PROJECT REPORT

COURSE**:**	**CDS101**
TOPIC:	**Flight Reservation System**
DATE OF SUBMISSION:	**23 November, 2024**
INSTRUCTORS:	**Dr. Sweta Kumari, Dr. Sumit Shekhar**

STUDENT DETAILS

Name:	Adya Singhal	Name:	Rishit Kamboj
Roll No.:	2410110027	Roll No.:	2410110598
SNU-Id:	as401@snu.edu.in	SNU-Id:	rk846@snu.edu.in

INTRODUCTION OF THE TOPIC

A Flight Reservation System (HMS) is a real-time problem of a person relatable getting the tension to book the flight tickets at the cheapest price to book tickets for their journey.

We have written a C code that does the following:

1. Stores the Flight Details: such as Flight name, from where to where its flying, no. of seats etc.
2. Book the flight ticket
3. Cancel the flight ticket
4. Update the passenger information on Booking (such as name, age, phone number, etc.)
5. View the passengers that are travelling in the flight

This application uses structured data to manage flights and passenger details effectively. It demonstrates key concepts of the C programming language, including arrays, structures, string manipulation, and control flow.

FUNCTIONALITY

1. BOOKING FLIGHTS
   Feature: Allows passengers to book seats on a flight.
   Process:
   - Users input the flight number, passenger name, and the number of seats to book.
   - The system checks the availability of seats and updates the data structures accordingly.
   - Passenger details are stored in an array.

2. CANCELLING BOOKINGS
   Feature: Handles cancellation requests for previously booked seats.
   Process:
   - Users input the flight number and the number of seats to cancel.
   - The system verifies and adjusts the available seats for the specified flight.

3. UPDATING PASSENGER INFORMATION
   Feature: Allows modifications to a passenger's name, age, and contact number.
   Process:
   - The system searches for a specific passenger in the passenger array.
   - The corresponding record is updated with new information.

4. VIEWING PASSENGERS
   Feature: Displays a list of all passengers on a specific flight.
   Process:
   - Users input the flight number, and the system retrieves and prints passenger details for that flight.

KEY C CONCEPTS USED

1. DATA STRUCTURES
   Structures (`struct`) :
   - Used to define custom data types for flights (`Flight`) and passengers (`Passenger`).
   - Encapsulates attributes such as `flightNumber`, `from`, `to`, `totalSeats`, `availableSeats` for flights, and `name`, `flightNumber`, `age`, `phone` for passengers.

2. ARRAYS
   - Arrays are used to store a fixed number of flights (`flights[MAX_FLIGHTS]`) and passengers (`passengers[MAX_PASSENGERS]`).

3. STRING HANDLING
   Functions like `strcpy` and `strcmp`:
   - `strcpy`: Used for copying strings, e.g., flight numbers and passenger names.
   - `strcmp`: Used for comparing strings to match flight numbers or passenger names.

4. CONTROL FLOW
   Loops:
   - `for` loops iterate over arrays of flights and passengers to locate or update records.

   Conditional Statements:
   - `if` statements are used to validate inputs, check seat availability, and manage user choices.

5. INPUT AND OUTPUT
   `scanf` and `printf`:
   Used to take user input and display messages or results to the console.

6. FUNCTIONS
   Modular programming is used with separate functions for each feature (`bookFlight`, `cancelFlight`, `updatePassengerInfo`, `viewPassengers`). This enhances readability, maintainability, and reusability of code.

7. MACROS
   `define`:
   Constants like `MAX_FLIGHTS` and `MAX_PASSENGERS` are defined using macros for better configurability and readability.

CODE HIGHLIGHTS

1. INITIALIZATION OF FLIGHTS:
   Predefined data for two flights is initialized in the `initializeFlights()` function, which demonstrates efficient use of string copying and structure manipulation.

2. DYNAMIC SEAT MANAGEMENT:
   Updates to `availableSeats` in both booking and cancellation processes ensure accurate seat tracking.

3. PASSENGER DATA MANAGEMENT:
   A structured array (`passengers[]`) maintains all passenger records with details like name, flight number, and age.

4. ERROR HANDLING:

Checks for invalid inputs (e.g., non-existent flight numbers or overbooking requests) ensure robustness.

5. MENU-DRIVEN INTERFACE:
   A simple loop-driven menu allows the user to perform operations interactively.

APPLICATION OF C PROGRAMMING KNOWLEDGE

This project showcases fundamental and intermediate C programming skills:

Logical Problem Solving: Breaking down the task into manageable sub-tasks with separate functions for each operation.

Data Management: Efficient use of arrays and structures to handle large data sets.

Code Optimization: Using loops and conditional checks to minimize redundancy.

User Interaction: Dynamic input and output for a seamless user experience.

This Airplane Management System serves as a foundational project for understanding real-world applications of C in system management and data handling, highlighting the power of structures, arrays, and modular programming.