CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

PRESENTATION ON AIRLINE RESERVATION SYSTEM

Program Name: BCA(Data Science)
SubjectName/Code:Data Structures Lab(24CAP-152)

Submitted by :
Vikita
24BCD-1(B)

Submitted to :
Ms. Mehak Bhatia

Airline Reservation System

The Airline Reservation System (ARS) is a software application designed to automate the process of booking airline tickets, managing flight schedules, and providing services to both passengers and staff. The system facilitates easy access for customers to view available flights, check prices, and book tickets. It also helps airlines manage flight inventory, customer details, reservations, and other essential data. The ARS provides various functionalities to streamline the flight booking process, making it efficient, user-friendly, and accessible.

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Core Functionalities

1 **Booking**

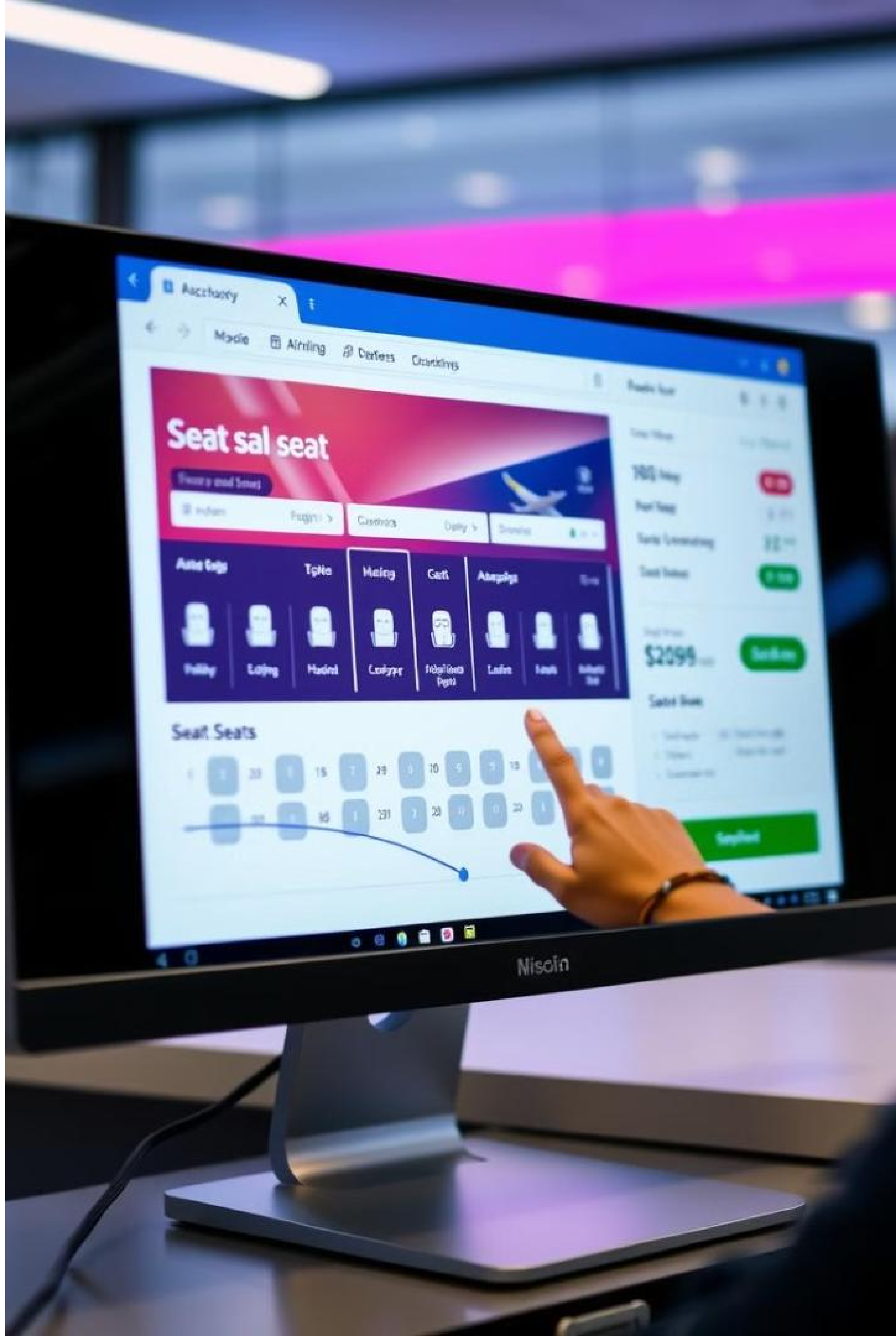
Implement a booking function to create new reservations. Capture passenger details and flight information.

