Data Science File

Q-1 Create package to manage railway ticket booking. [modules are the part of package]

Railway_reservation:

- __init__.py
- Booking.py
- Ticket.py
- Train.py
- User.py

__init__.py

```
from railway_reservation.train import Train
from railway_reservation.user import User
from railway_reservation.booking import book_ticket
from railway_reservation.ticket import print_tickets
```

#booking.py

```
from train import Train
from user import User
def book_ticket(user:User, train:Train, seats:int):
   if train.book_seats(seats):
      ticket={
        "train_number":train.train_number,
         "train_name":train.name,
         "seats":seats
      }
      user.add_ticket(ticket)
      print(f"Ticket booked on {train.name} ({train.train_number})
for {seats} seat(s).")
   else:
      print("Not enough seats available.")
```

#ticket.py

```
def print_tickets(user):
   print(f"\n Tickets for {user.name}: ")
   for t in user.tickets:
      print(f"Train:{t['train name']}
({['train_number']}),seats:{t['seats']}")
```

#train.py

```
class Train:
   def __init__(self, train_number, name, total_seats):
      self.train_number=train_number
      self.name=name
      self.total_seats=total_seats
      self.available_sears=total_seats
   def book_seats(self, count):
      if self.available_seats>=count:
         self.available_seats-=count
         return True
      return False
```

#user.py

```
class User:
   def __init__(self, name, age, gender):
      self.name=name
      self.age=age
      self.gender=gender
      self.tickets=[]
   def add_ticket(self, ticket):
      self.tickets.append(ticket)
```

#main.py

```
from railway_reservation import train, user, book_ticket,
print_tickets
train=Train("22890","Saurashtra Express", 60)
user=User("Krishna",19,"M")
```

```
book_ticket(user, train, 4)
print_tickets(user)
```

Q-2 Create package to manage movie ticket booking. [modules are the part of package]

Movie_booking

- __init__.py
- Booking.py
- showtime.py
- User.py
- Main.py

#__init__.py

```
from Movie_booking.movie import Movie
from Movie_booking.user import User
from Movie_booking.showtime import ShowTime
from Movie_booking.booking import book_ticket
```

#booking.py

```
from user import User
from showtime import ShowTime
def book_ticket(user:User, showtime:ShowTime, num_seat):
   if showtime.book_seat(num_seat):
      booking_info={
      "movie":showtime.movie.title,
      "time":showtime.time,
      "seats":num_seat
      }
      user.add_booking(booking_info)
      print(f"Booked {num_seat} seat for '{showtime.movie.title}' at
{showtime.time}")
   else:
      print("Not enough seat available.")
```

#movie.py

```
class Movie:
   def __init__(self, title, duration, rating):
```

```
        self.title=title
        self.duartion=duration #in minutes
        self.rating=rating

#showtime.py
from movie import Movie
class ShowTime:
    def __init__(self, movie=Movie, time, seat=100):
        self.movie=movie
        self.time=time
        self.available_seat=seat
    def book_seats(self, count=1):
        if self.available_seat>=count:
            self.available_seat-=count
            return True
#user.py
class User:
    def __init__(self, name):
        self.name=name
        self.bookings=[]
    def add_booking(self, booking):
        self.bookings.append(booking)

#main.py
from Movie_booking import Movie, User, ShowTime, book_ticket
#create movie and showtime
movie=Movie("RRR", 180,"R")
showtime=ShowTime(movie,"10:30 PM",52)

#create a user
user=User("Krishna")
#Book some tickets
book_ticket(user, showtime, 3)
#print user's bookings
print(f"\n {b['movie']} at {b['time']} ({b['seat']} seat)")
```

Q-3 Create package to manage phone book. [modules are the part of package]

Phone_book:

- __init__.py
- operations.py

#operations.py

```
directory={}
def add_contact(name, number):
   directory[name]=number
   print(f"Contact saved:{name} -> {number}")
def search_contact(name):
   if name in directory:
      print(f"{name}: {directory[name]}")
   else:
      print(f"{name} not found in the directory")

def delete_contact(name):
   if name in directory:
      del directory[name]
      print(f"{name} deleted from directory")
   else:
      print(f"{name} not found")
```

#mainfile.py

```
from Phonebook import operations as pb
pb.add_contact("yash", "9589762426")
pb.add_contact("vikas", "9954046804")
pb.search_contact("yash")
pb.delete_contact("yash")
pb.search_contact("yash")
```

Q-4 Create modules for college management system.

