

**Kanpur**

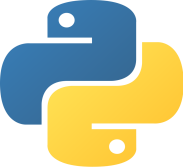
**2020-2021**

**Computer Science Project on**

Railway Reservation System

**Using Python**

u

****

**Submitted to: Submitted by:**

Mr. Rakesh Goswami Sir Swatambra SahuXII-‘A’

**Rollno.:**

**Certificate**

This is to certify the project titled on **Railway Reservation System** under my guidence and supervision has been completed by **Swatambra Sahu** student of class **XII-A, Dr. Virendra**

**Swarup Education Centre** as per the

requirement of **Central Board of Secondary Education** for the year 2020-21.

**Signature Signature Signature**

**(Subject Teacher) (Principal) (External Examiner)Acknowledgment**

I take this opportunity to express my profound gratitude and indebtedness to my teacher and faculty guide **Mr. Rakesh Goswami** for his

sincere guidance and motivation in completion

of this project.

I would like to express my heartfelt obligation to our principal **Mrs. Vijayshree Pandey**, for her coordination in extending every possible support for the completion of this project.

.

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

**Synopsis**

**INTRODUCTION**

This project introduces railway reservation

system. It explains **how reservation is being**

**done** in Indian Railways. The step by step procedure is explained.

**Proper** **comments** have been given at desired location to make the code **user friendly**.

**Various functions and structures are used** to

make a complete use of this language, this

project is well versed with the programming. Railway reservation can easily accompany with

the help of this.

**OBJECTIVE**

Write a program using Python and MY-SQL connectivity for Railway Reservation.





**About python**

Python is a popular programming language. It

was created by **Guido van Rossum**, and released

in **1991**.

It is used for:

* **web development** (server-side),
* **software development**,
* **system scripting**.

### Why Python?

* Python has a **simple syntax** similar to the

English language.

* Python has syntax that allows developers to

write programs with **fewer lines** than some

other programming languages.

* Python runs on an **interpreter system**, meaning that code can be executed as soon as it is

written.

**File handling in python**

**File Handling** is the storing of data in a **file**

using a program. Python has a **built-in open() function** to open a file. This function returns a

**file object**, also called a handle, as it is used to **read or modify the file** accordingly. We can

specify the **mode** while opening a file. In mode,

we specify whether we want to **read ‘r’ , write ‘w’** or **append ‘a’** to the file.

Three types of files in python:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Description** | **Extention** |
| 1. | Text file | Only contains text and has no special formatting | .txt |
| 2. | Binary file | Store data in a binary format(machine language). | .dat |
| 3. | CSV file | Comma Seperated Values File  can be store in a tabular form. | .csv |

**About modules**

A module is a file containing **a set of functions**

you want to include in your application.

Some advantages of modules are:

1. Programming **errors** can easily be **detected.**
2. Allows **reuse** of code.
3. Improves **manageability.**
4. Collaboration in projects made **efficient** and easy.

Following modules are used in the project:

1. pickle: Fuctions for binary file to pickle and unpickle like load( ) and dump( )
2. time: Time related functions like sleep( )
3. random: Functions to generate random numbers like randint( ),random( )and randrange( )
4. os: Functions to interact with OS like system( )
5. tickets:User defined module
6. train: User defined module

DATABASE USED

railways

**Functions used**

1. It is a named unit
2. Set of excutable statements
3. It can be invoked from the other parts of program.

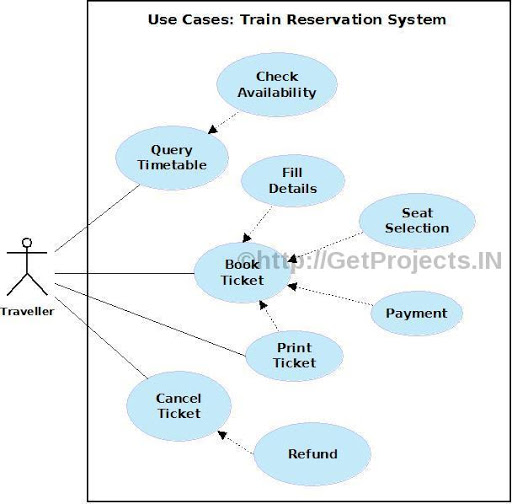
Types of Functions

|  |  |  |
| --- | --- | --- |
| 1. | Module Based | From random module, functions like **randint ( )** and **randint( )** are used and from pickle module, functions like **load( )** and **dump( )** are used. |
| 2. | Built-in | Functions like **int( ),input( ),float( ),print( ) , seek( )** and **tell( )** etc. |
| 3. | User defined | In tickets module,functions like **display( ), pending( )** and **confirmation( )** etc and in train module, functions like **getinput( ), startingpt( )** and **getdestination( )** etc. |

**Railway**

**reservation system**

The Railway Reservation System facilitates the passengers to enquire about the trains available

 on the basis of source and destination, booking and cancellation of tickets, enquire about the status of the booked ticket, etc.

The main objective is :

* To enquiry about **availability of trains**
* To **reserve and cancel their seats**
* To **modify** the information related to

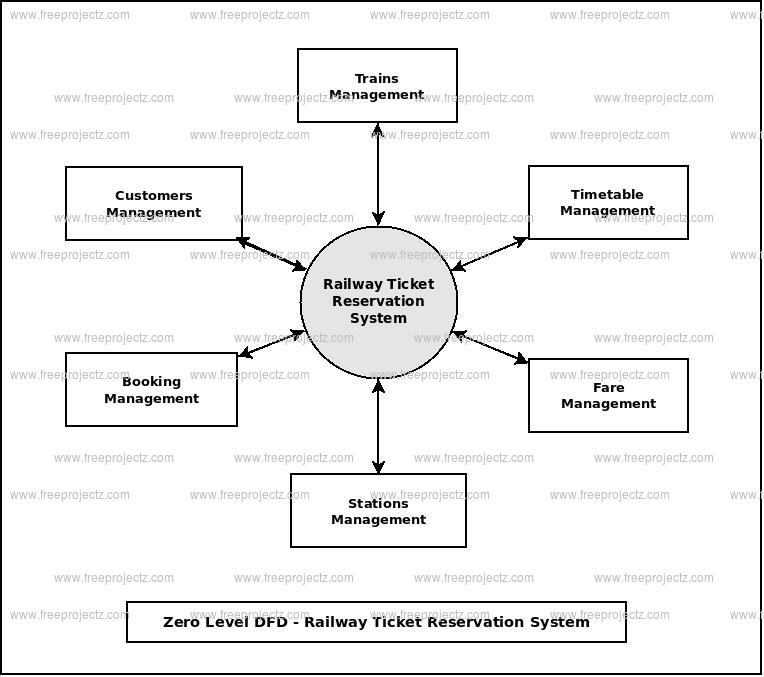
a)Trains

1.Time Table

2.Train Name

3.Train Number

b)Ticket Fare



**ADVANTAGES**

1. Convenient – You can book or cancel your tickets sitting in the comfort of your home or office.
2. Saves Time and Effort - You can save

the time needed to travel to the railway reservation office and waiting in the queue for your turn

1. Towards a Greener Planet – Instead of printing your ticket you can also choose to

travel with the SMS or soft copy of your

booked ticket in your laptop or even on your mobile.

1. Minimize Manual Data Entry –

Integrity of data is maintained and redundancy reduces.

**SOURCE**

**CODE**

**CODING**

**MODULES USED**

tickets.py

def \_\_init\_\_(): #"function definition"

no\_ofac1stclass=0

total=0

no\_ofac2ndclass=0

no\_ofac3rdclass=0

no\_ofsleeper=0

no\_oftickets=0

name=""

age=""

resno=0

status="" #"function body"

def display(): #"to display PNR status"

import train as tr

import time

import random

import os

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",passwd="dell",database="railways")

cursor=mycon.cursor()

cursor.execute("SELECT \* FROM tickets")

data=cursor.fetchone()

name,age,PNRno,type\_ofclass,status,booked\_seats=data[0],int(data[1]),int(data[2]),data[3],data[4],int(data[5])

f=0

n=int(input("ENTER PNR NUMBER :"))

print("\n\n")

print("FETCHING DATA..".center(80))

time.sleep(1) #"Time module used"

print('PLEASE WAIT...!!'.center(80))

time.sleep(1)

os.system('cls') #"os module used"

if(n==PNRno):

f=1

print("="\*80)

print("PNR STATUS".center(80))

print ("="\*80)

print("PASSENGER'S NAME:",name)

print("PASSENGER'S AGE:",age)

print("PNR NO:",PNRno)

print("TYPE\_OFCLASS:",type\_ofclass)

print("STATUS:",status)

print("NO OF SEATS BOOKED:",booked\_seats)

mycon.close()

input("PRESS ENTER TO GO TO BACK MENU".center(80))

if (f==0):

print("WRONG PNR NUMBER...!")#"if PNRno doesn't matches "

input("PRESS ENTER TO GO TO BACK MENU".center(80))

def pending(resno,no\_oftickets,x):#"if ticket in waiting list"

import time

status="WAITING LIST"

print ("PNR NUMBER :",resno)

time.sleep(1.2)

print ("STATUS=",abs(x),"seat(s) on",status)

print ("NO OF SEATS BOOKED : ",no\_oftickets)

input("PRESS ENTER TO GO TO BACK MENU".center(80))

def confirmation(resno): #"if ticket comfirmed"

import time

status="CONFIRMED"

print ("PNR NUMBER",resno)

time.sleep(1.2)

print("STATUS=",status)

input("PRESS ENTER TO GO TO BACK MENU".center(80))

return status

def cancellation():

import time

import os

import pickle

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",passwd="dell",database="railways")

cursor=mycon.cursor()

cursor.execute("SELECT \* FROM tickets")

data=cursor.fetchone()

name,age,PNRno,type\_ofclass,status,booked\_seats=data[0],int(data[1]),int(data[2]),data[3],data[4],int(data

