Delhi Public School, Hapur

PROJECT FILE
ON
COMPUTER SCIENCE (083)

SESSION:2024-25

Submitted By:

STUDENT'S NAME: Khushi Umrao

ROLL NO. : 10

CLASS & SECTION: XII-C

SUBMITTED TO:

Mr. Vivek Maheshwari

PROJECT

BASED ON

TRAVEL

PLANNER

APP

CERTIFICATE

This is to certify that Computer Science project on Travel Planner App has been successfully completed by ***Khushi Umrao*** of Class *XII C*, Delhi Public School, Hapur for consideration in partial fulfillment of curriculum of Central Board of Secondary Education (CBSE) of Computer Science (083) for the AISSCE Practical Examination 2024-25. I certify that this project is up to my expectation and as per the guidelines issued by the CBSE.

_____

(External Examiner)

_____ _____

(Internal Examiner) (Principal)

Acknowledgment

I take this opportunity to express my deep sense of gratitude to all those who have been instrumental in preparation of this project. I feel great pleasure to express my obligation to Mrs. Meena Anand, Principal of Delhi Public School, Hapur. I am also sincerely grateful to Mr. Vivek Maheshwari PGT(Computer Science), Delhi Public School for his encouragement and valuable guidance during the entire period of work. I would also like to express my gratitude towards my peer, Kinjal Jain, Class XII - A, for being a valuable addition to the team. Without all of them, the project wouldn’t have been successful. I could not forget Internet and Textbooks which provided me with sufficient matter for reference.

Index

INTRODUCTION

Travel Planner App

The objective of the travel planner app is to provide users with a comprehensive, user- friendly, and personalized platform to plan, organize, and manage their trips efficiently.

SCOPE

This app allows users to log their trips, share itineraries and store their travel data for later.

REQUIREMENTS

Hardware Requirements:

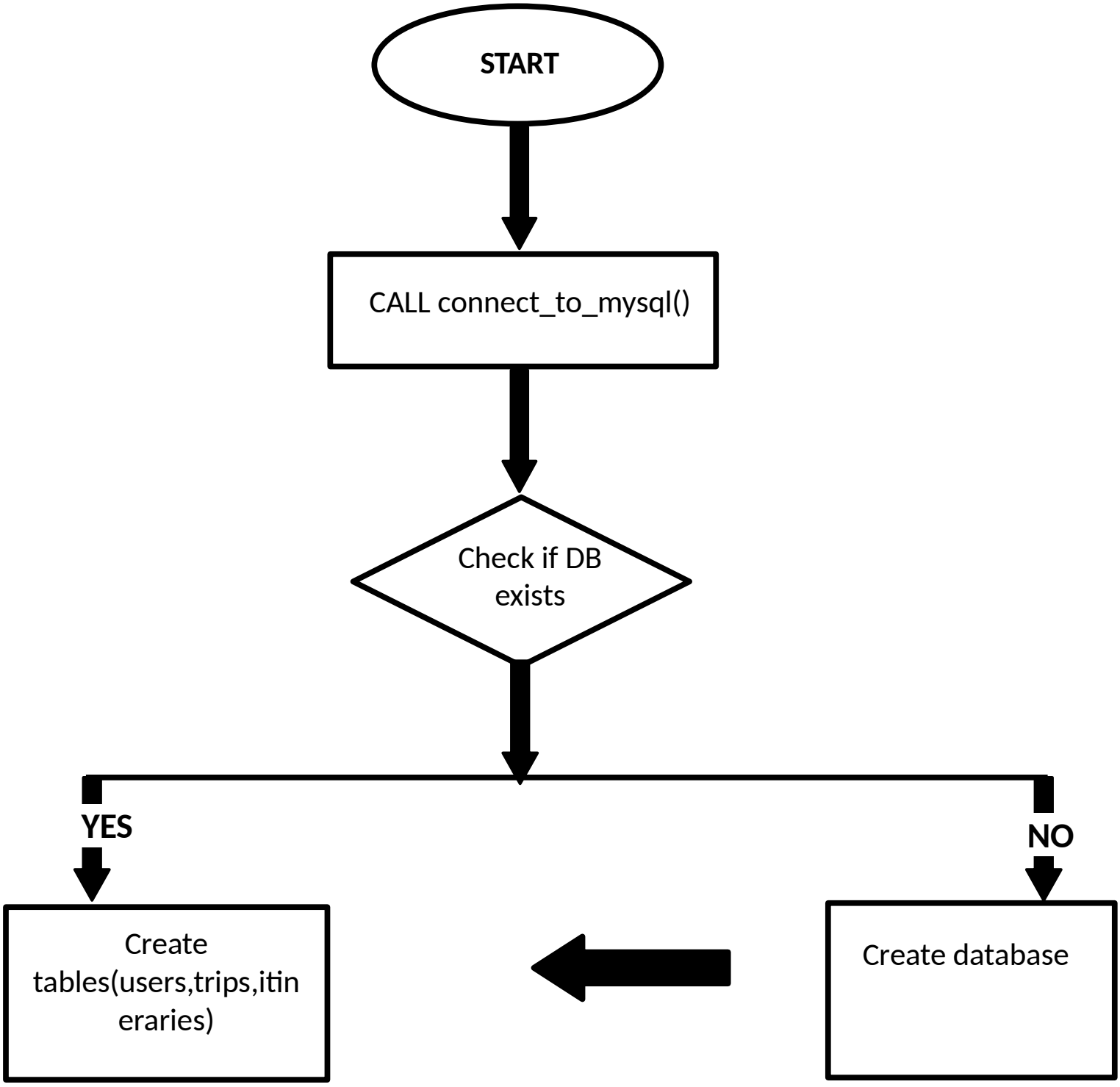
- CPU – Intel i3 (recommended)
- RAM – 4GB (recommended)