2 **Cancelling**

Enable users to cancel existing reservations. Ensure proper record deletion and seat availability updates.

3 **Modifying**

Allow modifications to existing reservations. Accommodate changes to dates, seats, or passenger information.



Technologies Used

1. Programming Language: C

•Used for its efficiency, speed, and structured approach.

2. Data Structures & Concepts

•**Arrays** – Store seat availability and reservations.

•**Structures (struct**) – Hold passenger details (name, flight number, seat number).

•**Loops & Functions** – Manage the booking, cancellation, and retrieval of flight details.

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

CODE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_FLIGHTS
#define MAX_SEATS 5

typedef struct {
    int flightNumber;
    char passengerName[50];
    int seatNumber;
} Reservation;

Reservation reservations[MAX_FLIGHTS][MAX_SEATS];
int seatAvailability[MAX_FLIGHTS] =
 {MAX_SEATS, MAX_SEATS, MAX_SEATS};
// Function to display available flights
void displayFlights() {
    printf("\nAvailable Flights:\n");
    printf("1. Flight 101\n");
    printf("2. Flight 202\n");
    printf("3. Flight 303\n");
}
```

```
// Function to book a ticket
void bookTicket() {
    int flight, seat;
    char name[50];

    displayFlights();
    printf("\nEnter Flight Number (1-3): ");
    scanf("%d", &flight);

    if (flight < 1 || flight > MAX_FLIGHTS ||
seatAvailability[flight - 1] == 0) {
        printf("Invalid selection or no available
seats!\n");
        return;
    }

    printf("Enter Passenger Name: ");
    scanf(" %[^\n]", name);
seat = MAX_SEATS - seatAvailability[flight - 1] + 1;
    reservations[flight - 1][seat - 1].flightNumber =
flight;
```

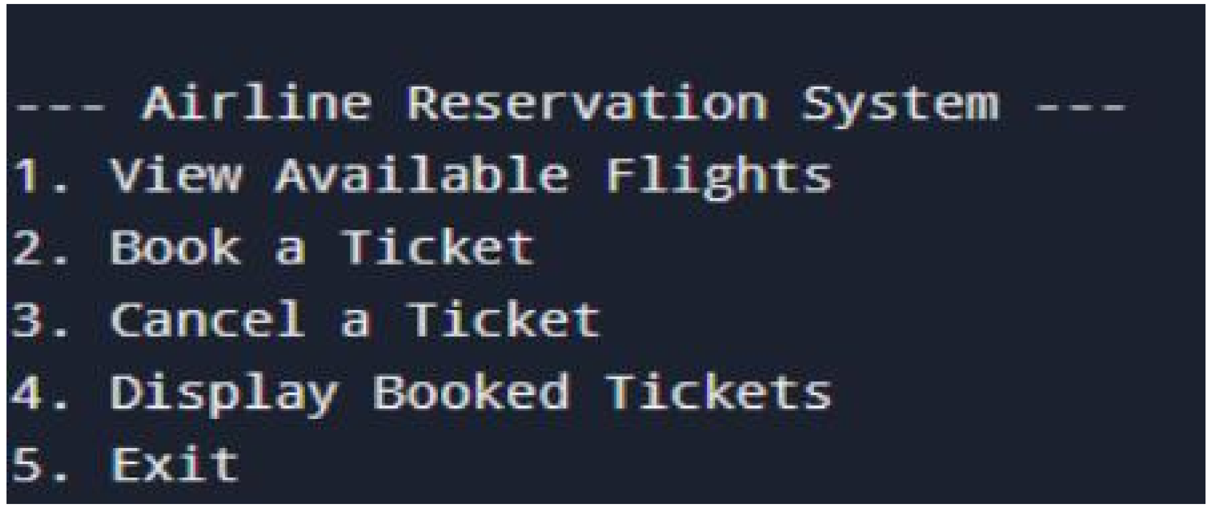
```
strcpy(reservations[flight - 1][seat - 1].passengerName,
name);
   reservations[flight - 1][seat - 1].seatNumber = seat;
seatAvailability[flight - 1]--;
   printf("Ticket booked successfully! Seat Number:
%d\n", seat);
}

