

Certificate



This is to certify that **Kapil Singh Negi** student of Class XII has successfully completed project on topic, **Railway Management System** under the guidance of Mr. Deepak Pithadia, during the Academic year 2023-2024, for partial fulfilment of Computer Science (083), Practical Examination conducted by AISSCE 2023-2024.

Subject: Computer Science (083)

Board Roll number:

Subject Teacher

Examiner

Date: November 9, 2023

Index

S. No.	Particulars	Page No.
1.	Acknowledgement	2
2.	Introduction	3
3.	System Requirement Hardware/ Software	4
4.	Purpose of Creating Project	5
5.	Coding	6
6.	Screen Shots	44
7.	Conclusion	63
8.	Bibliography / References	64

Acknowledgement

I would like to thank my computer teacher Mr. Deepak Pithadia for guiding me throughout this project work.

A special acknowledgement goes to our principal Ms. Sabina Sawhney who gave me the golden opportunity of this wonderful project, which also helped me in doing a lot of research and I came to know about so many new things.

I wish to thank my parents as well for their support and encouragement without which I could not have completed this project in the limited time frame.

Introduction

Railway Management System provides a way to book tickets conveniently and make the overall process easier for the customer as well as the administrator.

System Requirements

Hardware Specification:

- X86 64-bit CPU (intel/AMD architecture)
- 8 GB RAM
- 5 GB free disk space

Software:

Languages Used: Python Programming Language
Sequence Query Language

Python version (Recommended): 3.8

SQL version (Recommended): 5.5

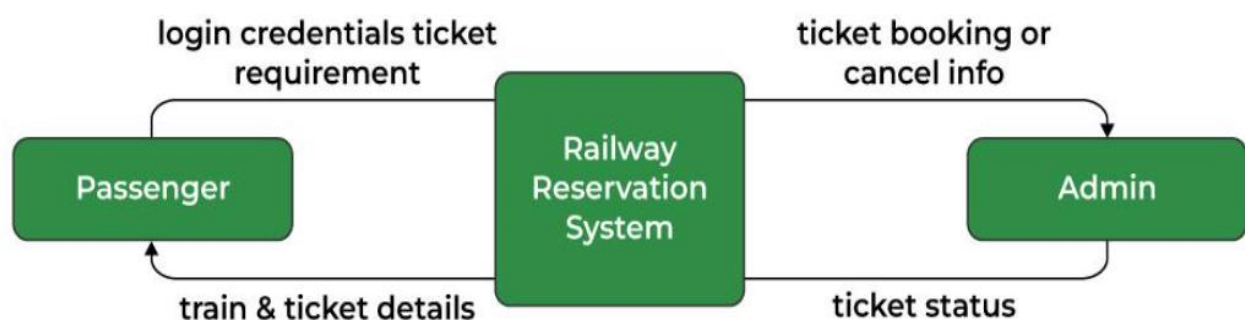
Purpose of Creating Project

The objective of the Railway Management System is to make the booking of trains more convenient and user-friendly.

It eases the process of booking tickets for everyone, from a single person travelling or a company using it for transporting goods in bulk.

It has a main administrator which has the ability to oversee all the operations that go about in the railways and make changes accordingly.

Data-Flow Diagram (DFD):



Coding

```
import mysql.connector
```

```
import datetime
```

```
def train_fig():
```

```
    return ""
```

```
        """
```

```
        """
```

```
        """
```

```
        """
```

```
        """
```

```
        """
```

```
        """
```

```
        """
```

```
        """
```

```
        """
```

```
        """
```

```
    """
```

```
def rules():
    print('\n', '\t'*3, 'RULES AND REGULATIONS TO FOLLOW WHILE
    TRAVELLING')
    print("""\n1. Please maintain decorum and cleanliness while travelling.
2. In case of a medical emergency, please follow these series of steps:
\t a. Pull the chain immediately, dont wait for the next station
\t b. Dial "919" and inform them of the situation.
\t c. Calm the patient and wait until help arrives.
3. Buying tickets from vendors/agents is a punishable offense with penalty
being 6 months
    jail/paying a fine of Rs. 5000
4. Pulling the chain unnecessarily is a punishable offense and can lead to
penalty of a
    fine of Rs. 2000 or 1 month in prison
5. Carrying contraband, alcohols, weapons and other illegal items is illegal and
can lead to
    differing sentences for prison.
6. For inconveniences/complements while travelling, call at "97XXXXXX08" and
submit your
    complaint there\n""")
```

```
date = str(datetime.datetime.now())[10:]
time = str(datetime.datetime.now())[11:19]
sqldb = mysql.connector.connect(host='localhost', user='root',
passwd='kapil2006@')
csr = sqldb.cursor()
try:
```



```

        csr.execute('use cs_project')
except mysql.connector.errors.ProgrammingError:
    csr.execute('create database cs_project')
    csr.execute('use cs_project')
while True:
    print("\nwelcome!!

what do you wish to do..?
1. check rules and regulations
2. login
3. sign up
4. about us")
    choice_1 = 0
    try:
        choice_1 = int(input('\nenter choice: '))
    except ValueError:
        print('invalid choice, you can only choose from 1 to 4')
        temp = input('press enter to go back')
        print('='*92, '\n')
        continue
    if choice_1 == 1:
        print('-' * 92)
        rules()
        temp = input('press enter to go back')
    elif choice_1 == 2:
        username = input('enter username: ')
        password = input('enter password: ')
        if username == 'Admin' and password == 'kapil2006@':

```

```

while True:
    print('-'*92, '\n\nwhat do you wish to do..?')
    print('\n1. alter train sheet\n2. alter user list\n3. server shut
down\n4. logout')
    try:
        choice = int(input('\nenter choice: '))
    except ValueError:
        print('invalid choice, you can only choose from 1 to 4')
        temp = input('press enter to go back')
        continue
    if choice == 1:
        while True:
            print('-'*92, '\n\nwhat do you wish to do..?')
            print('\n1. view train_sheet\n2. append train_sheet\n3. update
status\n4. go back')
            try:
                choice = int(input('\nenter choice: '))
            except ValueError:
                print('invalid choice, you can only choose from 1 to 4')
                temp_1 = input('press enter to go back')
                continue
            if choice == 1:
                csr.execute('select * from train_sheet')
                temp = csr.fetchall()
                k = ' '
                print('-'*92, '\n\n')
                print('train_num', ' start_point', k*12+'destination',
k*12+'departure', k*3+'arrival',
                        k*5+'train_type')

```

```

print()
for i in temp:
    print(i[0], k*6+i[1], ' '*(22 - len(i[1])), i[2], ' '*(22 - len(i[2])),
          str(i[3]))+' ', str(i[5]))+' ', i[8])
temp = input('\npress enter to go back')
elif choice == 2:
    while True:
        print('-' * 92, '\n\nplease input values carefully')
        print('if you enter an invalid value, you have to fill all values
again')

        train_details = []
        csr.execute('select city_name from cities')
        cities = csr.fetchall()
        print('\n')
        try:
            temp = int(input('enter train_number: '))
            csr.execute('select train_num from train_sheet')
            if (temp,) in csr.fetchall():
                print('invalid input, train number already exists in
database')

                temp_1 = input('press enter to go back')
                continue
            if len(str(temp)) == 5:
                train_details.append(temp)
            else:
                print('invalid input, train number can only be 5 digit')
                temp_1 = input('press enter to go back')
                continue

```

```

except ValueError:
    print('invalid train_num')
    temp_1 = input('press enter to go back')
    continue
temp = input('enter start_point: ').title().strip()
if len(temp) > 20:
    print('length of start_point should not exceed 20 letters')
    temp_1 = input('press enter to go back')
    continue
train_details.append(temp)
temp = input('enter destination: ').title().strip()
if len(temp) > 20:
    print('length of destination should not exceed 20 letters')
    temp_1 = input('press enter to go back')
    continue
if (train_details[1],) not in cities and (temp,) not in cities:
    print('entered start_point or destination not in database')
    temp_1 = input('press enter to continue')
    continue
train_details.append(temp)
temp = input('enter departure date: ')
try:
    temp_2 = datetime.date.fromisoformat(temp)
    train_details.append(temp)
except ValueError:
    print('the date entered is in wrong format use "YYYY-MM-DD"')
    temp_1 = input('press enter to continue')

```

```

        continue
temp = input('enter departure time: ')
try:
    temp_2 = datetime.time.fromisoformat(temp)
    train_details.append(temp)
except ValueError:
    print('the time entered is in wrong format use
"HH:MM:SS"')
    temp_1 = input('press enter to continue')
    continue
try:
    temp_1 = int(input('enter avg speed of train(Km/H): '))
except ValueError:
    print('invalid input, speed should only be a numerical
value')
    temp_2 = input('press enter to go back')
    continue
csr.execute(f'select * from cities where city_name =
"{train_details[1]}" or city_name = "{train_details[2]}"')
temp_2 = csr.fetchall()
temp_3 = round((((temp_2[0][2]-temp_2[1][3])**2 +
(temp_2[0][3]-temp_2[1][3])**2)**0.5)*4.9/temp_1, 2)//24
temp = datetime.datetime.strptime(train_details[3], '%Y-
%m-%d').date() + datetime.timedelta(days=temp_3)
train_details.append(str(temp))
coach_sheet = (str(train_details[0]) + '_coach_sheet_' +
                str(train_details[3].replace('-', '_')))
train_details.append(coach_sheet)
train_details.append('on_time')

