

PROJECT ON:

**RAILWAY
RESERVATION
SYSTEM**



AUTHOR : ARUNESH BASAK

CONTENTS

- Railway Reservation System
- Data Dictionary
- Source Code Output
- Bibliography

RAILWAY RESERVATION SYSTEM

Introduction:

The Railway Reservation System is basically a database based project done with the help of Python Language. This Project is very useful for the people to book or cancel train tickets by sitting at home with one cell phone in their hand. This project can be modified for various reservations.

Objectives:

The Objective of this Project is to let the students apply their programming knowledge into a real world situation/problem and expose the students how programming skills helps in developing a good software.

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

Hardware Requirements:

Operating System	:	Windows 10 (OR ABOVE)
Motherboard	:	Gigabyte H61M-S1 (OR ABOVE)
RAM	:	8GB STARLITE (OR ABOVE)
Hard Disk	:	1024GB WD GREEN SSD (OR ABOVE)
CD/DVD	:	FLOPPY DRIVE 1.44 MB (OR ABOVE)
Monitor	:	DELL S2204L 23 inches (OR ABOVE)
Keyboard & Mouse	:	CosmicByte VERITAS & REDGEAR A20 (OR BETTER)
Printer	:	HP DESKJET 2331 (OR BETTER)

Software Requirements:

Windows OS

Python

MySQL Connector Module

Proposed System

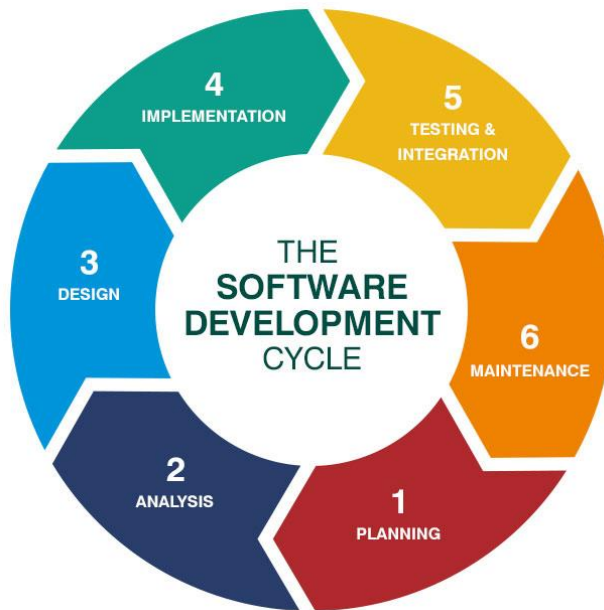
Today one cannot afford to rely on the fallible human beings of be really wants to stand against today's merciless competition where not to wise saying "**to err is human**" no longer valid, it's outdated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of files with a much sophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomization various organizations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paper work has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done.

This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it's an age of computers of and automating such an organization gives the better look.



System Development Life Cycle (SDLC)



The systems development life cycle is a project management technique that divides complex projects into smaller, more managed segments or phases. Segmenting project allows managers to verify the successful completion of project phases before allocating resources to subsequent phases.

Software development projects typically include initiation, planning, design, development, testing, implementation and maintenance phases. However, the phases maybe divided differently depending on the organizations involved.

For example, initial project activities might be divided as request, requirements-definition, planning phases, initiation, concept-development, and planning phases. End users of the system under development should be involved in reviewing the output of each phase to ensure the system is being built to deliver the needed functionality.

Phases of System Development Life Cycle

Initiation Phase

The Initiation Phase begins when a business sponsor identifies a need or an opportunity.

The purpose of the Initiation Phase is to:

- Identify and validate an opportunity to improve business accomplishments of the organization or a deficiency related to a business need.
- Identify significant assumptions and constraints on solutions to that need.
- Recommend the exploration of alternative concepts and methods to satisfy the need including questioning the need for technology, i.e., will a change in the business process offer a solution?
- Assure executive business and executive technical sponsorship. The Sponsor designates a Project Manager and the business need is documented in a Concept Proposal. The Concept Proposal includes information about the business process and the relationship to the Agency/Organization.
- Infrastructure and the Strategic Plan. A successful Concept Proposal results in a Project Management Charter which outlines the authority of the project manager to begin the project.

Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organization's enterprise architecture. The initiation phase begins when an opportunity to add, improve, or correct a system is identified and formally requested through the presentation of a business case. The business case should, at a minimum, describe a proposal's purpose, identify expected benefits, and explain how the proposed system supports one of the organization's business strategies. The business case should also identify alternative solutions and detail as many informational, functional, and network requirements as possible.

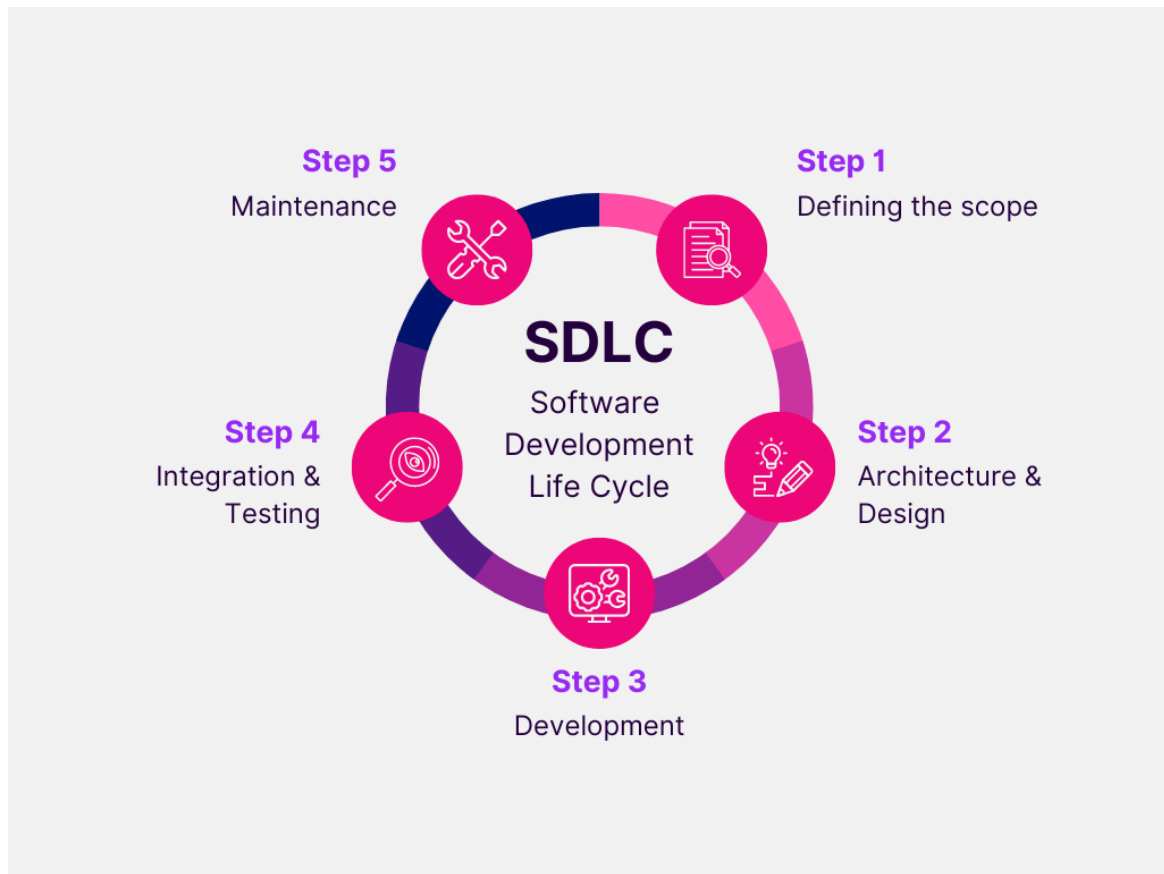
System Concept Development Phase

The System Concept Development Phase begins after a business need or opportunity is validated by the Agency/Organization Program Leadership and the Agency/Organization CIO.

