m:
            pass
        else:
            lst.append(Rec)
    f.close()
    f=open('Passenger_Details.obj','wb')
    pickle.dump(lst,f)
    f.close()
    print('BOOKING CANCELLED SUCESSFULLY')
    print('.'*187)
```

Main Code:

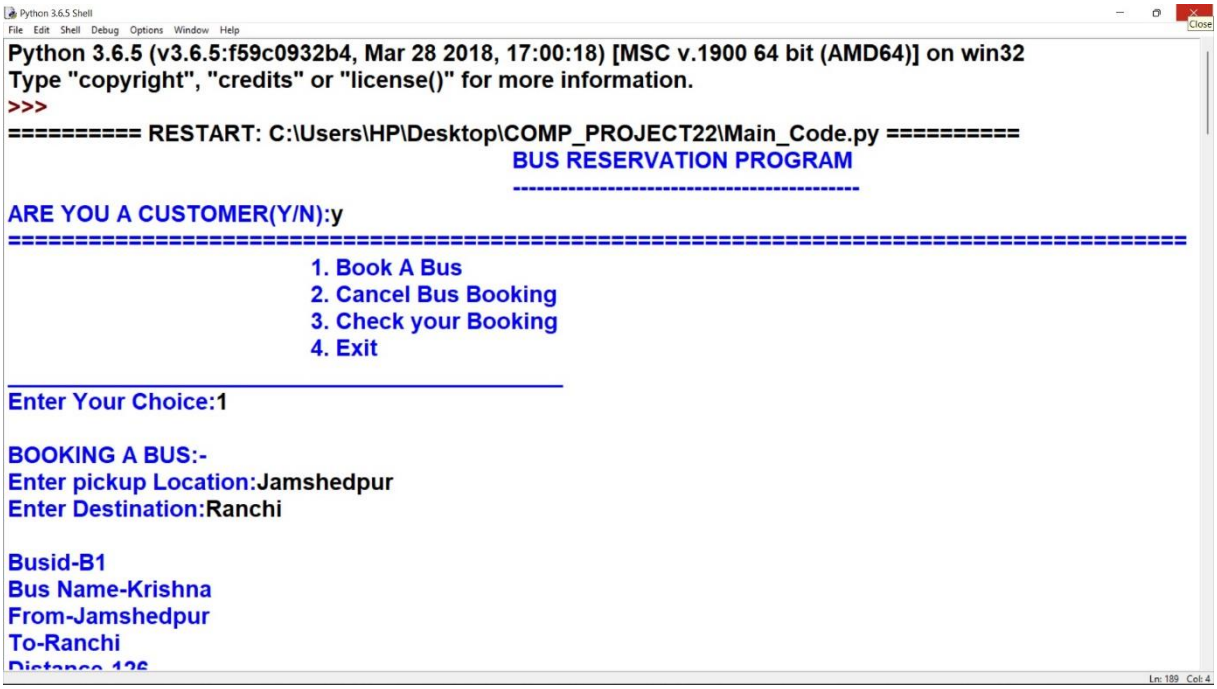
```
from bus_details import
Distance,NewBus,DeleteBus,DisplayBus,AllBookin
g,Cost

from Booking_Code import
FindDay,login,FindBus,Bookingid,SeatCheck,NewB
ooking,DisplayBooking,CancelBooking


def Customer():
    print('\t\t\t1. Book A Bus \n\t\t\t2. Cancel Bus
Booking \n\t\t\t3. Check your Booking\n\t\t\t4.
Exit')
    print('_'*44)
    ch=int(input('Enter Your Choice:'))
    while ch in [1,2,3]:
        if ch==1:
            NewBooking()
        elif ch==2:
            CancelBooking()
```

```
        elif ch==3:
            DisplayBooking()
        print('\t\t\t1. Book A Bus \n\t\t\t2. Cancel
Bus Booking \n\t\t\t3. Check your
Booking\n\t\t\t4. Exit')
        print('_'*44)
        ch=int(input('Again enter Your
Choice(4.exit):'))