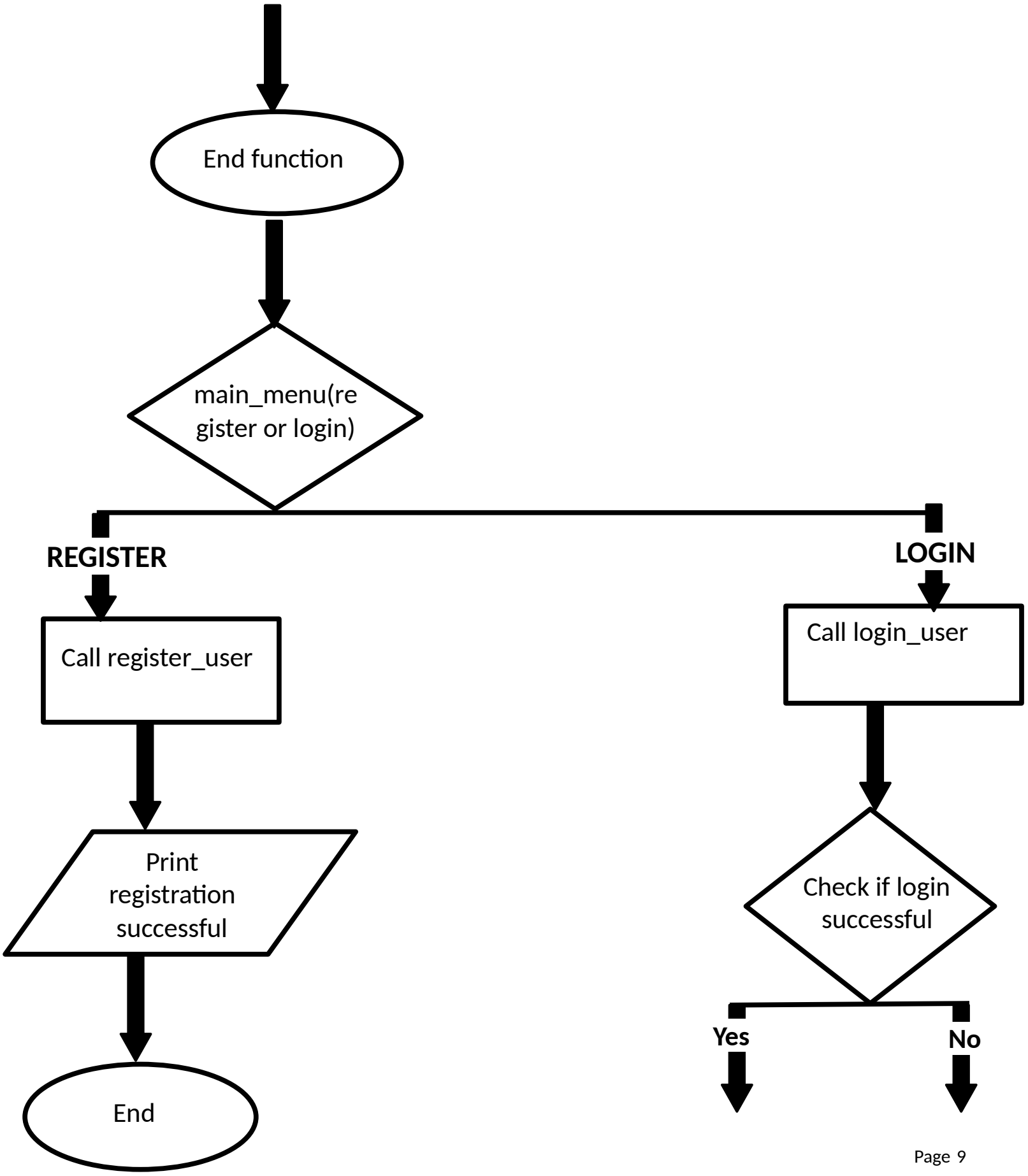
Software Requirements:

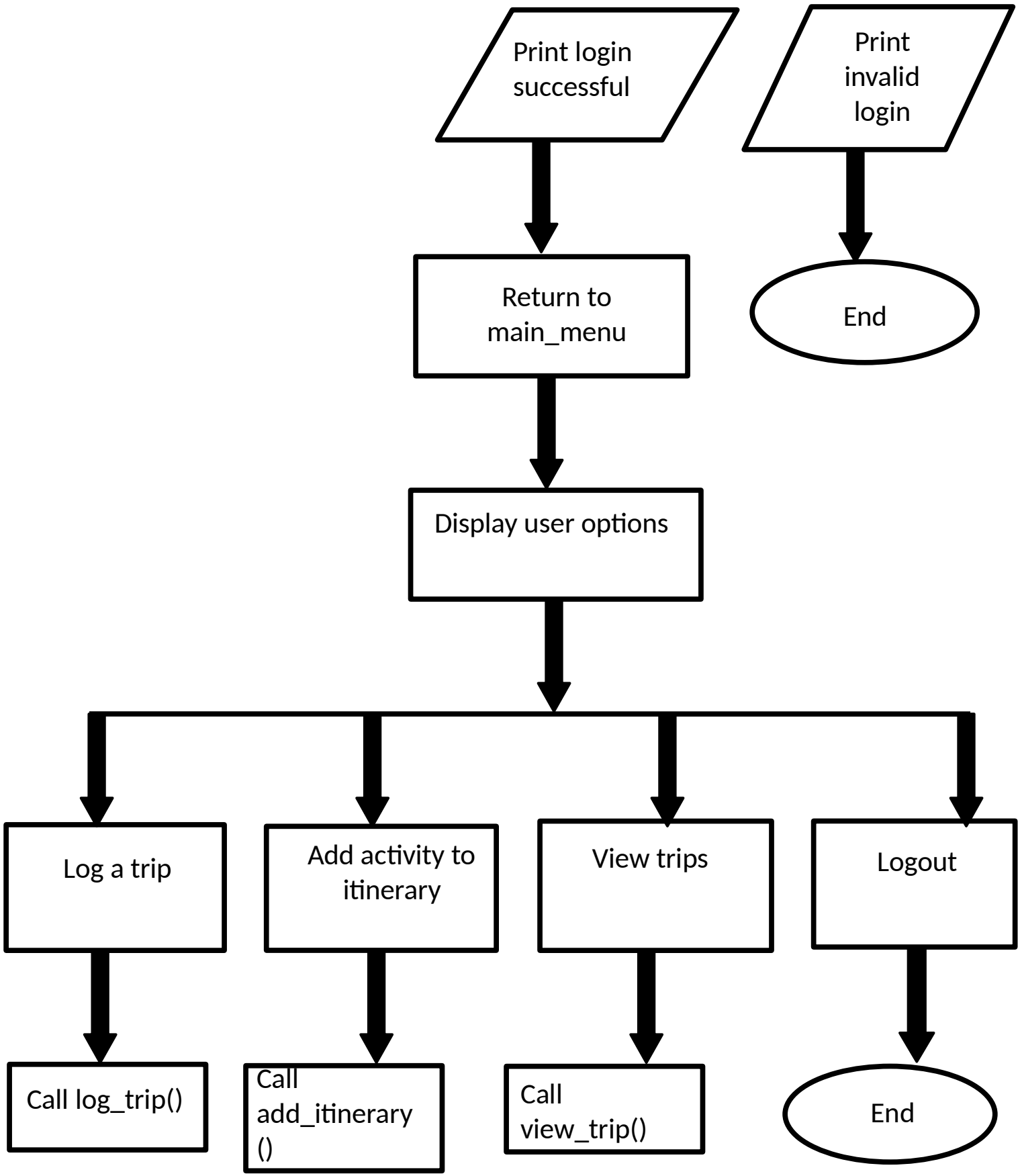
- Operating System – any (Linux/Windows/Mac)
- Python (>=3.5.x)
- mysql-connector-python (>=9.1.0)
- Database - MySQL