The Purpose of the System Concept Development Phase is to:

- Determine the feasibility and appropriateness of the alternatives.
- Identify the System Interfaces.
- Identify basic functional and data requirements to satisfy the business need.
- Establish system boundaries; identify goals, objectives, critical success factors, and performance measures.
- Evaluate costs and benefits of alternative approaches to satisfy the basic functional requirements.
- Assess Project Risks.
- Identify and initiate risk mitigation actions, and develop high-level technical architecture, process models, data models, and a Concept of operations. This phase explores potential technical solutions within the context of the business need.
- It may include several trade-off decisions such as the decision to use COTS software products as opposed to developing custom software or reusing software components, or the decision to use an incremental delivery versus a complete, onetime deployment.
- Construction of executable prototypes is encouraged to evaluate technology to support the business process. The System Boundary Document serves as an important reference document to support the Information Technology Project Request (ITPR) process.
- The ITPR must be approved by the State CIO before the project can move forward.

Pictorial Representation of SDLC



Planning Phase

The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, is necessary to coordinate activities and manage project risks effectively. The depth and formality of project plans should be commensurate with the characteristics and risks of a given project. Project plans refine the information gathered during the initiation phase by further identifying the specific activities and resources required to complete a project.

A critical part of a project manager's job is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and network requirements as possible. During this phase, a plan is developed that documents the approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and user input. Personnel assignments, costs, project schedule, and target dates are established.

A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and validation, and systems engineering Management planning.

Requirements Analysis Phase

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system performance, security, and maintainability requirements for the system. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and Evaluation Master Plan.

The purposes of this phase are to:

- Further define and refine the functional and data requirements and document them in the Requirements Document.
- Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it)
- Develop detailed data and process models (system inputs, outputs, and the process).
- Develop the test and evaluation requirements that will be used to determine acceptable system performance.

Design Phase

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design. Audit, security, and quality assurance personnel should be involved in the review and approval process. During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the software development, a variety of elements are considered in the design to mitigate risk.

These Includes:

- Identifying potential risks and defining mitigating design features.
- Performing a security assessment.
- Developing a conversion plan to migrate current data to the new system.
- Determining the Operating Environment.
- Defining major systems and their inputs and outputs.
- Allocating processes to resources.

- Preparing detailed logic specifications for each software module. The result is a draft System Design Document which captures the preliminary design for the system.
- Everything requiring user input or approval is documented and reviewed by the user. Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system.
- This document receives a rigorous review by Agency technical and functional representatives to ensure that it satisfies the business requirements. Concurrent with the development of the system design, the Agency Project Manager begins development of the Implementation Plan, Operations and Maintenance Manual, and the Training Plan.

Development Phase

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements. Programmers use various techniques to develop computer programs. The large transaction oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Effective completion of the previous stages is a key factor in the success of the Development phase.

The Development Phase consists of:

- Translating the detailed requirements and design into system components.
- Testing individual elements (units) for usability.
- Preparing for integration and testing of the IT.

Integration and Test Phase

Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. OIT Security staffs assess the system security and issue a security certification and accreditation prior to installation/implementation.

Multiple levels of testing are performed including:

- Testing at the development facility by the contractor and possibly supported by end users.
- Testing as a deployed system with end users working together with contract personnel.
- Operational testing by the end user alone performing all functions. Requirements are traced throughout testing, a final Independent Verification & Validation evaluation is performed and all documentation is reviewed and accepted prior to acceptance of the system.

Implementation Phase

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

Operation Phase

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization's needs. When modifications or changes are identified, the system may reenter the planning phase.

The purpose of this Phase is to:

- Operate, maintain, and enhance the system.
- Certify that the system can process sensitive information.
- Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.
- Determine when the system needs to be modernized, replaced, or retired.

DATA DICTIONARY

Variable Table

<i>Sl No.</i>	<i>Variable Name</i>	<i>Type</i>	<i>Size</i>	<i>Description</i>
1.	Train_No	Int	2	To store the Train Number.
2.	Train_Name	String(Char)	10	To store the name of the Train.
3.	Arrival_Station	Char	45	To store the Name of the Arrival Station.
4.	Destination_Station	Char	46	To store the Name of the Destined Station.
5.	Distance_Covered	Int	10	To store distance covered by the Train.
6.	PassengerID	Int	15	To store the Passenger ID of the Passenger.
7.	No_of_Tickets	Int	4	To store the Number of Tickets issued by the Passenger.

