FLIGHT RESERVATION SYSTEM

Project report submitted in partial fulfillment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

24KB1A05DW

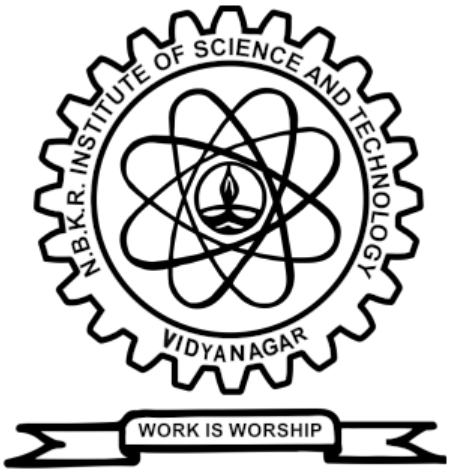
24KB1A05BB

24KB1A05K6

24KB1A05M1

Under the Guidance of

SMT.B.SRUTHI

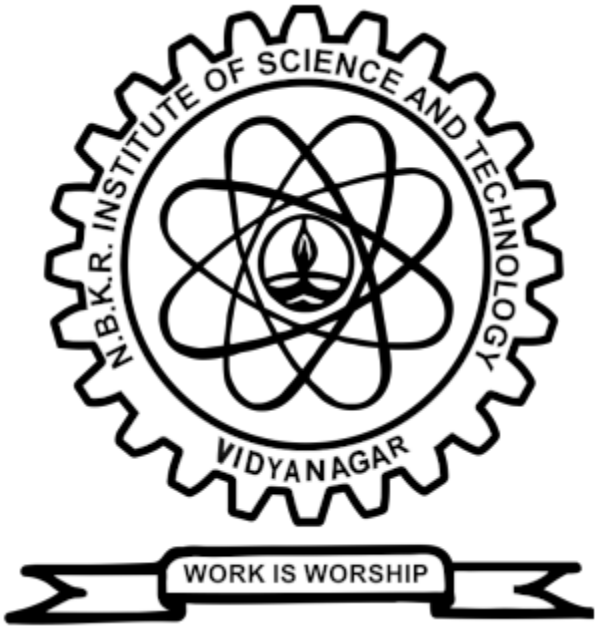


DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NBKRIST

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project report entitled Flight Reservation System being submitted by

24KB1A05DW

24KB1A05BB

24KB1A05K6

24KB1A05M1

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Jawaharlal Nehru Technological University, Kakinada is a

record of bonafied work carried out under my guidance and supervision.

SMT.B.SRUTHI	**Dr. HOD RAJSHEKAR REDDY**
	M.Tech, Ph.D
Designation	**Head of the Department**

DECLARATION

I hereby declare that the dissertation entitled **Flight Reservation System** submitted for the B.Tech Degree is my original work and the dissertation has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

24KB1A05DW

Place:Vidyanagr **24KB1A05BB**

24KB1A05K6

24KB1A05M1

Date: 05-05-25

Acknowledgment

I would like to express my sincere gratitude to all those who supported and guided me throughout the development of this project titled "Flight Reservation System using Arrays and Linked List in C Programming Language".

First and foremost, I would like to thank I my project guide SMT.B.SRUTH, for their valuable guidance, encouragement, and continuous support during the course of this project. Their insights and suggestions greatly enhanced the quality of my work.

I am also thankful to NBKRIST for providing the resources and a conducive environment for carrying out this project effectively.

My appreciation extends to my classmates, friends, and family who offered support, motivation, and helpful feedback during the various stages of the project.

Lastly, I am grateful for the opportunity to apply and deepen my understanding of data structures and programming in C, which has significantly contributed to my learning experience.

Abstract

This project is about creating a simple **Flight Reservation System** using the **C programming language**. The main goal is

to help users book and cancel flight tickets, and also view available seats. The system uses **arrays** to store flight details and **linked lists** to manage passenger information.

The project includes basic features like adding a new passenger, checking available seats, booking a seat, and canceling a reservation. It helps in understanding how data structures like arrays and linked lists can be used to manage information in a program.