[5])

r=int(input("ENTER PNR NUMBER :"))

if(r==PNRno):

cursor.execute("DROP table IF EXISTS tickets;")

mycon.commit()

mycon.close()

print("TICKET CANCELLED")

else:

print("NO SUCH RESERVATION NUMBER FOUND")

time.sleep(2)

os.system("cls")

def reservation():

import mysql.connector

import train as tr

import time

import random

import os

train\_no=int(input("ENTER THE TRAIN NO:"))

f=0

mycon=mysql.connector.connect(host="localhost",user="root",passwd="dell",database="railways")

cursor=mycon.cursor()

cursor.execute("SELECT \* FROM trdetails")

data=cursor.fetchone()#"train data imported"

n=data[0]

no\_of1stACclass=int(data[2])

no\_of2ndACclass=int(data[3])

no\_of3rdACclass=int(data[4])

no\_ofsleeperclass=int(data[5])

if(train\_no==n):

print("TRAIN NAME IS :",data[1])

f=1

print("-"\*80)

if(f==1):

cursor.execute("create table tickets (name varchar(50),age integer,PNRno integer,type\_ofclass varchar(50),status varchar(30),booked\_seats integer);")

mycon.commit()

name=input("ENTER THE PASSENGER'S NAME: ")

age=int(input("PASSENGER'S AGE: "))

print("\t\t SELECT A CLASS YOU WOULD LIKE TO TRAVEL IN:- ")

print("1.AC FIRST CLASS")

print("2.AC SECOND CLASS")

print("3.AC THIRD CLASS")

print("4.SLEEPER CLASS")

c=int(input("\t\t\tENTER YOUR CHOICE="))

os.system("cls")

amt1=0

if(c==1): #"for ac 1st class"

no\_oftickets=int(input("ENTER NO\_OF FIRST CLASS AC SEATS TO BE BOOKED :"))

for i in range(1,no\_oftickets+1):

amt1=1000\*no\_oftickets

print("PROCESSING..",time.sleep(0.5))

print(".",time.sleep(0.3))

print(".")

time.sleep(2)

os.system("cls")

print ("TOTAL AMOUNT TO BE PAID=",amt1)

resno=random.randint(1000,2546)

x=no\_of1stACclass-no\_oftickets

if(x>0):

status=confirmation(resno)#"function calling"

query="INSERT INTO tickets(name,age,PNRno,type\_ofclass,status,booked\_seats) VALUES('{}',{},{},'{}','{}',{})".format(name,age,resno,"AC FIRST CLASS","COMFIRMED",no\_oftickets)

cursor.execute(query)

mycon.commit()

mycon.close()

else:

status=pending(resno,no\_oftickets,x)

query="INSERT INTO tickets(name,age,PNRno,type\_ofclass,status,booked\_seats) VALUES('{}',{},{},'{}','{}',{})".format(name,age,resno,"AC FIRST CLASS","WAITING LIST",no\_oftickets)

cursor.execute(query)

mycon.commit()

mycon.close()

elif(c==2):#"for ac 2nd class"

no\_oftickets=int(input("ENTER NO\_OF SECOND CLASS AC SEATS TO BE BOOKED : "))

for i in range(1,no\_oftickets+1):

amtl=900\*no\_oftickets

print("PROCESSING..",time.sleep(0.5))

print(".",time.sleep(0.3))

print (".",time.sleep(2))

os.system("cls")

print("TOTAL AMOUNT TO BE PAID= ",amtl)

resno=random.randint(1000,2546)

x=no\_of2ndACclass-no\_oftickets

if(x>0):

status=confirmation(resno)#"function calling"

query="INSERT INTO tickets (name,age,PNRno,type\_ofclass,status,booked\_seats) VALUES('{}',{},{},'{}','{}',{})".format(name,age,resno,"AC SECOND CLASS",status,no\_oftickets)

cursor.execute(query)

mycon.commit()

mycon.close()

else:

status=pending(resno,no\_oftickets,x)

query="INSERT INTO tickets (name,age,PNRno,type\_ofclass,status,booked\_seats) VALUES('{}',{},{},'{}','{}',{})".format(name,age,resno,"AC SECOND CLASS","WAITING LIST",no\_oftickets)

cursor.execute(query)

mycon.commit()

mycon.close()

elif(c==3):#"for ac 3rd class"

no\_oftickets=int(input("ENTER NO OF THIRD CLASS AC SEATS TO BE BOOKED : "))

for i in range(1,no\_oftickets+1):

amtl=800\*no\_oftickets

print("PROCESSING..",time.sleep(0.5))

print(".",time.sleep(0.3))

print (".")

time.sleep(2)

os.system("cls")

print("TOTAL AMOUNT TO BE PAID= ",amtl)

resno=random.randint(1000,2546)

x=no\_of3rdACclass-no\_oftickets

if(x>0):

status=confirmation(resno)#"function calling"

query="INSERT INTO tickets (name,age,PNRno,type\_ofclass,status,booked\_seats) VALUES('{}',{},{},'{}','{}',{})".format(name,age,resno,"THIRD CLASS AC",status,no\_oftickets)

cursor.execute(query)

mycon.commit()

mycon.close()

else:

status=pending(resno,no\_oftickets,x)

query="INSERT INTO tickets (name,age,PNRno,type\_ofclass,status,booked\_seats) VALUES('{}',{},{},'{}','{}',{})".format(name,age,resno,"THIRD CLASS AC","WAITING LIST",no\_oftickets)

cursor.execute(query)

mycon.commit()

mycon.close()

elif(c==4):#"for sleeper class"

no\_oftickets=int(input("ENTER NO\_OF SLEEPER CLASS SEATS TO BE BOOKED : "))

for i in range(1,no\_oftickets+1):

amtl=550\*no\_oftickets

print("PROCESSING..",time.sleep(0.5))

print(".",time.sleep(0.3))

print (".")

time.sleep(2)

os.system("cls")

print("TOTAL AMOUNT TO BE PAID= ",amtl)

resno=random.randint(1000,2546)

x=no\_ofsleeperclass-no\_oftickets

if(x>0):

status=confirmation(resno)#"function calling"

query="INSERT INTO tickets (name,age,PNRno,type\_ofclass,status,booked\_seats) VALUES('{}',{},{},'{}','{}',{})".format(name,age,resno,"SLEEPER CLASS",status,no\_oftickets)

cursor.execute(query)

mycon.commit()

mycon.close()

else:

status=pending(resno,no\_oftickets,x)

query="INSERT INTO tickets (name,age,PNRno,type\_ofclass,status,booked\_seats) VALUES('{}',{},{},'{}','{}',{})".format(name,age,resno,"SLEEPER CLASS","WAITING LIST",no\_oftickets)

cursor.execute(query)

mycon.commit()

mycon.close()

else:

print("Please Enter valid choice number") #"for no. greater than 4"

if(f==0):#"for wrong train no."

time.sleep(2)

print("\n\n\n\n\n\n\t\t\t\t NO SUCH TRAINS FOUND !!")

time.sleep(2)

train.py

def \_\_init\_\_():#"Declaration of module through \_init\_"

trainno=0

no\_ofaclstclass=0

no\_ofac2ndclass=0

no\_ofac3rdclass=0

no\_ofsleeper=0

totalseats=0

trainname=""

startingpt=""

destination=""

def getinput():

import os

import mysql.connector#"importing module required for MySQL and Python connection"

mycon=mysql.connector.connect(host="localhost",user="root",passwd="dell",database="railways")#"using database railways"

cursor=mycon.cursor()

cursor.execute("create table trdetails (trainno integer,trainname varchar(50), no\_ofac1stclass integer, no\_ofac2ndclass integer,no\_ofac3rdclass integer,no\_ofsleeper integer,startingpt varchar(50),destination varchar(50));")

#"creating table in MySQL through Python"

print("="\*80)

print("\t\t\t ENTER THE TRAIN DETAILS")

print("="\*80)

trainname=input("ENTER THE TRAIN NAME : ").upper()

trainno=int(input("ENTER THE TRAIN NUMBER: "))

no\_ofac1stclass=int(input("ENTER NO\_OF AC FIRST CLASS SEATS TO BE RESERVED :"))

no\_ofac2ndclass=int(input("ENTER NO OF AC SECOND CLASS SEATS TO BE RESERVED:"))

no\_ofac3rdclass=int(input("ENTER NO OF AC THIRD CLASS SEATS TO BE RESERVED : "))

no\_ofsleeper=int(input("ENTER NO OF SLEEPER CLASS SEATS TO BE RESERVED:"))

startingpt=input("ENTER THE STARTING POINT :")

destination=input("ENTER THE DESTINATION POINT:")

#"input train details from user"

query="INSERT INTO trdetails(trainno,trainname, no\_ofac1stclass, no\_ofac2ndclass,no\_ofac3rd

class,no\_ofsleeper,startingpt,destination) VALUES({},'{}',{},{},{},{},'{}','{}')".format(trainno,trainname,

no\_ofac1stclass, no\_ofac2ndclass,no\_ofac3rdclass,no\_ofsleeper,startingpt,destination)

#"inserting data in table in MySQL through Python"

cursor.execute(query)

mycon.commit()

os.system('cls')

mycon.close()

def output():

import os

import mysql.connector

mycon=mysql.connector.connect(host="localhost",user="root",passwd="dell",database="railways")

cursor=mycon.cursor()

cursor.execute("SELECT \* FROM trdetails")

data=cursor.fetchone()#"for fetching 1 data from database"

print("\*"\*80)

print("\t\t\t\tTRAIN DETAILS")

print("\*"\*80)

input("PRESS ENTER TO VIEW NEXT TRAIN DETAILS")

os.system('cls')

print("="\*80)

print("THE TRAIN NUMBER IS : ",data[0])

print("THE ENTERED TRAIN NAME IS :", data[1])

print("NO\_OF AC FIRST CLASS SEATS RESERVED ARE ",data[2])

print("NO\_OF AC SECOND CLASS SEATS RESERVED ARE:",data[3])

print("NO\_OF AC THIRD CLASS SEATS RESERVED ARE:",data[4])

print("NO\_OF SLEEPER CLASS SEATS RESERVED ARE :",data[5])

print("STARTING POINT ENTERED IS:",data[6])

print("DESTINATION POINT ENTERED IS: ",data[7])

print("="\*80)

mycon.close()#"linking through MySQL closed"

input("PRESS ENTER TO GO TO MAIN MENU")

MAIN CODE USED

def menu(): #"Function declaration"

print ("WELCOME TO INDIAN RAILWAY MANAGEMENT SYSTEM".center(80))#"Heading"

while True:

print("="\*80)

print("\t\t\t\t RAILWAY")#"Display menu for user"

print("="\*80)

print("\t\t\t1. UPDATE TRAIN DETAILS.")

print("\t\t\t2. TRAIN DETAILS. ")

print("\t\t\t3. RESERVATION OF TICKETS.")

print("\t\t\t4. CANCELLATION OF TICKETS.")

print("\t\t\t5. DISPLAY PNR STATUS.")

print("\t\t\t6. QUIT.")

print("\*\* -office use.")

ch=int(input("\t\t\t ENTER YOUR CHOICE : "))

os.system('cls')

print("\n\n\n\n\n\n\r\t\t\t\t LOADING..")