Function Table

<i>Sl No.</i>	<i>Function name</i>	<i>Type</i>	<i>Size</i>	<i>Description</i>
1.	Train_NAME	Dictionary	40	Store Train's Name and Number.
2.	Train_NO	Input	2	Input Train's Number.
3.	Train_Name_And_No	If	40	To check the Train's Name and Number.
4.	Time	Int	8	Departure, Arrival Time.
5.	Input_Class	Class	9	Input Class
6.	Print	Print	18	Display the Fare and Passenger ID.

Jan-Shatabdi Express

Train No.: 17169

Train Name: Jan-Shatabdi Express

Arrival Time: 22:15

Departure Time: 14:05

Destination Station: Patna Jn.

Fare:

₹ 3591/- (Executive)

₹ 2316/- (Economical)

Total Distance Covered: 532 km



Kamrup Express

Train No.: 15959

Train Name: Kamrup Express

Arrival Time: 15:45

Departure Time: 01:34

Destination Station: Dibrugarh Jn.

Fare:

₹ 5367/- (Executive)

₹ 3526/- (Economical)

Total Distance Covered: 1,529 km



Duronto Express

Train No.: 11246

Train Name: Duronto Express

Arrival Time: 16:00

Departure Time: 11:11

Destination Station: Howrah Jn.

Fare:

₹ 5796/- (Executive)

₹ 3228/- (Economical)

Total Distance Covered: 1,531 km



Gatimaan Express

Train No.: 12050

Train Name: Gatimaan Express

Arrival Time: 12:01

Departure Time: 05:23

Destination Station: Hazrat Nizamuddin Jn.

Fare:

₹ 3624/- (Executive)

₹ 1690/- (Economical)

Total Distance Covered: 403 km



Rajdhani Express

Train No.: 64845

Train Name: Rajdhani Express

Arrival Time: 13:56

Departure Time: 06:14

Destination Station: Delhi Cant.

Fare:

₹ 5988/- (Executive)

₹ 3327/- (Economical)

Total Distance Covered: 1,450 km



Sampark Kranti Express

Train No.: 12329

Train Name: Sampark Kranti Express

Arrival Time: 13:10

Departure Time: 08:40

Destination Station: Anand Vihar Terminal

Fare:

₹ 5793/- (Executive)

₹ 3526/- (Economical)

Total Distance Covered: 1,441 km



Garib Rath Express

Train No.: 12877

Train Name: Garib Rath Express

Arrival Time: 16:25

Departure Time: 11:10

Destination Station: Delhi Cant.

Fare:

₹ 5235/- (Executive)

₹ 3098/- (Economical)

Total Distance Covered: 1,343 km



Vivek Express

Train No.: 15906

Train Name: Vivek Express

Arrival Time: 15:06

Departure Time: 01:56

Destination Station: Kanyakumari

Fare:

₹ 8437/- (Executive)

₹ 5980/- (Economical)

Total Distance Covered: 4,200 km



Teesta Torsha Express

Train No.: 13141

Train Name: Teesta Torsha Express

Arrival Time: 07:56

Departure Time: 10:32

Destination Station: New Alipurduar Jn.

Fare:

₹ 2986/- (Executive)

₹ 1270/- (Economical)

Total Distance Covered: 718 km



Jammu Tawi Express

Train No.: 13151

Train Name: Jammu Tawi Express

Arrival Time: 23:06

Departure Time: 13:56

Destination Station: Jammu Tawi Jn.

