Computer Science

A Project Report
    For

AISSCE 2024 Examination
[As a part of the Computer Science (083)]

PROJECT TITLE -   **"RAILWAY RESERVATION SYSTEM"**

By
Ashish Kumar(Admn No:21639)
&
FALAK SHABAN (Admn No: 1564)

School Address   :   **St. Xavier's School**
P.T.C Road
Post Box-19
Hazaribagh– 825301

Under the guidance of (Internal Guide) :   **Mr. SAMIR CHATTARAJ**
PGT
Dept.of Computer Science
St. Xavier's School
Hazaribagh– 825301

TABLE OF CONTENTS

Sl. No.	Particulars	Page No.
1.	Certificate	3
2.	Acknowledgement	4
3.	Introduction	5
4.	Objective & scope of the project	6
5.	Theoretical Background	7
6.	Modules and their purpose	13
7.	Coding	14
8.	Output	22
9.	Requirements	26
10.	Bibliography	27

<u>Certificate</u>

This is to certify that **Ashish Kumar** of class twelve, **St.Xavier's School, Hazaribagh** has successfully completed his project under my guidance in computer practical for the AISSCE as prescribed by CBSE in the year 2024-2025.

Date :

Registration No. :

Signature of Internal Examiner   **Signature of External Examiner**

<u>Acknowledgement</u>

I thank my Computer Science teacher **Mr. Samir Chattaraj** for guidance and support. I also thank my **Principal Father Rosner Xalxo**. I would also like to thank my parents and my sister for encouraging me during the course of this project. Finally I would like to thank **CBSE** for giving me this opportunity to undertake this project.

Ashish Kumar
XII A

Introduction

The **'Railway Reservation System'** is basically a database –based project done with help of python language. This project is very useful for the people to book or cancel train tickets by seating at home with one cell phone in their hand. This project can be modified for various reservation.

This project is to lets the students apply the programming knowledge into a real world situation/problem and exposed the students how programming skills help in developing a good software.

This project is dedicated to model the existing railway reservation system that aims at development of railway reservation system that facilitates the railway customer to manage their reservations and to modify the backend database in a user- friendly manner. The customer interact with the database, who have different 'view label schemas' to the database information. The software provides a comprehensive set of features to enhance the operational limits.

__Objective and scope of the Project__

The objective of this software is to develop a computerized Information system to automate the functions of Railway. This software is also aimed to enhance the current record keeping system, which will manage to retrieve the up-to-date information at right time at right place.
The Proposed software system is expected to do the following functionality-

❖ To Provide a user friendly Interface to its user.

❖ The proposed software will maintain all the records and transactions and will generate the required reports and information when required.

In its current scope , the software enables user to retrieve and update the information from database using mysql and python.

Despite of the best effort of the developer, the following limitations and functional boundaries are visible which limits the scope of this application software

1. This software can store records and produce reports in pre-designed format in soft copy. There is no facility yet to produce customized reports. Only specified reports are covered.

Theoretical Background

What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Features of Python

As a programming language, the features of Python brought to the table are many. Some of the most significant features of Python are:

Easy to Code

Python is a very developer-friendly language which means that anyone and everyone can learn to code it in a couple of hours or days. As compared to other object-oriented programming languages like Java, C, C++, and C#, Python is one of the easiest to learn.

Open Source and Free

Python is an open-source programming language which means that anyone can create and contribute to its development. Python has an online forum where thousands of coders gather daily to improve this language further. Along with this Python is free to download and use in any operating system, be it Windows, Mac or Linux.

Support for GUI

GUI or Graphical User Interface is one of the key aspects of any programming language because it has the ability to add flair to code and make the results more visual. Python has support for a wide array of GUIs which can easily be

imported to the interpreter, thus making this one of the most favorite languages for developers.

Object-Oriented Approach

One of the key aspects of Python is its object-oriented approach. This basically means that Python recognizes the concept of class and object encapsulation thus allowing programs to be efficient in the long run.

High-Level Language

Python has been designed to be a high-level programming language, which means that when you code in Python you don't need to be aware of the coding structure, architecture as well as memory management.

Integrated by Nature

Python is an integrated language by nature. This means that the python interpreter executes codes one line at a time. Unlike other object-oriented programming languages, we don't need to compile Python code thus making the debugging process much easier and efficient. Another advantage of this is, that upon execution the Python code is immediately converted into an intermediate form also known as byte-code which makes it easier to execute and also saves runtime in the long run.