os.system('cls')

if ch==1: #"UPDATE TRAIN DETAILS choice"

j="password"

r=input("\n\n\n\n\n\n\n\n\n\n\n\t\t\t\tENTER THE PASSWORD: ")

print("\n\n\n\n\n\n\r\t\t\t\t ")

os.system('cls')

if(j==r):

x='y'

while (x.lower()=='y'):

tr.getinput() #"Function calling from module train"

print("\n\n\n\n\n\n\n\n\n\n\n\t\t UPDATING TRAIN LIST PLEASE WAIT..",time.sleep(1))

print (("."),time.sleep(0.5))

print(("."), time.sleep(2))

os.system('cls')

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

x=input("\t\t DO YOU WANT TO ADD ANY MORE TRAINS DETAILS ?")

os.system('cls')

continue

elif(j!=r):

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

print("WRONG PASSWORD".center(80))

elif ch==2:#"TRAIN DETAILS choice"

tr.output()

elif ch==3:#"RESERVATION OF TICKETS choice"

print("="\*80)

print("\t\t\tRESERVATION OF TICKETS")

print('='\*80)

tick.reservation()

elif ch==4:#"CANCELLATION OF TICKETS choice"

print("="\*80)

print("\t\t\t\tCANCELLATION OF TICKETS")

print("="\*80)

tick.cancellation()

elif ch==5:#"DISPLAY PNR STATUS choice"

print ("="\*80)

print("PNR STATUS".center(80))

print("="\*80)

tick.display()

elif ch==6:#"QUIT choice"

os.system('cls')

print(" THANK YOU.....".center(80))

print("\n\t\t\t\tDONE BY:-")

print("\t\t\t\t SWATAMBRA SAHU")

print("\t\t\t\t XII-A")

print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\tLOADING",time

.sleep(1))

print (".")

time.sleep(0.5)

print(".")

time.sleep(2)

os.system('cls')

quit()

else:#"for no. greater than 6"

print("Please Enter valid choice number")

#MAIN CODE USED

#\_\_main\_\_

#"Importing module to be used in program"

import time

import random

import os

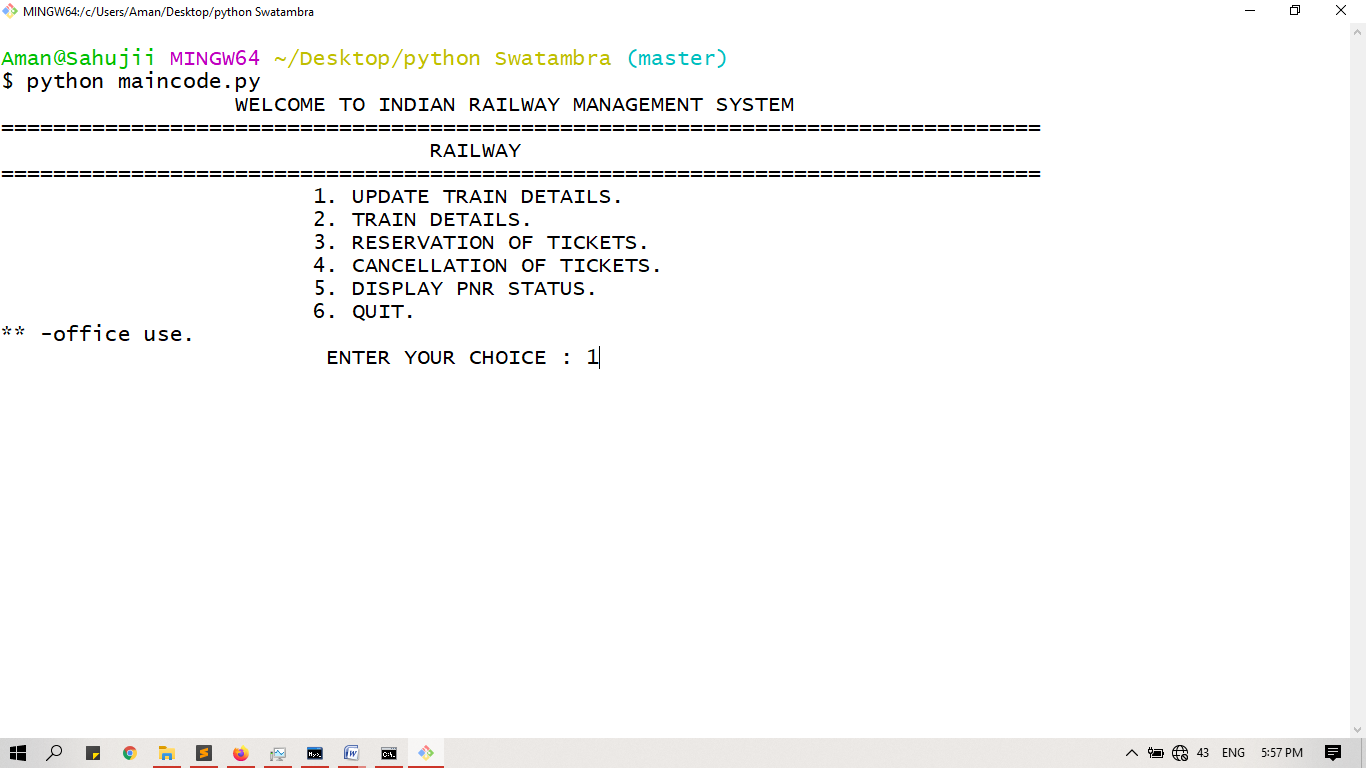
import tickets as tick

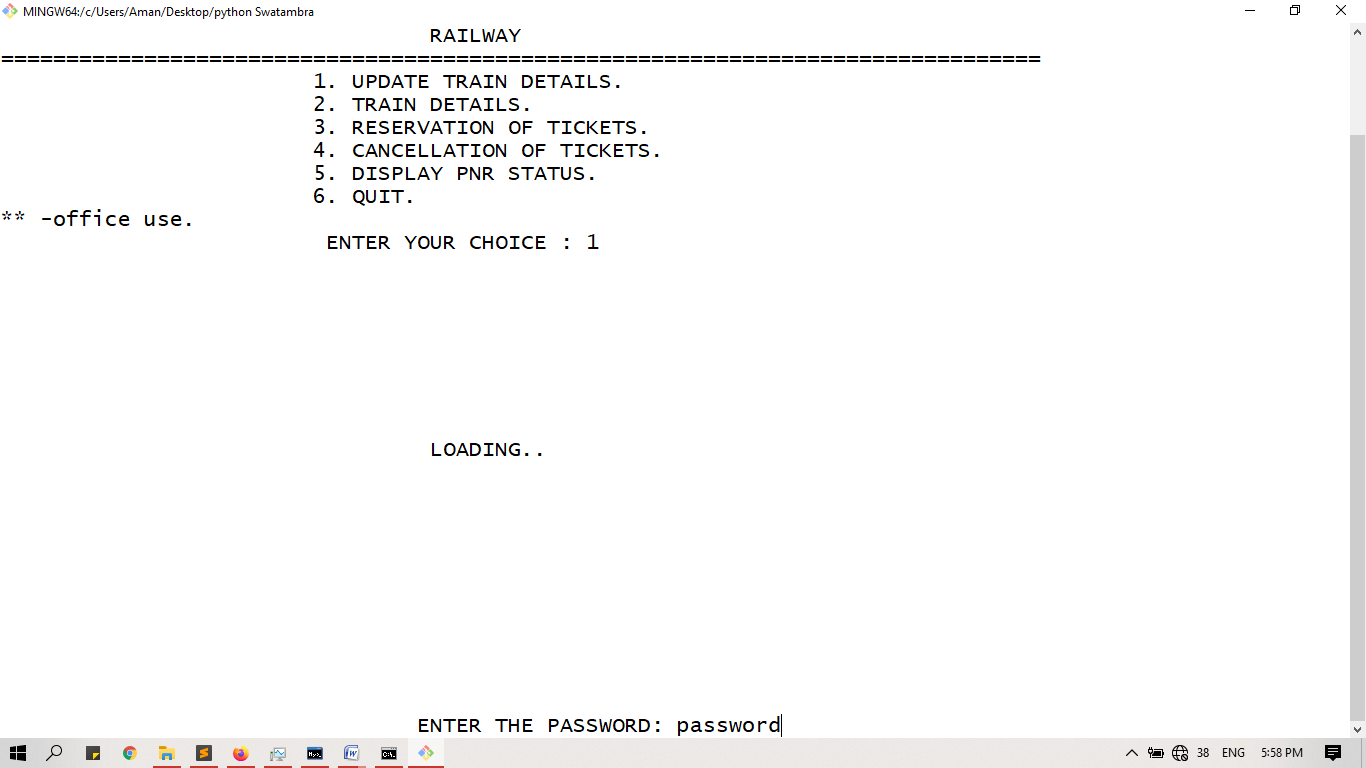
import train as tr

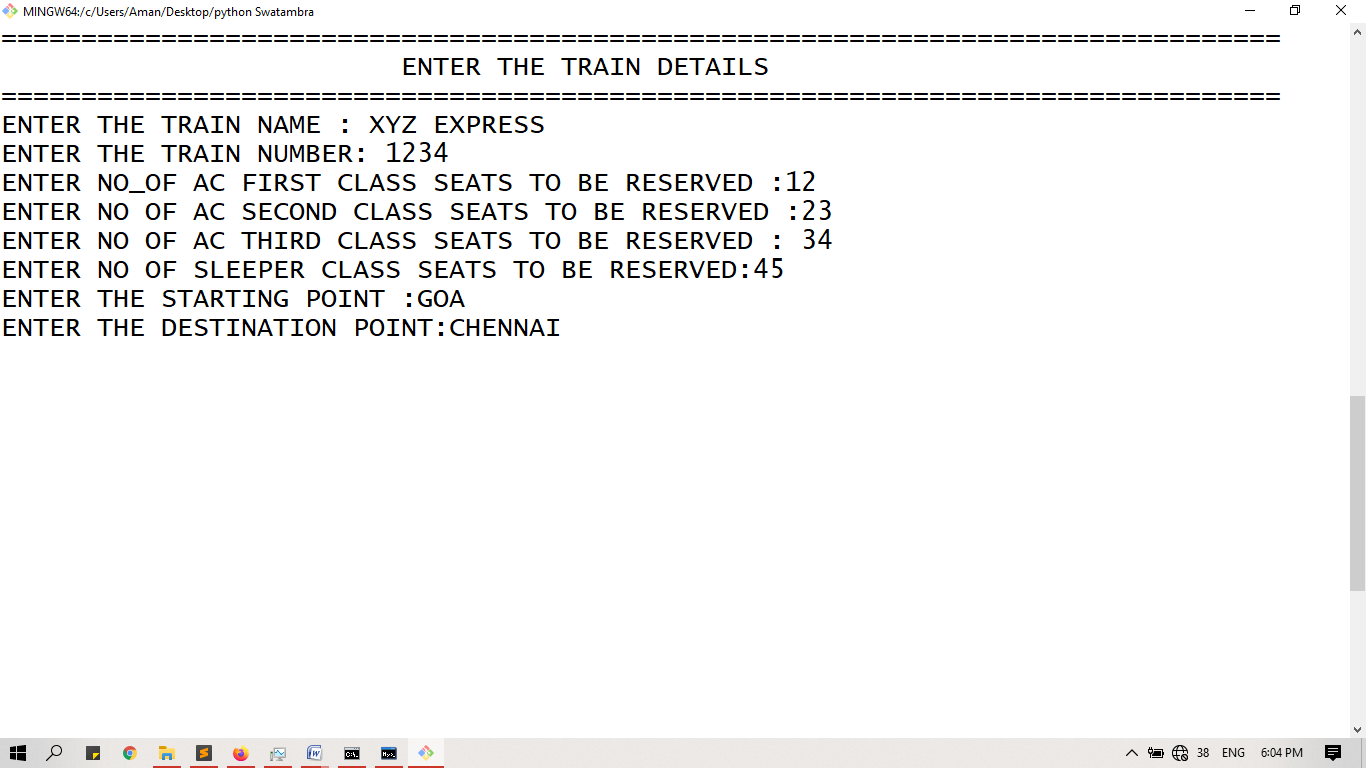
menu()#"Function calling"