```

```

        temp = input('enter type of train(passenger/freight):
    ').lower()

    if temp == 'passenger' or temp == 'freight':
        pass
    else:
        print('type is invalid')
        temp_1 = input('press enter to go back')
        continue
    train_details.append(temp)
    if train_details[8] == 'passenger':
        csr.execute(f'create table {coach_sheet} select * from
False_Passenger_Coach_Sheet')
        try:
            ac_coaches = int(input('\nenter number of ac coaches:
'))

            ac_seats = int(input('enter number of ac seats per coach:
'))

            ac_fee = int(input('enter booking fee of one ac seat: '))
            if ac_seats > 99:
                print('invalid input, too many seats')
                continue

            sleeper_coaches = int(input('\nenter number of sleeper
coaches: '))

            sleeper_seats = int(input('enter number of sleeper seats
per coach: '))

            sleeper_fee = int(input('enter booking fee of one sleeper
seat: '))

            if sleeper_seats > 99:
                print('invalid input, too many seats')
                continue

```

```

except ValueError:
    print('invalid input, you can only enter numerical values')
    temp_1 = input('press enter to go back')
    continue
k = 65
temp_2 = temp_1 = 0
if ac_coaches % 9 == 0:
    temp_3 = 0
else:
    temp_3 = 1
for i in range(k, k + ac_coaches // 9 + temp_3):
    temp = chr(i)
    for i in range(1, 10):
        csr.execute(f'insert into {coach_sheet} values("{temp}
+ str(i)}", "AC", {ac_fee}, {ac_seats}, {ac_seats})')
        sqlldb.commit()
    temp_1 += 1
    if temp_1 == ac_coaches:
        k = ord(temp) + 1
        break
if sleeper_coaches % 9 == 0:
    temp_3 = 0
else:
    temp_3 = 1
for i in range(k, k + sleeper_coaches // 9 + temp_3):
    temp = chr(i)
    for i in range(1, 10):

```

```

        csr.execute(f'insert into {coach_sheet} values("{temp
+ str(i)}", "SLEEPER",{sleeper_fee},{sleeper_seats}, {sleeper_seats})')
        sqlldb.commit()
        temp_2 += 1
        if temp_2 == sleeper_coaches:
            break
    if train_details[8] == 'freight':
        csr.execute(f'create table {coach_sheet} select * from
false_freight_coach_sheet')
        try:
            wagon_nums = int(input('enter number of wagons: '))
            if wagon_nums > 99:
                print('invalid input, number of wagons can only be
chosen from 1-99')
                continue
        except ValueError:
            print('invalid input, number of wagons can only be
chosen from 1-99')
            continue
    k = 65
    temp_1 = 0
    if wagon_nums % 9 == 0:
        temp_2 = 0
    else:
        temp_2 = 1
    for i in range(k, k + wagon_nums // 9 + temp_2):
        temp = chr(i)
        for i in range(1, 10):

```



```

        csr.execute(f'insert into {coach_sheet} values("{temp
+ str(i)}",NULL, "VACANT", NULL)')
        sqlldb.commit()
        temp_1 += 1
        if temp_1 == wagon_nums:
            break
while True:
    print('-'*92)
    print("\nhere is the listing of entered data, please
recheck")

    print('\ntrain_num:  ', train_details[0])
    print('start_point:  ', train_details[1])
    print('destination:  ', train_details[2])
    print('departure_date: ', train_details[3])
    print('departure_time: ', train_details[4])
    print('arrival_date:  ', train_details[5])
    print('coach_sheet:  ', train_details[6])
    print('status:      ', train_details[7])
    print('train_type:   ', train_details[8])
    if train_details[8] == 'passenger':
        print('\nnumber of ac coaches: ', ac_coaches, ',
\t\tnumber of seats per coach: ', ac_seats)
        print('number of sleeper coaches:', sleeper_coaches,
',\t\tnumber of seats per coach: ', sleeper_seats)
        print('booking fee of one ac_seat:   ', ac_fee)
        print('booking fee for one sleeper seat: ', sleeper_fee)
    elif train_details[8] == 'freight':
        print('number of empty wagons: ', wagon_nums)

```

```

temp = input('\ndo you wish to continue with the following
data(y/n): ')

if temp == 'y':
    print('-'*92, '\n\n---TRAIN DETAILS HAVE BEEN ADDED---
')

    temp_2 = input('\npress enter to go back')
    k = 0
    break
elif temp == 'n':
    k = 1
    try:
        csr.execute(f'drop table {coach_sheet}')
        sqlldb.commit()
    except mysql.connector.errors.ProgrammingError:
        pass
    temp_3 = input('do you wish to (1)re-enter data or
(2)exit: ')

    if temp_3 == '1':
        pass
    elif temp_3 == '2':
        k = 2
    else:
        print('invalid input you can only select from 1 or 2')
        temp_2 = input('press enter to go back')
        continue
    break
else:
    print('invalid choice, you can only select from "y" or "n"')
    temp_2 = input('press enter to go back')

```

```

        if k == 1:
            continue
        elif k == 2:
            break

        csr.execute('insert into train_sheet
values(%s,"%s","%s","%s","%s","%s","%s","%s")' % tuple(train_details))

        sqldb.commit()

        break
    elif choice == 3:
        csr.execute('select train_num from train_sheet')
        train_nums = csr.fetchall()
        try:
            print('-'*92)
            train_num = int(input('\nenter train number: '))
            if (train_num,) not in train_nums:
                print('the entered train number does not exist')
                temp = input('press enter to go back')
                continue
        except ValueError:
            print('the entered train number does not exist')
            temp = input('press enter to go back')
            continue

        csr.execute(f'select * from train_sheet where train_num =
"{train_num}"')

        temp = csr.fetchone()

        while True:
            print('\nhere are the detailed info on train number',
train_num)

            print('\nstart point: ', temp[1])

```

```

print('destination: ', temp[2])
print('departure date: ', temp[3])
print('arrival date: ', temp[5])
print('current status: ', temp[7])
print('\nwhat do you wish to do..?')
print('1. set status = "on_time"\n2. set status = "delayed"\n3.
set status = "complete"\n4. go back')
try:
    temp_2 = int(input('\nenter choice: '))
    status = "
except ValueError:
    print('invalid choice, you can only select from 1 to 4')
    temp_3 = input('press enter to go back')
    print('-'*92)
    continue
if temp_2 == 1:
    csr.execute(f'update train_sheet set status = "on_time"
where train_num = {train_num}')
    sqlldb.commit()
    status = 'on_time'
    print('\n\t\t---STATUS HAS BEEN SET TO ON_TIME---\n')
    break
elif temp_2 == 2:
    csr.execute(f'update train_sheet set status = "delayed"
where train_num = {train_num}')
    sqlldb.commit()
    status = 'delayed'
    print('\n\t\t---STATUS HAS BEEN SET TO DELAYED---\n')
    break

```

```

        elif temp_2 == 3:
            csr.execute(f'update train_sheet set status = "complete"
where train_num = {train_num}')
            sqlldb.commit()
            status = 'complete'
            print('\n\t\t---STATUS HAS BEEN SET TO COMPLETE---\n')
            break
        elif temp_2 == 4:
            break
        else:
            print('invalid choice, you can only select from 1 to 4')
            temp_3 = input('press enter to go back')
            print('-'*92)
            csr.execute('select user_booking from user_list where account
= "business"')
            temp = csr.fetchall()
            for i in temp:
                csr.execute(f'update {i[0]} set journey_status = "{status}"
where train_number = {train_num}')
                sqlldb.commit()
                temp_3 = input('press enter to go back')
            elif choice == 4:
                break
            else:
                print('invalid choice, you can only choose from 1 to 4')
                temp_1 = input('press enter to go back')
        elif choice == 2:
            print('-'*92)
            while True:

```

```

print('\nwhat do you wish to do..?')
print('\n1. view user list\n2. update sus value\n3. go back')
try:
    choice_1 = int(input('\nenter choice: '))
except ValueError:
    print('invalid choice, you can only choose from 1 to 3')
    choice_1 = input('press enter to go back')
    print('-' * 92)
    continue
if choice_1 == 1:
    csr.execute('select * from user_list')
    temp = csr.fetchall()
    k = ''
    print('-'*92)
    print('\nuser id', k*6, 'username', k*16, 'aadhar number', k*3,
'age', k*3, 'sus value', k*4, 'account\n')
    for i in temp:
        print(i[0], k*4, i[1], k*(24-len(i[1])), i[3], k*4,
datetime.datetime.now().year - int(str(i[4])[:4]), k*5, i[6], k*12, i[7])
        temp_1 = input('\npress enter to go back')
elif choice_1 == 2:
    csr.execute('select user_id from user_list')
    user_ids = csr.fetchall()
    try:
        print('-' * 92)
        user_id = int(input('\nenter user_id: '))
        if (user_id,) not in user_ids:
            print('the entered user_id does not exist')