This system is a basic model and can be improved in the future by adding more features like saving data to files or adding a user login system. It is a good way to learn how real-life reservation systems work and how to build them using simple C programming.

1.INTRODUCTION

The Flight Reservation System is a simple C program designed to simulate the process of booking flight tickets. It helps users book seats, cancel bookings, and view flight information. The system uses arrays to store fixed flight and seat details, while linked lists are used to manage dynamic passenger records. This project is a good example of how data structures can be used in real-life applications, especially in transport and booking systems.

2.PROBLEM STATEMENT

In today's fast-moving world, people prefer booking tickets online or through digital systems rather than going to the airport or using manual booking services. Without a proper system, managing bookings, cancellations, and passenger data can become difficult and confusing. There is a need for a simple and easy-to-use reservation system that helps in handling such tasks smoothly, even in a basic form.

3.SCOPE OF PROJECT

- **This system allows users to book and cancel flight tickets.**
- **It can display available flights and reserved passengers.**
- **The program is written in C language using basic arrays and linked lists, so it is best suited for small-scale use or educational purposes.**
- **This system does not include real-time payment, online access, or advanced features but lays the foundation for future development.**

4.OBJECTIVES

- **To create a basic flight reservation system using C programming.**

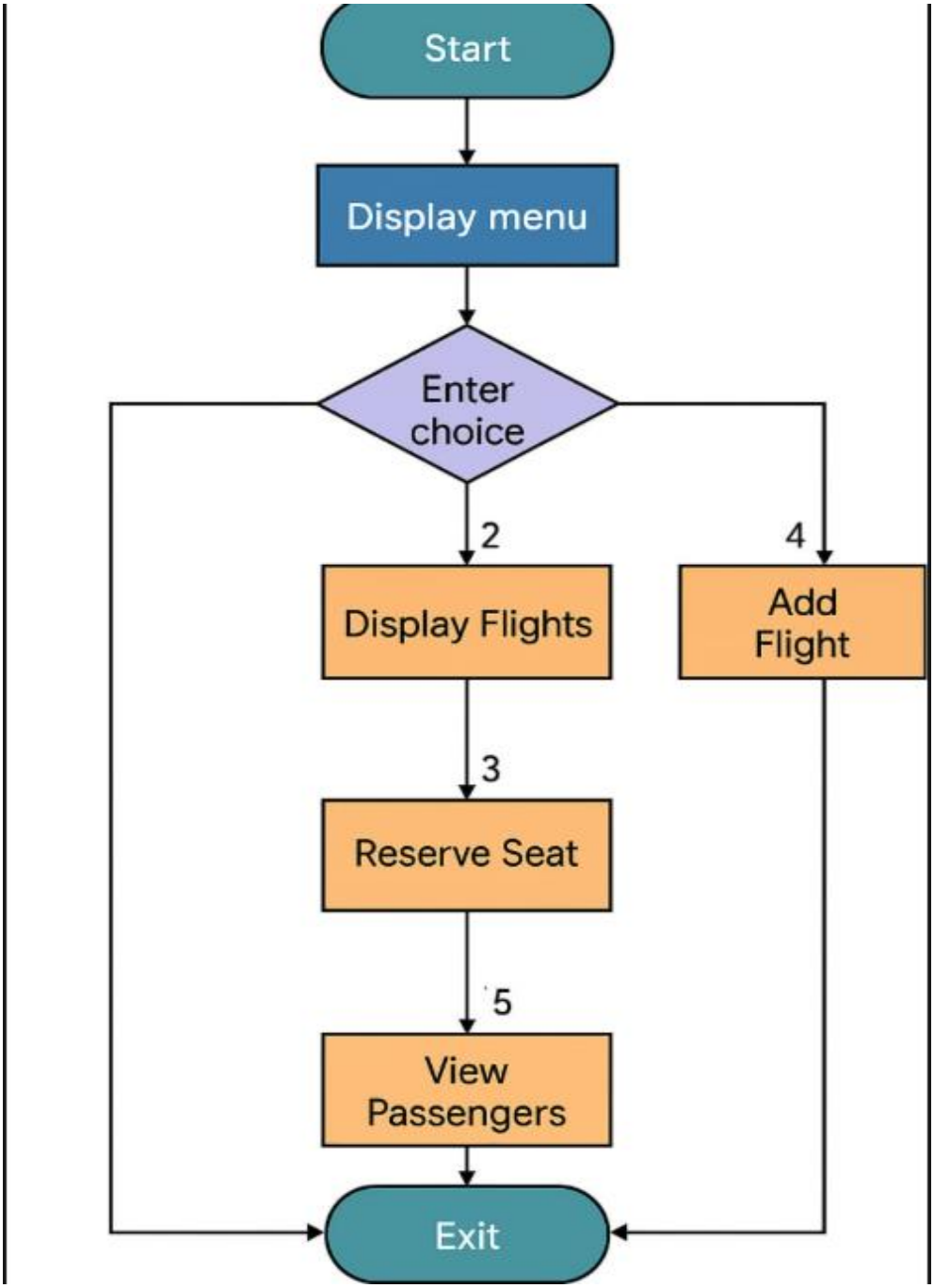
  To learn and implement arrays and linked lists in a practical project.
- **To allow users to perform actions like booking, canceling, and viewing reservations.**