// Function to cancel a ticket
void cancelTicket() {
   int flight, seat;
   displayFlights();

   printf("\nEnter Flight Number: ");
   scanf("%d", &flight);
   printf("Enter Seat Number: ");
   scanf("%d", &seat);
if (flight < 1 || flight > MAX_FLIGHTS || seat < 1 || seat
> MAX_SEATS || reservations[flight - 1][seat -
1].seatNumber == 0) {
      printf("Invalid flight or seat number!\n");
      return;
   }
```

```
// Cancel the ticket (reset reservation)
   reservations[flight - 1][seat - 1].flightNumber = 0;
   strcpy(reservations[flight - 1][seat -
1].passengerName, "");
   reservations[flight - 1][seat - 1].seatNumber = 0;

   seatAvailability[flight - 1]++;

   printf("Ticket canceled successfully!\n");
}

// Main Menu
int main() {
   int choice;

   do {
      printf("\n--- Airline Reservation System ---\n");
      printf("1. View Available Flights\n");
      printf("2. Book a Ticket\n");
      printf("3. Cancel a Ticket\n");
      printf("4. Display Booked Tickets\n");
      printf("5. Exit\n");
      printf("Enter your choice: ");
      scanf("%d", &choice);
switch (choice) {
            case 1: displayFlights(); break;
            case 2: bookTicket(); break;
            case 3: cancelTicket(); break;
            case 4: displayBookedTickets(); break;
            case 5: printf("Exiting... Thank you!\n");
break;
            default: printf("Invalid choice! Try again.\n");
        }
    } while (choice != 5);
    return 0;
}
```

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

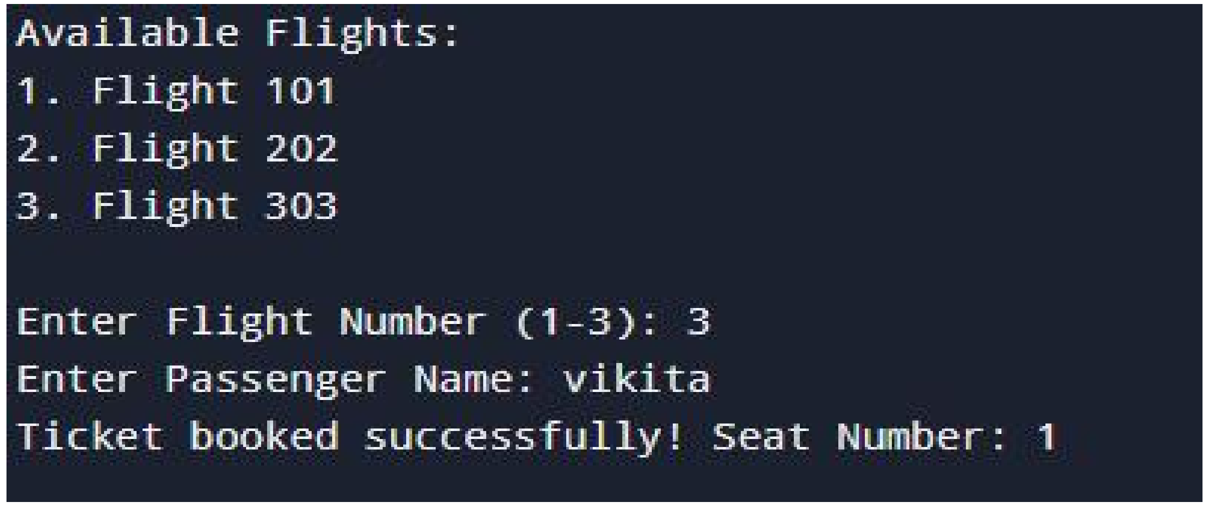
Output

Main Menu:



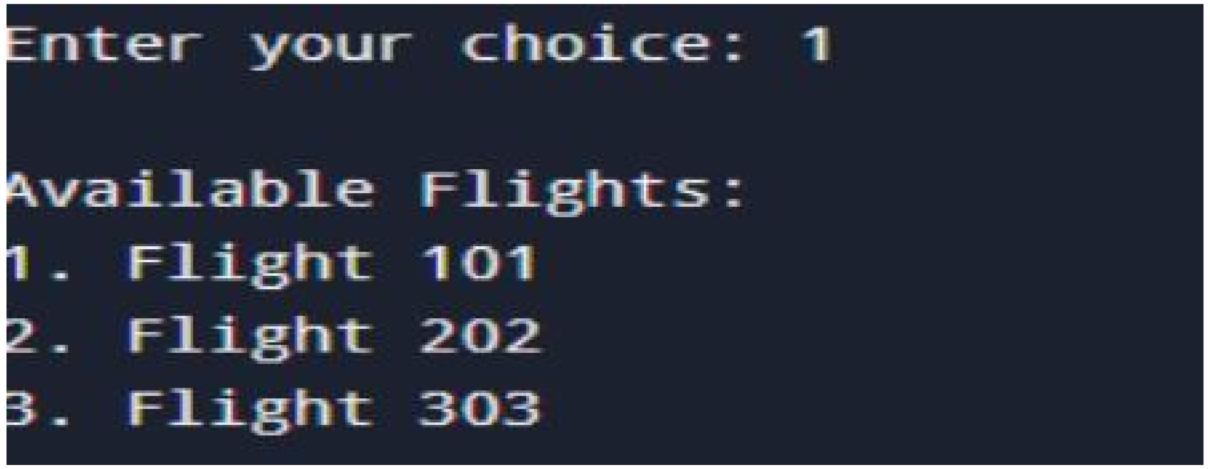
```
--- Airline Reservation System ---
1. View Available Flights
2. Book a Ticket
3. Cancel a Ticket
4. Display Booked Tickets
5. Exit
```

Book a Ticket:



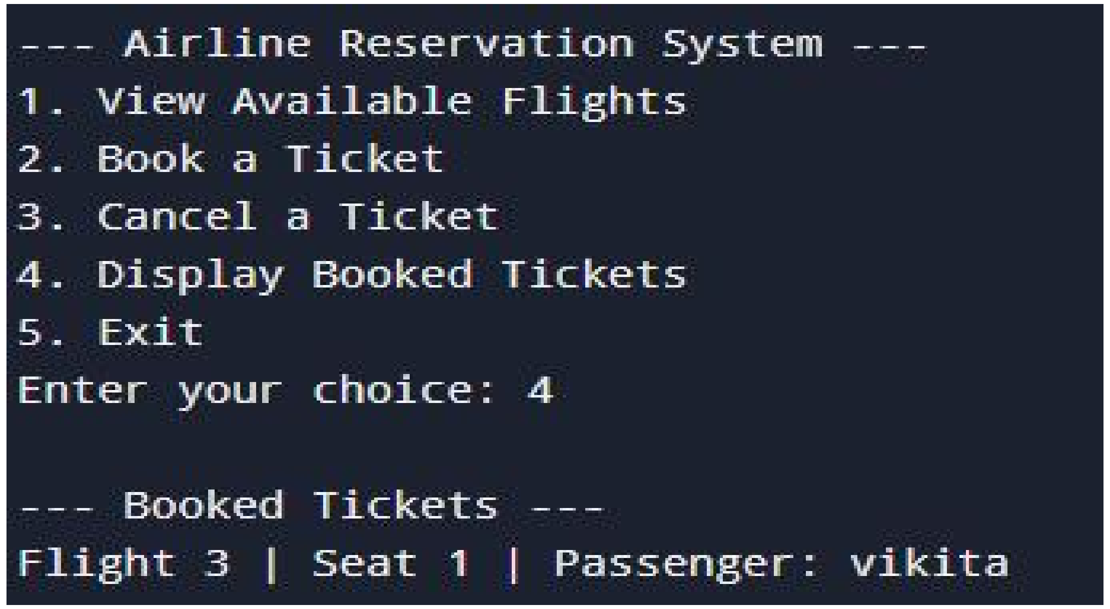
```
Available Flights:
1. Flight 101
2. Flight 202
3. Flight 303