```

```

        temp = input('press enter to go back')
        continue
except ValueError:
    print('the entered user_id does not exist')
    temp = input('press enter to go back')
    continue
csr.execute(f'select * from user_list where user_id =
"{user_id}"')
temp = csr.fetchone()
while True:
    print('\nhere are the detailed info on user_id', user_id)
    print('\nusername:    ', temp[1])
    print('password:    ', temp[2])
    print('aadhar number: ', temp[3])
    print('date of birth: ', temp[4])
    print('sus value:    ', temp[6])
    print('account type: ', temp[7])
    if temp[7] == 'business':
        print('company name: ', temp[8])
    temp_1 = input('\ndo you wish to reset the sus value(y/n): ')
    if temp_1 == 'y':
        csr.execute(f'update user_list set sus_val = 0 where
user_id = "{user_id}"')
        sqlldb.commit()
        print('\n\t\t---SUS VALUE HAS BEEN SET TO ZERO---\n')
        break
    elif temp_1 == 'n':
        break

```

```

        else:
            print('invalid input, you can only choose from "y" or "n"')
            temp_3 = input('press enter to go back')
            print('-'*92)
            temp_3 = input('press enter to go back')
    elif choice_1 == 3:
        break
    else:
        print('invalid choice, you can only choose from 1 to 3')
        temp = input('press enter to go back')
        print('-'*92)
elif choice == 3:
    print('-' * 92)
    temp = input('\nenter "CONFIRM" if you really want to shut down
the server: ')
    if temp != "CONFIRM":
        print('\nas you have written', temp, 'instead of "CONFIRM",')
        print('the server shut down has been aborted')
        temp = input('\npress enter to go back')
    else:
        print('\n\n', '\t'*4, '---SERVER HAS BEEN SHUT DOWN---\n')
        print('*'*92)
        csr.close()
        sqlldb.disconnect()
        exit()
elif choice == 4:
    print('-'*92, '\n\n\t\t\t\t\t--you have been logged out--')
    temp = input('\npress enter to go back')

```



```

        print('='*92)
        break
    else:
        print('invalid choice, you can only choose from 1 to 4')
        temp = input('press enter to go back')
    continue
csr.execute('select * from user_list')
user_list = csr.fetchall()
csr.execute('select username from user_list')
username_list = csr.fetchall()
if (username,) in username_list:
    i = user_list[username_list.index((username,))]
    user_id = i[0]
    if i[6] < 4:
        if password == i[2] and i[7] == 'Personal':
            csr.execute('update user_list set sus_val = 0 where user_id = %s' %
(user_id,))
            sqldb.commit()
            while True:
                print('='*92, '\n\nwelcome, ' + username, '\n\n1. check profile')
                csr.execute(f'select user_booking from user_list where user_id =
"{user_id}"')
                if csr.fetchone()[0] == None:
                    print('2. book new seat')
                    temp = 0
                else:
                    print('2. view ticket details')
                    temp = 1

```

```

print('3. log out')
try:
    choice_2 = int(input('\nenter choice: '))
    print('-'*92)
except ValueError:
    print('invalid choice, you can only choose from 1 to 3')
    temp = input('press enter to go back')
    continue
if choice_2 == 1:
    csr.execute('select * from user_list where user_id = "' +
str(user_id) + '"')
    i = csr.fetchone()
    print('\nuser id : ', i[0])
    print('username : ', i[1])
    print('aadhar number:', i[3])
    print('age : ', datetime.datetime.now().year - int(str(i[4])[4:]))
    temp = input('\npress enter to go back')
elif choice_2 == 2 and temp == 0:
    print("\nhere are the listings of the train stations all over
India:\n")

    csr.execute('select city_name from cities')
    temp = csr.fetchall()
    temp_1 = temp_2 = None
    for i in range(0, len(temp), 3):
        try:
            temp_1 = 25 - len(temp[i][0])
            print(temp[i][0], end=' '*temp_1)
            temp_2 = 25 - len(temp[i + 1][0])

```

```

        print(temp[i + 1][0], end=' ' * temp_2)
        print(temp[i+2][0], end='\n')
    except IndexError:
        continue
    start_point = input('\n\nenter name of start point (or exit):
').title().strip()
    if start_point == 'Exit':
        temp = input('press enter to go back')
        continue
    destination = input('enter name of destination: ').title().strip()
    csr.execute('select city_name from cities')
    temp = csr.fetchall()
    if (start_point,) in temp and (destination,) in temp:
        csr.execute('select * from train_sheet where start_point =
"%s" and destination = "%s" and departure_date > "%s" and train_type
="passenger"' % (start_point, destination, date))
        temp = csr.fetchall()
        train_nums = []
        while True:
            if not temp == []:
                print('-'*92, '\n\nfollowing trains are available for
travel:\n')
                for i in temp:
                    csr.execute('select sum(seats_available) from %s
where coach_type = "ac"' % (i[6]))
                    ac_seats = csr.fetchone()[0]
                    if ac_seats == None:
                        ac_seats = 0
                    csr.execute('select sum(seats_available) from %s
where coach_type = "sleeper"' % (i[6]))

```

```

        sleeper_seats = csr.fetchone()[0]
        if sleeper_seats == None:
            sleeper_seats = 0
        if ac_seats + sleeper_seats != 0:
            print('train number -', i[0])
            print('\tdate of departure:    ', i[3])
            print('\ttime of departure:    ', i[4])
            print('\testimated date of arrival: ', i[5])
            print('\tavailable ac seats:    ', ac_seats)
            print('\tavailable sleeper seats: ', sleeper_seats,
'\n')

            train_nums.append(i[0])
        try:
            booking_train = input('enter train number to proceed
(or exit): ').lower().strip()
            if booking_train == 'exit':
                temp = input('press enter to go back')
                break
            booking_train = int(booking_train)
        except ValueError:
            print('the entered train number is invalid')
            temp_1 = input('press enter to go back')
            continue
        if booking_train in train_nums:
            csr.execute('select coach_sheet from train_sheet
where train_num = %s' %(booking_train,))
            coach_sheet = csr.fetchone()[0]
            while True:

```

```

!= 0' % coach_sheet)

coaches:\n')

csr.execute('select * from %s where seats_available

print('-'*92, '\n\nyou can select from the following

temp_1 = 1
coach_nums = []
for i in csr.fetchall():
    coach_nums.append(i[0])
    print(str(temp_1)+' coach number -', i[0])
    print('\tcoach type:      ', i[1])
    print('\tbooking fee(one seat): ', i[2])
    print('\tseats available    ', i[3], '\n')
    temp_1 += 1
temp = input('\nenter the coach number to book a
seat: ').upper()

if temp not in coach_nums:
    print('entered coach number is invalid')
    temp = input('press enter to go back')
    continue

csr.execute(f'select coach_fee from {coach_sheet}
where coach_num = "{temp}"')

print('-'*92, '\n\nplease deposit Rs.',
csr.fetchone()[0], 'into the counter')

temp_1 = input('\n\npress enter to continue')
print('-'*92, '\n')
train_fig()
print('\t*7, '---TICKET BOOKING WAS SUCCESSFUL---

\n\n')

temp_1 = input('press enter to go back')

```

```

        csr.execute(f'update {coach_sheet} set
seats_available = seats_available - 1 where coach_num = "{temp}"')

        csr.execute(f'update user_list set user_booking =
"{coach_sheet+"_" +str(temp)}" where user_id = "{user_id}"')

        sqlldb.commit()

        break

    break

else:

    print('the entered train number is invalid')

    temp_1 = input('press enter to go back')

else:

    print('\nApologies.. it seems that no trains are up for the
journey')

    temp = input('press enter to go back')

    break

else:

    print('\nneither start point or destination is chosen
incorrectly')

    temp = input('press enter to go back')

elif choice_2 == 2 and temp == 1:

    csr.execute(f'select * from user_list where user_id =
"{user_id}"')

    booking_code = csr.fetchone()[5]

    csr.execute(f'select * from {booking_code[:-3]} where
coach_num = "{booking_code[-2:]}"')

    coach_details = csr.fetchone()

    csr.execute(f'select * from train_sheet where train_num =
"{booking_code[:5]}"')

    train_details = csr.fetchone()

    while True:

```

```

print('\nTICKET DETAILS -\n')
print('train number:   ', booking_code[:5])
print('date of departure: ', train_details[3])
print('time of departure: ', train_details[4])
print('coach number:    ', booking_code[-2:])
print('train status:    ', train_details[7])
print('\nwhat do you wish to do..?')
print('1. download ticket\n2. cancel booking\n3. go back')
try:
    choice_3 = int(input('\nenter choice: '))
except ValueError:
    print('invalid choice, you can only choose from 1 to 3')
    continue
if choice_3 == 1:
    print()
    path = input(r'enter path address to save ticket file:
')+'\train_ticket_'+booking_code+'.txt'
    try:
        f = open(path, 'w', encoding='UTF-8')
        f.write(('t'*7)+'---TRAIN TICKET---'+train_fig())
        f.write('\ntrain num: '+str(train_details[0]))
        f.write('\nstart point: '+str(train_details[1]))
        f.write('\ndestination: '+str(train_details[2]))
        f.write('\ndeparture date : '+str(train_details[3]))
        f.write('\narrival date: '+str(train_details[5]))
        f.write('\ndeparture time: '+str(train_details[4]))
        f.write('\ntrain status: '+str(train_details[7]))
        f.write('\n\nbooking code: '+booking_code)