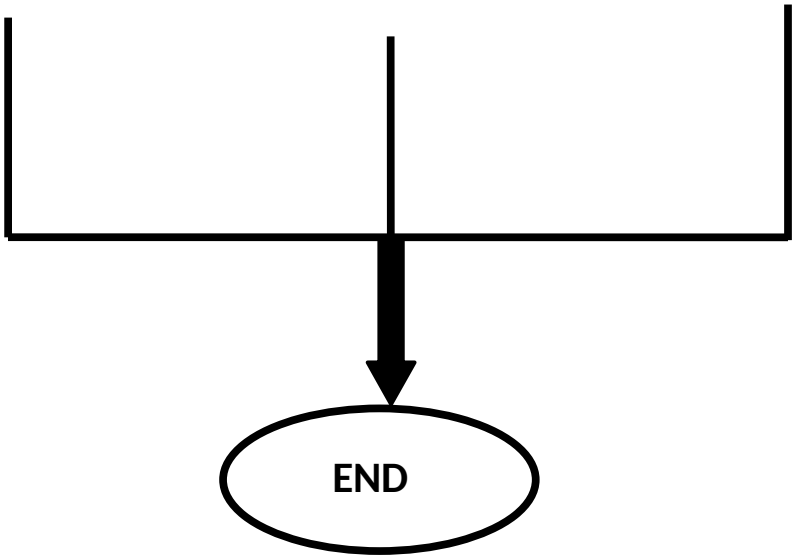
SYSTEM DESIGNS

FLOWCHART:




End function
main_menu(re gister or login)
REGISTER
LOGIN
Call register_user
Print registration successful
End
Call login_user
Check if login successful
Yes
No


Print login successful
Print invalid login
Return to main_menu
End
Display user options
Log a trip
Add activity to itinerary
View trips
Logout
Call log_trip()
Call add_itinerary ()
Call view_trip()
End



DATABASE DESIGNS:

Tables:

- users(user_id, username, password)
- trips(trip_id, user_id, destination, start_date, end_date)
- itineraries(itinerary_id, trip_id, activity)