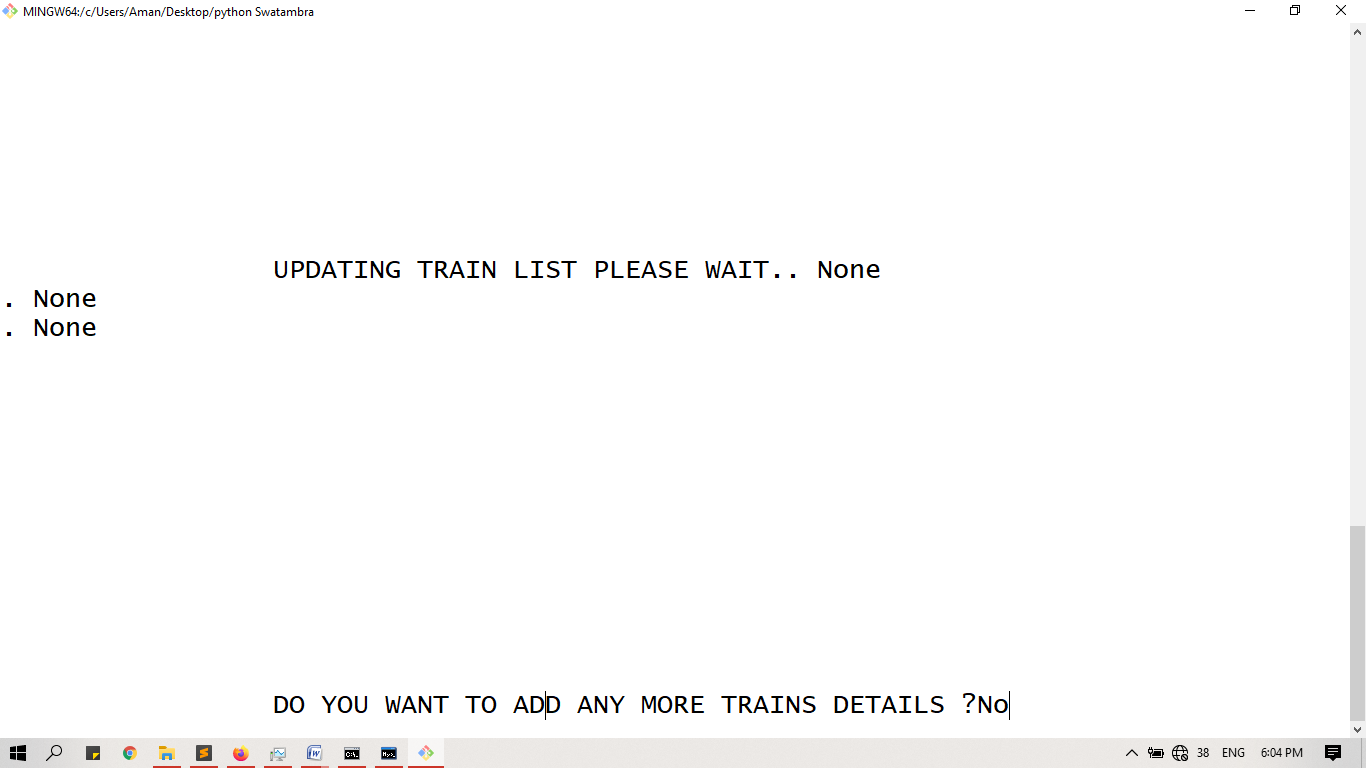
**Output**

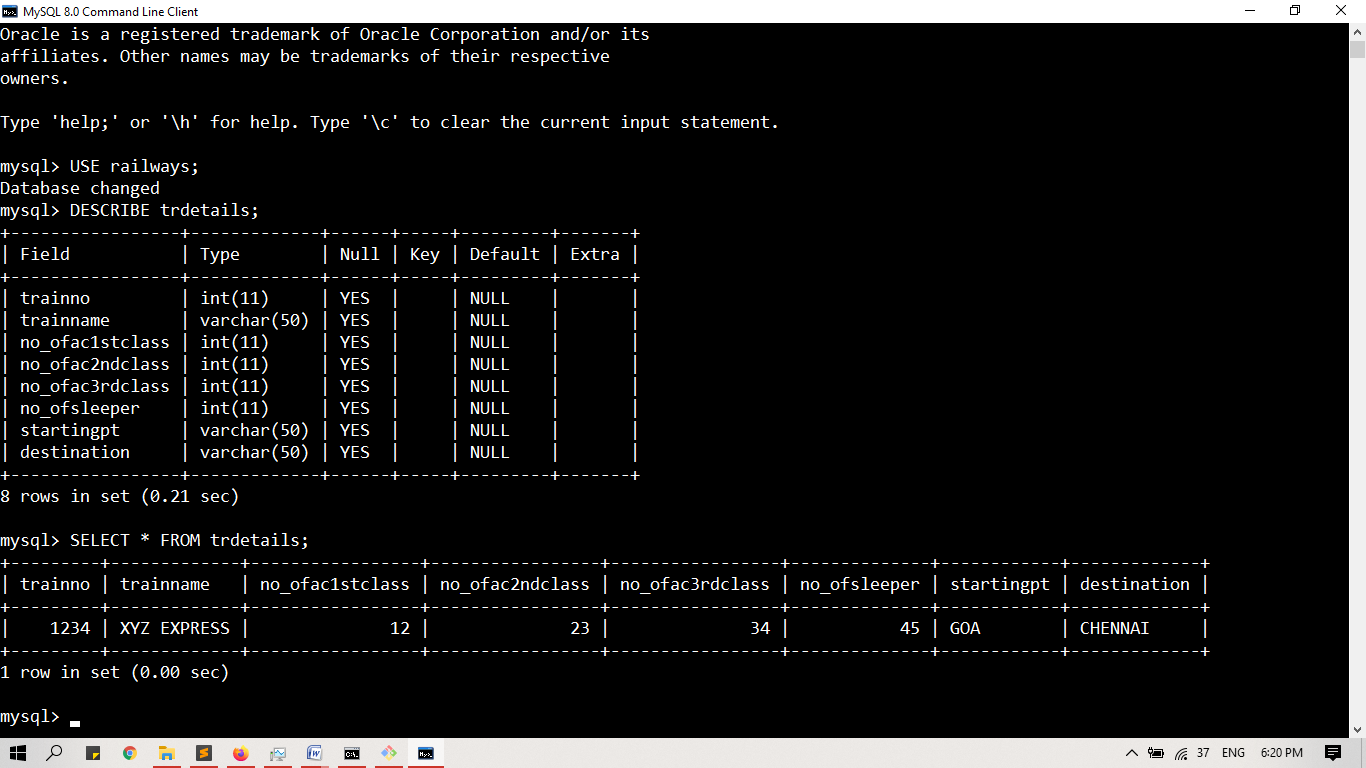
**screens**

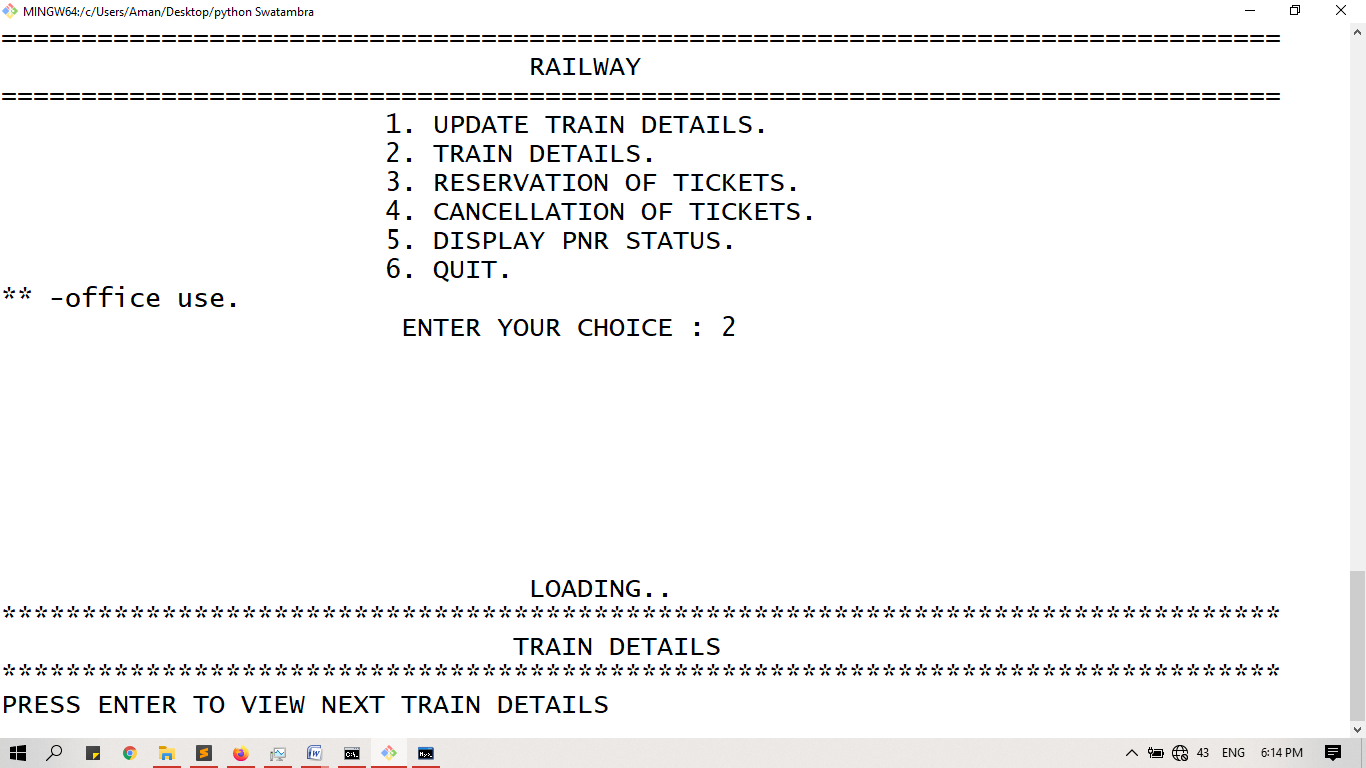


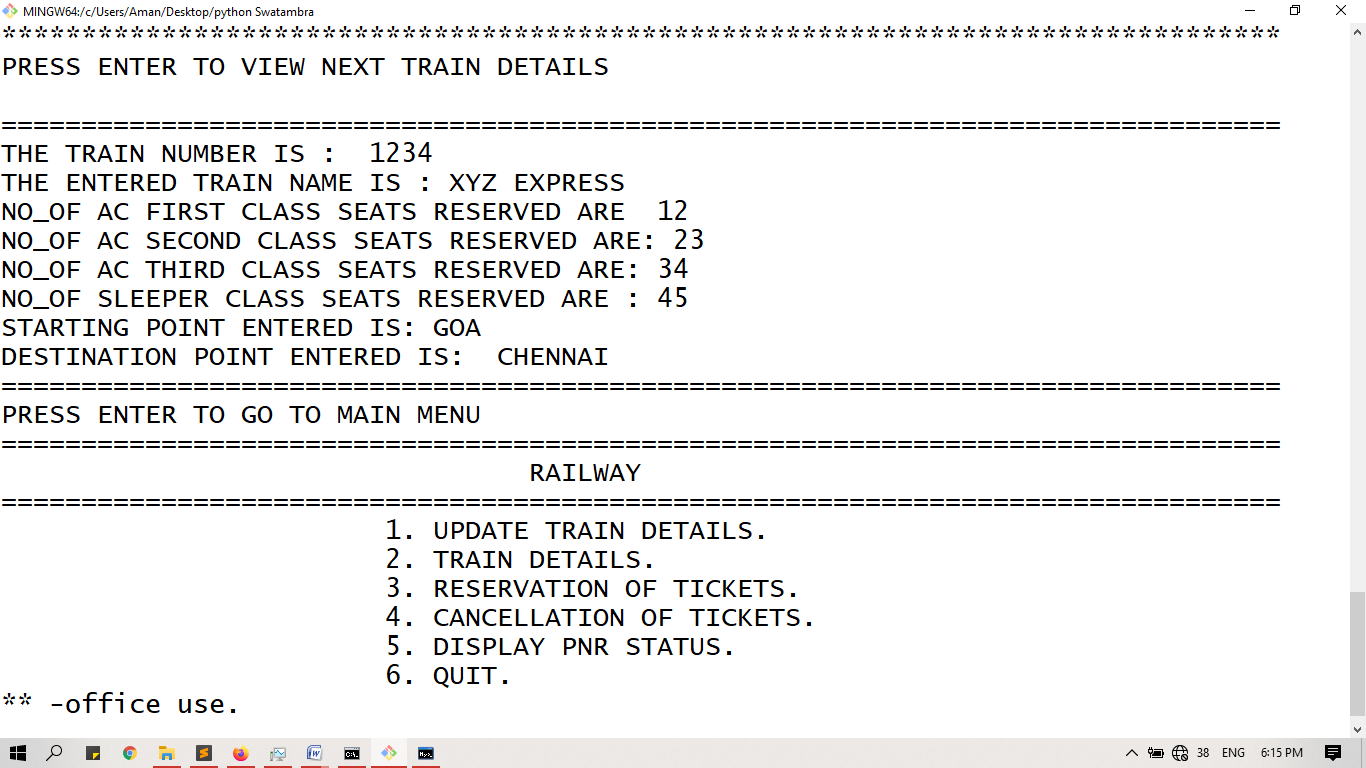