Highly Portable

Suppose you are running Python on Windows and you need to shift the same to either a Mac or a Linux system, then you can easily achieve the same in Python without having to worry about changing the code. This is not possible in other programming languages, thus making Python one of the most portable languages available in the industry.

Highly Dynamic

As mentioned in an earlier paragraph, Python is one of the most dynamic languages available in the industry today. What this basically means is that the type of a variable is decided at the run time and not in advance. Due to the presence of this feature, we do not need to specify the type of the variable during coding, thus saving time and increasing efficiency.

Extensive Array of Library

Out of the box, Python comes inbuilt with a large number of libraries that can be imported at any instance and be used in a specific program. The presence of

libraries also makes sure that you don't need to write all the code yourself and can import the same from those that already exist in the libraries.

Support for Other Languages

Being coded in C, Python by default supports the execution of code written in other programming languages such as Java, C, and C#, thus making it one of the versatile in the industry.

.

File Handling in Python

Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files. The concept of file handling has stretched over various other languages, but the implementation is either complicated or lengthy, but alike other concepts of Python, this concept here is also easy and short. Python treats file differently as text or binary and this is important. Each line of code includes a sequence of characters and they form text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun. Let's start with Reading and Writing files.

Working of open() function

We use **open ()** function in Python to open a file in read or write mode. As explained above, open ( ) will return a file object. To return a file object we use **open()** function along with two arguments, that accepts file name and the mode, whether to read or write. So, the syntax being: **open(filename, mode)**. There are three kinds of mode, that Python provides and how files can be opened:

- " **r** ", for reading.
- " **w** ", for writing.
- " **a** ", for appending.
- " **r+** ", for both reading and writing

One must keep in mind that the mode argument is not mandatory. If not passed, then Python will assume it to be " r " by default. Let's look at this program and try to analyze how the read mode works:
filter_none
edit
play_arrow
brightness_4

```
# a file named "geek", will be opened with the reading mode.
file = open('geek.txt', 'r')
# This will print every line one by one in the file
for each in file:
    print (each)
```

The open command will open the file in the read mode and the for loop will print each line present in the file.

MYSQL

MySQL is a relational database management system (RDBMS) based on the SQL (Structured Query Language) queries. It is one of the most popular languages for accessing and managing the records in the table. MySQL is open-source and free software under the GNU license. Oracle Company supports it.
The following are the most important features of MySQL:
Relational Database Management System (RDBMS)
MySQL is a relational database management system. This database language is based on the SQL queries to access and manage the records of the table.
Easy to use
MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.
It is secure
MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.
Client/ Server Architecture
MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.
Free to download
MySQL is free to use so that we can download it from MySQL official website without any cost.
It is scalable
MySQL supports multi-threading that makes it easily scalable. It can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, we can increase this number to a theoretical limit of 8 TB of data.

Speed
MySQL is considered one of the very fast database languages, backed by a large number of the benchmark test.
High Flexibility
MySQL supports a large number of embedded applications, which makes MySQL very flexible.
Compatible on many operating systems
MySQL is compatible to run on many operating systems, like Novell NetWare, Windows* Linux*, many varieties of UNIX* (such as Sun* Solaris*, AIX, and DEC* UNIX), OS/2, FreeBSD*, and others. MySQL also provides a facility that the clients can run on the same

computer as the server or on another computer (communication via a local network or the Internet).

Allows roll-back

MySQL allows transactions to be rolled back, commit, and crash recovery.

Memory efficiency

Its efficiency is high because it has a very low memory leakage problem.

High Performance

MySQL is faster, more reliable, and cheaper because of its unique storage engine architecture. It provides very high-performance results in comparison to other databases without losing an essential functionality of the software. It has fast loading utilities because of the different cache memory.

High Productivity

MySQL uses Triggers, Stored procedures, and views that allow the developer to give higher productivity.

Platform Independent

It can download, install, and execute on most of the available operating systems.

Partitioning

This feature improves the performance and provides fast management of the large database.

GUI Support

MySQL provides a unified visual database graphical user interface tool named "**MySQL Workbench**" to work with database architects, developers, and Database Administrators. MySQL Workbench provides SQL development, data modeling, data migration, and comprehensive administration tools for server configuration, user administration, backup, and many more. MySQL has a fully GUI supports from MySQL Server version 5.6 and higher.