def employee():
    print('\t\t\t1. Add a New Bus \n\t\t\t2. Delete a
Bus Booking \n\t\t\t3. Display All Bus\n\t\t\t4.
Display All Booking \n\t\t\t5. Display Profit of each
Date 6. Exit')
    print('_'*44)
    ch=int(input('Enter Your Choice:'))
    while ch in [1,2,3,4,5]:
        if ch==1:
            NewBus()
```

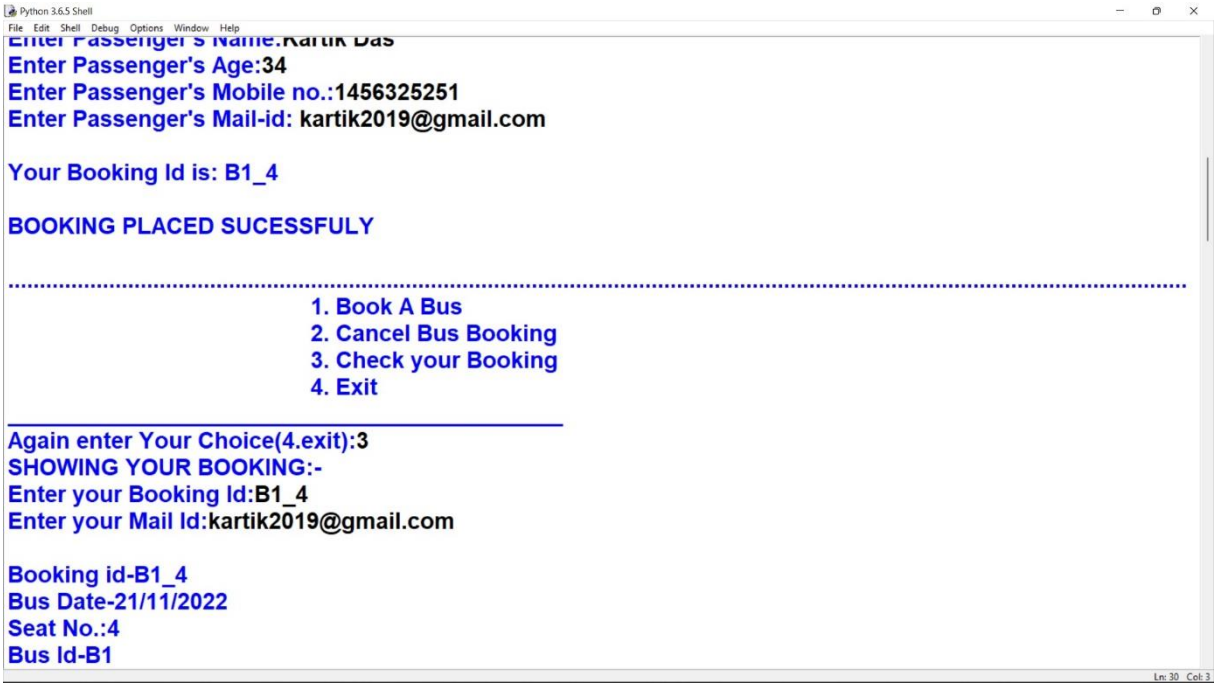
```
        elif ch==2:
            DeleteBus()
        elif ch==3:
            DisplayBus()
        elif ch==4:
            AllBooking()
        elif ch==5:
            Cost()
        print('\t\t\t1. Add a New Bus \n\t\t\t2. Delete a Bus Booking \n\t\t\t3. Display All Bus\n\t\t\t4. Display All Booking \n\t\t\t5. Display Profit of each Date 6. Exit')
        print('_'*44)
        ch=int(input('Again enter Your Choice(6.exit):'))

#MAIN PROGRAM
print('\t\t\t\t\tBUS RESERVATION PROGRAM')
print('\t\t\t\t\t----------------------------------------')
```

