

SHREE SWAMINARAYAN ENGLISH MEDIUM SCHOOL (CBSE), SALVAV

A PROJECT REPORT

ON

Railway Station management



FOR

AISSCE 2023-24 EXAMINATION

(AS A PART OF INFORMATION PRACTICES)

SUBMITTED BY: PiyushKumar Thummar

ROLL NUMBER: 18

SEAT NUMBER: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

UNDER THE GUIDANCE OF: <name of the teacher>

CERTIFICATE

This is to certify that the project entitled Railway Station Management

(Is a Bona-fide work done by)

Piyush Thummar  
Of class XII-Science From

SHREE SWAMINARAYAN GURUKUL, SALVAV

For session 2023-24 In partial fulfilment of

CBSE’s AISSCE Examination 2023-2024

Has been carried out under my direct supervision and guidance.

This report or a similar report on this topic has not been submitted for any other examination and does not from a part of any other course undergone by the candidate.

SIGN. OF EXTERNAL EXAMINER. SIGN. OF INTERNAL EXAMINER

SIGN. OF PRINCIPAL SCHOOL SEAL

ACKNOWLEDGEMENT

In the accomplishment of this project successfully, many people have best owned upon me their blessings and the heart pledged support, this time I am utilizing to thank all the people who have been concerned with this project.

Primarily I would thank God for being able to complete this project with success. Then I would like to thank my principal <name of principal> and IP teacher <name of teacher> , whose valuable guidance has been the ones that helped me patch this project and make it full proof success. Her suggestions and her instructions have served as the major contributor towards the completion of the project.

Then I would like to thank my parents who have helped me with their valuable suggestions and guidance has been very helpful in various phases of the completion of the project.

Piyush Thummar  
12th SCIENCE A

REFERENCES

Text book by Sumita Arora  
Text book by NCERT  
[www.Google.com](http://www.Google.com)

|  |  |  |
| --- | --- | --- |
| Index | | |
| Sr. No. | **Description** | **Page no.** |
| 1 | Certificate | 2 |
| 2 | Acknowledgment | 3 |
| 3 | References | 4 |
| 4 | Introduction | 6 |
| 5 | Objective of the project | 6 |
| 6 | CSV files | 7 |
| 7 | Source Code | 9 |
| 8 | Output Screen | 17 |
| 9 | Hardware and software requirement | 23 |

Kljsdf.k.jjdf;l

PROJECT ON Railway Station Management

**INTRODUCTION**

This project is all about a software for railway station. It helps people to book tickets in a very short time.

It also have some other functionalities such as ticket viewing, view latency, etc.