```

```

        f.write('\nprint date:
'+str(datetime.datetime.now())[0:10])
        f.close()
    except FileNotFoundError or PermissionError:
        print('\nthere occurred a problem in saving the file,')
        print('please recheck the path address and try again')
        temp = input('press enter to go back')
        print('-'*92)
        continue
    print('-'*92, '\n\n\ticket has been saved as', path)
    temp = input('press enter to go back')
    break
elif choice_3 == 2:
    print('-'*92, '\n\nAre you sure you want to cancel the
ticket,')

    print('you would have to deposit 15% of the booking fee as
cancellation fee')

    temp = input('enter "CONFIRM" to cancel booking: ')
    if temp != 'CONFIRM':
        print('\nas you have written', temp, 'instead of
"CONFIRM"')

        print('the ticket cancellation will not take place')
        temp = input('press enter to go back')
        break

    csr.execute(f'select coach_fee from {booking_code[:-3]}
where coach_num = "{booking_code[-2:]}"')

    print('-'*92, '\n\nplease deposit Rs.', csr.fetchone()[0] *
0.15, 'into the counter')

    temp = input('\n\n\npress enter to continue')

```



```

        print('-'*92, '\n\n---YOUR TICKET HAS SUCCESSFULLY BEEN
CANCELLED---')

        csr.execute(f'update user_list set user_booking = NULL
where user_id = {user_id}')

        csr.execute(f'update {booking_code[:-3]} set
seats_available = seats_available + 1 where coach_num = "{booking_code[-
2:]}"')

        sqldb.commit()

        temp = input('\n\n\npress enter to go back')
        break
    elif choice_3 == 3:
        break
    else:
        print('invalid choice, you can only choose from 1 to 3')
        temp = input('press enter to go back')

elif choice_2 == 3:
    print('\n\t\t\t\t--you have been logged out--')
    temp = input('\npress enter to go back')
    break
else:
    print('invalid choice, you can only choose from 1 to 3')
    temp = input('press enter to go back')

elif password == i[2] and i[7] == 'Business':
    csr.execute('update user_list set sus_val = 0 where user_id = "%s"'
% (user_id,))
    sqldb.commit()
    while True:
        print('-'*92, '\n\nwelcome,', username)
        print('\n1. check profile\n2. book coaches\n3. check bookings\n4.
log out')

```

```

try:
    choice_4 = int(input('\nenter choice: '))
except ValueError:
    print('invalid choice, you can only choose from 1 to 4')
    temp = input('press enter to go back')
    continue
csr.execute('select * from user_list where user_id = "' +
str(user_id) + '"')
user_details = csr.fetchone()
if choice_4 == 1:
    print('-'*92, '\n\nuser id :      ', user_details[0])
    print('username :      ', user_details[1])
    print('aadhar number:    ', user_details[3])
    print('age :            ', datetime.datetime.now().year -
int(str(user_details[4])[:4]))
    print('company affiliated: ', user_details[8])
    temp = input('\npress enter to go back')
elif choice_4 == 2:
    print('-'*92, "\n\nhere are the listings of the train stations all over
India:\n")
    csr.execute('select city_name from cities')
    temp = csr.fetchall()
    temp_1 = temp_2 = None
    for i in range(0, len(temp), 3):
        try:
            temp_1 = 25 - len(temp[i][0])
            print(temp[i][0], end=' ' * temp_1)
            temp_2 = 25 - len(temp[i + 1][0])
            print(temp[i + 1][0], end=' ' * temp_2)

```

```

        print(temp[i + 2][0], end='\n')
    except IndexError:
        continue

start_point = input('\n\nenter name of start point (or exit):
').title().strip()

if start_point == 'Exit':
    temp = input('press enter to go back')
    continue

destination = input('enter name of destination: ').title().strip()
csr.execute('select city_name from cities')
temp = csr.fetchall()

if (start_point,) in temp and (destination,) in temp:
    csr.execute('select * from train_sheet where start_point =
"%s" and destination = "%s" and departure_date > "%s" and train_type =
"freight" % (start_point, destination, date))

    temp = csr.fetchall()
    train_nums = []
    while True:
        False_train_nums = []
        if not temp == []:
            print('-' * 92, '\n\nfollowing trains are available for
travel:\n')

            for i in temp:
                csr.execute(f'select count(*) from {i[6]} where status =
"VACANT"')

                coc_avail = csr.fetchone()[0]
                print('train number -', i[0])
                print('\tdate of departure:      ', i[3])
                print('\ttime of departure:      ', i[4])

```

```

        print('\testimated date of arrival: ', i[5])
    if coc_avail != 0:
        print('\tcoaches available:      ', coc_avail, '\n')
        train_nums.append(i[0])
    else:
        print('\tcoaches available:      NONE\n')
        False_train_nums.append(i[0])
    try:
        booking_train = int(input('enter train number to
proceed: '))

    except ValueError:
        print('the entered train number is invalid')
        temp_1 = input('press enter to go back')
        continue
    if booking_train in train_nums:
        csr.execute('select coach_sheet from train_sheet
where train_num = %s' % (booking_train,))
        coach_sheet = csr.fetchone()[0]
        csr.execute(f'select count(*) from {coach_sheet}
where Status = "VACANT"')
        vacant_coaches = csr.fetchone()[0]
        while True:
            csr.execute('select * from coach_types')
            temp = csr.fetchall()
            print('-'*92, '\n\nhere are the types of coaches you
can choose from:\n')

            try:
                for i in temp:

```

```

coach =',
                                print(str(i[0])+'.', i[1], ' '*(25-len(i[1])), 'fee per
                                i[2], ' '*(11 - len(str(i[2]))))
                                type_coach = int(input('\nselect the type of coach
you wish to book: '))
                                if type_coach > i[0]:
                                    print('invalid input, type of coach coach can
only be selected from 1-'+str(i[0]))
                                    temp_1 = input('press enter to go back')
                                    continue
                                num_coach = int(input('enter the number of
coaches you wish to book: '))
                                if num_coach > coc_avail:
                                    print('invalid input, no.of available coaches are
only', coc_avail)
                                    temp_1 = input('press enter to go back')
                                    continue
                                csr.execute(f'select * from coach_types where
sr_no = {type_coach}')
                                temp = csr.fetchone()
                                print('-' * 92, '\n\nplease deposit Rs.',
temp[2]*num_coach, 'into the counter')
                                temp_1 = input('\n\npress enter to continue')
                                print('-' * 92, '\n\n---YOUR BOOKING WAS
SUCCESSFUL---')
                                csr.execute(f'select coach_num from
{coach_sheet} where status = "VACANT"')
                                coach_nums = csr.fetchall()
                                for i in range(num_coach):

```

```

        csr.execute(f'update {coach_sheet} set
coach_type = "{temp[1]}", status = "BOOKED", booked_for = "{user_details[8]}"
where coach_num = "{coach_nums[i][0]}"')

        sqldb.commit()

        csr.execute(f'select status from train_sheet where
train_num = {booking_train}')

        temp_1 = csr.fetchone()[0]

        csr.execute(f'insert into {user_details[5]}
values({booking_train},{num_coach},"{temp[1]}", "{temp_1}")')

        sqldb.commit()

        while True:

            temp = input('\n\ndo you wish to
continue(y/n): ').lower()

            if temp == 'y':

                break

            elif temp == 'n':

                temp = 0

                break

            else:

                print('invalid choice, you can only choose
from "y" or "n"')

                temp = input('press enter to go back')

                print('-'*92)

            if not temp:

                break

        except ValueError or IndexError:

            print('invalid input, type of coach coach can only
be selected from 1-'+str(i[0]))

            temp_1 = input('press enter to go back')

            continue

```

```

        break
    elif booking_train in False_train_nums:
        print('it seems that there are no coaches available in
the train')

        if train_nums:
            temp_1 = input('press enter to go back')
            continue
        else:
            temp_1 = input('press enter to go to home page')
            break
    else:
        print('the entered train number is invalid')
        temp_1 = input('press enter to go back')
    else:
        print('\nApologies.. it seems that no trains are up for the
journey')

        temp = input('press enter to go back')
        break
    else:
        print('\neither start point or destination is chosen
incorrectly')

        temp = input('press enter to go back')
    elif choice_4 == 3:
        csr.execute(f'select train_number from {user_details[5]} group
by train_number ')
        train_nums = csr.fetchall()
        temp = 1
        if train_nums:
            while True:

```

```

trains-')
print('-' * 92, '\n\nthe bookings were made in the following

for i in train_nums:
    csr.execute(f'select * from train_sheet where train_num

= {i[0]}')

    temp_1 = csr.fetchone()
    print('\n', str(temp)+'.', i[0])
    print('\tstart point:  ', temp_1[1])
    print('\tdestination:  ', temp_1[2])
    print('\tarrival date:  ', temp_1[5])
    print('\tjourney status: ', temp_1[7])
    temp+= 1

try:
    temp_1 = int(input('\nenter train_number to proceed: '))
except ValueError:
    print('invalid input, entered train_number is incorrect')
    temp_3 = input('press enter to go back')
    continue

if (temp_1,) not in train_nums:
    print('invalid input, entered train_number is incorrect')
    temp_3 = input('press enter to go back')
    continue

csr.execute(f'select coach_sheet from train_sheet where
train_num = {temp_1}')

temp_2 = csr.fetchone()[0]

csr.execute(f'select coach_type, count(*) from {temp_2}
where booked_for = "{user_details[8]}" group by coach_type')

temp_2 = csr.fetchall()

print('-'*92, '\n')

```



```

        for i in temp_2:
            print('coach_type: ', i[0], ' '*(25-len(i[0])), 'coaches
booked: ', i[1])

            temp_3 = input('\npress enter to go back')
            break
        else:
            print('-'*92, '\n\n\nit seems that there are no bookings yet..')
            temp = input('\npress enter to go back')
        elif choice_4 == 4:
            print('-'*92, '\n\n', '\t'*4, '---you have been logged out---')
            temp = input('\npress enter to go back')
            break
        else:
            print('invalid choice, you can only choose from 1 to 4')
            temp = input('press enter to go back')

    else:
        csr.execute('update user_list set sus_val = sus_val + 1 where
user_id = '+str(i[0]))
        sqlldb.commit()
        print('either username or password is incorrect')
        temp = input('press enter to go back')

    else:
        print('\nthere have been numerous suspicious attempts to log in this
account')

        print('hence, you will not be able to log in to this account for a while')
        print('\nplease contact the server manager to resolve this issue')
        print('contact info:\n\temail- xyz@gmail.com\n\tp.h.no.- 915XX
XXXX4')

    else:

```

```

        print('either username or password is incorrect')
        temp = input('press enter to go back')
elif choice_1 == 3:
    while True:
        print('-'*92)
        username = input('\nenter username: ')
        csr.execute('select username from user_list')
        username_list = csr.fetchall()
        if not username.isalnum():
            print('\nusername can not have any special characters or spaces')
            temp = input('press enter to go back')
            continue
        if (username,) in username_list:
            print('\nusername is already taken, please choose another username')
            temp = input('press enter to go back')
            continue
        password = input('enter password: ')
        temp = input('enter aadhaar number: ')
        if temp.isdigit() and len(temp) == 12:
            aadhar_num = temp
        else:
            print('\naadhar number can not have spaces and should contain only
12 digits')
            temp = input('press enter to go back')
            continue
        account = input('enter account type(personal/business): ').title().strip()
        csr.execute('select max(user_id) from user_list')
        user_id = csr.fetchone()[0] + 1

```

```

if account not in ('Personal', 'Business'):
    print('\naccount can only be Personal or Business rest are invalid')
    temp = input('press enter to go back')
    continue
if account == 'Business':
    company_name = input('enter name of (company/brand): ')
    csr.execute('select company_name from user_list where account =
"business"')
    temp = csr.fetchall()
    if (company_name,) in temp:
        print('there is already an account made for', company_name)
        print('you can only make one account for a particular
company/brand')
        temp = input('press enter to go back')
        continue
    if len(company_name) > 25:
        print('name of company is too long please abbreviate it')
        temp = input('press enter to go back')
        break
    csr.execute(f'create table {str(user_id)+"_bookings"} select * from
false_business_booking_sheet')
    DOB = input('enter DOB: ')
    try:
        if account == 'Business':
            csr.execute(f'insert into user_list
values({user_id},"{username}", "{password}", "{aadhar_num}", "{DOB}", "{str(use
r_id)+"_bookings"}",{0},"{account}", "{company_name}")')
        else:

```

```

        csr.execute(f'insert into user_list
values({user_id},"{username}","{password}","{aadhar_num}","{DOB}",NULL,0,"
{account}",NULL)')

        print('\n---YOUR ACCOUNT HAS BEEN CREATED---')

    except mysql.connector.errors.DataError:

        print('\nDOB is to be written YYYY-MM-DD format')

        continue

    sqldb.commit()

    temp = input('press enter to go back')

    break

elif choice_1 == 4:

    print('-'*92)

    print("""\n\nwe hope to create software that makes the customer
experience easier and more satisfying.

        We prioritize customer satisfaction over anything else.""")

    temp = input('\n\npress enter to go back')

else:

    print('invalid choice, you can only choose from 1 to 4')

    temp = input('press enter to go back')

print('=' * 92, '\n')

```

Screen Shots

MySQL Screen Shots

(before running the program)

1. Tables in database

```
mysql> show tables;
+-----+
| Tables_in_cs_project |
+-----+
| 100000004_bookings   |
| 10001_coach_sheet_2023_11_11 |
| 10002_coach_sheet_2023_12_10 |
| 10003_coach_sheet_2023_12_24 |
| 10004_coach_sheet_2024_01_02 |
| cities               |
| coach_types          |
| false_business_booking_sheet |
| false_freight_coach_sheet  |
| false_passenger_coach_sheet |
| train_sheet          |
| user_list            |
+-----+
12 rows in set (0.00 sec)
```

2. Table for user details

```
mysql> desc user_list;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| user_id    | int(10)   | NO   | PRI | NULL    |       |
| username   | varchar(20) | YES  |     | NULL    |       |
| password   | varchar(20) | YES  |     | NULL    |       |
| aadhar_num | varchar(12) | YES  |     | NULL    |       |
| DOB        | date      | YES  |     | NULL    |       |
| user_booking | varchar(31) | YES  |     | NULL    |       |
| sus_val    | int(1)    | YES  |     | NULL    |       |
| account    | varchar(8) | YES  |     | NULL    |       |
| company_name | varchar(25) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.02 sec)

mysql> select * from user_list;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| user_id | username   | password   | aadhar_num | DOB        | user_booking | sus_val | account | company_name |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 100000001 | Kapil Negi | 123       | 123456789012 | 2006-09-15 | NULL        | 0       | Personal | NULL        |
| 100000002 | Aditya Rathore | rainme@123 | 123456789012 | 2005-09-28 | NULL        | 0       | Personal | NULL        |
| 100000003 | Renesh Damre | me@123    | 123456789012 | 2006-09-15 | NULL        | 0       | Personal | NULL        |
| 100000004 | sanskar    | book      | 989087666781 | 2006-08-11 | 100000004_bookings | 0       | Business | adani        |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

3. Empty table for user_booking (business)

```
mysql> desc false_business_booking_sheet;
```

Field	Type	Null	Key	Default	Extra
train_number	smallint(6)	YES		NULL	
number_of_coaches	int(2)	YES		NULL	
type_of_coaches	varchar(25)	YES		NULL	
journey_status	varchar(8)	YES		NULL	

```
4 rows in set (0.02 sec)
```

```
mysql> select * from false_business_booking_sheet;
```

```
Empty set (0.00 sec)
```

4. Table for train details

```
mysql> desc train_sheet;
```

Field	Type	Null	Key	Default	Extra
train_num	smallint(6)	YES		NULL	
start_point	varchar(20)	YES		NULL	
destination	varchar(20)	YES		NULL	
departure_date	date	YES		NULL	
departure_time	time	YES		NULL	
arrival_date	date	YES		NULL	
coach_sheet	varchar(30)	YES		NULL	
status	varchar(8)	YES		NULL	
train_type	varchar(9)	YES		NULL	

```
9 rows in set (0.01 sec)
```

```
mysql> select * from train_sheet;
```

train_num	start_point	destination	departure_date	departure_time	arrival_date	coach_sheet	status	train_type
10001	Dehradun	Gandhinagar	2023-11-11	18:00:00	2023-11-11	10001_coach_sheet_2023_11_11	on_time	passenger
10002	Ranchi	Aizawl	2023-12-10	12:00:00	2023-12-11	10002_coach_sheet_2023_12_10	on_time	freight
10003	Thiruvananthapuram	Bengaluru	2023-12-24	06:15:00	2023-12-26	10003_coach_sheet_2023_12_24	on_time	passenger
10004	Ranchi	Aizawl	2024-01-02	23:45:00	2024-01-02	10004_coach_sheet_2024_01_02	on_time	freight

```
4 rows in set (0.00 sec)
```

5. Empty table for coach_sheet (passenger)

```
mysql> desc false_passenger_coach_sheet;
```

Field	Type	Null	Key	Default	Extra
coach_num	varchar(2)	NO	PRI	NULL	
coach_type	varchar(7)	YES		NULL	
coach_fee	smallint(6)	YES		NULL	
seats_available	int(2)	YES		NULL	
total_seats	int(2)	YES		NULL	

```
5 rows in set (0.01 sec)
```

```
mysql> select * from false_passenger_coach_sheet;
```

```
Empty set (0.00 sec)
```

6. Filled table for coach_sheet (10001- passenger)

