EDS Assignment 1

Topic: Railway Management System

Group Members:

- Amrik Bhadra (203)
- Suryansh Ambekar (202)
- Vaibhav Dingalwar (216)

Code:

```
import csv

# function to find maximum age
def max_age():
    name = passengerName[age.index(max(age))]
    a = max(age)
    print(name, a, end="\n\n")

# function to find minimum age
def min_age():
    name = passengerName[age.index(max(age))]
    a = min(age)
    print(name, a, end="\n\n")

# function to find names starting with 'S'
def name_start_with_S():
    name_S = []
    for i in range(len(passengerName)):
        n = passengerName[i]
        if n[0] == 'S':
            name_S.append(n)

    if len(name_S) == 0:
        print("No Passengers found with 'S'", end="\n\n")
    else:
        print("Passengers name : ", name_S)
    print(end="\n\n")

# function to count number of passengers having age below 18
def age_below_18():
    countBelow18 = 0
    for i in range(len(age)):
        if age[i] < 18:
            countBelow18 += 1
```

```
    if countBelow18 == 0:
        print("No passenger with age under 18", end="\n\n")
    else:
        print("Number of passengers below 18 : ", countBelow18, end="\n\n")

# function to count number of upper berths which are confirmed
def count_confirmed_upper_berths():
    countUpperBerthConfirmed = 0
    for i in range(len(berth)):
        if(berth[i] == 'UPPER' and status[i] == 'CONFIRMED'):
            countUpperBerthConfirmed += 1

    if countUpperBerthConfirmed == 0:
        print("No confirmed upper berths", end="\n\n")
    else:
        print("Number of upper berths which are confirmed : ",
countUpperBerthConfirmed, end="\n\n")


# function to count the number of trains which are going to depart between 5pm and
10pm
def count_5_10():
    countTime = 0
    for i in range(len(departureTime)):
        dt = departureTime[i]
        hd = dt[0:3]
        if(hd >= '17' and hd <= '22'):
            countTime += 1

    if countTime == 0:
        print("No trains departing between 5pm and 10pm", end="\n\n")
    else:
        print("Number of train going to depart between 5pm and 10pm : ",
countTime, end="\n\n")

# function to print passenger names paying ticket fare above Rs. 1000
def fare_above_1000():
    peopleAbove1000 = []
    fareAbove1000 = []
    for i in range(len(ticketFare)):
        if(ticketFare[i] > 1000):
            peopleAbove1000.append(passengerName[i])
            fareAbove1000.append(ticketFare[i])
    print("Passengers paying fare above 1000:")
    for i in range(len(peopleAbove1000)):
        print(peopleAbove1000[i], "\t\t", fareAbove1000[i])
    print(end="\n\n")

# function to enter train Number and get its details
def get_details():
    trnNo = int(input("Enter train no.: "))
```

```
    flag = 0
    for i in range(len(trainNo)):
        if(trnNo == trainNo[i]):
            flag = 1
            break
    if(flag == 0):
        print("\nTrain No. not available in the list", end="\n\n")
    else:
        print("\nTrain details:")
        print("Train No. : ", trainNo[i])
        print("Train Name : ", trainName[i])
        print("Departure station : ", startStation[i])
        print("Destination station : ", endStation[i])
        print("Departure time : ", departureTime[i])
        print("Arrival time : ", arrivalTime[i])
    print(end="\n\n")

# function to find the trains departing from csmt
def depart_from_csmt():
    trainsFromCsmt = []
    for i in range(len(startStation)):
        if(startStation[i] == "CSMT"):
            trainsFromCsmt.append(trainName[i])

    if len(trainsFromCsmt) == 0:
        print("No train departing from csmt", end="\n\n")
    else:
        print("Trains from CSMT : ", trainsFromCsmt, end="\n\n")

# function to enter a passengers name and print their travel details
def print_passenger_details():
    name = input("Enter passengers name : ")
    name = name.upper()
    for i in range(len(passengerName)):
        if(name == passengerName[i]):
            flag = 1
            break

    if flag == 0:
        print("\nPassenger not present in the list", end="\n\n")
    else:
        print("\nPassenger details:-")
        print("Passenger name : ", passengerName[i])