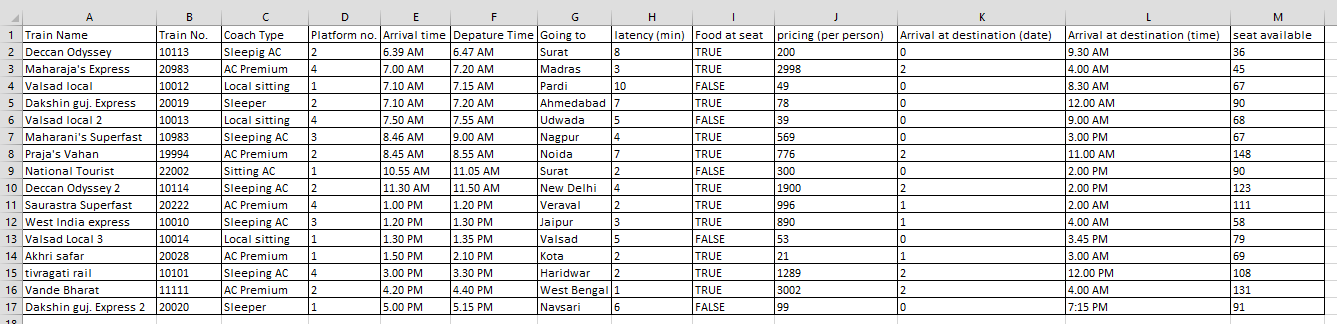
**OBJECTIVES OF THE PROJECT**

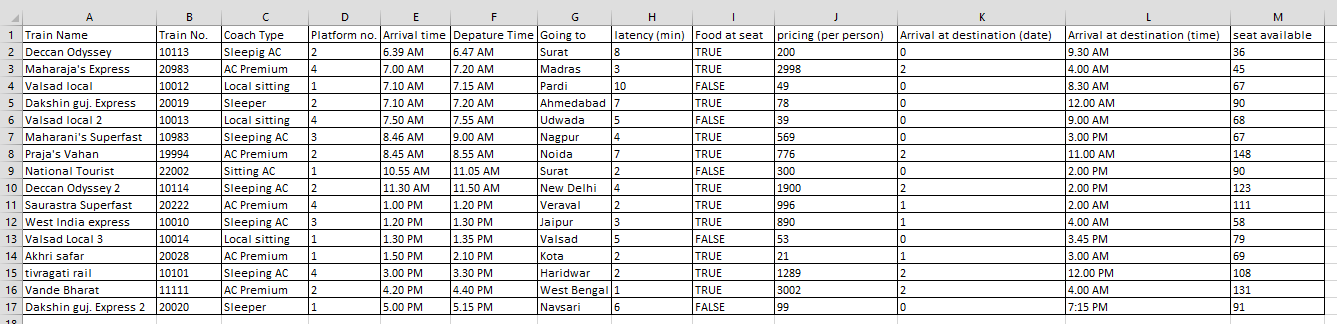
The objective of this project is to let the students apply the programming knowledge into a real-world situation/problem and exposed the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.
2. Apply object oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
5. Students will demonstrate ability to conduct a research or applied Computer Science Project, requiring writing and presentation skills which exemplify scholarly style in Computer science.

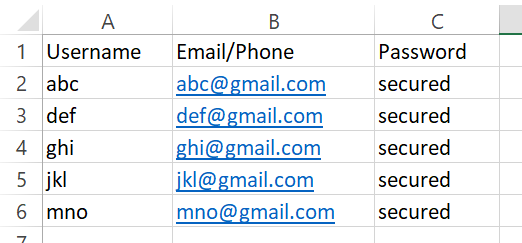
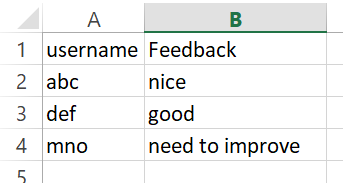
# CSV FILE IN EXCEL

Info.csv

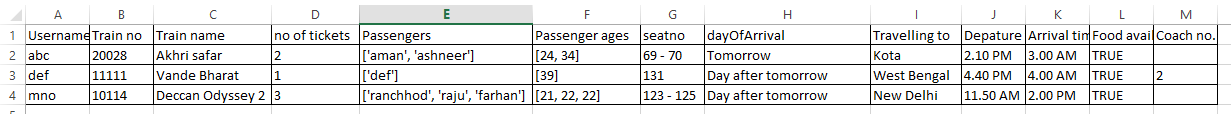


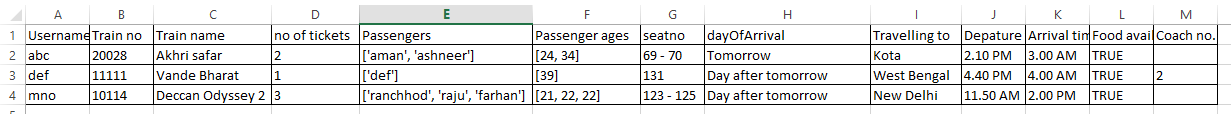


logins.csv Feedback.csv

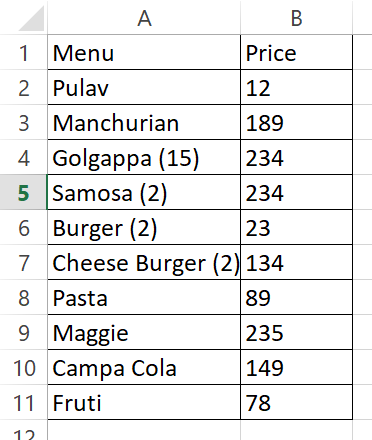
 

Booking.csv





Food menu.csv



SOURCE CODE

# Importing necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as pl

# Printing the welcome menu

print(' =================================================================================\n \

||\t\t\t\tWelcome to Vapi Station\t\t\t\t||\n \

=================================================================================')