```
ans1='y'
while ans1 in 'yY':
    q=input('ARE YOU A CUSTOMER(Y/N):')
    print('='*88)
    if (q in 'yY'):
        Customer()
    else:
        an=login()
        if an==1:
            print('^^LOGIN SUCESSFULL^^')
            employee()
            break
        else:
            print('WRONG PASSWORD')
    ans1=input('Want to continue(Y/N):')
    print('='*88)
print('!!! THE PROGRAM IS SUCCESSFULLY CLOSED !!!')
```

OUTPUT

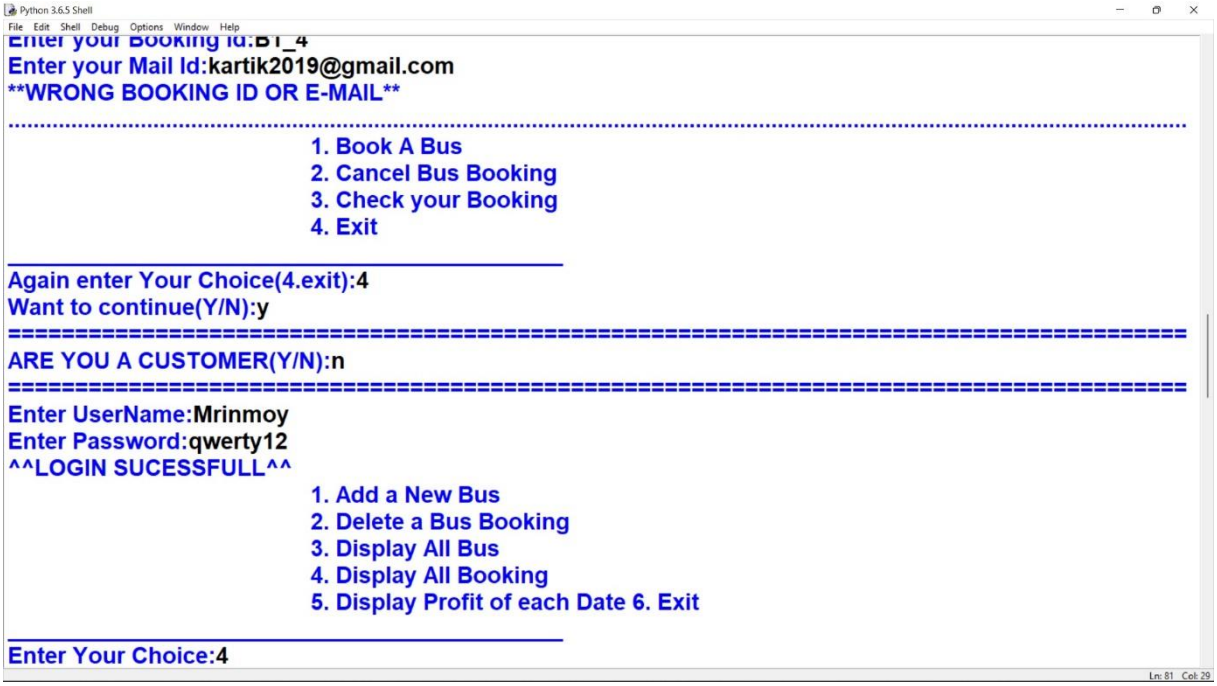

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\COMP_PROJECT22\Main_Code.py ==========
BUS RESERVATION PROGRAM
ARE YOU A CUSTOMER(Y/N):y
1. Book A Bus
2. Cancel Bus Booking
3. Check your Booking
4. Exit
Enter Your Choice:1
BOOKING A BUS:-
Enter pickup Location:Jamshedpur
Enter Destination:Ranchi
Busid-B1
Bus Name-Krishna
From-Jamshedpur
To-Ranchi


Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Busid-B1
Bus Name-Krishna
From-Jamshedpur
To-Ranchi
Distance-126
Time-6:00
Teriff-20
TotalFare-2520
AVAILABLE DAYS:
Mon
Wed
Thr
Want to Book this Bus(Y/N):y
Enter Your applicable Date(D/M/Y):22/11/2022
This Date is not availabe
Please write another Date(D/M/Y):21/11/2022
The Date 21/11/2022 is available for this bus.
Enter Passenger's Name:Kartik Das
Enter Passenger's Age:34
Enter Passenger's Mobile no.:1456325251
Enter Passenger's Mail-id: kartik2019@gmail.com
Your Booking Id is: B1_4

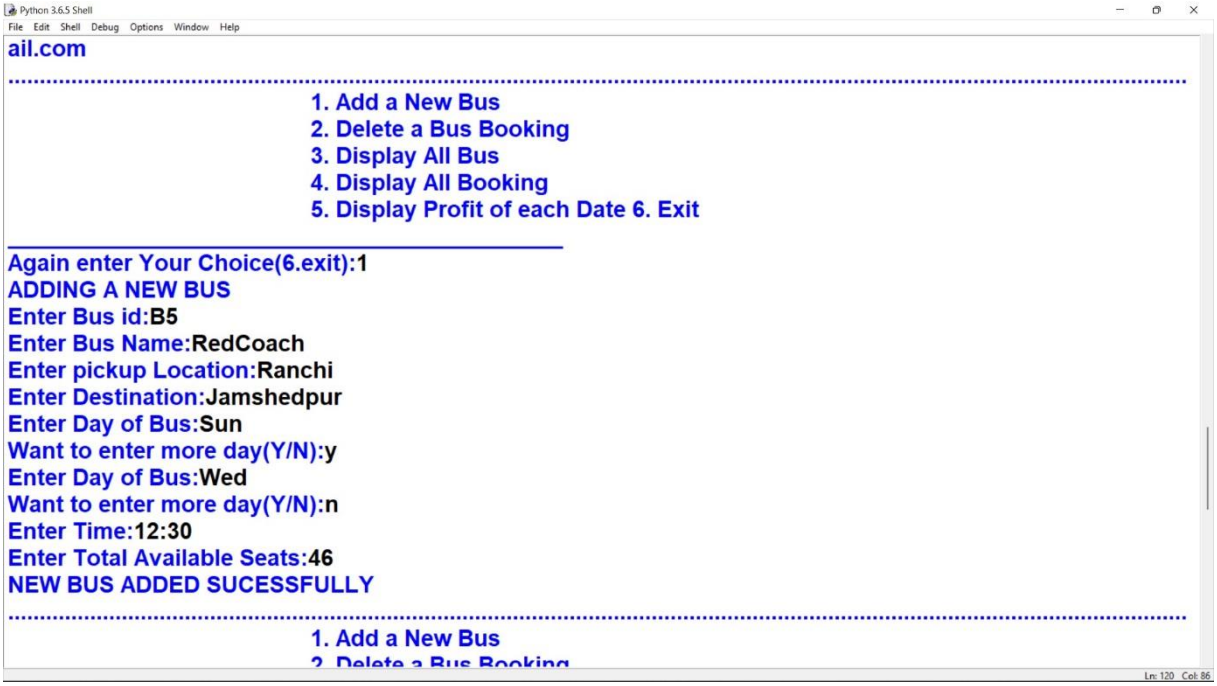

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Enter Passenger's Name:Kartik Das
Enter Passenger's Age:34
Enter Passenger's Mobile no.:1456325251
Enter Passenger's Mail-id: kartik2019@gmail.com
Your Booking Id is: B1_4
BOOKING PLACED SUCESSFULY
1. Book A Bus
2. Cancel Bus Booking