Enter Flight Number (1-3): 3
Enter Passenger Name: vikita
Ticket booked successfully! Seat Number: 1
```

View Available Flights:



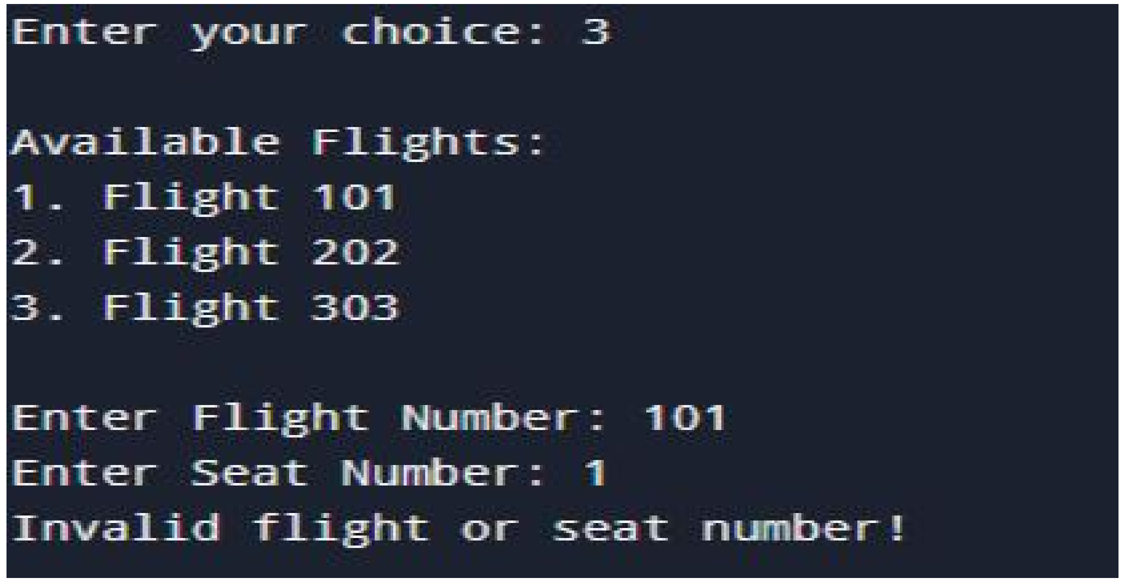
```
Enter your choice: 1

Available Flights:
1. Flight 101
2. Flight 202
3. Flight 303
```

Display Booked Tickets:



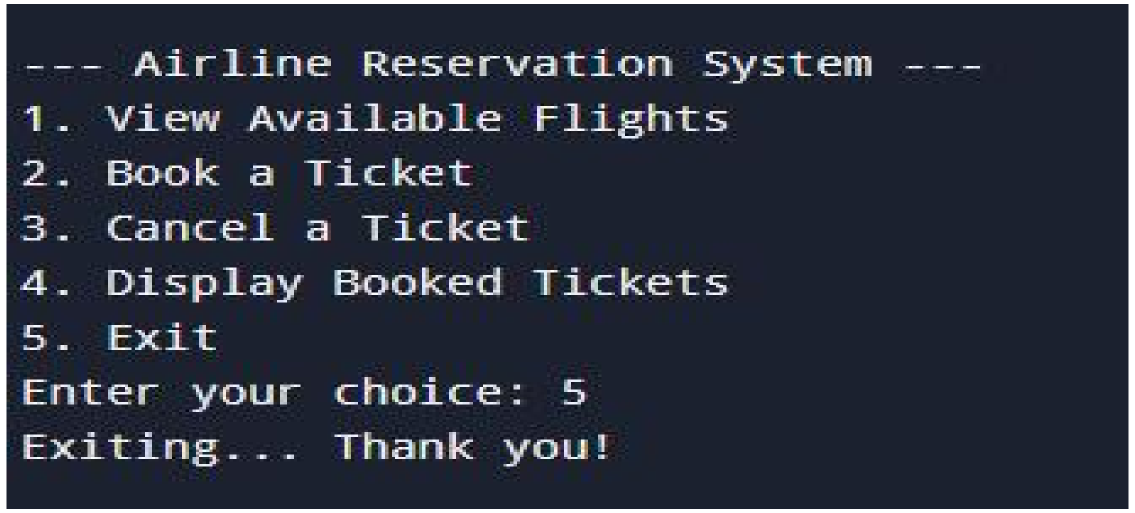
```
--- Airline Reservation System ---
1. View Available Flights
2. Book a Ticket
3. Cancel a Ticket
4. Display Booked Tickets
5. Exit
Enter your choice: 4