# Predeclaring some variables

login = False

book = False

# Getting user's choice

inpt= '0'

while inpt != '8':

    # Dataframe That stores all the info

    infoDft = pd.read\_csv('Data/info.csv')

    # Printing the Menus

    print('\n\t\t==================================================\n\t\t\t\t\tMenus \

          \n\t\t==================================================')

    print("\n\n 1. Login / Sign up \n \

2. View today's Schedule \n \

3. Book a ticket \n \

4. View your ticket \n \

5. View latency of trains\n \

6. Order food on your seat\n \

7. Give Feedback\n \

8. Exit")

    # Sorting functions according to the choice

    inpt = '0'

    inpt = input('\n\nPlease select one of these:\n\t')

    # To login or Sign up

    if inpt == '1':

        inpt2= '0'

        while inpt2 not in ('1','2'):

            # Getting the second choice

            inpt2= input('\n\n1) Sign up\n2) Login\n\n')

            # To Sign up

            if inpt2 == '1':

                # Predeclaring some variables

                Username, Email, password = np.NaN, np.NaN, np.NaN

                # Taking user details

                print('Please fill these details')

                Username = input('Username: \n')

                Email = input('Email: \n')

                password= input('Create a password: \n')

                # If details are valid

                if Username != np.NaN and Email != np.NaN and password != np.NaN:

                    # Adding the user to csv file file

                    lgn= pd.read\_csv('Data/logins.csv')

                    lgn.loc[len(lgn)]= [Username, Email, password]

                    lgn.to\_csv('Data/logins.csv', index=False)

                    login = True

                    print('\n||\tSign up and Login success\t||')

                # If details are not valid

                else:

                    print('\nThe details must not be empty..\n')

                input('\n\n Press Enter to continue!')

            # To Login

            elif inpt2 == '2':

                # Getting the usernames from csv file

                lgn= pd.read\_csv('Data/logins.csv')

                userlist= list(lgn['Username'])

                Username= input('Enter your Username:\n')

                # Verifying Username

                if Username in userlist:

                    # Getting user index

                    indx= userlist.index(Username)

                    # getting password from user

                    password= input('Enter your Passsword:\n')

                    # If password is correct

                    if password==lgn.at[indx,'Password']:

                        login= True

                        print('\n|\tYou are now logged in\t|\n')

                        input('\n\n Press Enter to continue!')

                        # To know the booking status of user

                        booking= pd.read\_csv('Data/booking.csv', sep=':')

                        travellers= list(booking['Username'].values)

                        if Username in travellers:

                            book= True

                    # If password is not true

                    else:

                        print('|\nPassword is incorrect\n|')

                        input('\n\n Press Enter to continue!')

                # If person do not have an account

                else:

                    print('\nThe Username you entered does not exist')

                    input('\n\n Press Enter to continue!')

            # Correcting user's mistake

            else:

                print('\nEnter 1 if you want to create an account\nOr enter 2 if want to login an existing account\n')

                input('Press Enter to continue\t')

    # To get the schedule

    elif inpt == '2':

        print("\n\nHere's the schedule for today:\n\n")

        print(infoDft[['Train Name', 'Train No.', 'Depature Time', 'Going to', 'Coach Type', 'pricing (per person)']])

        input('\n\n Press Enter to continue!')