IMPLEMENTATION

CODE SNIPPETS

```
import mysql.connector
import config

def connect_to_mysql():
    conn = mysql.connector.connect(host = config.DB_HOST, user = config.DB_USER, database = config.DB_NAME, password = config.DB_PASSWORD)
    cursor = conn.cursor()

    # Check if the database exixts
    cursor.execute("show databases;")

        existing_databases = [row[0] for row in cursor.fetchall()]

        if config.DB_NAME not in existing_databases:

            # Create the database if it doesn't exist
            cursor.execute(f"create database {config.DB_NAME}")

            print(f"Database '{config.DB_NAME}' created successfully")

        # Create users table
        cursor.execute('''create table if not exists users(
            user_id int primary key auto_increment,
            username varchar(20),
            password varchar(15)
            );''')

```

```
        # Create trips table
        cursor.execute('''create table if not exists trips(
            trip_id int primary key auto_increment,
            user_id int,
            destination varchar(20),
            start_date date,
            end_date date,
            foreign key(user_id) references users(user_id)
            );''')

        # Create itineraries table
        cursor.execute('''create table if not exists itineraries(
            itinerary_id int primary key auto_increment,
            trip_id int,
            activity varchar(100),
            foreign key(trip_id) references trips(trip_id)
            );''')

        # Close cursor and connection to the MySql server
        cursor.close()
        conn.close()
    connect_to_mysql()

    import mysql.connector

    # Function to register a new user
    def register_user(username,password):
        conn = mysql.connector.connect(host = config.DB_HOST, user = config.DB_USER, database = config.DB_NAME, password = config.DB_PASSWORD)
        cursor = conn.cursor()

        cursor.execute('''insert into users(username,password)
            values(%s,%s)''' , (username,password))

```

```
60.         conn.commit()
61.         conn.close()
62.
63.         print(f"User '{username}' registered
  successfully")
64.
65.     # Function to log in a user
66.     def login_user(username,password):
67.         conn = mysql.connector.connect(host =
  config.DB_HOST, user = config.DB_USER, database =
  config.DB_NAME, password = config.DB_PASSWORD)
68.         cursor = conn.cursor()
69.
70.         cursor.execute('''select * from users
71.             where username = %s and password = %s''',
72.             (username,password))
73.
74.         user = cursor.fetchone()
75.
76.         conn.close()
77.
78.         if user:
79.             print(f"Welcome, {user[1]}")
80.             return user[0]       # Return user_id
81.
82.         else:
83.             print("Invalid login... Please check username
  or password and try again!")
84.             return None
85.
86.     # Function to log a trip
87.     def log_trip(user_id,destination,start_date,end_date):
88.         conn = mysql.connector.connect(host =
  config.DB_HOST, user = config.DB_USER, database =
  config.DB_NAME, password = config.DB_PASSWORD)
89.         cursor = conn.cursor()
90.
```

```
91.         cursor.execute('''insert into
  trips(user_id,destination,start_date,end_date)
92.               values(%s, %s, %s, %s)''',
  (user_id,destination,start_date,end_date))
93.
94.         conn.commit()
95.         conn.close()
96.
97.         print(f"Trip to '{destination}' logged
  successfully")
98.
99.    # Function to add an activity to the itinerary for a
  trip
100.   def add_itinerary(trip_id,activity):
101.        conn = mysql.connector.connect(host =
  config.DB_HOST, user = config.DB_USER, database =
  config.DB_NAME, password = config.DB_PASSWORD)
102.        cursor = conn.cursor()
103.
104.        cursor.execute('''insert into
  itineraries(trip_id,activity)
105.                  values(%s, %s)''', (trip_id,activity))
106.
107.        conn.commit()
108.        conn.close()
109.
110.        print("Activity added to itinerary")
111.
112.   # Function to display user's trips
113.   def view_trips(user_id):
114.        conn = mysql.connector.connect(host =
  config.DB_HOST, user = config.DB_USER, database =
  config.DB_NAME, password = config.DB_PASSWORD)
115.        cursor = conn.cursor()
116.
117.        cursor.execute('''select * from trips
118.                       where user_id = %s''', (user_id,))
119.
```

```
120.       trips = cursor.fetchall()
121.
122.       if trips:
123.           print("\nYour Trips:")
124.           for trip in trips:
125.               print(f"Trip ID: {trip[0]} , Destination:
  {trip[2]}, Dates: {trip[3]} to {trip[4]}")
126.
127.       else:
128.           print("No trips found!")
129.
130.       conn.close()
131.
132.   def main_menu():
133.       print("Welcome to the Travel Planner")
134.
135.       action = input("Do you want to: (1)Register or
  (2)Login? ---> ")
136.
137.       if action == "1":
138.            username = input("Enter username: ")
139.            password = input("Enter password: ")
140.            register_user(username,password)
141.
142.       elif action == "2":
143.           username = input("Enter username: ")
144.           password = input("Enter password: ")
145.           user_id = login_user(username,password)
146.
147.           if user_id:
148.               while True:
149.                   print("\nOptions:")
150.                   print("1. Log a new trip")
151.                   print("2. Add activity to itinerary")
152.                   print("3. View your trips")
153.                   print("4. Logout")
154.
```

```
                    choice = input("Choose an option from above: ")

                    if choice == "1":
                        destination = input("Enter destination: ")
                        start_date = input("Enter start date (YYYY-MM-DD): ")
                        end_date = input("Enter end date (YYYY-MM-DD): ")
                        log_trip(user_id, destination, start_date, end_date)

                    elif choice == '2':
                        trip_id = int(input("Enter trip ID to add activity to: "))
                        activity = input("Enter activity: ")
                        add_itinerary(trip_id, activity)

                    elif choice == '3':
                        view_trips(user_id)

                    elif choice == '4':
                        print("Logging out...")
                        break

                    else:
                        print("Invalid option, try again.")

main_menu()
```