        print("Age : ", age[i])
        print("Contact No. : ", contactNo[i])
        print("Train name : ", trainName[i])
        print("Train number : ", trainNo[i])
        print("Ticket booked from : ", startStation[i])
        print("Ticket booked till : ", endStation[i])
        print("Ticket status : ", status[i])
        print("Berth : ", berth[i])
```

```
        print("Berth No. : ", berthNo[i])
        print("Coach : ", coach[i])
        print("Ticket fare : ", ticketFare[i], end="\n\n")

# function to enter name of passenger to print his/her journey duration
def journey_duration():
    name = input("Enter passengers name : ")
    name = name.upper()
    for i in range(len(passengerName)):
        if(name == passengerName[i]):
            flag = 1
            break

    if flag == 0:
        print("\nPassenger not present in the list", end="\n\n")
    else:
        depTime = departureTime[i]
        depTime_hrs = int(depTime[0:2])
        depTime_min = int(depTime[3:])
        arrTime = arrivalTime[i]
        arrTime_hrs = int(arrTime[0:2])
        arrTime_min = int(arrTime[3:])

        # convert both times to minutes
        depTime_total_min = (depTime_hrs * 60) + depTime_min
        arrTime_total_min = (arrTime_hrs * 60) + arrTime_min

        # find the difference between the two times in minutes
        timeDifference_min = abs(depTime_total_min - arrTime_total_min)

        # convert the difference back to hours and minutes
        difference_hours = timeDifference_min // 60
        difference_minutes = timeDifference_min % 60

        print("\nPassenger name : ", passengerName[i])
        print("Departure time : ", departureTime[i])
        print("Arrival time : ", arrivalTime[i])
        print("Journey duration : ", difference_hours, " hours ",
difference_minutes, " minutes")


# function to print passenger names with RAC ticket status
def passengers_with_rac():
    racPassengers = []
    for i in range(len(status)):
        if(status[i] == 'RAC'):
            racPassengers.append(passengerName[i])

    if len(racPassengers) == 0:
        print("\nNo passengers with RAC ticket status", end="\n\n")
    else:
        print("\nPassengers with RAC : ", racPassengers, end="\n\n")
```

```
# function to convert fare from inr to dollars
def inr_to_dollar():
    fareInDollar = []
    for i in range(len(ticketFare)):
        dollar = round(ticketFare[i]/81.73, 2)
        fareInDollar.append(dollar)

    print("\nTicket fare in dollar:")
    for i in range(len(passengerName)):
        print(passengerName[i], "\t\tRs. ", ticketFare[i], "\t\t",
fareInDollar[i], "$")
    print(end="\n\n")

#function to sort the age of passengers
def sort_age():
    age1 = []
    for i in range(len(age)):
        age1.append(age[i])
    age1.sort()
    print("\nAge in ascending order : ", age1, end="\n\n")


# opening the files in read mode
passengerfile = open("passengerDetails.csv", "r")
stationfile = open("stationDetails.csv", "r")
ticketfile = open("ticketDetails.csv", "r")
trainFile = open("trainDetails.csv", "r")

# retreiving the data from the csv files and stroing it in form of list using csv
reader
passengerData = list(csv.reader(passengerfile))
stationData = list(csv.reader(stationfile))
ticketData = list(csv.reader(ticketfile))
trainData = list(csv.reader(trainFile))

#  creating empty lists for different data items
passengerName = []
age = []
contactNo = []
startStation = []
endStation = []
berth = []
berthNo = []
status = []
ticketFare = []
trainName = []
trainNo = []
coach = []
arrivalTime = []
departureTime = []
```

```
# storing data into their respective lists
for i in range(1, len(passengerData)):
    passengerName.append(passengerData[i][0])     # storing passengers name
    age.append(int(passengerData[i][1]))          # storing age of passengers
    contactNo.append(int(passengerData[i][2]))    # storing contact no. of
passenger
    startStation.append(stationData[i][1])        # storing start station
    endStation.append(stationData[i][2])          # storing destination station
    berth.append(ticketData[i][1])                # storing the berth type
    berthNo.append(int(ticketData[i][2]))         # storing berth no.