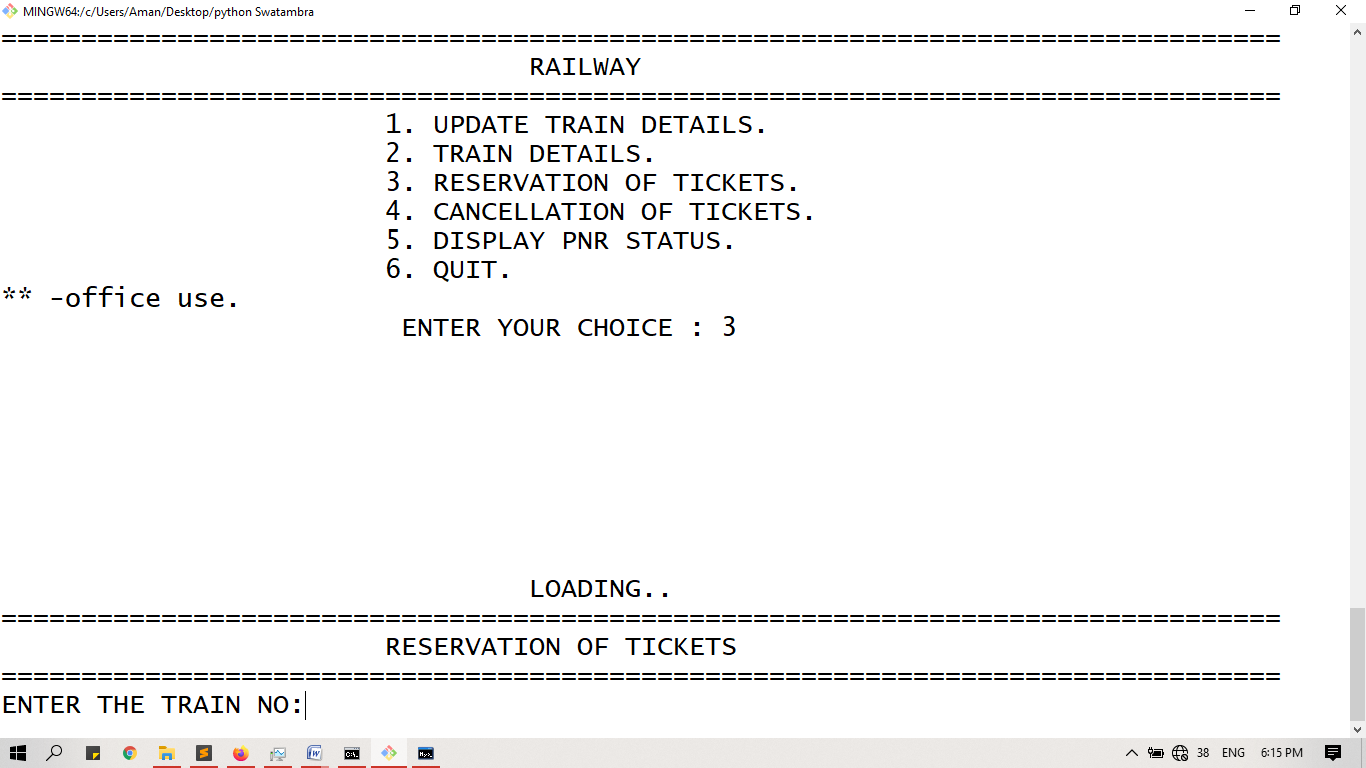


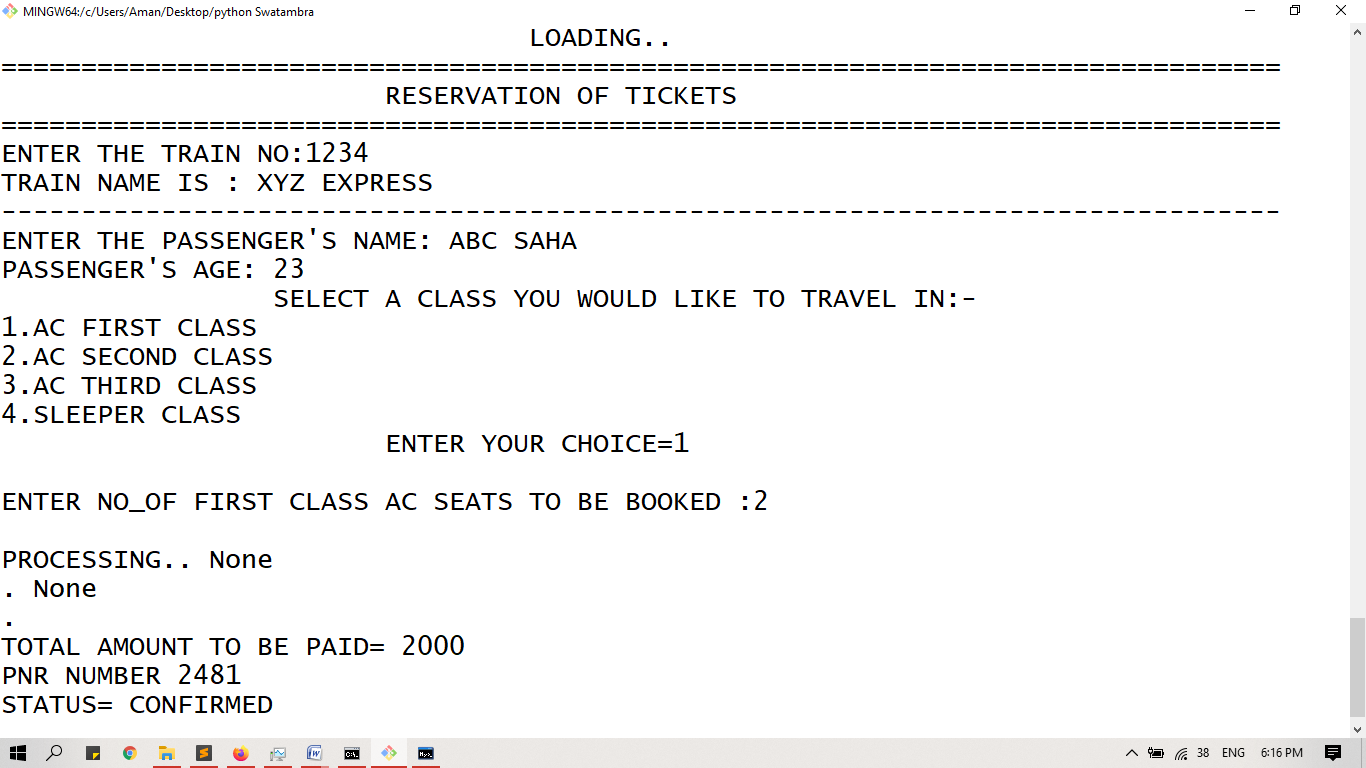


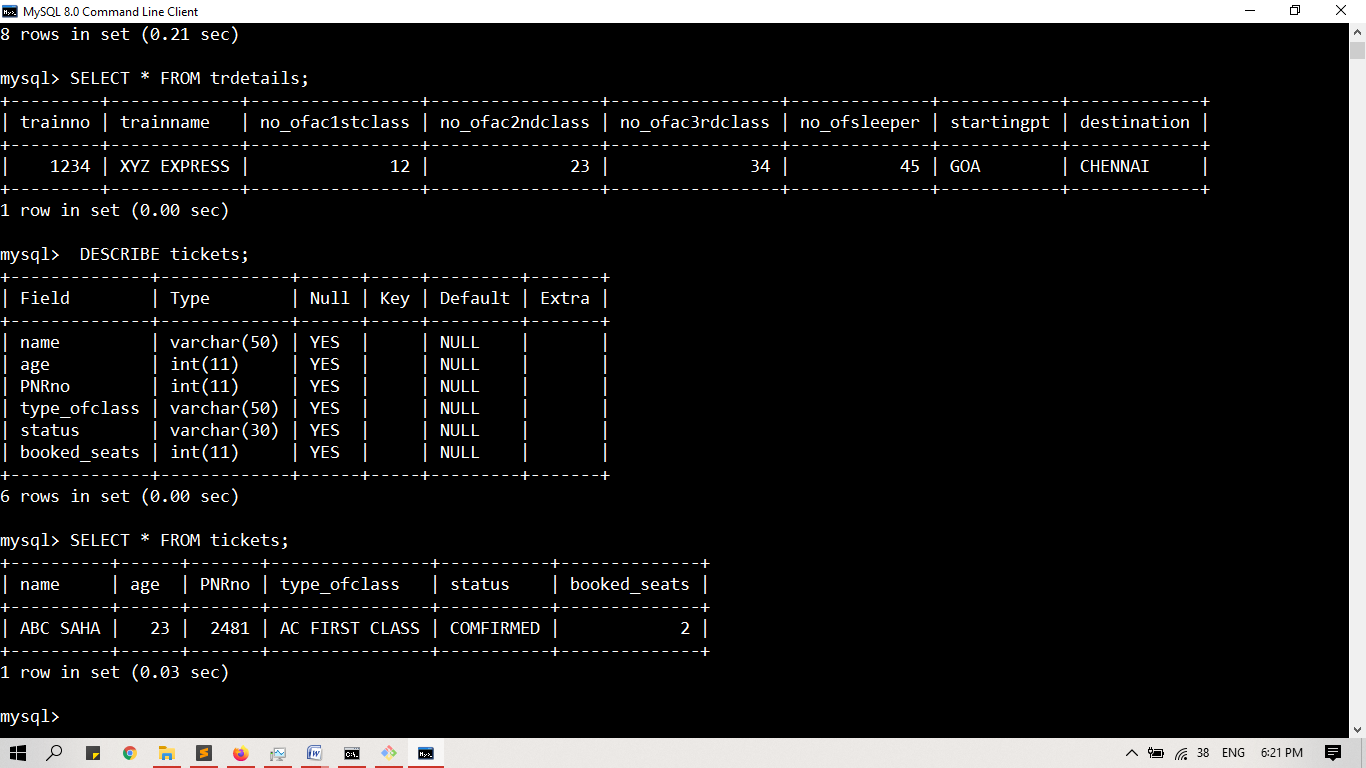


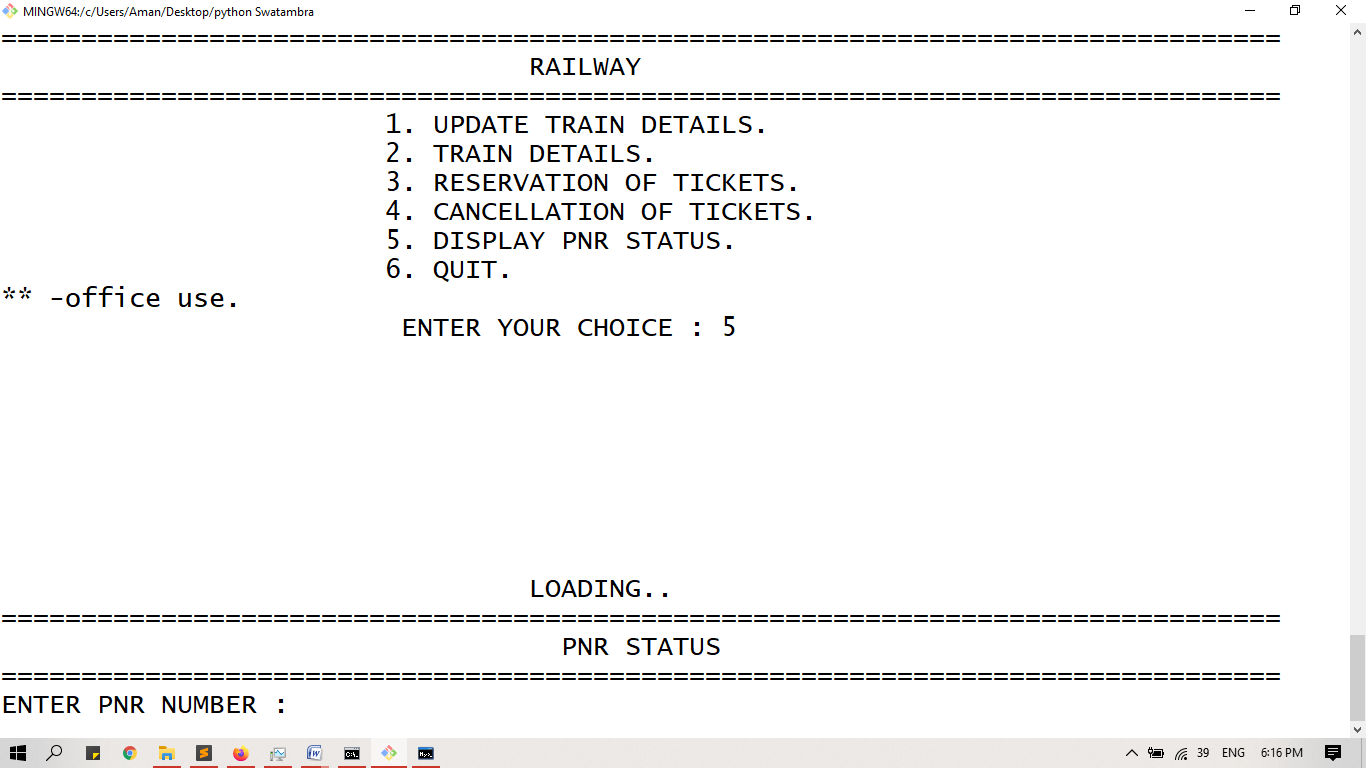


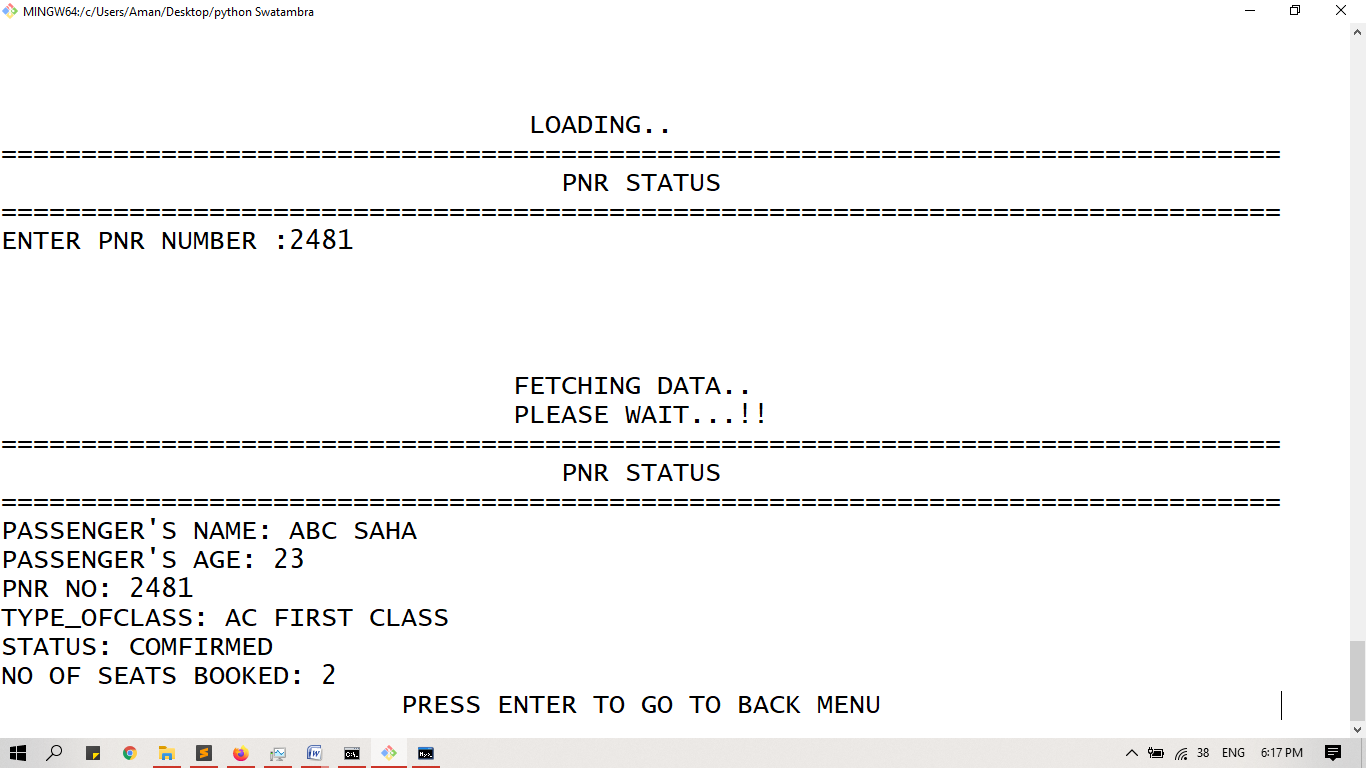