Fare:

₹ 5730/- (Executive)

₹ 3986/- (Economical)

Total Distance Covered: 1,981 km



SOURCE CODE

```
import mysql.connector

# Database and Table Setup

mycon = mysql.connector.connect(host='localhost', user='root',
                                passwd='ss@2007')

cursor = mycon.cursor()

mycon.autocommit = True

# Create database and tables

cursor.execute("CREATE DATABASE IF NOT EXISTS railway")
cursor.execute("USE railway")

# Railway table

cursor.execute("""
CREATE TABLE IF NOT EXISTS 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)
)
""")
```

```
# User accounts table
```

```
cursor.execute("""
```

```
CREATE TABLE IF NOT EXISTS user_accounts(
```

```
    fname VARCHAR(100),
```

```
    lname VARCHAR(100),
```

```
    user_name VARCHAR(100) PRIMARY KEY,
```

```
    password VARCHAR(100),
```

```
    phno VARCHAR(15),
```

```
    gender VARCHAR(50),
```

```
    dob VARCHAR(50),
```

```
    age VARCHAR(4)
```

```
)
```

```
""")
```

```
print("Database and tables set up successfully.")
```

```
# Menu function
```

```
def menu():
```

```
    while True:
```

```
        print('1. YES')
```

```
        print('2. NO')
```

```
        ch = int(input('DO YOU WANT TO CONTINUE OR NOT: '))
```

```
        if ch == 1:
```

```
            print('WELCOME TO ONLINE RAILWAY RESERVATION  
SYSTEM')
```

```
print('1. SIGN IN')
print('2. SIGN UP')
print('3. DELETE ACCOUNT')
print('4. EXIT')
chi = int(input('ENTER YOUR CHOICE: '))
if chi == 1:
    if checking():
        print('WELCOME')
        main()
    else:
        print('Incorrect credentials. Try again.')
elif chi == 2:
    if checking_1():
        main()
    else:
        print('PASSWORD ALREADY EXISTS')
elif chi == 3:
    if checking_2():
        print('ACCOUNT DELETED')
    else:
        print('YOUR PASSWORD OR USER NAME IS
INCORRECT')
elif chi == 4:
    print('THANK YOU')
    break
else:
```

```
        print('ERROR 404: PAGE NOT FOUND!')
elif ch == 2:
    print('THANK YOU')
    break
else:
    print('Invalid choice. Please enter 1 or 2.')

# Main function
def main():
    while True:
        print('1. TICKET BOOKING')
        print('2. TICKET CHECKING')
        print('3. TICKET CANCELLING')
        print('4. ACCOUNT DETAILS')
        print('5. LOG OUT')
        ch = int(input('Enter your 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('ERROR 404: PAGE NOT FOUND!')
```

Ticket booking function

```
def ticket_booking():
```

```
    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: ').upper()
```

```
    fr = input('Enter your starting point: ')
```

```
    to = input('Enter your destination: ')
```

```
    datel = input('Enter date (dd): ')
```

```
    date2 = input('Enter month (mm): ')
```

```
    date3 = input('Enter year (yyyy): ')
```

```
    date = f"{datel}/{date2}/{date3}"
```

```
    gender_full = {'M': 'MALE', 'F': 'FEMALE', 'N': 'NOT TO MENTION'}
```

```
    v = gender_full.get(gender, 'NOT TO MENTION')
```

```
    sl = f"INSERT INTO railway VALUES ('{nm}', '{phno}', {age}, '{v}',  
'{fr}', '{to}', '{date}')
```

```
    cursor.execute(sl)
```

```
    print('BOOKED SUCCESSFULLY')
```