    # To book a ticket

    elif inpt == '3':

        # If the user has logged in

        if login == True:

            # Predeclaring some variables

            names= []

            ages= []

            print(infoDft[['Train Name', 'Train No.', 'Depature Time', 'Going to', 'Coach Type', 'pricing (per person)']], "\n\n")

            # Getting the train to be booked

            trnno= int(input('Enter the Index no.\n\n'))

            no= int(input('Enter the number of tickets:\n'))

            # Getting the day of arrival

            dayOfArrival= infoDft.iat[trnno,10]

            if dayOfArrival==0:

                dayOfArrival= 'Today'

            elif dayOfArrival==1:

                dayOfArrival= 'Tomorrow'

            elif dayOfArrival==2:

                dayOfArrival= 'Day after tomorrow'

            # Getting the seat no

            if no==1:

                seatno= infoDft.iat[trnno,12]

            else:

                seatno= f'{infoDft.iat[trnno,12]} - {infoDft.iat[trnno,12]+no-1}'

            # Getting the names and ages of passengers

            for i in range(1,no+1):

                name= input(f'Enter name of passenger {i}: ')

                age= int(input(f'Enter age of passenger {i}: '))

                names.append(name)

                ages.append(age)

            # Booking the seat

            print(f'\nIt will cost you {infoDft.iat[trnno,9]\*no} INR')

            input('Press Enter to continue:')

            # Getting coach no. from seat no.

            booking= pd.read\_csv('Data/booking.csv', sep=':')

            coachNo= np.NaN

            if seatno in range(1,71):

                coachNo = 1

            elif seatno in range(71,141):

                coachNo = 2

            elif seatno in range(141,211):

                coachNo = 3

            elif seatno in range(211,281):

                coachNo = 4

            elif seatno in range(281,351):

                coachNo = 5

            # Adding this booking in csv file

            booking.loc[len(booking)]= [Username, infoDft.at[trnno,'Train No.'], infoDft.at[trnno,'Train Name'], no, names, ages, seatno, dayOfArrival, infoDft.at[trnno,'Going to'], infoDft.at[trnno,'Depature Time'] ,infoDft.at[trnno,'Arrival at destination (time)'],infoDft.at[trnno,'Food at seat'], coachNo]

            booking.to\_csv('Data/booking.csv', index=False, sep=':')

            booking= pd.read\_csv('Data/booking.csv', sep=':')

            booking.index= booking['Username']

            # Printing the journey details

            print('\n\nPassenger names:', booking.at[Username,'Passengers'],

                  '\nSeat no:', booking.at[Username,'seatno'],

                  '\nCoach no:', booking.at[Username,'Coach no.'],

                  '\nTrain name:',booking.at[Username,'Train name'],

                  '\nTrain No.:',booking.at[Username,'Train no'],

                  '\nTravelling to:',booking.at[Username,'Travelling to'],

                  '\nDeparture time:',booking.at[Username,'Depature time'],

                  '\nArrival date:', dayOfArrival,

                  '\nArrival time:', booking.at[Username,'Arrival time']

                  )

            book= True

            # Changing seat availability in info.csv

            infoDft.iat[trnno, 12]= infoDft.iat[trnno, 12]+no

            infoDft.to\_csv('Data/info.csv', index= False)

            input('\n\n Press Enter to continue!')

        # If the person has not logged in

        else:

            print('\n\nYou first have to login..')

            input('\n\n Press Enter to continue!')

    # To display the ticket

    elif inpt == '4':

        # Getting the booking info

        if book == True:

            # Importing the dataframe

            booking= pd.read\_csv('Data/booking.csv', sep= ':')

            booking.index= booking['Username']

            # Getting the day of arrival

            dayOfArrival= infoDft.iat[trnno,10]

            if dayOfArrival==0:

                dayOfArrival= 'Today'

            elif dayOfArrival==1:

                dayOfArrival= 'Tomorrow'

            elif dayOfArrival==2:

                dayOfArrival= 'Day after tomorrow'

            # Printing the ticket details

            print('\n\nPassenger names:', booking.at[Username,'Passengers'],

                  '\nSeat no:', booking.at[Username,'seatno'],

                  '\nCoach no:', booking.at[Username,'Coach no.'],

                  '\nTrain name:',booking.at[Username,'Train name'],

                  '\nTrain No.:',booking.at[Username,'Train no'],

                  '\nTravelling to:',booking.at[Username,'Travelling to'],

                  '\nDeparture time:',booking.at[Username,'Depature time'],

                  '\nArrival date:', dayOfArrival,

                  '\nArrival time:', booking.at[Username,'Arrival time']

                  )

            input('\n\n Press Enter to continue!')