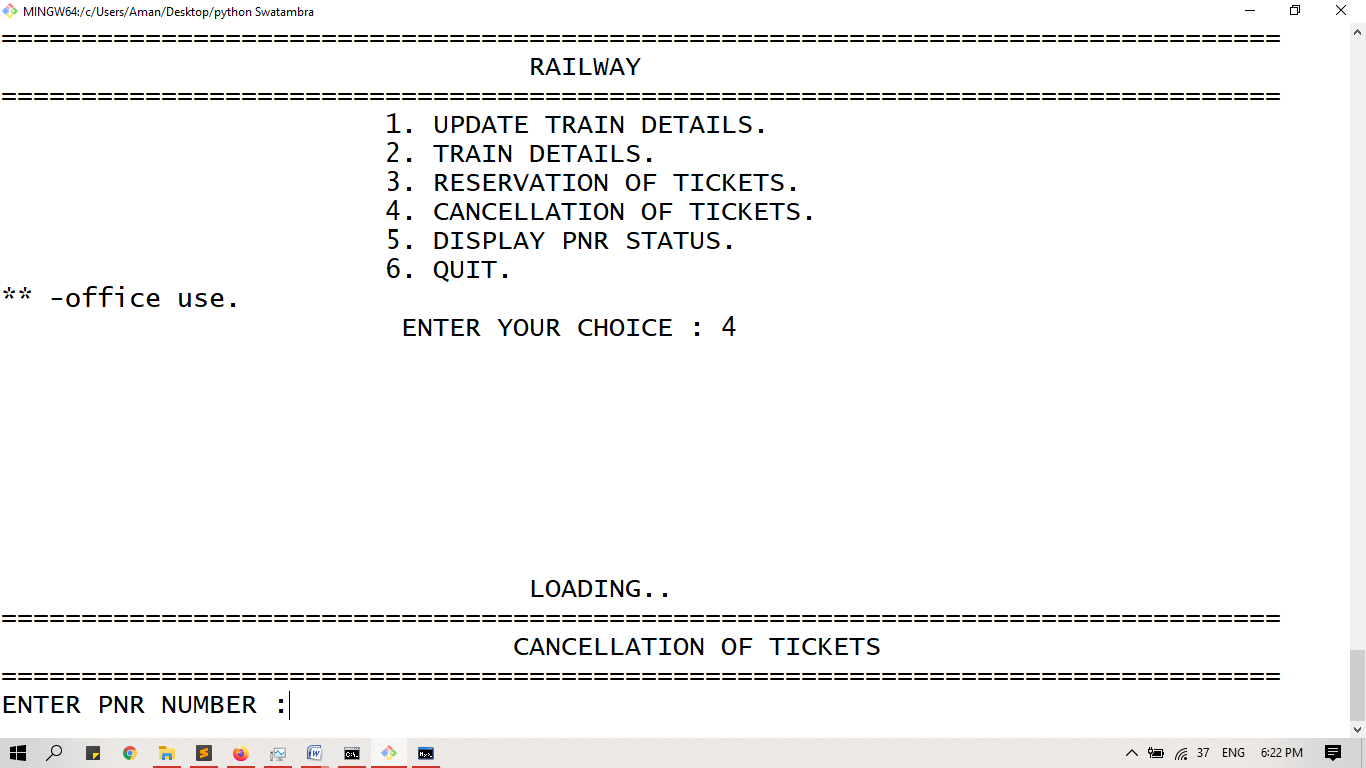


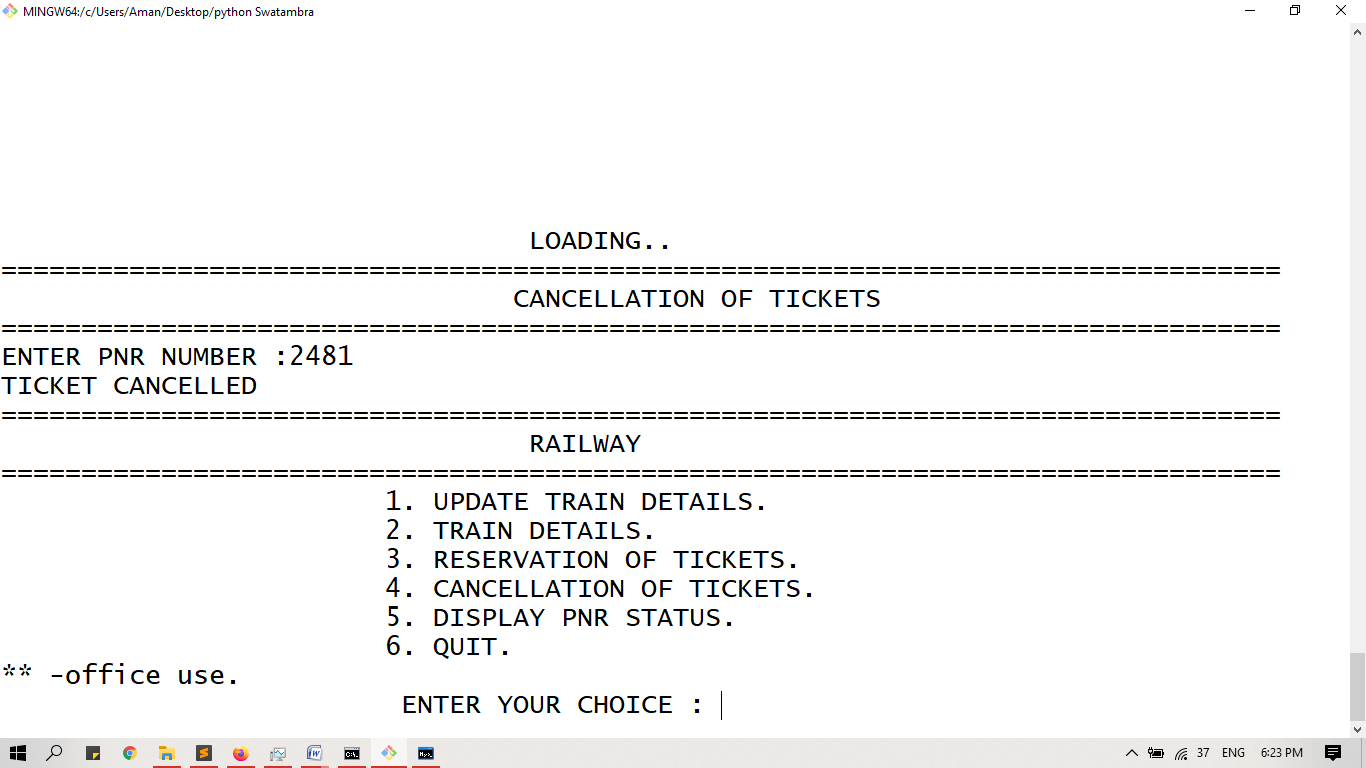


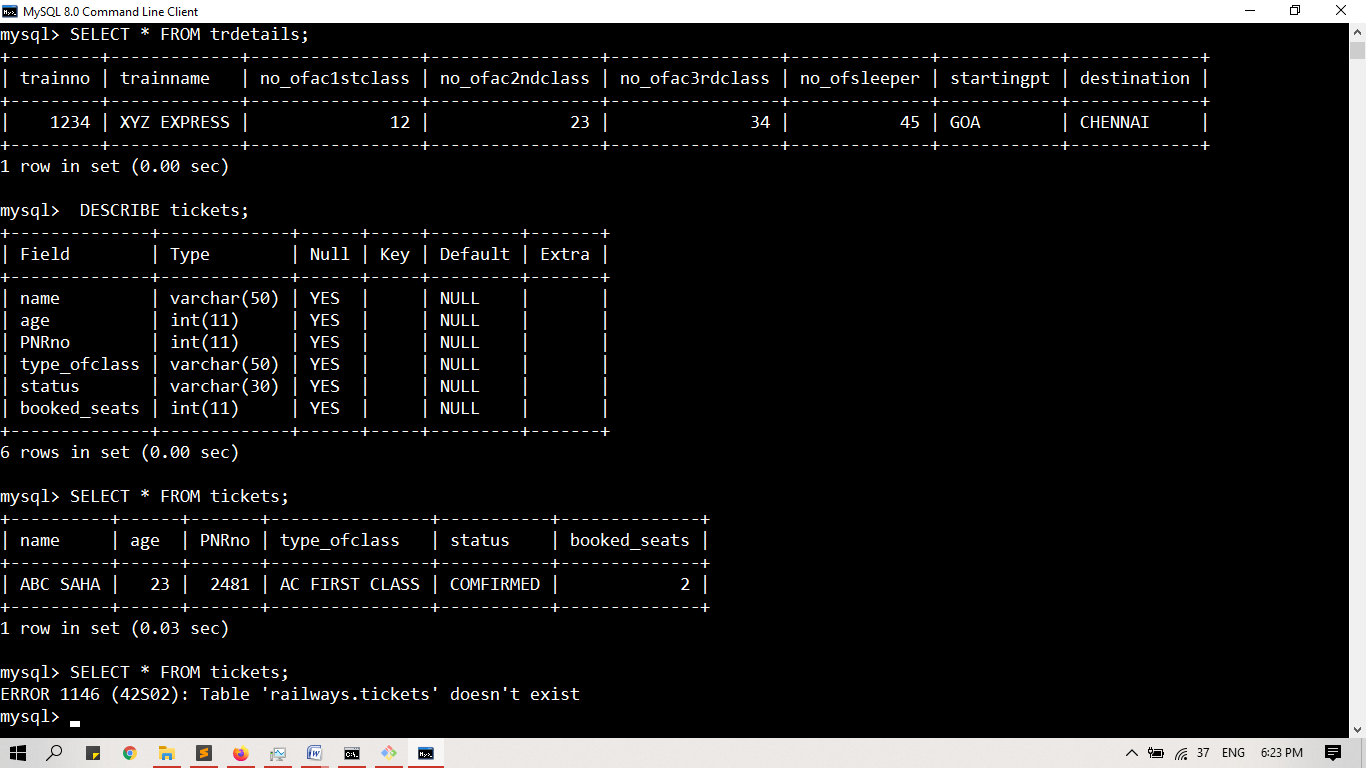


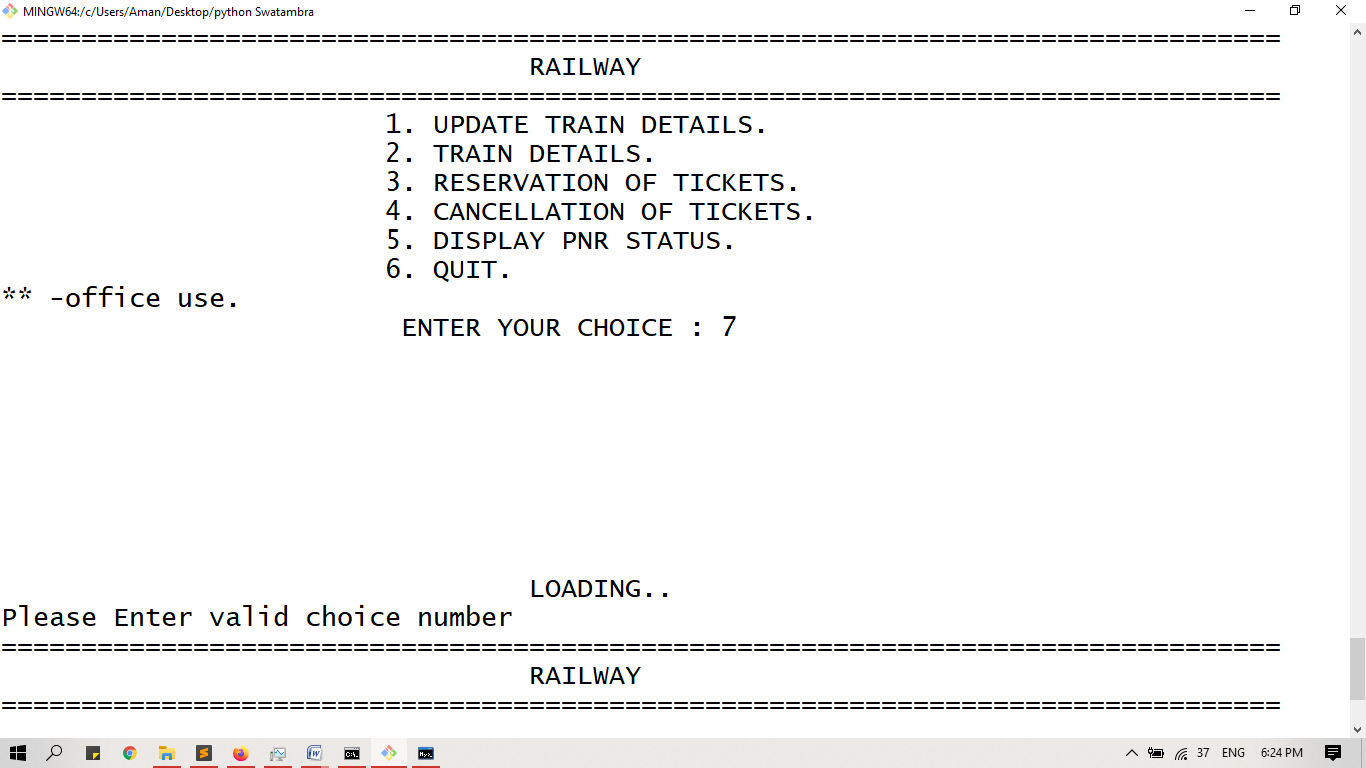