Ticket checking function

```
def ticket_checking():
```



```
phno = input('Enter your phone number: ')
```

```
try:
```

```
    sl = f"SELECT * FROM railway WHERE phno = '{phno}'"
```

```
    cursor.execute(sl)
```

```
    data = cursor.fetchone()
```

```
    if data:
```

```
        fields = ['NAME', 'PHONE NUMBER', 'AGE', 'GENDER',  
'STARTING POINT', 'DESTINATION', 'DATE']
```

```
        for field, value in zip(fields, data):
```

```
            print(f"{field}: {value}")
```

```
    else:
```

```
        print('TICKET DOES NOT EXIST')
```

```
except Exception as e:
```

```
    print('Error:', e)
```

```
# Ticket cancelling function
```

```
def ticket_cancelling():
```

```
    phno = input('Enter your phone number: ')
```

```
    sl = f"DELETE FROM railway WHERE phno = '{phno}'"
```

```
    cursor.execute(sl)
```

```
    print('TICKET CANCELLED')
```

```
# Sign-up function
```

```
def checking_1():
```

```
    f = input("FIRST NAME: ")
```

```
    l = input("LAST NAME: ")
```

```

a = input('USER NAME: ')
b = input('PASSWORD: ')
c = input('RE-ENTER YOUR PASSWORD: ')
ph = input('PHONE NUMBER: ')
print('M=MALE', '\n', 'F=FEMALE', '\n', 'N=NOT TO MENTION')
gen = input('ENTER YOUR GENDER: ').upper()
d = input("DATE OF BIRTH (dd/mm/yyyy): ")
age = input('YOUR AGE: ')
gender_full = {'M': 'MALE', 'F': 'FEMALE', 'N': 'NOT TO MENTION'}
v = gender_full.get(gen, 'NOT TO MENTION')
if b == c:
    try:
        sl = f"INSERT INTO user_accounts VALUES ('{f}', '{l}', '{a}',
        '{b}', '{ph}', '{v}', '{d}', '{age}')"
        cursor.execute(sl)
        print(f'WELCOME {f} {l}')
        return True
    except:
        print('PASSWORD ALREADY EXISTS')
        return False
else:
    print('BOTH PASSWORDS ARE NOT MATCHING')
    return False

# Sign-in function
def checking():

```

```
a = input('USER NAME: ')
b = input('PASSWORD: ')

try:
    sl = f"SELECT user_name FROM user_accounts WHERE password = '{b}'"
    cursor.execute(sl)
    data = cursor.fetchone()
    if data and data[0] == a:
        print('HII, WELCOME!')
        return True
    else:
        return False
except:
    print('ACCOUNT DOES NOT EXIST')
    return False
```

Account deletion function

```
def checking_2():
    a = input('USER NAME: ')
    b = input('PASSWORD: ')

    try:
        sl = f"DELETE FROM user_accounts WHERE user_name = '{a}' AND password = '{b}'"
        cursor.execute(sl)
        print('ACCOUNT DELETED SUCCESSFULLY')
        return True
    except:
```

```
print('ACCOUNT DOES NOT EXIST')
```

```
return False
```

```
# Account details function
```

```
def checking_3():
```

```
    a = input('USER NAME: ')
```

```
    b = input('PASSWORD: ')
```

```
    try:
```

```
        sl = f"SELECT * FROM user_accounts WHERE user_name = '{a}'  
AND password = '{b}'"
```

```
        cursor.execute(sl)
```

```
        data = cursor.fetchone()
```

```
        if data:
```

```
            fields = ['FIRST NAME', 'LAST NAME', 'USER NAME', 'PHONE  
NUMBER', 'GENDER', 'DOB', 'AGE']
```

```
            for field, value in zip(fields, data):
```

```
                print(f"{field}: {value}")
```

```
        else:
```

```
            print('ACCOUNT DOES NOT EXIST')
```

```
    except Exception as e:
```

```
        print('Error:', e)
```

```
# Start the program
```

```
menu()
```

OUTPUTS

Choice Window

```
Database and tables set up successfully.
1. YES
2. NO
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
ENTER YOUR CHOICE:
```

If the Choice was: 1(Sign In Window)

```
Database and tables set up successfully.
1. YES
2. NO
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
ENTER YOUR CHOICE: 1
USER NAME: ss123
PASSWORD: ss321
HII, WELCOME!
```

If the Choice was: 2(Sign Up Window)

```
Database and tables set up successfully.
1. YES
2. NO
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
ENTER YOUR CHOICE: 2
FIRST NAME: Souptim
LAST NAME: Saha
USER NAME: ss123
PASSWORD: ss321
RE-ENTER YOUR PASSWORD: ss321
PHONE NUMBER: 8252525849
M=MALE
  F=FEMALE
  N=NOT TO MENTION
ENTER YOUR GENDER: M
DATE OF BIRTH (dd/mm/yyyy): 01/01/2006
YOUR AGE: 18
WELCOME Souptim Saha
1. TICKET BOOKING
2. TICKET CHECKING
3. TICKET CANCELLING
4. ACCOUNT DETAILS
5. LOG OUT
Enter your choice: 1
```

If the Choice was: 3(Account Deletion)

```
Database and Tables set up successfully.
1. YES
2. NO
DO YOU WANT TO CONTINUE OR NOT: 1
WELCOME TO RAILWAY RESERVATION SYSTEM
1. SIGN IN
2. SIGN UP
3. DELETE ACCOUNT
4. EXIT
ENTER YOUR CHOICE: 3
ACCOUNT DELETED
```

If the Choice was: 4(Exiting)

Database and Tables set up successfully.