        # If the user has not booked a ticket

        else:

            print('Enter first you have to book a ticket')

    # To see the graph of latency of trains

    elif inpt == '5':

        # Getting the x and y axis

        xaxis= list(infoDft.index)

        yaxis= list(infoDft['latency (min)'])

        # Plotting the graph

        pl.plot(xaxis,yaxis)

        pl.ylabel('Latency in mins')

        pl.xlabel('Train index')

        pl.show()

    # To order food at the seat

    elif inpt == '6':

        # Getting the booking info

        if book == True:

            booking= pd.read\_csv('Data/booking.csv', sep= ':')

            booking.index= booking['Username']

            if booking.at[Username,'Food availability'] == True:

                # Getting info of user's train

                trnno= booking.at[Username,'Train no']

                food= pd.read\_csv('Data/food menu.csv')

                # Printing the menu items

                print(food)

                menuitems= int(input('Enter no of menu items you want to order: \n'))

                order= pd.DataFrame([], columns=['Menus', 'Price', 'no of orders', 'Total price'])

                # Getting the order

                for i in range(1,menuitems+1):

                    menuNo= int(input(f'Enter the index of order {i}: \n'))

                    no= int(input(f'Enter no of plates of {food.at[menuNo,"Menu"]}: \n'))

                    order.loc[len(order)]= [food.at[menuNo,'Menu'], food.at[menuNo,'Price'], no, food.at[menuNo,'Price']\*no]

                totalPrice= order['Total price'].sum()

                # Confirming the order

                print('Your Order is: ')

                print(order)

                print(f'\n\n Total bill amount is {totalPrice} rupees.')

                input('\n Press Enter to continue')

                print(f'Your order is being prepared. \nYour food will be on your seat (seat no. {booking.at[Username,"seatno"]}) before 30 mins. \nThank you! ')

                input('\n\n Press Enter to continue!')

            # If the facility is not available in user's train

            else:

                print('Currently this facility is unavailable in your train.\nWe are sorry for the inconvenience')

                input('\n\n Press Enter to continue!')

        else:

            print('\nYour first have to book a ticket')

            input('\n\n Press Enter to continue!')

    # To give a Feedback

    elif inpt == '7':

        if login== True:

            # Getting the feedback

            feedback = input('\nEnter your Feedback here: \n\n')

            # Storing the feedback in csv file

            feedbk= pd.read\_csv('Data/feedback.csv', sep='=')

            feedbk.loc[len(feedbk)]= [Username, feedback]

            feedbk.to\_csv('Data/feedback.csv', index=False, sep='=')

            print("Thank you for your valuable feedback. \n\n")

            input('\n\n Press Enter to continue!')

        else:

            print('First, you have to login')

    # To Exit

    elif inpt == '8':

        print('\n\nThank you !')

        input('\n\n\tPress Enter to exit\t')

    # Correcting user's Mistake

    else:

        input('\n\nSelect one from the serial numbers\n\n Press enter to continue')

# OUTPUT SCREEN

# 

# 

# 

# 

# 

# 

# 

# 

# 

HARDWARE AND SOFTWARE

REQUIREMENTS

HARDWARE REQUIREMENT

* Pentium 3/4/Core 2 Duo / Dual core/ i3 / i5 / i7
* With at least 256 MB RAM
* 2 MB free space on Hard Disk Color
* Monitor/LCD

OPERATING SYSTEM & COMPILER

* MS Windows
* Python with related libraries used for Data Analysis Pandas
* Numpy
* Matplotlib (Pyplot)