College_mngt:

- __init__.py
- Details.py
- Result.py
- Mainfile.py

#details.py

```
def display_student(name, roll):
   print(f"Name:{name}")
   print(f"Roll No.:{roll}")
```

#result.py

```
def calculate_average(marks):
   """Return average of the marks."""
   return sum(marks)/len(marks)
def calculate_grade(average):
   """Return grade based on average marks."""
   if average>=90:
      return "A"
   elif average>=75:
      return "B"
   elif average>=60:
      return "C"
   else:
      return "D"
```

#mainfile.py

```
from college_mngt import details, result
name="Yash "
roll=78
marks=[75, 70, 72]
```

```
#display student info
details.display_student(name, roll)
#process results
average=result.calculate_average(marks)
grade=result.calculate_grade(average)
print(f"Marks:{marks}")
print(f"Average:{average:.2f}")
print(f"Grade:{grade}")
```

Q-5 Write python script to read and write text file.

```
fileobject=open("report.text","w+")
print("Writing data in the file")
print()
while True:
   line=input("Enter a sentences:")
   fileobject.write(line)
   fileobject.write('\n')
   choice=input("Do you wish to enter more data?(y/n):")
   if choice in('n','N'): break
   print ("the byte position of file object is", fileobject.tell())
   fileobject.seek(o)
   print()
   print("Reading data from the file")
   str=fileobject.read()
   print(str)
```

Q-6 Write python script to read and write binary file.

```
f=open("bfile.bin","wb+")
message="Learning Python"
file_encode=message.encode("ASCII")
f.write(file_encode)
f.seek(0)
bdata=f.read()
print("Binary Data:", bdata)
ntext=bdata.decode("ASCII")
print("Normal data:", ntext)
```

Q-7 Write python script to read and write CSV file.

```
import csv
with open('info.csv','r') as file:
   csv_reader=csv.reader(file)
   #read each row of the csv file
   for row in csv_reader:
      print(row)

with open('info.csv','w',newline='') as file:
   csv_writer=csv.writer(file)
   csv_writer.writerow(['Name','Age','Country'])
   csv_writer.writerow([Krishna,19,'India'])
   csv_writer.writerow(['Yash',19,'India'])
   csv_writer.writerow(['Vikas',20,'India'])
```

Q-8 Write a python script to read and write the employee record.

```
import pickle
print("WORKING WITH BINARY FILES")
bfile=open("mpfile.dat","ab")
recno=1
print("Enter records of employees")
print()
while True:
   print("RECORD NO.",recno)
   eno=int(input("\t Employee Number:"))
   ename=input("\t Employee Name:")
   ebasic=int(input("\t Basic Salary:"))
   allow=int(input("\t Allowances:"))
   totsal=ebasic+allow
   print("\t TOTAL SALARY:",totsal)
   edata=[eno, ename, ebasic, allow, totsal]
   pickle.dump(edata, bfile)
   ans=input("Do you want to enter more records(y/n)")
   recno=recno+1
   if ans.lower()=='n':
      print("RECORD ENTRY OVER")
      print()
      break
print("Size of binary file(in bytes):", bfile.tell())
bfile.close()
print("Now reading the employee records from the file")
print()
readrec=1
try:
   with open("empfile.dat","rb") as bfile:
      while True:
         edata=pickle.load(bfile)
         print("Record Number:", readrec)
         print(edata)
```

```
        readrec=readrec+1
except EOFError:
   pass
bfile.close()
```