3. Check your Booking
4. Exit
Again enter Your Choice(4.exit):3
SHOWING YOUR BOOKING:-
Enter your Booking Id:B1_4
Enter your Mail Id:kartik2019@gmail.com
Booking id-B1_4
Bus Date-21/11/2022
Seat No.:4
Bus Id-B1
Ln: 30 Col: 3


Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Enter your Mail Id:kartik2019@gmail.com
Booking id-B1_4
Bus Date-21/11/2022
Seat No.:4
Bus Id-B1
Busname-Krishna
From-Jamshedpur
To-Ranchi
Time-6:00
TotalFare-2520
Name-Kartik Das
Age-34
Mobile-1456325251
Mail Id-kartik2019@gmail.com
1. Book A Bus
2. Cancel Bus Booking
3. Check your Booking
4. Exit
Again enter Your Choice(4.exit):2
CANCELING YOUR BOOKING
Ln: 52 Col: 32

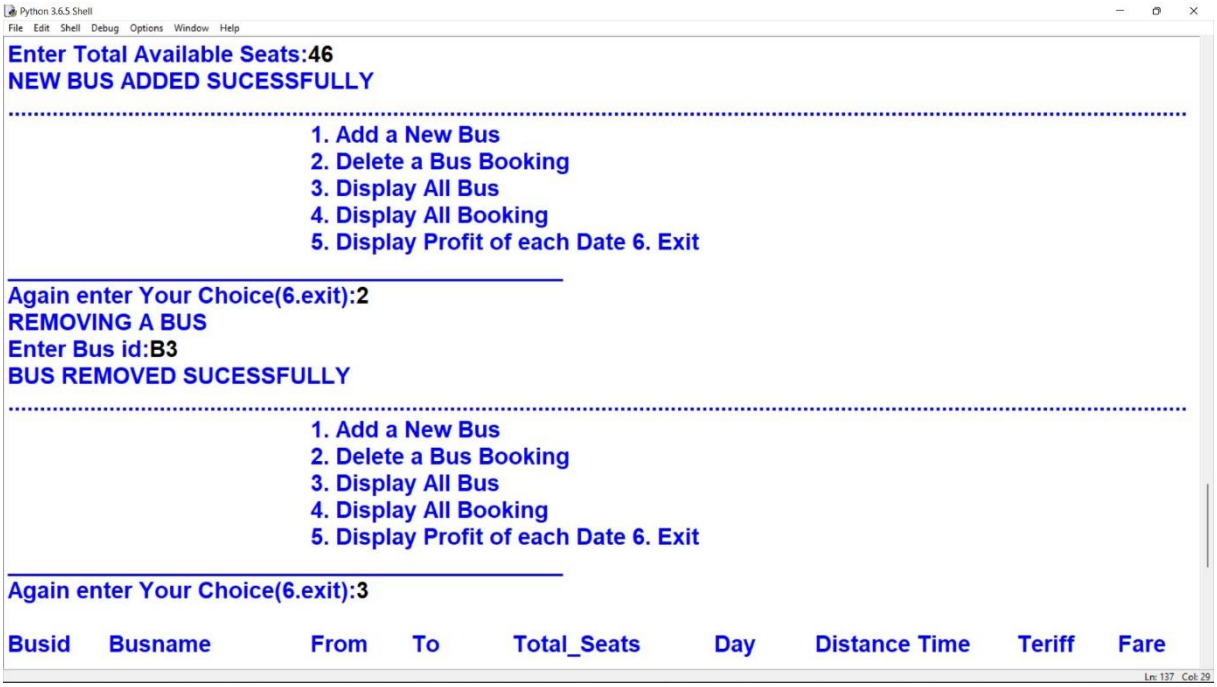

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
1. Book A Bus
2. Cancel Bus Booking
3. Check your Booking
4. Exit
Again enter Your Choice(4.exit):2
CANCELING YOUR BOOKING
Enter Booking id:B1_4
Enter your Mail Id:kartik2019@gmail.com
BOOKING CANCELLED SUCESSFULLY
1. Book A Bus
2. Cancel Bus Booking
3. Check your Booking
4. Exit
Again enter Your Choice(4.exit):3
SHOWING YOUR BOOKING:-
Enter your Booking Id:B1_4
Enter your Mail Id:kartik2019@gmail.com
WRONG BOOKING ID OR E-MAIL
Ln: 66 Col: 14


Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Enter your Booking Id:B1_4
Enter your Mail Id:kartik2019@gmail.com
WRONG BOOKING ID OR E-MAIL
1. Book A Bus
2. Cancel Bus Booking
3. Check your Booking
4. Exit
Again enter Your Choice(4.exit):4
Want to continue(Y/N):y
ARE YOU A CUSTOMER(Y/N):n
Enter UserName:Mrinmoy
Enter Password:qwerty12
^^LOGIN SUCESSFULL^^
1. Add a New Bus
2. Delete a Bus Booking
3. Display All Bus
4. Display All Booking
5. Display Profit of each Date 6. Exit
Enter Your Choice:4
Ln: 81 Col: 29