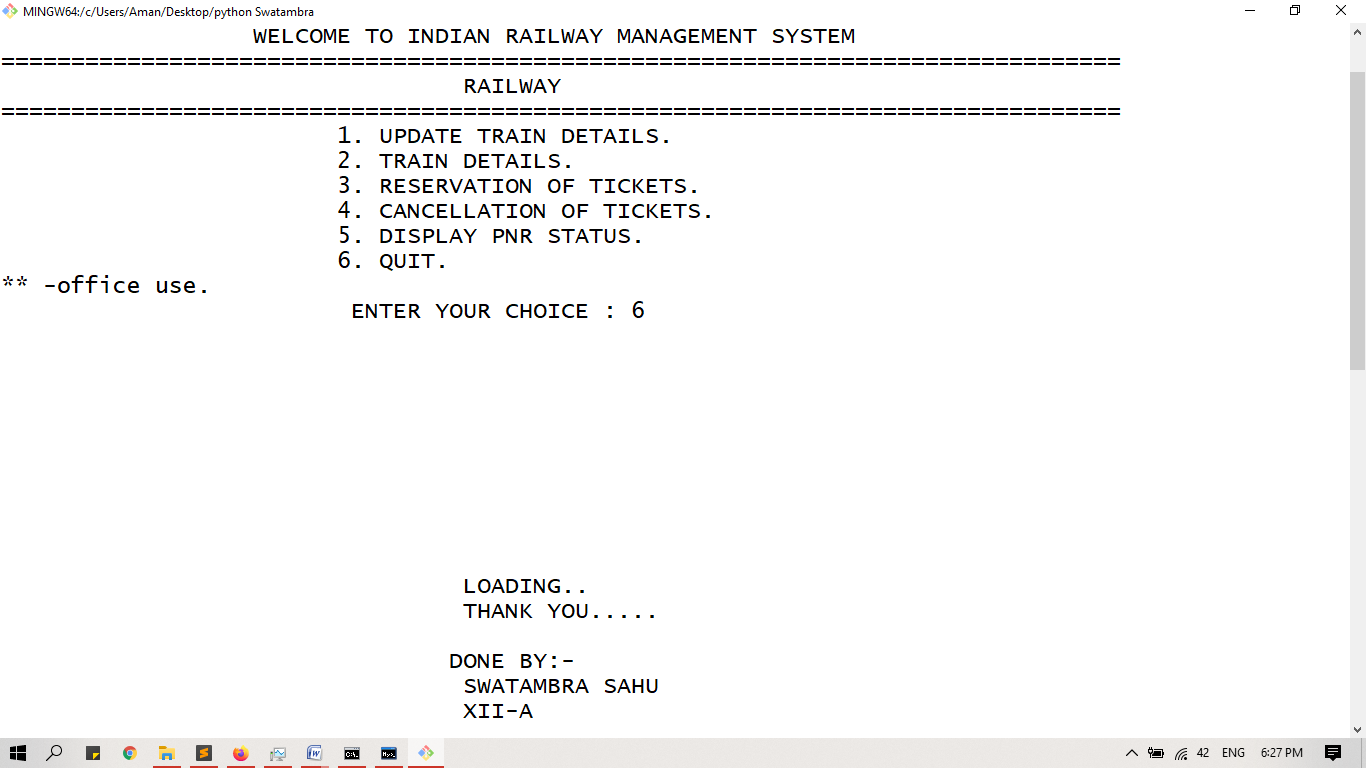


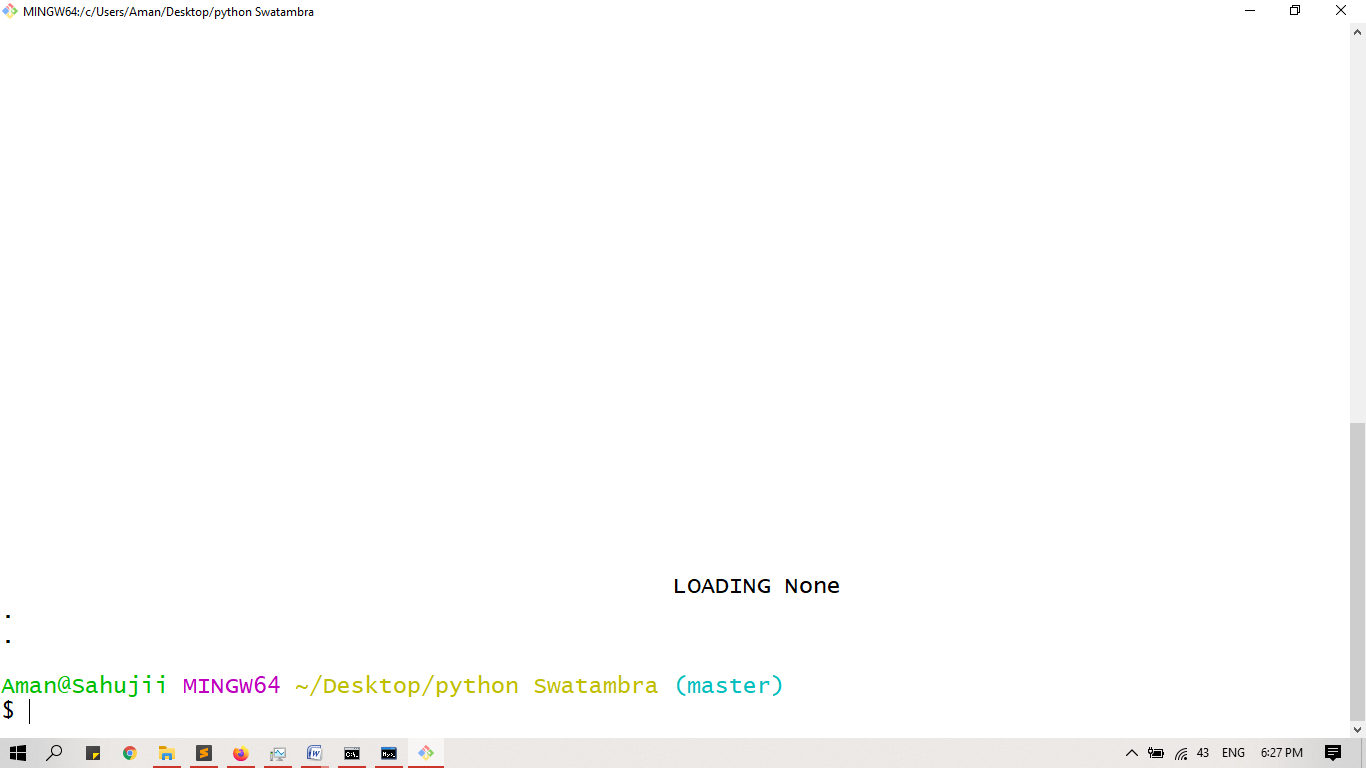












**Bibliography**

* Computer Science with Python by Sumita Arora Class 11
* Computer Science with Python by Sumita Arora Class 12
* [www.google.co.in](http://www.google.co.in)
* <https://www.scribd.com/doc/440085272>
* [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
* <https://www.w3schools.com/python>