- **To understand memory handling, dynamic data storage, and simple user interface design in C.**

5.Software Requirements

- **Programming Language: C**
- **Compiler: GCC, Turbo C, Code::Blocks, or any C-supported IDE**
- **Operating System: Windows, Linux, or macOS**
-
- **Editor/IDE: Code::Blocks, Dev C++, Turbo C++, or Visual Studio Code**

6.CONTROL FLOW CHART



7.Modules And Their Functionalities

1. Flight Management Module

Functions Involved:

- addFlight(int flightID, char destination[])

- displayFlights()

Functionality:

- Allows the admin/user to add new flights by entering a unique Flight ID and destination.
- Displays a list of all available flights stored in the flights[] array.

2. Passenger Reservation Module

Function Involved:

- reserveSeat(int flightID, char name[], int seatNumber)

Functionality:

- Allows a user to reserve a seat for a passenger by entering flight ID, passenger name, and seat number.
- Dynamically creates a Passenger node and adds it to the flight's linked list of passengers.

3. Passenger Display Module

Function Involved:

- viewPassengers(int flightID)

Functionality:

- Displays all passengers who have reserved seats on a specific flight.

- Traverses the linked list of passengers for the specified flight and prints each passenger's name and seat number.

4. User Interface / Menu Module

Function Involved:

- main()

Functionality:

- Acts as the main control loop that presents a text-based menu to the user.
- Handles user input and calls the appropriate functions based on the selected option:
  - Add Flight
  - Display Flights
  - Reserve Seat
  - View Passengers
  - Exit the program

