INTRODUCTION TO COMPUTER PROGRAMMING

Project Report: Data Management of Airport in C Language

Names	Muneeb Ahmad
Registration Numbers	FA21-BEE-145
Class	BEE-2C
Instructor's Name	Dr. Sayed Junaid Nawaz

Lab Assessment

Pre-Lab	**In-Lab**	**Post Lab**			**Total**
		Data Presentation	**Data Analysis**	**Writing Style**	

Project Report

Airport Data Management System in C

1. Objective:

To design and implement a C program that manages data for an airport system, handling both:

> **Flight management** (storing and retrieving flight data like flight number, destination, and departure time), and
>
> **Passenger passport checks**, allowing the system to track passengers linked to specific flights.

2. Introduction

Airports deal with a large amount of data, including scheduling flights and verifying passenger identity and bookings.
This project simulates a **basic airport data management system in the C programming language**, which helps understand concepts such as:

a) Structure data types,
b) Array-based databases,
c) Searching and displaying data.

3. Modules Implemented

1. Flights Management

a) Add new flight records.
b) Display all flights.
c) Search for a flight by flight number.

2. Passengers Management with Passport Check

a) Add new passengers with passport number, name, and booked flight.
b) Display all passengers.
c) Search passengers by passport number.

4. Working Principle

1. Uses **structs** to define Flight and Passenger data types.
2. Maintains two arrays:

> flights[] to store flight records.

passengers[] to store passenger records.

3. Supports adding, listing, and searching data.
4. Ensures a **simple menu-driven interface** using a do-while loop and switch statements.
5. Demonstrates the use of string operations like strcmp, strcpy, fgets, and strcspn to handle text input safely.

5. Features of the C Code

1. Uses **structures (**struct**)** to logically organize flight and passenger data.
2. Handles string inputs with fgets to avoid buffer overflow.
3. Provides modular functions:

a. addFlight() to input and store flight data.
b. displayFlights() to show all flights in tabular format.
c. searchFlight() to find flights by flight number.
d. addPassenger() to check passport and store passenger info.
e. displayPassengers() to list all passengers.
f. searchPassenger() to find by passport number.

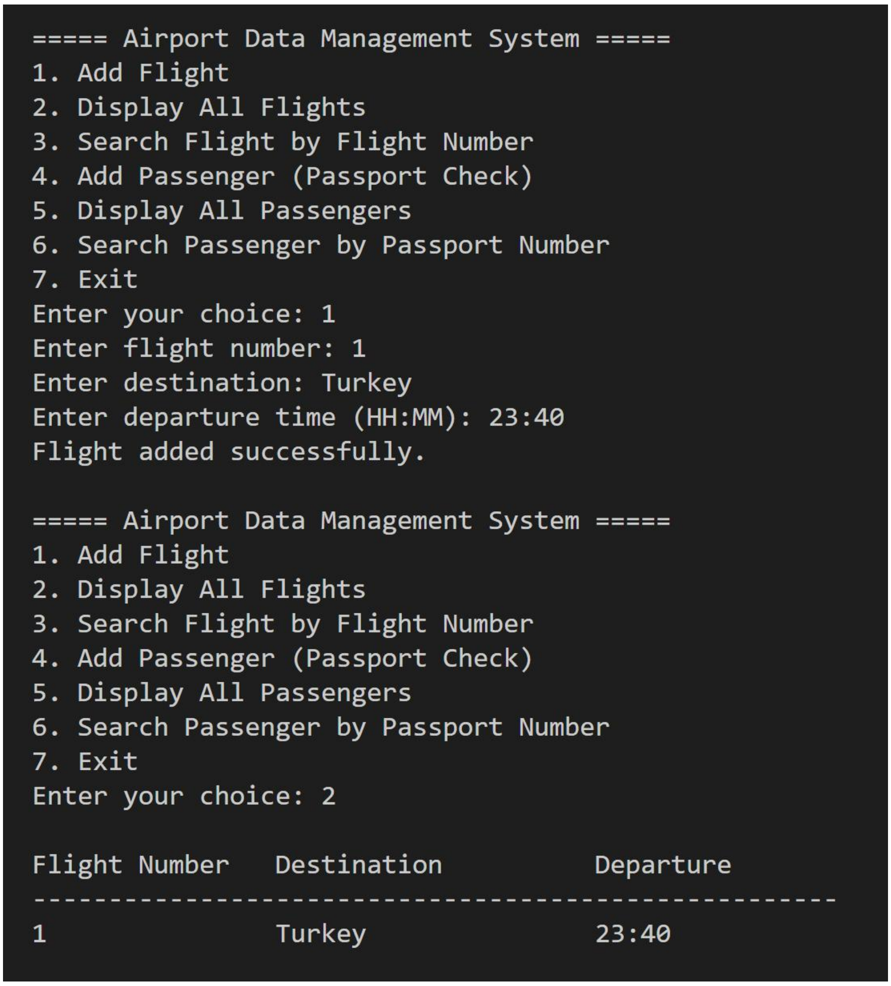
6. Advantages

1. Easy to expand: can add more fields (like seat number, gate, etc.).
2. Demonstrates fundamental **data management concepts** (arrays, searching, structures).
3. Serves as a good educational tool for learning simple, file-less databases in C.

7. Limitations

1. Data is lost when the program closes (as it currently does not use file storage).
2. No advanced checks (like verifying if a flight exists before booking a passenger on it).

8. Output



```
===== Airport Data Management System =====
1. Add Flight
2. Display All Flights
3. Search Flight by Flight Number
4. Add Passenger (Passport Check)
5. Display All Passengers
6. Search Passenger by Passport Number
7. Exit
Enter your choice: 1
Enter flight number: 1
Enter destination: Turkey
Enter departure time (HH:MM): 23:40
Flight added successfully.

===== Airport Data Management System =====
1. Add Flight
2. Display All Flights
3. Search Flight by Flight Number
4. Add Passenger (Passport Check)
5. Display All Passengers
6. Search Passenger by Passport Number
7. Exit
Enter your choice: 2

Flight Number   Destination         Departure
-----------------------------------------------------
1               Turkey              23:40
```

10. Conclusion

This project demonstrates how to build a **basic airport data management system in C** that integrates flight scheduling and passport checks for passengers.
It highlights important programming concepts such as structures, string handling, arrays, and simple database-like operations.
It can easily be expanded with **file handling or dynamic memory allocation** to manage larger datasets.