Dual Password Support

MySQL version 8.0 provides support for dual passwords: one is the current password, and another is a secondary password, which allows us to transition to the new password.

Disadvantages/Drawback of MySQL

Following are the few disadvantages of MySQL:

MySQL version less than 5.0 doesn't support ROLE, COMMIT, and stored procedure.

MySQL does not support a very large database size as efficiently.

MySQL doesn't handle transactions very efficiently, and it is prone to data corruption.

MySQL is accused that it doesn't have a good developing and debugging tool compared to paid databases.

MySQL doesn't support SQL check constraints.

MODULES USED AND THEIR PURPOSE

Module

math	*Mathematical functions (sin() etc.).*
random	*Generate pseudo-random numbers with various common distributions.*
operator	*Functions corresponding to the standard operators.*
Os	*Miscellaneous operating system interfaces.*
enum	*Implementation of an enumeration class.*

CODING

 # DATABASE CREATION.py

```python
def Create_Database_Railway():
    import mysql.connector

mycon=mysql.connector.connect(host='localhost',user='root',passwd='',charset=
'utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    s1="create database railway"
    cursor.execute(s1)
Create_Database_Railway()
```

#connection.py
```python
def connection():
    import mysql.connector

mycon=mysql.connector.connect(host='localhost',user='root',passwd='',databas
e='railway',charset='utf8')
    if mycon.is_connected():
        print("successfully connected")
connection()
```

#tablecreation.py

```python
def table_creation_railway():
    import mysql.connector
    mycon=mysql.connector.connect(host='localhost',user='root',
                    passwd='',database='railway',
                    charset='utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    s1="create table railway(name varchar(100),phno varchar(15)  primary
key,age int(4),gender varchar(50),from_f varchar(100),to_t varchar(100),date_d
varchar(20))"
    cursor.execute(s1)
#table_creation_railway()
```

```python
def table_creation_user_accounts():
    import mysql.connector

mycon=mysql.connector.connect(host='localhost',user='root',passwd='',databas
e='railway',
                              charset='utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    s1="create table user_accounts(fname varchar(100),lname
varchar(100),user_name varchar(100) ,password varchar(100) primary key,
phno varchar(15),gender varchar(50),dob varchar(50),age varchar(4))"
    cursor.execute(s1)
table_creation_user_accounts()
```