1. YES

2. NO

DO YOU WANT TO CONTINUE OR NOT: 1

WELCOME TO RAILWAY RESERVATION SYSTEM

1. SIGN IN

2. SIGN UP

3. DELETE ACCOUNT

4. EXIT

ENTER YOUR CHOICE: 4

THANK YOU

If the Choice was: 5 OR MORE(Error Page)

Database and Tables set up successfully.

1. YES

2. NO

DO YOU WANT TO CONTINUE OR NOT: 1

WELCOME TO RAILWAY RESERVATION SYSTEM

1. SIGN IN

2. SIGN UP

3. DELETE ACCOUNT

4. EXIT

ENTER YOUR CHOICE: 5

ERROR 404:PAGE NOT FOUND

Second Choice List after Login

Database and tables set up successfully.

1. YES

2. NO

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

ENTER YOUR CHOICE: 1

USER NAME: ss123

PASSWORD: ss321

HII, WELCOME!

WELCOME

1. TICKET BOOKING

2. TICKET CHECKING

3. TICKET CANCELLING

4. ACCOUNT DETAILS

5. LOG OUT

Enter your choice:

If the Choice was: 1(Ticket Booking)

WELCOME Souptim Saha

1. TICKET BOOKING

2. TICKET CHECKING

3. TICKET CANCELLING

4. ACCOUNT DETAILS

5. LOG OUT

Enter your choice: 1

Enter your name: Souptim Saha

Enter your phone number: 8252525849

Enter your age: 18

M=MALE

F=FEMALE

N=NOT TO MENTION

Enter your gender: M

Enter your starting point: Sealdah

Enter your destination: NJP

Enter date (dd): 12

Enter month (mm): 11

Enter year (yyyy): 2024

BOOKED SUCCESSFULLY

If the Choice was: 2(Ticket Checking)

WELCOME

1. TICKET BOOKING
2. TICKET CHECKING
3. TICKET CANCELLING
4. ACCOUNT DETAILS
5. LOG OUT

Enter your choice: 2

Enter your phone number: 8252525849

NAME: Souptim Saha

PHONE NUMBER: 8252525849

AGE: 18

GENDER: MALE

STARTING POINT: Sealdah

DESTINATION: NJP

DATE: 12/11/2024/11/2024

If the Choice was: 3(Ticket Cancellling)

1. TICKET BOOKING
2. TICKET CHECKING
3. TICKET CANCELLING
4. ACCOUNT DETAILS
5. LOG OUT

Enter your choice: 3

Enter your phone number: 8252525849

TICKET CANCELLED

If the Choice was: 4(Account Details)

1. TICKET BOOKING
2. TICKET CHECKING
3. TICKET CANCELLING
4. ACCOUNT DETAILS
5. LOG OUT

Enter your choice: 4

USER NAME: ss123

PASSWORD: ss321

FIRST NAME: Souptim

LAST NAME: Saha

USER NAME: ss123

PHONE NUMBER: ss321

GENDER: 8252525849

DOB: MALE

AGE: 01/01/2006

If the Choice was: 5(Logging Out)

1. TICKET BOOKING
2. TICKET CHECKING
3. TICKET CANCELLING
4. ACCOUNT DETAILS
5. LOG OUT

Enter your choice: 5

THANK YOU

1. YES

2. NO

DO YOU WANT TO CONTINUE OR NOT: 2

THANK YOU

BIBLIOGRAPHY

The Project has been done with the help of my Friends, Family, Friends and Subject Teacher.

References:

- Microsoft Word
- IDLE Python
- Microsoft SQL Server Management Studio
- Google Chrome
- Wikipedia
- Textbook of Computer Science with Python

And finally I am honored to present the Project on
“**RAILWAY RESERVATION SYSTEM**”

Thank You