```
Python 3.6.5 Shell
File  Edit  Shell  Debug  Options  Window  Help
                              4. Display All Booking
                              5. Display Profit of each Date 6. Exit
_____________________________________________
Enter Your Choice:4

BookingID | Date       Seat_No.   Busname      Time    Fare     Name            Phone          E-mail
--------- ------       ---------- ---------    -----   -----    ----            -----          -----
B6_1      20/11/2022   1          Decker 6:45  520     Shreya rajput    6716783100     sr2022@gmail.c
om
B1_1      21/11/2022   1          Krishna      6:00    2520     Virat Singh     8638738010     virat20
19@gmail.com
B1_2      21/11/2022   2          Krishna      6:00    2520     Harsh Kumar     7507582500     harsh4
49@gmail.com
B4_1      20/11/2022   1          Neptune      12:30   2900     Kajal Agarwal   5430664358     kajal19
@gmail.com
B1_3      21/11/2022   3          Krishna      6:00    2520     Shiv Prakash    2101025409     sp23@
gmail.com
B2_1      22/11/2022   1          Geetanjali   3:30    2660     Ammu     2103879059     ammu2022@gm
ail.com
.....................................................................................................................

                              1. Add a New Bus
                              2. Delete a Bus Booking
                              3. Display All Bus
                              4. Display All Booking
                                                                                          Ln: 99  Col: 33
```



```
Python 3.6.5 Shell
File  Edit  Shell  Debug  Options  Window  Help
ail.com
.....................................................................................................................