```
mysql> select * from 10001_coach_sheet_2023_11_11;
```

coach_num	coach_type	coach_fee	seats_available	total_seats
A1	AC	6000	40	40
A2	AC	6000	40	40
A3	AC	6000	40	40
A4	AC	6000	40	40
A5	AC	6000	40	40
A6	AC	6000	40	40
A7	AC	6000	40	40
A8	AC	6000	40	40
A9	AC	6000	40	40
B1	AC	6000	40	40
B2	AC	6000	40	40
B3	AC	6000	40	40
C1	SLEEPER	5000	60	60
C2	SLEEPER	5000	60	60
C3	SLEEPER	5000	60	60
C4	SLEEPER	5000	60	60
C5	SLEEPER	5000	60	60
C6	SLEEPER	5000	60	60
C7	SLEEPER	5000	60	60
C8	SLEEPER	5000	60	60
C9	SLEEPER	5000	60	60

```
21 rows in set (0.00 sec)
```

7. Empty table for coach_sheet (freight)

```
mysql> desc false_freight_coach_sheet;
```

Field	Type	Null	Key	Default	Extra
coach_num	varchar(2)	NO	PRI		
coach_type	varchar(20)	YES		NULL	
status	varchar(6)	YES		NULL	
booked_for	varchar(20)	YES		NULL	

```
4 rows in set (0.01 sec)

mysql> select * from false_freight_coach_sheet;
Empty set (0.00 sec)
```

8. Filled table for coach_sheet (10002- freight)

```
mysql> select * from 10002_coach_sheet_2023_12_10;
```

coach_num	coach_type	status	booked_for
A1	FLAT CAR	BOOKED	adani
A2	FLAT CAR	BOOKED	adani
A3	FLAT CAR	BOOKED	adani
A4	FLAT CAR	BOOKED	adani
A5	FLAT CAR	BOOKED	adani
A6	BOX CAR	BOOKED	adani
A7	BOX CAR	BOOKED	adani
A8	BOX CAR	BOOKED	adani
A9	BOX CAR	BOOKED	adani
B1	BOX CAR	BOOKED	adani
B2	BOX CAR	BOOKED	adani
B3	NULL	VACANT	NULL
B4	NULL	VACANT	NULL
B5	NULL	VACANT	NULL
B6	NULL	VACANT	NULL
B7	NULL	VACANT	NULL
B8	NULL	VACANT	NULL

```
17 rows in set (0.00 sec)
```


9. Table for different train stations

```
mysql> desc cities;
```

Field	Type	Null	Key	Default	Extra
sr_num	int(2)	NO	PRI	NULL	
city_name	varchar(20)	YES		NULL	
x_coordinate	int(4)	YES		NULL	
y_coordinate	int(4)	YES		NULL	

```
4 rows in set (0.01 sec)
```



```
mysql> select * from cities;
```

sr_num	city_name	x_coordinate	y_coordinate
1	Srinagar	153	80
2	Jammu	154	111
3	Shimla	196	151
4	Chandigarh	189	160
5	Dehradun	211	170
6	New Delhi	199	206
7	Gangtok	425	233
8	Itanagar	531	229
9	Jaipur	167	246
10	Lucknow	271	251
11	Dispur	497	253
12	Kohima	544	261
13	Shillong	499	265
14	Patna	359	277
15	Manipur	542	280
16	Agartala	491	309
17	Aizawl	521	309
18	Gandhinagar	96	327
19	Bhopal	198	329
20	Ranchi	365	327
21	Kolkata	428	342
22	Chhattisgarh	287	379
23	Bhubaneswar	381	396
24	Mumbai	98	428
25	Hyderabad	220	466
26	Panaji	117	510
27	Begaluru	198	568
28	Amaravati	256	489
29	Chennai	259	567
30	Puducherry	247	593
31	Thiruvananthapuram	180	673
32	Dadra & Nagar Haveli	96	394

```
32 rows in set (0.00 sec)
```

10. Table for different type of coaches

```
mysql> desc coach_types;
```

Field	Type	Null	Key	Default	Extra
sr_no	int(2)	NO	PRI	NULL	
coach_types	varchar(20)	YES		NULL	
coach_fee	smallint(6)	YES		NULL	

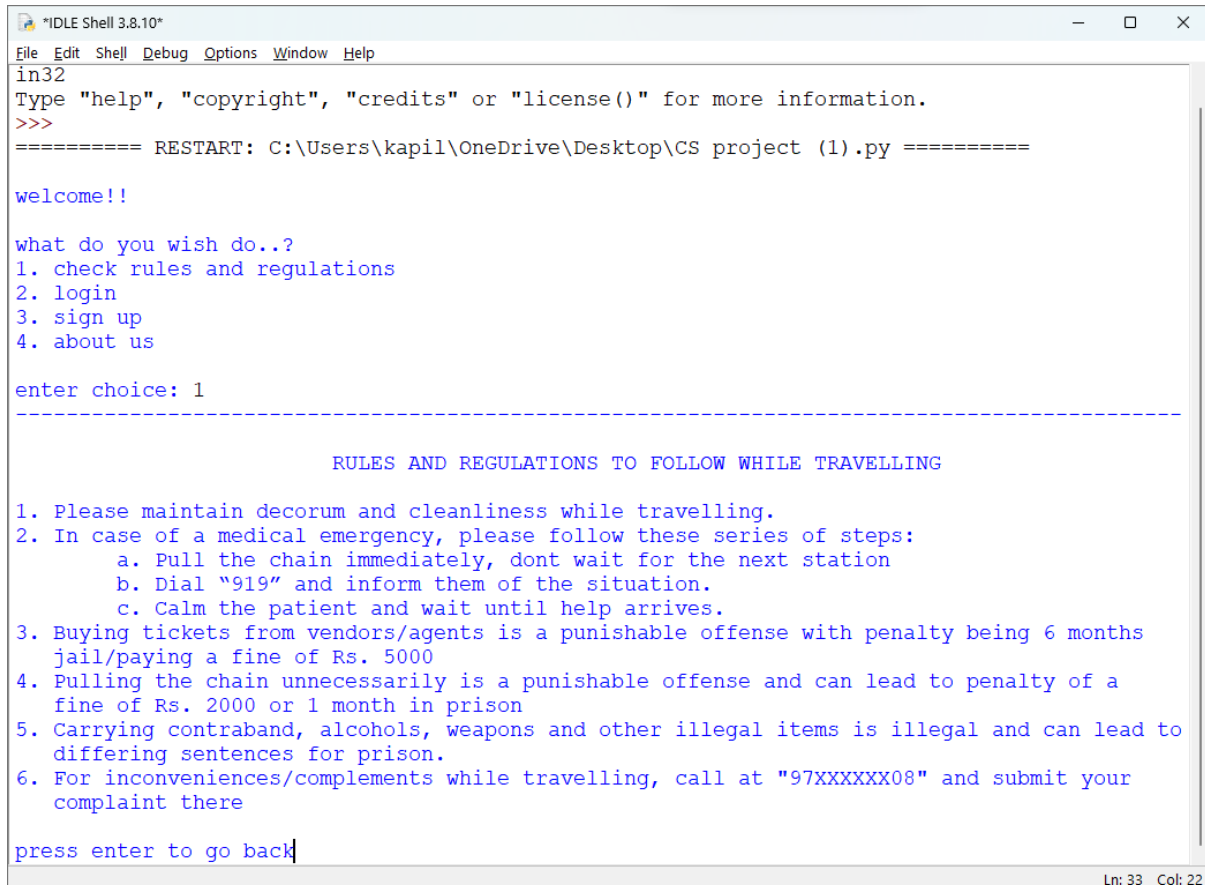
```
3 rows in set (0.02 sec)
```

```
mysql> select * from coach_types;
```

sr_no	coach_types	coach_fee
1	BOX CAR	10000
2	FLAT CAR	8000
3	GONDOLA	13000
4	REFRIGERATED BOX CAR	20000
5	TANK CAR	20000

```
5 rows in set (0.00 sec)
```

Python Program Screen Shots



```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
in32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\kapil\OneDrive\Desktop\CS project (1).py =====

welcome!!

what do you wish do..?
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 1
-----

                RULES AND REGULATIONS TO FOLLOW WHILE TRAVELLING

1. Please maintain decorum and cleanliness while travelling.
2. In case of a medical emergency, please follow these series of steps:
    a. Pull the chain immediately, dont wait for the next station
    b. Dial "919" and inform them of the situation.
    c. Calm the patient and wait until help arrives.
3. Buying tickets from vendors/agents is a punishable offense with penalty being 6 months
   jail/paying a fine of Rs. 5000
4. Pulling the chain unnecessarily is a punishable offense and can lead to penalty of a
   fine of Rs. 2000 or 1 month in prison
5. Carrying contraband, alcohols, weapons and other illegal items is illegal and can lead to
   differing sentences for prison.
6. For inconveniences/complements while travelling, call at "97XXXXXX08" and submit your
   complaint there

press enter to go back|
```

Ln: 33 Col: 22

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help

=====

welcome!!

what do you wish do..?
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 4
-----

we hope to create software that makes the customer experience easier and more satisfying.
    We prioritize customer satisfaction over anything else.

press enter to go back
=====

welcome!!

what do you wish do..?
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 2
enter username: Admin
```

Ln: 66 Col: 21

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help

enter username: Admin
enter password: kapil2006@
-----

what do you wish to do..?

1. alter train sheet
2. alter user list
3. server shut down
4. logout

enter choice: 1
-----

what do you wish to do..?

