Sprint 4

Date	11-Nov-22
Team ID	PNT2022TMID18159
Project Name	Smart solutions for railways

Ticket Cancellation:

```
from pickle import load,dump
import time
import random
import os
class tickets:
    def __init__(book):
        book.no_ofac1stclass=0
        book.totaf=0
        book.no_ofac2ndclass=0
        book.no_ofac3rdclass=0
        book.no_ofsleeper=0
        book.no_oftickets=0
        book.name=''
        book.age=''
        book.resno=0
        book.status=''
    def ret(book):
        return(book.resno)
    def retname(book):
        return(book.name)
    def display(book):
        f=0
        fin1=open("tickets.dat","rb")
        if not fin1:
            print "ERROR"
        else:
            print
            n=int(raw_input("Enter PNR number : "))
            print "\n\n"
            print ("Fetching Data . . .".center(80))
            time.sleep(1)
            print
            print('PLEASE WAIT...!!'.center(80))
            time.sleep(1)
            os.system('cls')
            try:
                while True:
                    tick=load(fin1)
                    if(n==tick.ret()):
                        f=1
```

```
                print "="*80
                print("PNR Status".center(80))
                print"="*80
                print
                print "Passenger Name :",tick.name
                print
                print "Passenger Age :",tick.age
                print
                print "PNR No :",tick.resno
                print
                print "Status :",tick.status
                print
                print "Ticket Count : ",tick.no_oftickets
                print
        except:
            pass
        fin1.close()
        if(f==0):
            print
            print "Wrong PNR Number !!"
            print
def pending(self):
    book.status="Waiting List"
    print "PNR Number :",book.resno
    print
    time.sleep(1.2)
    print "Status = ",book.status
    print
    print "Ticket Count: ",book.no_oftickets
    print
def confirmation (book):
    book.status="Confirmed"
    print "PNR Number : ",book.resno
    print
    time.sleep(1.5)
    print "Status = ",book.status
    print
def cancellation(book):
    z=0
    f=0
    fin=open("tickets.dat","rb")
    fout=open("temp.dat","ab")
    print
    r= int(raw_input("Enter PNR Number : "))
    try:
        while(True):
```

```
            tick=load(fin)
            z=tick.ret()
            if(z!=r):
                dump(tick,fout)
            elif(z==r):
                f=1
        except:
            pass
        fin.close()
        fout.close()
        os.remove("tickets.dat")
        os.rename("temp.dat","tickets.dat")
        if (f==0):
            print
            print "Invalid Reservation Number"
            print
            time.sleep(2)
            os.system('cls')
        else:
            print
            print "Ticket Cancelled"
            print "Refund Assured"
    def reservation(book):
        trainno=int(raw_input("Enter Train Number:"))
        z=0
        f=0
        fin2=open("tr1details.dat")
        fin2.seek(0)
        if not fin2:
            print "ERROR"
        else:
            try:
                while True:
                    tr=load(fin2)
                    z=tr.gettrainno()
                    n=tr.gettrainname()
                    if (trainno==z):
                        print
                        print "TRAIN NAME IS : ",n
                        f=1
                        print
                        print "-"*80
                        no_ofac1st=tr.getno_ofac1stclass()
                        no_ofac2nd=tr.getno_ofac2ndclass()
                        no_ofac3rd=tr.getno_ofac3rdclass()
                        no_ofsleeper=tr.getno_ofsleeper()
```

```
if(f==1):
    fout1=open("tickets.dat","ab")
    print
    book.name=raw_input("Enter Passenger Name:")
    print
    book.age=int(raw_input("Passenger Age : "))
    print
    print"\t\t Select a Class to Travl: "
    print "1.AC FIRST CLASS"
    print
    print "2.AC SECOND CLASS"
    print
    print "3.AC THIRD CLASS"
    print
    print "4. SLEEPER CLASS"
    print
    c=int(raw_input("\t\t\t Enter The Choice:"))
    os.system('cls')
    amt1=0
    if(c==1):
        book.no_oftickets=int(raw_input("Enter AC First Class Ticket Count:"))
        i=1
        while(i<=book.no_oftickets):
            book.totaf=book.totaf+1
            amt1=1000*book.no_oftickets
            i=i+1
        print
        print "Processing…...",
        time.sleep(0.5)
        print ".",
        time.sleep(0.3)
        print'.'
        time.sleep(2)
        os.system('cls')
        print "Total Amount: ",amt1
        book.resno=int(random.randint(1000,2546))
        x=no_ofac1st-book.totaf
        print
        if(x>0):
            book.confirmation()
            dump(book,fout1)
            break
        else:
            book.pending()
            dump(tick,fout1)
            break
```

```
                elif(c==2):
                    book.no_oftickets=int(raw_input("Enter AC Second Class Ticket Count: "))
                    i=1

def menu():
    tr=train()
    tick=tickets()
    print
    print "WELCOME".center(80)
    while True:
            print
            print "="*80
            print " \t\t\t\t  Railway"
            print
            print "="*80
            print
            print "\t\t\t1. **Update Train Details"
            print
            print "\t\t\t2. Train Details"
            print
            print "\t\t\t3. Ticket Reservation"
            print
            print "\t\t\t4. Ticket Cancellation "
            print
            print "\t\t\t5. PNR Status"
            print
            print "\t\t\t6. Quit."
            print"** - office use....."
            ch=int(raw_input("\t\t\t Enter Choice : "))
            os.system('cls')
            print "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\nn\n\n\n\n\n\n\t\t\t\t\t\t\tLOADING…. .",
            time.sleep(1)
            print (".",
            time.sleep(0.5)
            print (".")
            time.sleep(2)
            os.system('cls')
            if ch==1:
                j="*****"
                r=raw_input("\n\n\n\n\n\n\n\n\n\n\t\t\t\t Enter Password: ")
                os.system('cls')
                if (j==r):
                    x='y'
                    while (x.lower()=='y'):
                        fout=open("tr1details.dat","ab")
                        tr.getinput()
```

```
            dump(tr,fout)
            fout.close()
            print"\n\n\n\n\n\n\n\n\n\t\t\t List of Train Availablity.",
            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=raw_input("\t\t Addiction of details needed? ")
            os.system('cls')
         continue
      elif(j<>r):
         print"\n\n\n\n\n"
         print "Invalid  Password".center(80)
elif ch==2:
   fin=open("tr1details.dat",'rb')
   if not fin:
      print "ERROR"
   else:
      try:
         while True:
            print"*"*80
            print"\t\t\t\t Train Details  "
            print"*"*80
            print
            tr=load(fin)
            tr.output()



            raw_input("Press ENTER to review other train details;")
            os.system('cls')
      except EOFError:
          pass
elif ch==3:
   print'='*80
   print "\t\t\t\t Ticket Reservation"
   print'='*80
   print
   tick.reservation()
elif ch==4:
   print"="*80
   print"\t\t\t\t Ticket Cancellation"
   print
```

```
            print"="*80
            print
            tick.cancellation()
        elif ch==5:
            print "="*80
            print("PNR Status".center(80))
            print"="*80
            print
            tick.display()
        elif ch==6:
            quit()

        raw_input("Press ENTER to go home page".center(80))
        os.system('cls')

menu()
```