    status.append(ticketData[i][3])               # storing ticket status
    ticketFare.append(float(ticketData[i][4]))    # storing ticket fare
    trainName.append(trainData[i][1])             # storing train name
    trainNo.append(int(trainData[i][2]))          # storing train no.
    coach.append(trainData[i][3])                 # storing coach no.
    departureTime.append(trainData[i][4])         # storing train's departure time
    arrivalTime.append(trainData[i][5])           # storing train's arrival time

print("\nAnalysis Menu:-")
print("\n1. Find the passenger name having maximum age")
print("2. Find the passenger name having minimum age")
print("3. Find the passenger name(s) whose name starts with 'S'")
print("4. Count the number of passenger(s) whose age is below 18")
print("5. Count the number of confirmed upper berths")
print("6. Count number of trains departing between 5pm and 10pm")
print("7. Print passenger names paying travel fare above Rs. 1000")
print("8. Enter train Number and get its details")
print("9. Print the trains departing from csmt")
print("10. Enter a passenger's name and print his/her travel details")
print("11. Enter a passenger's name and print his/her travel time duration")
print("12. Print passenger names with RAC ticket status")
print("13. Convert ticket fare from INR to Dollars")
print("14. Sort age of all passengers in ascending order")

ch = 'Y'
while(ch == 'Y' or ch == 'y'):
    choice = int(input("\nEnter option which you want to perform : "))
    print(end="\n")

    match choice:
        case 1:
            max_age()
        case 2:
            min_age()
        case 3:
            name_start_with_S()
        case 4:
            age_below_18()
        case 5:
            count_confirmed_upper_berths()
        case 6:
```

```
            count_5_10()
        case 7:
            fare_above_1000()
        case 8:
            get_details()
        case 9:
            depart_from_csmt()
        case 10:
            print_passenger_details()
        case 11:
            journey_duration()
        case 12:
            passengers_with_rac()
        case 13:
            inr_to_dollar()
        case 14:
            sort_age()
        case _:
            print("Invalid option")

    ch = input("Want to perform any more task? (Y/N): ")
    print("_____
_____")

# Merging all files into single file
railwayFile = open("railwayDetails.csv", "w")
railwayData = csv.writer(railwayFile)

for i in range(len(passengerData)):
    rowData = passengerData[i] + stationData[i] + ticketData[i] + trainData[i]
    railwayData.writerow(rowData)

# closing all the files
passengerfile.close()
stationfile.close()
ticketfile.close()
trainFile.close()
railwayFile.close()
```

Output:

```
Analysis Menu:-

1. Find the passenger name having maximum age
2. Find the passenger name having minimum age
3. Find the passenger name(s) whose name starts with 'S'
4. Count the number of passenger(s) whose age is below 18
5. Count the number of confirmed upper berths
6. Count number of trains departing between 5pm and 10pm
7. Print passenger names paying travel fare above Rs. 1000
8. Enter train Number and get its details
9. Print the trains departing from csmt
10. Enter a passenger's name and print his/her travel details
11. Enter a passenger's name and print his/her travel time duration
12. Print passenger names with RAC ticket status
13. Convert ticket fare from INR to Dollars
14. Sort age of all passengers in ascending order

Enter option which you want to perform : 1

KIRAN MALHOTRA 60

Want to perform any more task? (Y/N): y
--------------------------------------------------------------------------------

Enter option which you want to perform : 2

KIRAN MALHOTRA 12

Want to perform any more task? (Y/N): y
--------------------------------------------------------------------------------
```

```
Enter option which you want to perform : 3

Passengers name :  ['SOHAM GANGULY', 'SANIKA KAPOOR', 'SAVITRI BOLAJ', 'SAMARTH BOLAJ', 'SAMEER SEN']


Want to perform any more task? (Y/N): y
--------------------------------------------------------------------------------