--- Booked Tickets ---
Flight 3 | Seat 1 | Passenger: vikita
```

Cancel a Ticket:



```
Enter your choice: 3

Available Flights:
1. Flight 101
2. Flight 202
3. Flight 303

Enter Flight Number: 101
Enter Seat Number: 1
Invalid flight or seat number!
```

Exit:

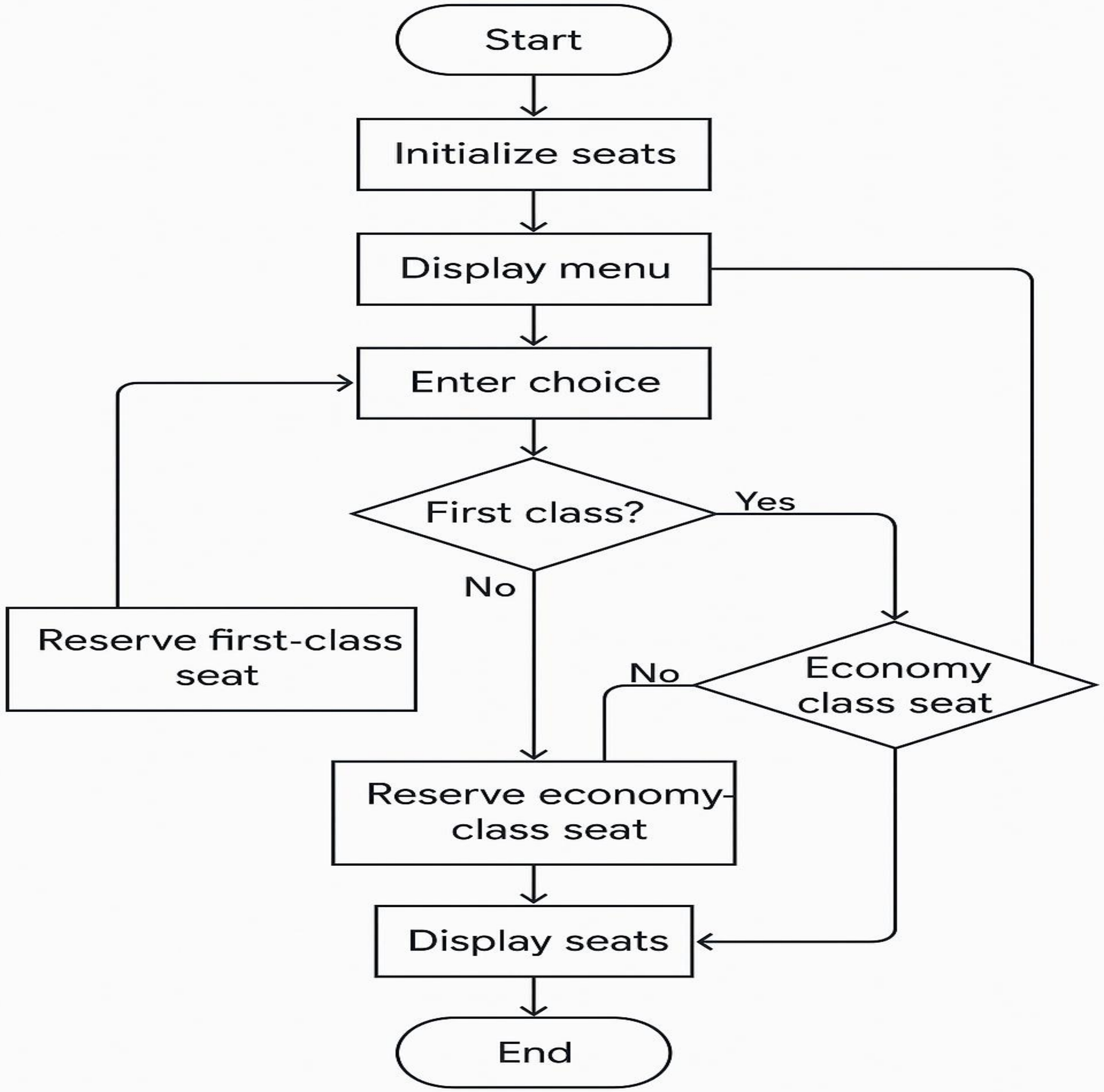