#railwayreservation.py

```python
def menu():
    print('1.YES')
    print('2.NO')
    ch=int(input('DO YOU WANT TO CONTINUE OR NOT:'))
    while ch==1:
        print('WELCOME TO ONLINE RAILWAY RESERVATION SYSTEM')
        print('1.SIGN IN')
        print('2.SIGN UP')
        print('3.DELETE ACCOUNT')
        print('4.EXIT')
        ch1=int(input('ENTER YOUR CHOICE:'))
        if ch1==1:
            a=checking()
            if a==True:
                print('WELCOME')
                main()
            else:

                continue
        elif ch1==2:
            a=checking_1()
            if a==True:
                main()
            else:
                print('PASSWORD ALREADY EXISTS')
                continue
```

```python
        elif ch1==3:
            c=checking_2()
            if c==True:

                print('ACCOUNT DELETED')
                continue
            else:
                print('YOUR PASSWAORD OR USER_NAME IS INCORRECT')
                continue
        elif ch1==4:
            print('THANK YOU')
            break
        else:
            print('ERROR 404:PAGE NOT FOUND')
            break
def main():
    print('1.yes')
    print('2.no')
    c=int(input("do you want to continue or not:"))
    while (c==1):
        print(' 1.TICKET BOOKING',"\n", '2.TICKET CHECKING',"\n",'3.TICKET
CANCELLING',"\n",'4.ACCOUNT DETAILS',"\n",'5.LOG OUT')
        ch=int(input('enter ur choice:'))
        if ch==1:
            ticket_booking()
        elif ch==2:
            ticket_checking()
        elif ch==3:
            ticket_cancelling()
        elif ch==4:
            checking_3()


        elif ch==5:
            print('THANK YOU')
            break
        else:
            print('WRONG INPUT')
    else:
        print('ERROR 404: ERROR PAGE NOT FOUND')
def ticket_booking():
    import mysql.connector
    mycon=mysql.connector.connect(host='localhost',user='root',passwd='',
                        database='railway',
                        charset='utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    nm=input('enter your name:')
    phno=input('enter your phone number:')
    age=int(input('enter your age:'))
    print(' M=MALE','\n','F=FEMALE','\n','N=NOT TO MENTION')
    gender=input('enter your gender:')
```

```python
    Gender=gender.upper()
    fr=input('enter ur starting point:')
    to=input('enter your destination:')
    date1=input('enter date(dd):')
    date2=input('enter month(mm):')

    date3=input('enter year(yyyy):')
    date=date1+"/"+date2+"/"+date3
    a={'M':'MALE','F':'FEMALE','N':'NOT TO MENTION'}
    v=a[Gender]
    s1="insert into railway
values('{}',{},{},'{}','{}','{}','{}')".format(nm,phno,age,v,fr,to,date)
    cursor.execute(s1)
    print('BOOKED SUCCESSFULLY')
def ticket_checking():
    import mysql.connector
    mycon=mysql.connector.connect(host='localhost',user='root',passwd=''
                        ,database='railway',
                        charset='utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    print('1.yes')
    print('2.no')
    ch=int(input("do you want to continue or not:"))

    if ch==1:
        phno=int(input('enter your phnone number:'))
        try:
            s1="select * from railway where phno=phno"
            cursor.execute(s1)
            data=cursor.fetchall()[0]
            Data=list(data)
            a=['NAME','PHONE NUMBER','AGE','GENDER','STARTING
POINT','DESTINATION','DATE',]
            print(a[0],':::::',Data[0].upper())
            print(a[1],':::::',Data[1])
            print(a[2],':::::',Data[2])
            print(a[3],':::::',Data[3].upper())
            print(a[4],':::::',Data[4].upper())
            print(a[5],':::::',Data[5].upper())
            print(a[6],':::::',Data[6])
        except:
            print('TICKET DOES NOT EXISTS')
    elif ch==2:
        print('THANK YOU')
    else:
        print('ERROR 404:PAGE NOT FOUND')


def ticket_cancelling():
    import mysql.connector
```

```python
    mycon=mysql.connector.connect(host='localhost',user='root',passwd='',
                    database='railway',charset='utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    print('1.yes')

    print('2.no')
    ch=int(input("do you want to continue or not:"))
    if ch==1:
        phno=input('enter your phone number:')
        s1="delete from railway where phno=phno"
        cursor.execute(s1)
        print('TICKET CANCELLED')
    elif ch==2:
        print('THANK YOU')
    else:
        print('ERROR 404:PAGE NOT FOUND')

def checking_2():
    import mysql.connector
    mycon=mysql.connector.connect(host='localhost',user='root'
                    ,passwd='',database='railway'
                    ,charset='utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    a=input('USER NAME:')
    b=input('PASS WORD:')
    try:
        s1="select user_name from user_accounts where password='{}'".format(b)
        cursor.execute(s1)
        data=cursor.fetchall()[0]