Enter option which you want to perform : 4

Number of passengers below 18 :  3

Want to perform any more task? (Y/N): y
--------------------------------------------------------------------------------

Enter option which you want to perform : 5

Number of upper berths which are confirmed :  2

Want to perform any more task? (Y/N): y
Number of train going to depart between 5pm and 10pm :  3

Want to perform any more task? (Y/N): y
--------------------------------------------------------------------------------
```

```
Enter option which you want to perform : 7

Passengers paying fare above 1000:
MEENAL SINGH              1870.0
SANIKA KAPOOR             2300.0
KIRAN MALHOTRA            1275.0
SAVITRI BOLAJ             4800.0
SAMARTH BOLAJ             4800.0
AMAN BOLAJ                4800.0
ESHITA SEN                1250.0
SAMEER SEN                1250.0


Want to perform any more task? (Y/N): y
------------------------------------------------------------

Enter option which you want to perform : 8

Enter train no.: 121234

Train details:
Train No. :  121234
Train Name :  HUMSAFAR EXP
Departure station :  NAGPUR
Destination station :  PUNE
Departure time :  09:20
Arrival time :  17:05


Want to perform any more task? (Y/N): y
------------------------------------------------------------
```

```
Enter option which you want to perform : 9

Trains from CSMT :  ['RAJDHANI EXP', 'VANDE BHARAT', 'DURONTO EXP']

Want to perform any more task? (Y/N): y
------------------------------------------------------------------------

Enter option which you want to perform : 10

Enter passengers name : Samarth bolaj

Passenger details:-
Passenger name :  SAMARTH BOLAJ
Age :  12
Contact No. :  7867982340
Train name :  RAJDHANI EXP
Train number :  122456
Ticket booked from :  PUNE
Ticket booked till :  HOWRAH
Ticket status :  CONFIRMED
Berth :  UPPER
Berth No. :  68
Coach :  B7
Ticket fare :  4800.0

Want to perform any more task? (Y/N): y
------------------------------------------------------------------------
```

```
Enter option which you want to perform : 11

Enter passengers name : samarth bolaj

Passenger name :  SAMARTH BOLAJ
Departure time :  07:25
Arrival time :  12:10
Journey duration :  4  hours  45  minutes
Want to perform any more task? (Y/N): y
------------------------------------------------------------------------

Enter option which you want to perform : 12


Passengers with RAC :  ['SOHAM GANGULY', 'DEV PANDEY']

Want to perform any more task? (Y/N): y
------------------------------------------------------------------------
```

```
Enter option which you want to perform : 13


Ticket fare in dollar:
MEENAL SINGH            Rs.  1870.0             22.88 $
RAJESH PRATAP           Rs.  750.0              9.18 $
MAHESH SAXENA           Rs.  890.0              10.89 $
APARNA MANDAL           Rs.  250.0              3.06 $
SOHAM GANGULY           Rs.  400.0              4.89 $
ANAMIKA DESHMUKH                Rs.  566.0              6.93 $
SANIKA KAPOOR           Rs.  2300.0             28.14 $
VARUN SHETTY            Rs.  890.0              10.89 $
KIRAN MALHOTRA          Rs.  1275.0             15.6 $
DEV PANDEY              Rs.  768.0              9.4 $
SAVITRI BOLAJ           Rs.  4800.0             58.73 $
SAMARTH BOLAJ           Rs.  4800.0             58.73 $
AMAN BOLAJ              Rs.  4800.0             58.73 $
ESHITA SEN              Rs.  1250.0             15.29 $
SAMEER SEN              Rs.  1250.0             15.29 $
Want to perform any more task? (Y/N): y
------------------------------------------------------------------------------------------------------------

Enter option which you want to perform : 14


Age in ascending order :  [12, 16, 17, 19, 21, 22, 28, 33, 34, 45, 45, 45, 50, 56, 60]
Want to perform any more task? (Y/N): y
------------------------------------------------------------------------------------------------------------

Enter option which you want to perform : 15

Invalid option
Want to perform any more task? (Y/N): n
------------------------------------------------------------------------------------------------------------
```