```
--- Airline Reservation System ---
1. View Available Flights
2. Book a Ticket
3. Cancel a Ticket
4. Display Booked Tickets
5. Exit
Enter your choice: 5
Exiting... Thank you!
```

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

FLOWCHART



CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Overview of the Airline Reservation System

This C program is a simple Airline Reservation System that allows users to:

- ✓ View available flights
- ✓ Book a ticket
- ✓ Cancel a ticket
- ✓ Display booked tickets

It uses arrays and structures to store reservations and track seat availability.

Key Components

Structure (Reservation): Stores flight number, passenger name, and seat number.
2D Array (reservations[MAX_FLIGHTS][MAX_SEATS]): Stores reservations.
Array (seatAvailability[MAX_FLIGHTS]): Tracks available seats.

Functions and Their Purpose

Function	Purpose
displayFlights()	Shows available flights
bookTicket()	Books a flight ticket
cancelTicket()	Cancels a reservation
displayBookedTickets()	Displays all booked tickets
main()	Handles user interaction

How It Works

1 ☐ View Flights → Lists available flights.
2 ☐ Book Ticket → User selects flight, enters name, and gets a seat.
3 ☐ Cancel Ticket → User enters flight & seat number to cancel.
4 ☐ Display Booked Tickets → Shows all reservations.
5 ☐ Exit System → Ends the program.

Reference Output

Sample Output

1 ☐ Viewing Available Flights

```
Available Flights:
1. Flight 101
2. Flight 202
3. Flight 303
```

2 ☐ Booking a Ticket

```
EnterFlightNumber(1-3):1
EnterPassengerName:JohnDoe
Ticketbookedsuccessfully!SeatNumber:1
```

3 ☐ Canceling a Ticket

```
EnterFlightNumber:1
EnterSeatNumber:1
Ticketcanceledsuccessfully.
```

4 ☐ Displaying Booked Tickets

```
Booked Tickets:
Flight 1:
Seat 2:Alice
Seat 3:Bob
```

5 ☐ Exiting the System

