Malnad College of Engineering

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

Hassan, Karnataka, India – 573202

Course Title: Data Structures

Course Code: 23AI304

Project Based-Learning

Project Title: "To simulate the working of Flight Information Display System (FIDS) present at the airport**"**

Submitted By:

Rakshak D	- 4MC23CI043
Shesha Prasad H	- 4MC23CI050
Nithin H C	- 4MC23CI036
Likith Shetty	- 4MC23CI024

Under the guidance of

Dr. Balaji Prabhu B V

Associate Professor and HOD,

Dept. of CSE (AI&ML)

Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning)

2024-2025

Problem Statement:

The details of flights such as its number, source, departure time, destination, arrival, travel duration, adult fare is given in a text file. Implement a program using appropriate data structure to compute:

1. Suggest Direct Flights: Filter and display flights where the source directly connects to the destination

2. Suggest 1-Stop and 2-Stop Flights: Use intermediate cities to find connecting flights (e.g., Bengaluru -> Hyderabad -> Delhi).

3. Suggest Cheapest Flights: Use the fare data to find the cheapest flight(s) for a given source and destination, considering all possible routes (direct, 1-stop, or 2-stop)

Objective of the Project:

The project aims to develop a user-friendly flight information system that allows users to:

1. View all available flights.
2. Search for flights based on source and destination cities.
3. Create and manage user accounts with secure login functionality.

Overview:

This program provides an interactive interface and allows users to check available flights, search for flights from specific sources to destinations, and manage user accounts. The system supports operations such as filtering flights by location, viewing direct and indirect flights, and calculating travel costs. The details of flights such as its number, source, departure time, destination, arrival, travel duration, adult fare can also be seen in this program.

Data Structure Used: Structure

Here we have used **Structures** because they're are particularly useful when you need to represent and work with complex data that consists of multiple components.

```
struct Flight
{
    char flightID[10];
    char source[20];
    char arrivalTime[10];
    char destination[20];
    char departureTime[10];
    float flightTime;
    float fare;
}flights[20];
```

Algorithm for FIDS:

1. Initialization

1. Start the program.
2. Declare and define structures:
    - Flight: To store flight details.
    - city: To store city names.
    - userdetails: To manage user account data.
3. Initialize arrays for flights, cities, and users.

2. Load Data

1. Open FLIGHTDETAILS.csv to extract flight details:
    - Read each line.
    - Load data into the flights array.

2. Open PLACES.csv to extract city names:
    - Read each city name into the city_list array.
3. Open Userdetails.csv to extract user data:
    - Read each line.
    - Parse data into the user array.

3. User Authentication

1. Ask the user:

   Asking the user whether he has an acc or not and selecting an option from “Y” or “N”.
2. If Y:
    - Call the login function:
        - Ask for username or email.
        - Validate input and match it with stored data.
        - Retry up to 3 times for incorrect passwords.
        - If successful, display a welcome message.
        - If failed, offer to create a new account.
3. If N:
    - Call the createnewaccount() function:
        - Validate and store a new username, email, and password.
        - Save new user data to Userdetails.csv file.

4. Display Menu

Present the user with options:

1: View all flights.

2: View flights from a specific source city.

3: Search flights between two cities.

4: Logout.

5. Execute User's Choice

1. **Option 1**: Display all flights.
    - Loop through the flights array.
    - Print details for each flight.
2. **Option 2**: Display flights from a specific source.
    - List available source cities from the city_list.
    - Accept user input for the source city.
    - Loop through the flights array:
        - Print flights matching the selected source city.
3. **Option 3**: Search flights between two cities.
    - List available source cities from the city_list.
    - Accept user input for the source city.
    - List remaining cities for destination selection.
    - **Direct Flights**:
        - Loop through flights to find direct matches.
        - Print matching flight details.
    - **Indirect Flights**:
        - Use nested loops to find connecting flights.
        - Calculate total time and fare for connections.
        - Print indirect flight details with calculations.

4. **Option 4**: Logout.
    - Display a logout message.
    - Exit the program.

6. Repeat Menu

1. Continue displaying the menu until the user selects "Logout."
2. If an invalid input is provided, prompt the user to re-enter a valid option.

7. Termination

1. Save any changes (e.g., new user data) to files.
2. Close all open files.
3. End the program.

Result:

Display of MENU:

```
Welcome!! user123
How may i help you
Select a option
1.To check all the flights available.
2.To check all the flights from once place.
3.To check all the flights from one place to another place.
4.Logout
```

This is the menu that is displayed when the user successfully logs in/successfully creates an account in the flight finder.