                              1. Add a New Bus
                              2. Delete a Bus Booking
                              3. Display All Bus
                              4. Display All Booking
                              5. Display Profit of each Date 6. Exit
_____________________________________________
Again enter Your Choice(6.exit):1
ADDING A NEW BUS
Enter Bus id:B5
Enter Bus Name:RedCoach
Enter pickup Location:Ranchi
Enter Destination:Jamshedpur
Enter Day of Bus:Sun
Want to enter more day(Y/N):y
Enter Day of Bus:Wed
Want to enter more day(Y/N):n
Enter Time:12:30
Enter Total Available Seats:46
NEW BUS ADDED SUCESSFULLY
.....................................................................................................................

                              1. Add a New Bus
                              2. Delete a Bus Booking
                                                                                          Ln: 120  Col: 86
```



```
Python 3.6.5 Shell
File  Edit  Shell  Debug  Options  Window  Help
Enter Total Available Seats:46
NEW BUS ADDED SUCESSFULLY
.....
                        1. Add a New Bus
                        2. Delete a Bus Booking
                        3. Display All Bus
                        4. Display All Booking
                        5. Display Profit of each Date 6. Exit
_____
Again enter Your Choice(6.exit):2
REMOVING A BUS
Enter Bus id:B3
BUS REMOVED SUCESSFULLY
.....
                        1. Add a New Bus
                        2. Delete a Bus Booking
                        3. Display All Bus
                        4. Display All Booking
                        5. Display Profit of each Date 6. Exit
_____
Again enter Your Choice(6.exit):3

Busid   Busname          From      To        Total_Seats       Day       Distance Time   Teriff   Fare
                                                                                                    Ln: 137  Col: 29
```

```
Python 3.6.5 Shell
File  Edit  Shell  Debug  Options  Window  Help
                        2. Delete a Bus Booking
                        3. Display All Bus
                        4. Display All Booking
                        5. Display Profit of each Date 6. Exit
_____
Again enter Your Choice(6.exit):3

Busid   Busname          From      To        Total_Seats       Day       Distance Time   Teriff   Fare
--------  -------------  -------   ----  --------------      -----     ------------ -------  -------  --------
B1      Krishna  Jamshedpur       Ranchi    35        ['Mon', 'Wed', 'Thr']          126      6:00     20
2520
B2      Geetanjali      Jamshedpur       Bokaro    10        ['Tue', 'Thr', 'Sat']  133      3:30     20
2660
B4      Neptune  Purulia Ranchi    69        ['Sun']   145       12:30     20       2900
B6      Decker   Hata    Jamshedpur       55        ['Tue', 'Fri', 'Sun']  26       6:45     20       520

B5      RedCoach        Ranchi    Jamshedpur        46        ['Sun', 'Wed']         126      12:30    20
2520
.....
                        1. Add a New Bus
                        2. Delete a Bus Booking
                        3. Display All Bus
                        4. Display All Booking
                        5. Display Profit of each Date 6. Exit
                                                                                                    Ln: 137  Col: 29
```


Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
B5 RedCoach Ranchi Jamshedpur 46 ['Sun', 'Wed'] 126 12:30 20
2520
1. Add a New Bus
2. Delete a Bus Booking
3. Display All Bus
4. Display All Booking
5. Display Profit of each Date 6. Exit
Again enter Your Choice(6.exit):5
Total PROFIT on 20/11/2022 is ₹3420
Total PROFIT on 21/11/2022 is ₹7560
Total PROFIT on 22/11/2022 is ₹2660
1. Add a New Bus
2. Delete a Bus Booking
3. Display All Bus
4. Display All Booking
5. Display Profit of each Date 6. Exit
Again enter Your Choice(6.exit):6
!!! THE PROGRAM IS SUCCESSFULLY CLOSED !!!
>>>
Ln: 137 Col: 29

CONCLUSION

Nowadays in this generation every work is required to be donr in an easy way to efficiently use time and effort yet still having a productive report. An example of this is upgrading the traditional Bus Booking system to Python-Coding based Bus Reservation system. In Bus Reservation System, we have developed a secure, user-friendly Bus Reservation System. This Project basically provides Bus information and regarding bus information. First of all, a customer view our system and search the bus by its location and how many seats are available in our buses. It is user friendly and accurate.