1. view train sheet
2. append train sheet
3. update status
4. go back

enter choice: 1
-----

train_num    start_point    destination    departure    arrival    train_type
10001        Dehradun       Gandhinagar    2023-11-11   2023-11-11   passenger
10002        Ranchi          Aizawl         2023-12-10   2023-12-11   freight
10003        Thiruvananthapuram Begaluru       2023-12-24   2023-12-26   passenger
10004        Ranchi          Aizawl         2024-01-02   2024-01-02   freight

press enter to go back
```

Ln: 98 Col: 22

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
-----
what do you wish to do..?
1. view train_sheet
2. append train_sheet
3. update status
4. go back
enter choice: 2
-----
please input values carefully
if you enter an invalid value, you have to fill all values again

enter train_number: 10005
enter start_point: patna
enter destination: jammu
enter departure date: 2023-11-15
enter departure time: 18:30:00
enter avg speed of train(Km/H): 35
enter type of train(passenger/freight): passenger

enter number of ac coaches: 6
enter number of ac seats per coach: 40
enter booking fee of one ac seat: 6000

enter number of sleeper coaches: 12
enter number of sleeper seats per coach: 50
enter booking fee of one sleeper seat: 5500
-----
Ln: 27 Col: 0

*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
here is the listing of entered data, please recheck

train_num:      10005
start_point:    Patna
destination:     Jammu
departure_date: 2023-11-15
departure_time: 18:30:00
arrival_date:    2023-11-16
coach_sheet:     10005_coach_sheet_2023_11_15
status:          on_time
train_type:      passenger

number of ac coaches: 6 ,           number of seats per coach: 40
number of sleeper coaches: 12 ,       number of seats per coach: 50
booking fee of one ac_seat: 6000
booking fee for one sleeper seat: 5500

do you wish to continue with the following data(y/n): y
-----

---TRAIN DETAILS HAVE BEEN ADDED---

press enter to go back
-----

what do you wish to do..?
1. view train_sheet
2. append train_sheet
3. update status
4. go back
enter choice: 3
-----
Ln: 92 Col: 15
```

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help

-----

enter train number: 10005

here are the detailed info on train number 10005

start point:      Patna
destination:      Jammu
departure date:   2023-11-15
arrival date:     2023-11-16
current status:   on_time

what do you wish to do..?
1. set status = "on_time"
2. set status = "delayed"
3. set status = "complete"
4. go back

enter choice: 2

                ---STATUS HAS BEEN SET TO DELAYED---

press enter to go back
-----

what do you wish to do..?

1. view train sheet
2. append train sheet
3. update status
4. go back

enter choice: 4|
```

Ln: 125 Col: 15

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help

-----

what do you wish to do..?

1. alter train sheet
2. alter user list
3. server shut down
4. logout

enter choice: 2

-----

what do you wish to do..?

1. view user list
2. update sus value
3. go back

enter choice: 1

-----

user id      username      aadhar number      age      sus value      account
100000001    Kapil Negi      123456789012      17      0      Personal
100000002    Aditya Rathore  123456789012      18      0      Personal
100000003    Renesh Damre   123456789012      17      0      Personal
100000004    sanskar        989087666781      17      0      Business

press enter to go back
-----

what do you wish to do..?
```

Ln: 163 Col: 14

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
what do you wish to do..?

1. view user list
2. update sus value
3. go back

enter choice: 3
-----

what do you wish to do..?

1. alter train sheet
2. alter user list
3. server shut down
4. logout

enter choice: 4
-----

--you have been logged out--

press enter to go back
=====

welcome!!

what do you wish do..?
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 3
Ln: 189 Col: 11

*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
-----

enter username: user1
enter password: 456
enter aadhaar number: 123456789067
enter account type(personal/business): personal
enter DOB: 1990-11-15

---YOUR ACCOUNT HAS BEEN CREATED---
press enter to go back
=====

welcome!!

what do you wish do..?
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 2
enter username: user1
enter password: 456
-----

welcome, user1

1. check profile
2. book new seat
3. log out

enter choice: 1|
Ln: 228 Col: 15
```

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
-----
user id : 100000005
username : user1
aadhar number: 123456789067
age : 33

press enter to go back
-----

welcome, user1

1. check profile
2. book new seat
3. log out

enter choice: 2
-----

here are the listings of the train stations all over India:

Srinagar          Jammu             Shimla
Chandigarh        Dehradun          New Delhi
Gangtok           Itanagar          Jaipur
Lucknow           Dispur            Kohima
Shillong          Patna             Manipur
Agartala          Aizawl            Gandhinagar
Bhopal            Ranchi            Kolkata
Chhattisgarh      Bhubaneswar       Mumbai
Hyderabad         Panaji            Begaluru
Amaravati         Chennai           Puducherry
Thiruvananthapuram Dadra & Nagar Haveli

Ln: 260 Col: 45

*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
-----

enter name of start point (or exit): patna
enter name of destination: jammu
-----

following trains are available for travel:

train number - 10005
  date of departure: 2023-11-15
  time of departure: 18:30:00
  estimated date of arrival: 2023-11-16
  available ac seats: 240
  available sleeper seats: 600

enter train number to proceed (or exit): 10005
-----

you can select from the following coaches:

1. coach number - A1
   coach type: AC
   booking fee(one seat): 6000
   seats available 40

2. coach number - A2
   coach type: AC
   booking fee(one seat): 6000
   seats available 40

3. coach number - A3
   coach type: AC
   booking fee(one seat): 6000
   seats available 40

Ln: 261 Col: 0
```



```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help

4. coach number - A4
   coach type:          AC
   booking fee(one seat): 6000
   seats available      40

5. coach number - A5
   coach type:          AC
   booking fee(one seat): 6000
   seats available      40

6. coach number - A6
   coach type:          AC
   booking fee(one seat): 6000
   seats available      40

7. coach number - B1
   coach type:          SLEEPER
   booking fee(one seat): 5500
   seats available      50

8. coach number - B2
   coach type:          SLEEPER
   booking fee(one seat): 5500
   seats available      50

9. coach number - B3
   coach type:          SLEEPER
   booking fee(one seat): 5500
   seats available      50

10. coach number - B4
    coach type:          SLEEPER

Ln: 294 Col: 0

---

*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help

   coach type:          SLEEPER
   booking fee(one seat): 5500
   seats available      50

18. coach number - C3
    coach type:          SLEEPER
    booking fee(one seat): 5500
    seats available      50

enter the coach number to book a seat: A6
-----

please deposit Rs. 6000 into the counter

press enter to continue
-----

                        ---TICKET BOOKING WAS SUCCESSFUL---

press enter to go back
-----

welcome, user1

1. check profile
2. view ticket details
3. log out

enter choice: 2

Ln: 393 Col: 15
```

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
enter choice: 2
-----

TICKET DETAILS -

train number:      10005
date of departure: 2023-11-15
time of departure: 18:30:00
coach number:      A6
train status:      delayed

what do you wish to do..?
1. download ticket
2. cancel booking
3. go back

enter choice: 1

enter path address to save ticket file: f:\\cs_project
-----

ticket has been saved as f:\\cs_projecttrain_ticket_10005_coach_sheet_2023_11_15_A6.txt
press enter to go back
-----

welcome, user1

1. check profile
2. view ticket details
3. log out

enter choice: 3
```

Ln: 425 Col: 15

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
-----

--you have been logged out--

press enter to go back
=====

welcome!!

what do you wish do..?
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 3
-----

enter username: user2
enter password: 456
enter aadhaar number: 123456789087
enter account type(personal/business): business
enter name of (company/brand): AMUL
enter DOB: 1996-12-23

---YOUR ACCOUNT HAS BEEN CREATED---
press enter to go back
=====

welcome!!
```

Ln: 426 Col: 51

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 3
-----

enter username: user2
enter password: 456
enter aadhaar number: 123456789087
enter account type(personal/business): business
enter name of (company/brand): AMUL
enter DOB: 1996-12-23

---YOUR ACCOUNT HAS BEEN CREATED---
press enter to go back
=====

welcome!!

what do you wish do..?
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 2
enter username: user2
enter password: 456
-----

Ln: 437 Col: 0

*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
welcome, user2

1. check profile
2. book coaches
3. check bookings
4. log out

enter choice: 1
-----

user id :          100000006
username :         user2
aadhar number:     123456789087
age :             27
company affiliated: AMUL

press enter to go back
-----

welcome, user2

1. check profile
2. book coaches
3. check bookings
4. log out

enter choice: 2
-----

here are the listings of the train stations all over India:

Srinagar          Jammu             Shimla
Chandigarh        Dehradun          New Delhi
Gangtok           Itanagar          Jaipur