        data=list(data)
        if data[0]==a:
            print('IS THIS YOUR ACCOUNT')
            s1="select user_name from user_accounts where
password='{}'".format(b)
            c1="select fname,lname from user_accounts where
password='{}'".format(b)
            cursor.execute(c1)
            data1=cursor.fetchall()[0]
            data1=list(data1)
            data1=data1[0]+' '+data1[1]
            cursor.execute(s1)
            data=cursor.fetchall()[0]
            data=list(data)
            if data[0]==a:
                x=['FIRST NAME','LAST NAME','PHONE NUMBER','GENDER','DATE
OF BIRTH','AGE']
                s1="select fname,lname,phno,gender,dob,age from user_accounts
where password='{}'".format(b)
                cursor.execute(s1)
                data=cursor.fetchall()[0]
```

```python
            data=list(data)
            print(x[0],':::',data[0])
            print(x[1],':::',data[1])
            print(x[2],':::',data[2])
            print(x[3],':::',data[3])

            print(x[4],':::',data[4])
            print(x[5],':::',data[5])
            print('1.yes')
            print('2.no')
            vi=int(input('enter your choice:'))
            if vi==1:
                b1="delete from user_accounts where password = '{}'".format(b)
                cursor.execute(b1)
                return True
            elif vi==2:
                print('SORRY,RETRY')
            else:
                print('ERROR 404:PAGE NOT FOUND')
        else:
            return False
    except:
        print('ACCOUNT DOES NOT EXIST')


def checking_1():
    import mysql.connector
    mycon=mysql.connector.connect(host='localhost',user='root',
                    passwd='',database='railway',
                    charset='utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    f=input("FIRST NAME:")
    l=input("LAST NAME:")
#    n=f+" "+1
    a=input('USER NAME:')
    b=input('PASS WORD:')
    c=input('RE-ENTER YOUR PASS WORD:')
    ph=input("PHONE NUMBER:")
    print(' M=MALE','\n','F=FEMALE','\n','N=NOT TO MENTION')
    gen=input('ENTER YOUR GENDER:')
    print("ENTER YOR DATE OF BIRTH")
    d=input("DD:")
    o=input("MM:")
    p=input("YYYY:")
    dob=d+'/'+o+'/'+p
    age=input('YOUR AGE:')
    v={'m':'MALE','f':'FEMALE','n':'NOT TO MENTION'}
    if b==c:
        try:
            c1="insert into user_accounts
values('{}','{}','{}','{}','{}','{}','{}','{}')".format(f,l,a,b,ph,v[gen],dob,age)
```

```python
            print(c1)
            cursor.execute(c1)
            print('WELCOME',f,' ',l)
            return True
        except:

            print('PASSWORD ALREADY EXISTS')
            return False
    else:
        print('BOTH PASSWORDS ARE NOT MATCHING')



def checking():
    import mysql.connector
    mycon=mysql.connector.connect(host='localhost',user='root',
                        passwd='',database='railway',
                        charset='utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    a=input('USER NAME:')
    b=input('PASS WORD:')
    try:
        s1="select user_name from user_accounts where password='{}'".format(b)
        c1="select fname,lname from user_accounts where
password='{}'".format(b)
        cursor.execute(c1)
        data1=cursor.fetchall()[0]

        data1=list(data1)
        data1=data1[0]+' '+data1[1]
        cursor.execute(s1)
        data=cursor.fetchall()[0]
        data=list(data)[0]
        if data==a:
            print(' HII ',data1)
            return True
        else:
            return False
    except:
        print('ACCOUNT DOES NOT EXIST')

def checking_3():
    import mysql.connector
    mycon=mysql.connector.connect(host='localhost',user='root',passwd='',
                        database='railway',
                        charset='utf8')
    cursor=mycon.cursor()
    mycon.autocommit=True
    a=input('USER NAME:')
    b=input('PASS WORD:')
    try:
```

```python
        s1="select user_name from user_accounts where password='{}'".format(b)
        c1="select fname,lname from user_accounts where
password='{}'".format(b)
        cursor.execute(c1)
        data1=cursor.fetchall()[0]
        data1=list(data1)
        data1=data1[0]+' '+data1[1]
        cursor.execute(s1)
        data=cursor.fetchall()[0]
        data=list(data)
        if data[0]==a:

                x=['FIRST NAME','LAST NAME','PHONE NUMBER','GENDER','DATE OF
BIRTH','AGE']
                s1="select fname,lname,phno,gender,dob,age from user_accounts
where password='{}'".format(b)
                cursor.execute(s1)
                data=cursor.fetchall()[0]
                data=list(data)
                print(x[0],':::',data[0])
                print(x[1],':::',data[1])
                print(x[2],':::',data[2])
                print(x[3],':::',data[3])
                print(x[4],':::',data[4])
                print(x[5],':::',data[5])


        else:
                return False
    except:
        print('ACCOUNT DOES NOT EXIST')

menu()
```