6.Code

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

```
#define MAX_FLIGHTS 10
#define MAX_NAME 50

// Passenger linked list node
typedef struct Passenger {
    char name[MAX_NAME];
    int seatNumber;
    struct Passenger* next;
} Passenger;

// Flight structure
typedef struct {
    int flightID;
    char destination[50];
    Passenger* passengerList;
} Flight;

// Global flight array and count
Flight flights[MAX_FLIGHTS];
int flightCount = 0;
```

```
// Function to add a new flight
void addFlight(int flightID, char destination[]) {
    if (flightCount >= MAX_FLIGHTS) {
        printf("Flight limit reached.\n");
        return;
    }
    flights[flightCount].flightID = flightID;
    strcpy(flights[flightCount].destination, destination);
    flights[flightCount].passengerList = NULL;
    flightCount++;
    printf("Flight added successfully.\n");
}

// Display all flights
void displayFlights() {
    printf("\nAvailable Flights:\n");
    for (int i = 0; i < flightCount; i++) {
        printf("Flight ID: %d, Destination: %s\n",
flights[i].flightID, flights[i].destination);
    }
}
```

```
// Reserve a seat for a passenger on a flight
void reserveSeat(int flightID, char name[], int seatNumber) {
    for (int i = 0; i < flightCount; i++) {
        if (flights[i].flightID == flightID) {
            // Create new passenger node
            Passenger* newPassenger = (Passenger*)malloc(sizeof(Passenger));
            strcpy(newPassenger->name, name);
            newPassenger->seatNumber = seatNumber;
            newPassenger->next = NULL;

            // Add to passenger list
            if (flights[i].passengerList == NULL) {
                flights[i].passengerList = newPassenger;
            } else {
                Passenger* temp = flights[i].passengerList;
                while (temp->next != NULL)
                    temp = temp->next;
                temp->next = newPassenger;
            }
            printf("Reservation successful.\n");
```

```
            return;
        }
    }
    printf("Flight not found.\n");
}

// View all passengers of a flight
void viewPassengers(int flightID) {
    for (int i = 0; i < flightCount; i++) {
        if (flights[i].flightID == flightID) {
            Passenger* temp = flights[i].passengerList;
            printf("Passengers on Flight %d:\n", flightID);
            while (temp != NULL) {
                printf("Name: %s, Seat No: %d\n", temp->name,
temp->seatNumber);
                temp = temp->next;
            }
            return;
        }
    }
    printf("Flight not found.\n");
}
```

```
// Main function
int main() {
    int choice, flightID, seatNumber;
    char name[MAX_NAME], destination[50];

    while (1) {
        printf("\n--- Flight Reservation System ---\n");
        printf("1. Add Flight\n");
        printf("2. Display Flights\n");
        printf("3. Reserve Seat\n");
        printf("4. View Passengers\n");
        printf("5. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);
        getchar(); // clear newline

        switch (choice) {
            case 1:
                printf("Enter Flight ID: ");
                scanf("%d", &flightID);
                getchar();
```

```
            printf("Enter Destination: ");
            fgets(destination, sizeof(destination), stdin);
            destination[strcspn(destination, "\n")] = '\0';
            addFlight(flightID, destination);
            break;
        case 2:
            displayFlights();
            break;
        case 3:
            printf("Enter Flight ID: ");
            scanf("%d", &flightID);
            getchar();
            printf("Enter Passenger Name: ");
            fgets(name, sizeof(name), stdin);
            name[strcspn(name, "\n")] = '\0';
            printf("Enter Seat Number: ");
            scanf("%d", &seatNumber);
            reserveSeat(flightID, name, seatNumber);
            break;
        case 4:
            printf("Enter Flight ID to view passengers: ");
            scanf("%d", &flightID);
```