Display of all Available Flights:

Flight ID	Source	Arrival	Destination	Departure	Time	Fare
F001	Bengaluru	06:00	Mumbai	08:30	2.50	4500.00
F002	Mumbai	09:30	Delhi	12:00	2.50	5000.00
F003	Delhi	14:00	Kolkata	17:00	3.00	5500.00
F004	Bengaluru	07:00	Hyderabad	08:45	1.75	4000.00
F005	Hyderabad	10:00	Mumbai	12:30	2.50	4700.00
F006	Bengaluru	09:00	Chennai	10:30	1.50	3000.00
F007	Chennai	12:00	Kolkata	15:30	3.50	6000.00
F008	Kolkata	17:30	Guwahati	19:00	1.50	3500.00
F009	Delhi	11:00	Bengaluru	14:30	3.50	7000.00
F010	Hyderabad	15:00	Delhi	17:30	2.50	4800.00
F011	Mumbai	18:00	Bengaluru	20:30	2.50	4200.00
F012	Kolkata	19:30	Chennai	22:00	2.50	5200.00
F013	Bengaluru	16:00	Delhi	19:00	3.00	6500.00
F014	Mumbai	13:00	Hyderabad	15:30	2.50	4700.00
F015	Bengaluru	20:00	Kolkata	23:30	3.50	7200.00
F016	Chennai	08:00	Mumbai	10:30	2.50	5200.00
F017	Delhi	06:30	Hyderabad	09:00	2.50	5000.00
F018	Guwahati	10:30	Kolkata	12:00	1.50	3500.00
F019	Bengaluru	05:30	Delhi	08:30	3.00	6100.00
F020	Mumbai	19:00	Chennai	21:30	2.50	4600.00

When the user selects “Option 1”, it displays all the available flights from all places to one another. It doesn’t represent any indirect or direct flights and just flights from one place to another.

Display of all available flights from One Place:

Chennai

Flight ID	Source	Arrival	Destination	Departure	Time	Fare
F007	Chennai	12:00	Kolkata	15:30	3.50	6000.00
F016	Chennai	08:00	Mumbai	10:30	2.50	5200.00

When the user selects “Option 2”, it asks for a particular place and when we choose a particular place, it displays all the available flights from that place. Here also, there is no mention of direct or indirect flights but give all the flights from the selected place.

Display of all flights from one place to another:

```
                        DIRECT FLIGHTS
Flight ID  Source     Arrival    Destination Departure  Time       Fare
--------------------------------------------------------------------------
                      NO DIRECT FLIGHTS ):
--------------------------------------------------------------------------

                      INDIRECT FLIGHTS
Flight ID  Source     Arrival    Destination Departure  Time       Fare
--------------------------------------------------------------------------
**FLIGHT 1**

F015       Bengaluru  20:00      Kolkata    23:30      3.50       7200.00

**FLIGHT 2**

F008       Kolkata    17:30      Guwahati   19:00      1.50       3500.00

TOTAL TIME = 5.00 hours
TOTAL FARE = Rs. 10700.00
--------------------------------------------------------------------------

               CHEAPEST INDIRECT FLIGHT
--------------------------------------------------------------------------

FLIGHTID'S: F015, F008

CHECKPOINTS: Bengaluru->Kolkata->Guwahati

FARE = Rs. 10700.00
--------------------------------------------------------------------------
```

When a user option 3, the user must select a source and destination for it. Then, it gives all the direct and indirect flights available. It also gives the total time and fare required to complete that indirect flight. It also suggests the cheapest indirect flight and gives the route that is followed and fare of the the indirect flight.

Conclusion:

The **Flight Information System** is a well-structured program that provides a comprehensive platform for managing flight data and user accounts. The user authentication system is flexible, supporting both login and account creation with validation for usernames, email formats, and password strength. Data handling is efficient, with flight details and user accounts loaded from external CSV files, ensuring reusability and easier accessbility in terms of data operations. The flight search feature allows users to view all flights, filter flights based on a specific source city, search for both direct and indirect flights between two locations, complete with calculations for total fare and travel time for indirect routes.

Overall, our program is designed to ensure user-friendliness, with clear prompts and a structured flow of operations. It provides a good interface, guiding users through account setup, login, and flight queries. Hence, it is a good approach designed to tackle the problem statement given.