OUTPUT

Database Tables

Table- user_accounts

```
mysql> describe user_accounts;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| fname     | varchar(100) | YES  |     | NULL    |       |
| lname     | varchar(100) | YES  |     | NULL    |       |
| user_name | varchar(100) | YES  |     | NULL    |       |
| password  | varchar(100) | NO   | PRI | NULL    |       |
| phno      | varchar(15)  | YES  |     | NULL    |       |
| gender    | varchar(50)  | YES  |     | NULL    |       |
| dob       | varchar(50)  | YES  |     | NULL    |       |
| age       | varchar(4)   | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
```

Table- railway

```
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| name   | varchar(100) | YES  |     | NULL    |       |
| phno   | varchar(15)  | NO   | PRI | NULL    |       |
| age    | int(4)       | YES  |     | NULL    |       |
| gender | varchar(50)  | YES  |     | NULL    |       |
| from_f | varchar(100) | YES  |     | NULL    |       |
| to_t   | varchar(100) | YES  |     | NULL    |       |
| date_d | varchar(20)  | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
```

Output-1

1.YES
2.NO

DO YOU WANT TO CONTINUE OR NOT:

Output-2

DO YOU WANT TO CONTINUE OR NOT:1
WELCOME TO ONLINE RAILWAY RESERVATION SYSTEM
1.SIGN IN
2.SIGN UP
3.DELETE ACCOUNT
4.EXIT

Output-3

ENTER YOUR CHOICE:2

FIRST NAME:SAMIR

LAST NAME:CHATTARAJ

USER NAME:SAM

PASS WORD:12345

RE-ENTER YOUR PASS WORD:12345

PHONE NUMBER:897654
 M=MALE
 F=FEMALE
 N=NOT TO MENTION

ENTER YOUR GENDER:M
ENTER YOR DATE OF BIRTH

DD:04

MM:12

YYYY:1985

YOUR AGE:32

WELCOME SAMIR   CHATTARAJ

Output-4

ENTER YOUR CHOICE:1

USER NAME:SAM

PASS WORD:12345
 HII   SAMIR CHATTARAJ
WELCOME
1.YES
2.NO

DO YOU WANT TO CONTINUE OR NOT:1
 1.TICKET BOOKING
 2.TICKET CHECKING
 3.TICKET CANCELLING
 4.ACCOUNT DETAILS
 5.LOG OUT

ENTER UR CHOICE:1

ENTER YOUR NAME:SAMIR

ENTER YOUR PHONE NUMBER:897654

ENTER YOUR AGE:32
 M=MALE
 F=FEMALE
 N=NOT TO MENTION

ENTER YOUR GENDER:M

ENTER UR STARTING POINT:DHANBAD

ENTER YOUR DESTINATION:HOWRAH

ENTER DATE(DD):2

ENTER MONTH(MM):12

ENTER YEAR(YYYY):2024
BOOKED SUCCESSFULLY
 1.TICKET BOOKING
 2.TICKET CHECKING
 3.TICKET CANCELLING
 4.ACCOUNT DETAILS

5.LOG OUT

ENTER UR CHOICE:

Output-5

DO YOU WANT TO CONTINUE OR NOT:1
 1.TICKET BOOKING
 2.TICKET CHECKING
 3.TICKET CANCELLING
 4.ACCOUNT DETAILS
 5.LOG OUT

ENTER UR CHOICE:4

USER NAME:SAM

PASS WORD:12345
FIRST NAME ::: SAMIR
LAST NAME ::: CHATTARAJ
PHONE NUMBER ::: 897654
GENDER ::: MALE
DATE OF BIRTH ::: 04/12/1985
AGE ::: 32
 1.TICKET BOOKING
 2.TICKET CHECKING
 3.TICKET CANCELLING
 4.ACCOUNT DETAILS
 5.LOG OUT

ENTER UR CHOICE:

Output-6

ENTER UR CHOICE:5
THANK YOU
WELECOME TO ONLINE RAILWAY RESERVATION SYSTEM
1.SIGN IN
2.SIGN UP
3.DELETE ACCOUNT
4.EXIT

ENTER YOUR CHOICE:4
THANK YOU

<u>REQUIREMENTS</u>

❖ **HARDWARE REQUIRED**

    ☐ **Printer, to print the required documents of the project**
    ☐ **Compact Drive**
    ☐ **Processor : I5**
    ☐ **Ram : 4GB**
    ☐ **Harddisk : 80 Gb.**

❖ **SOFTWARE REQUIRED**

    ☐ **Operating system : Windows XP,Window-10**
    ☐ **Spyder, pycharm for execution of program and**
    ☐ **Ms word for documentation.**

<u>BIBLIOGRAPHY</u>

In order to work on this project titled-'**Railway Reservation System'** ,the following books and literature are referred by me during the various phases of development of the project.

1. **COMPUTER SCIENCE With Python BY :– SUMITA ARORA**

2. **Object Oriented Programming In python By- ******
3. **Various websites(You can mention name of websites)**

Other than the above mentioned books, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project.

<u>Thank You.....</u>