Output Window

```
Welcome to the Travel Planner
Do you want to: (1)Register or (2)Login? ---> 1
Enter username: Khushi
Enter password: abcd
User 'Khushi' registered successfully
```

```
Welcome to the Travel Planner
Do you want to: (1)Register or (2)Login? ---> 2
Enter username: Khushi
Enter password: abcd
Welcome, Khushi

Options:
1. Log a new trip
2. Add activity to itinerary
3. View your trips
4. Logout
Choose an option from above: 1
Enter destination: Delhi
Enter start date (YYYY-MM-DD): 2024-12-10
Enter end date (YYYY-MM-DD): 2024-12-17
Trip to 'Delhi' logged successfully
```

```
Options:
1. Log a new trip
2. Add activity to itinerary
3. View your trips
4. Logout
Choose an option from above: 2
Enter trip ID to add activity to: 1
Enter activity: Lotus Temple
Activity added to itinerary

Options:
1. Log a new trip
2. Add activity to itinerary
3. View your trips
4. Logout
Choose an option from above: 3

Your Trips:
Trip ID: 1 , Destination: Delhi, Dates: 2024-12-10 to 2024-12-17
```

```
Options:
1. Log a new trip
2. Add activity to itinerary
3. View your trips
4. Logout
Choose an option from above: 4
Logging out...
```

```
Welcome to the Travel Planner
Do you want to: (1)Register or (2)Login? ---> 2
Enter username: Khushi
Enter password: qwer
Invalid login... Please check username or password and try again!
```

CONCLUSION

- Enhances user engagement
- Streamlines travel planning
- Ensures user data privacy
- Personalized platform
- Time efficient
- Performance optimization: Achieved through code refactoring, caching and optimizing database queries.
- User testing feedback: Incorporated user feedback for overall experience

FUTURE ENHANCEMENTS

- If VR(Virtual Reality) feature is ever added so the users can have tours for pre-planning trips.
- AI (Artificial Intelligence) powered travel recommendations for better experience
- Personalized travel budgeting and expense tracking
- Real-time location sharing and safety tracking

REFERENCES

- Wikipedia
- Meta AI
- ChatGPT
- Computer Science with python Class-XII by Sumita Arora
- www.ncert.co.in