Q-9 Write a program to read and write pickle files.

```
import pickle
listvalues=[1, "Yash",'M']
fileobject=open("mybinary.dat","wb")
pickle.dump(listvalues, fileobject)
fileobject.close()

import pickle
print("The data that were stored in file are:")
fileobject=open("mybinary.dat","rb")
objectvar=pickle.load(fileobject)
fileobject.close()
print(objectvar)
```

Q-10 Write python script to add,modify,delete,search movie in CSV file.

```
import csv
def init_file():
   with open('movies.csv','w', newline='') as file:
      writer=csv.writer(file)
      writer.writerow(['moviename', 'actor', 'releaseyr', 'director'])

#add movie
def add_movie():
   print("\n ADD NEW MOVIE")
   movie=[
      input("Movie Name:"),
      input("Actor:"),
      input("Release Year:"),
      input("Director:"),
   ]
   with open('movies.csv','a', newline='') as file:
      writer=csv.writer(file)
      writer.writerow(movie)
   print("Movie added successfully!")

 #delete movie
def delete_movie():
   print("\n Delete Movie")
   name=input("Enter movie name to delete:")
   movies=[]
   found=False
   with open('movies.csv','r') as file:
      reader=csv.reader(file)
      header=next(reader)
      for row in reader:
         if row[0].lower()==name.lower():
            found=True
```

```
            else:
                movies.append(row)
        if found:
            with open('movies.csv','w', newline='') as file:
                writer.csv.writer(file)
                writer.writerow(header)
                writer.writerows(movies)
            print("Movie deleted successfully!")
        else:
            print("Movie not found!")

#search movie
def search_movie():
    print("\n  Search Movies")
    print("\n 1. By movie name")
    print("\n 2. By actor")
    choice=input("Choose search type(1-2):")
    term=input("Enter search term:").lower()
    found=False
    with open('movies.csv','r') as file:
        reader=csv.DictReader(file)
        print("\n Search results:")
        print("-"*50)
        for row in reader:
            if(choice=='1' and term in row['movie_name'].lower()) or
(choice=='2' and term in row['actor'].lower()):
                print(f"Movie:{row['moviename']}")
                print(f"Actor:{row['actor']}")
                print(f"Year:{row['releaseyr']}")
                print(f"Director:{row['director']}")
                print("-"*50)
                found=True
        if not found:
            print("No matching movies found.")
```

```
#sort movies
def sort_movies():
   with open('movies.csv','r') as file:
      reader=csv.DictReader(file)
      movies=sorted(reader, key=lambda
x:x['moviename'].lower())
   print("\n Movies sorted by name:")
   print("-"*50)
   for movie in movies:
         print(f"{movie['moviename']} ({movie['releaseyr']})")
         print(f"Starring:{movie['actor']}")
         print(f"Directed by:{movie['director']}")
         print("-"*50)

#view all movies
def view_movies():
   print("\n All movies in database:")
   print("-"*50)
   with open('movies.csv','r') as file:
      reader=csv.DictReader(file)
      for row in reader:
         print(f"Movie:{row['moviename']}")
         print(f"Actor:{row['actor']}")
         print(f"Year:{row['releaseyr']}")
         print(f"Director:{row['director']}")
         print("-"*50)

#main menu
def main():
   init_file()
   while True:
      print("\n MOVIE DATABASE MENU")
      print("1.Add movie")
      print("2.Delete movie")
      print("3.Search movie")
```

```
        print("4.Sort movies")
        print("5.View all movies")
        print("6.Exit")
        choice=input("Enter your choice(1-6):")
        if choice=="1":
            add_movie()
        elif choice=="2":
            delete_movie()
        elif choice=="3":
            search_movie()
        elif choice=="4":
            sort_movie()
        elif choice=="5":
            view_movies()
        elif choice=="6":
            print("Existing program...")
            break
        else:
            print("Invalid choice. Please try again...")
if __name__=="__main__":
    main()
```