Forwarding Queries:

```
import smtplib, ssl
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

sender_email = "user@gmail.com"
receiver_email = "rail@gmail.com"
password = input("Password:")

message = MIMEMultipart("alternative")
message["Subject"] = "multipart test"
message["From"] = sender_email
message["To"] = receiver_email

text = """\
Hi,
How are you?
Real Python has many great tutorials:
www.realpython.com"""
html = """\
<html>
  <body>
    <p>Hi,<br>
       How are you?<br>
       <a href="http://www.realpython.com">Real Python</a>
       has many great tutorials.
    </p>
  </body>
</html>
```

```
"""
part1 = MIMEText(text, "plain")
part2 = MIMEText(html, "html")

message.attach(part1)
message.attach(part2)

context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
    server.login(sender_email, password)
    server.sendmail(
        sender_email, receiver_email, message.as_string()
    )
```

Queries Response:

```
import email, smtplib, ssl

from email import encoders
from email.mime.base import MIMEBase
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

subject = "An email with attachment from Python"
body = "This is an email with attachment sent from Python"
sender_email = "rail@gmail.com"
receiver_email = "user@gmail.com"
password = input("Password:")

message = MIMEMultipart()
message["From"] = sender_email
message["To"] = receiver_email
message["Subject"] = subject
message["Bcc"] = receiver_email
message.attach(MIMEText(body, "plain"))

filename = "document.pdf"

with open(filename, "rb") as attachment:
    part = MIMEBase("application", "octet-stream")
    part.set_payload(attachment.read())

encoders.encode_base64(part)

part.add_header(
    "Content-Disposition",
    f"attachment; filename= {filename}",
)
```

```
message.attach(part)
text = message.as_string()

context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
   server.login(sender_email, password)
   server.sendmail(sender_email, receiver_email, text)
```

Feed Info:

```
import requests, json
api_key = "Your_API_key"
base_url = "https://api.railwayapi.com/v2/pnr-status/pnr/"
pnr_number = "6515483790"
complete_url = base_url + pnr_number + "/apikey/" + api_key + "/"
response_ob = requests.get(complete_url)
result = response_ob.json()
if result["response_code"] == 200:
        train_name = result["train"]["name"]
        train_number = result["train"]["number"]
        from_station = result["from_station"]["name"]
        to_station = result["to_station"]["name"]
        boarding_point = result["boarding_point"]["name"]
        reservation_upto = result["reservation_upto"]["name"]
        pnr_num = result["pnr"]
        date_of_journey = result["doj"]
        total_passengers = result["total_passengers"]
        passengers_list = result["passengers"]
        chart_prepared = result["chart_prepared"]
        print(" train name : " + str(train_name)
                + "\n train number : " + str(train_number)
                + "\n from station : " + str(from_station)
                + "\n to station : " + str(to_station)
                + "\n boarding point : " + str(boarding_point)
                + "\n reservation upto : " + str(reservation_upto)
                + "\n pnr number : " + str(pnr_num)
                + "\n date of journey : " + str(date_of_journey)
                + "\n total no. of passengers: " + str(total_passengers)
                + "\n chart prepared : " + str(chart_prepared))
        for passenger in passengers_list:
                passenger_num = passenger["no"]
                current_status = passenger["current_status"]
                booking_status = passenger["booking_status"]
                print(" passenger number : " + str(passenger_num)
                        + "\n current status : " + str(current_status)
                        + "\n booking_status : " + str(booking_status))
else:
        print("Record Not Found")
```