```
Exiting... Thank you!
```

Advantages of the System

"There are several advantages of this system:

•It reduces manual work and human errors.

•It saves time for both passengers and airline staff.

•It provides a simple interface that’s easy to understand.

•And it can be expanded later to support online payments, notifications, and much more."

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Future Scope

- **Dynamic Data Storage:**
  Replace arrays with linked lists or databases (like MySQL) to manage a larger number of reservations dynamically.
- **User Authentication:**
  Add login and registration systems for passengers and administrators.
- **Multiple Flights and Routes:**
  Support booking for many flights, cities, and different airlines, not just three hardcoded flights.
- **Payment Gateway Integration:**
  Add real-time payment methods for booking confirmations.
- **GUI (Graphical User Interface):**
  Move from console application to a web-based or mobile app with beautiful interfaces.
- **Real-time Seat Updates:**
  Show live seat maps where users can select available seats.

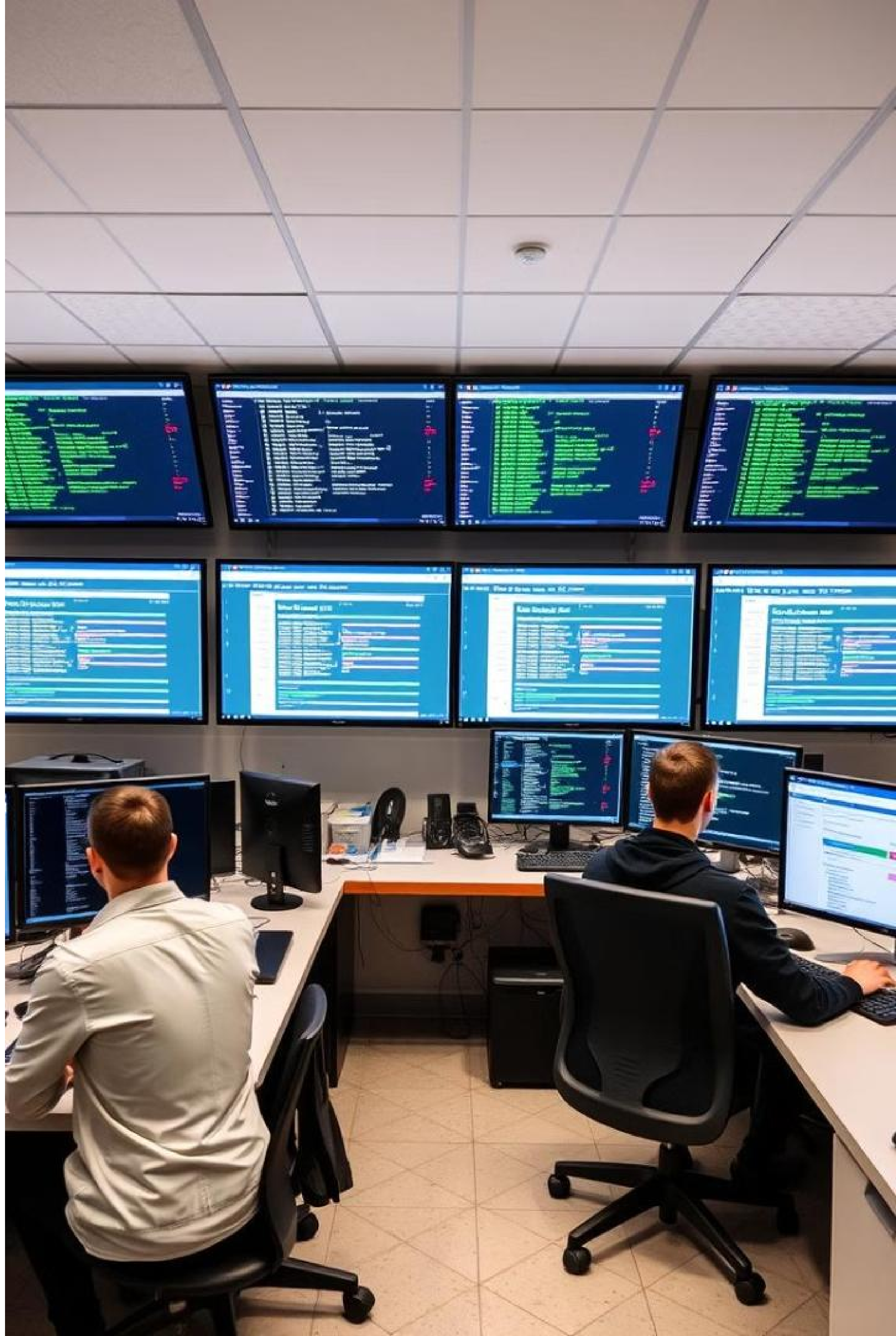
CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Result

✓ The system successfully books, cancels, and displays flight reservations.
✓ Seat availability is updated in real time.
✓ Provides user-friendly interaction via a menu-based approach.

Conclusion

1.This C-based Airline Reservation System efficiently manages flight bookings.
2.Uses arrays, structures, and user input validation for seamless operation.
3.Can be expanded with additional features like payment integration and online booking.



Case Study: C-Based Airline Reservation System

A domestic airline faced challenges in managing flight reservations due to manual booking processes. Customers often encountered overbooked flights, slow cancellations, and inaccurate seat availability. To address these issues, the airline developed a C-based Airline Reservation System.
The system introduced real-time seat availability updates, booking and cancellation options, and a user-friendly menu-based interface. By implementing arrays and structures, the airline ensured efficient data handling and validation. After deployment, booking efficiency improved, reducing overbooking errors and enhancing the customer experience

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Questions

Q1. What problem did the airline face before implementing the system?
Q2.What key features does the system offer?
Q3.How does the system update seat availability in real time?
Q4.What programming concepts were used to develop the system?
Q5.How did the system improve airline operations?

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Answers

A1. The airline struggled with manual booking errors, overbookings, slow cancellations, and lack of real-time seat availability updates.

A2. The system provides flight booking, reservation cancellations, real-time seat availability updates, and a user-friendly menu-based interface.

A3. The system dynamically updates seat counts whenever a passenger books or cancels a reservation, ensuring accurate availability data.

A4. The system was built using arrays, structures, loops, conditions, and user input validation to manage reservations efficiently.

A5. The system reduced booking errors, improved customer experience, and ensured real-time updates, making reservations faster and more reliable.

THANK YOU