Ln: 512 Col: 37
```

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
Lucknow          Dispur          Kohima
Shillong         Patna          Manipur
Agartala         Aizawl        Gandhinagar
Bhopal           Ranchi         Kolkata
Chhattisgarh     Bhubaseswar   Mumbai
Hyderabad        Panaji        Begaluru
Amaravati        Chennai       Puducherry
Thiruvananthapuram Dadra & Nagar Haveli

enter name of start point (or exit): ranchi
enter name of destination: aizawl
-----

following trains are available for travel:

train number - 10002
    date of departure:      2023-12-10
    time of departure:      12:00:00
    estimated date of arrival: 2023-12-11
    coaches available:      6

train number - 10004
    date of departure:      2024-01-02
    time of departure:      23:45:00
    estimated date of arrival: 2024-01-02
    coaches available:      16

enter train number to proceed: 10002
-----

here are the types of coaches you can choose from:

1. BOX CAR                      fee per coach = 10000
-----
Ln: 503 Col: 56

*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
2. FLAT CAR                      fee per coach = 8000
3. GONDOLA                      fee per coach = 13000
4. REFRIGERATED BOX CAR        fee per coach = 20000
5. TANK CAR                     fee per coach = 20000

select the type of coach you wish to book: 5
enter the number of coaches you wish to book: 3
-----

please deposit Rs. 60000 into the counter

press enter to continue
-----

---YOUR BOOKING WAS SUCCESSFUL---

do you wish to continue(y/n): y
-----

here are the types of coaches you can choose from:

1. BOX CAR                      fee per coach = 10000
2. FLAT CAR                      fee per coach = 8000
3. GONDOLA                      fee per coach = 13000
4. REFRIGERATED BOX CAR        fee per coach = 20000
5. TANK CAR                     fee per coach = 20000

select the type of coach you wish to book: 1
enter the number of coaches you wish to book: 2
-----
Ln: 536 Col: 52
```

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help

-----
please deposit Rs. 20000 into the counter

press enter to continue
-----

---YOUR BOOKING WAS SUCCESSFUL---

do you wish to continue(y/n): n
-----

welcome, user2

1. check profile
2. book coaches
3. check bookings
4. log out

enter choice: 3
-----

the bookings were made in the following trains-

1. 10002
    start point:    Ranchi
    destination:    Aizawl
    arrival date:    2023-12-11
    journey status: on_time

enter train number to proceed: 10002
Ln: 567 Col: 47

-----
coach_type:  BOX CAR           coaches booked:  2
coach_type:  TANK CAR          coaches booked:  3

press enter to go back
-----

welcome, user2

1. check profile
2. book coaches
3. check bookings
4. log out

enter choice: 4
-----

---you have been logged out---

press enter to go back
=====

welcome!!

what do you wish do..?
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 2
Ln: 632 Col: 15
```

```
IDLE Shell 3.8.10
File Edit Shell Debug Options Window Help

welcome!!

what do you wish do..?
1. check rules and regulations
2. login
3. sign up
4. about us

enter choice: 2
enter username: Admin
enter password: kapil2006@
-----

what do you wish to do..?
1. alter train sheet
2. alter user list
3. server shut down
4. logout

enter choice: 3
-----

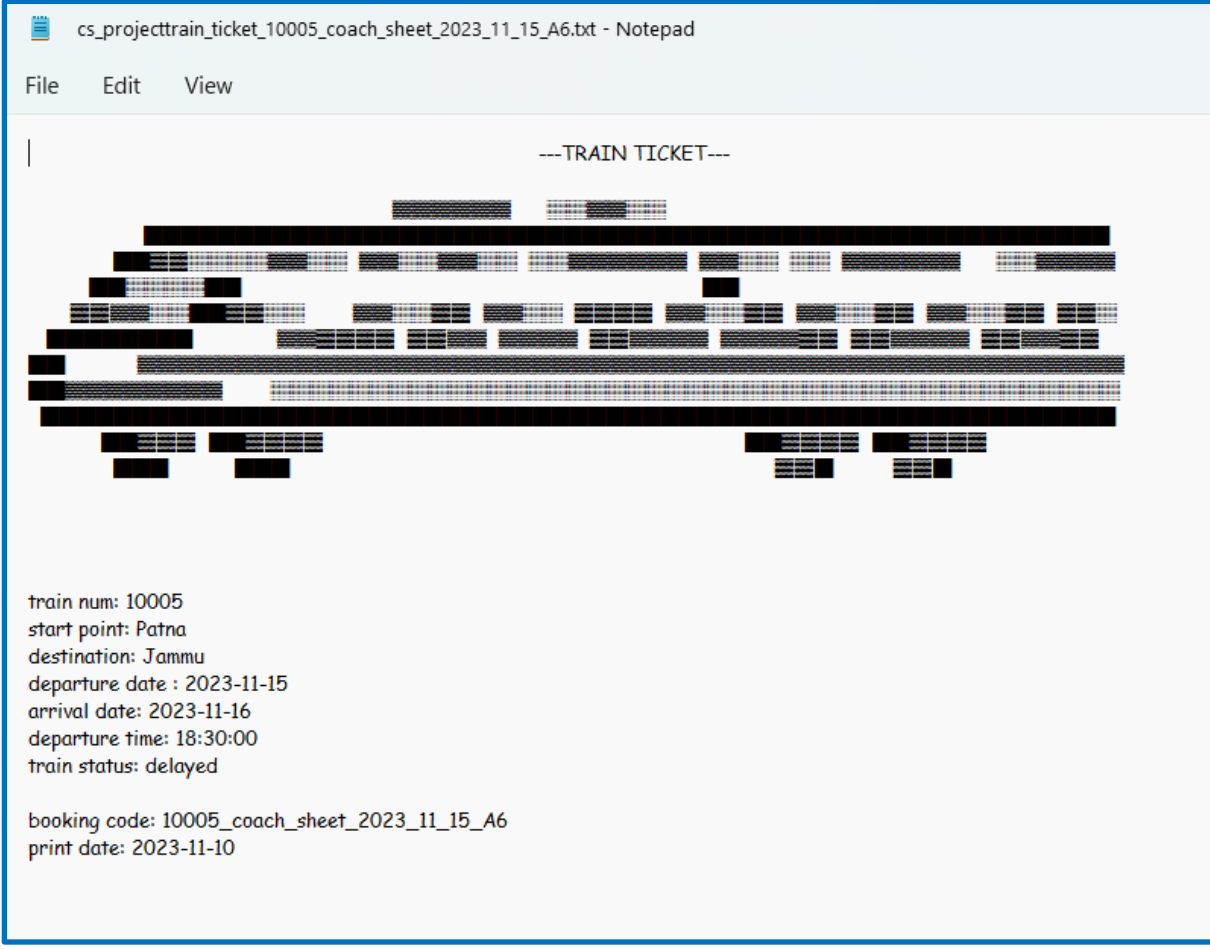
enter "CONFIRM" if you really want to shut down the server: CONFIRM

          ---SERVER HAS BEEN SHUT DOWN---

*****
>>>
>>>
>>> |
```

Ln: 687 Col: 4

Saved text file screen shot



The screenshot shows a Notepad window titled "cs_projecttrain_ticket_10005_coach_sheet_2023_11_15_A6.txt - Notepad". The window contains a text file with a train ticket layout. The layout includes a header "---TRAIN TICKET---", a large table of coach numbers (1A, 2A, 3A, 4A, 5A, 6A, 7A, 8A, 9A, 10A, 11A, 12A, 13A, 14A, 15A, 16A, 17A, 18A, 19A, 20A, 21A, 22A, 23A, 24A, 25A, 26A, 27A, 28A, 29A, 30A, 31A, 32A, 33A, 34A, 35A, 36A, 37A, 38A, 39A, 40A, 41A, 42A, 43A, 44A, 45A, 46A, 47A, 48A, 49A, 50A, 51A, 52A, 53A, 54A, 55A, 56A, 57A, 58A, 59A, 60A, 61A, 62A, 63A, 64A, 65A, 66A, 67A, 68A, 69A, 70A, 71A, 72A, 73A, 74A, 75A, 76A, 77A, 78A, 79A, 80A, 81A, 82A, 83A, 84A, 85A, 86A, 87A, 88A, 89A, 90A, 91A, 92A, 93A, 94A, 95A, 96A, 97A, 98A, 99A, 100A), and a section of train details.

```
---TRAIN TICKET---

train num: 10005
start point: Patna
destination: Jammu
departure date : 2023-11-15
arrival date: 2023-11-16
departure time: 18:30:00
train status: delayed

booking code: 10005_coach_sheet_2023_11_15_A6
print date: 2023-11-10
```

Conclusion

In conclusion, the implementation of the Railway Management System represents a significant leap forward in enhancing the efficiency, and overall performance of railway operations. By seamlessly integrating advanced technology and predictive analytics, this system not only optimizes train scheduling and routing but also mitigates potential risks and improves the overall user experience.

The comprehensive nature of the Railway Management System ensures that railway operators can make informed decisions, respond promptly and maintain a high level of service reliability.

Bibliography / References

1. Arora, Sumita., *Computer science with Python: Textbook for class XII*. New Delhi: Dhanpat Rai & Co. (P), Ltd., 2023
2. Arora, Preeti, *Computer Science with python: Textbook for class-XII*. New Delhi: Sultan Chand & Sons (P) Ltd., 2023
3. Uppal, Shweta, *Computer Science* ; National Council of Education and Research, Page no.:1-32 and 111-172 New Delhi: Publication Department, 2023.
4. “Railway ticket reservation” accessed 7/11/2023 accessed by <https://www.geeksforgeeks.org/online-railway-ticket-reservation-system/>