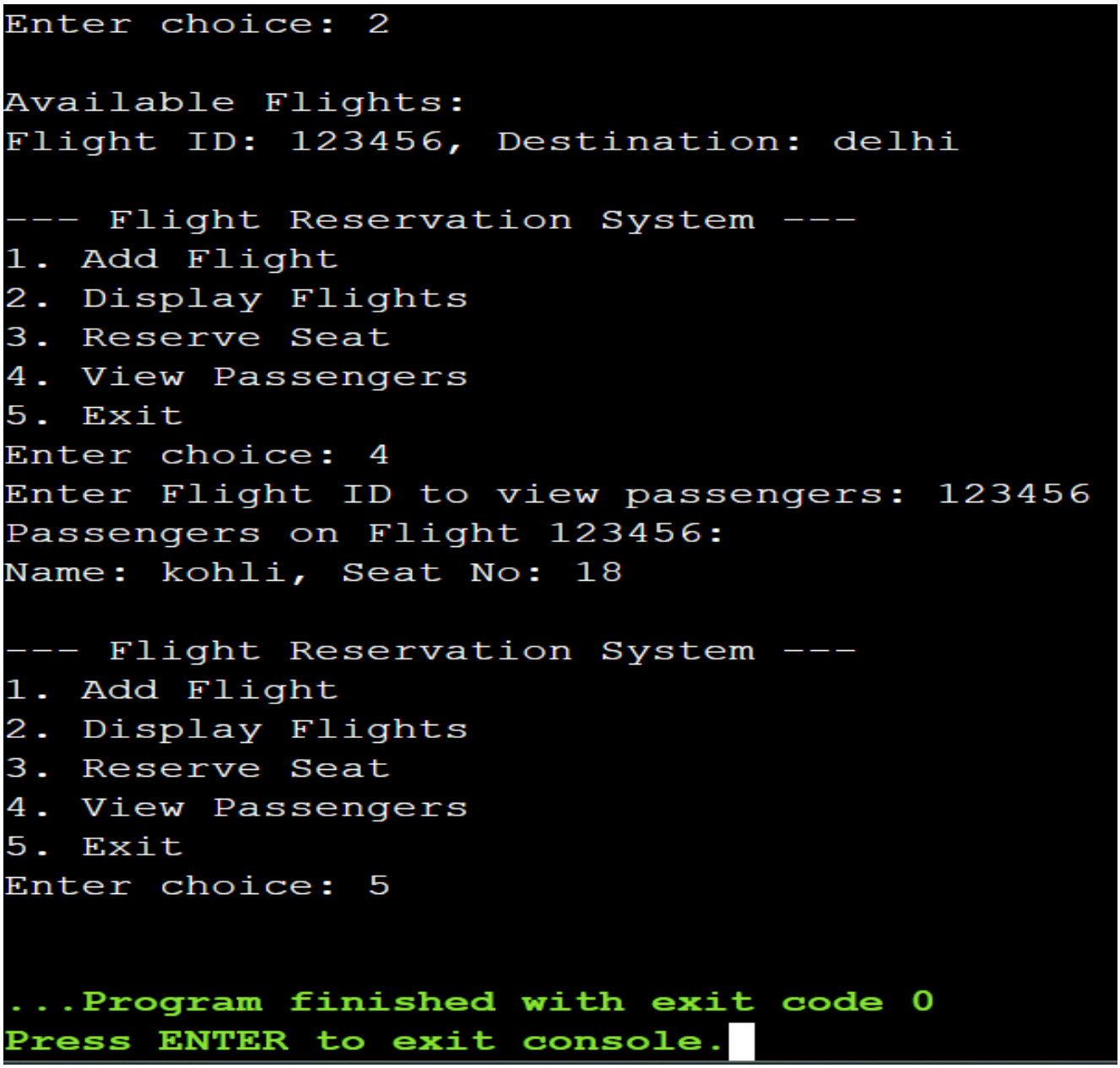
```
            viewPassengers(flightID);
            break;
        case 5:
            exit(0);
        default:
            printf("Invalid choice.\n");
      }
    }
    return 0;
}
```

5. Output Screens

```
--- Flight Reservation System ---
1. Add Flight
2. Display Flights
3. Reserve Seat
4. View Passengers
5. Exit
Enter choice: 1
Enter Flight ID: 123456
Enter Destination: delhi
Flight added successfully.

--- Flight Reservation System ---
1. Add Flight
2. Display Flights
3. Reserve Seat
4. View Passengers
5. Exit
Enter choice: 3
Enter Flight ID: 123456
Enter Passenger Name: kohli
Enter Seat Number: 18
Reservation successful.

--- Flight Reservation System ---
1. Add Flight
2. Display Flights
3. Reserve Seat
4. View Passengers
5. Exit
```



8. Conclusion

The **Flight Reservation System** developed using the **C programming language** successfully demonstrates how core data structures like **arrays** and **linked lists** can be applied to solve real-world problems. This system allows users to add new flights, reserve seats, and view passenger details in an organized manner.

By using arrays for managing fixed flight information and linked lists for dynamic passenger records, the project provides an efficient way to handle memory and data processing. Although this is a basic version of a reservation system, it forms a solid foundation for more complex applications involving databases, graphical interfaces, and network features.

Overall, this project helped in gaining practical knowledge of structured programming, memory management, and user interaction, while simulating the fundamental operations of a real-world airline booking system.