# Bus Ticket Reservation System

## Introduction

The **Bus Ticket Reservation System** is a menu-driven C application designed to automate the process of managing bus routes and ticket bookings.
Instead of maintaining records manually, this system stores bus details and booking information using **structures and file handling**, ensuring data is saved permanently.

## Objective of the Project

The main objectives of this project are:

To manage bus route details efficiently.

To track seat availability and bookings.

To allow ticket booking and cancellation.

To store data permanently using files.

To implement modular programming in C.

## Data Structure Used

A **structure** is used to store bus information.

Structure Members:

`busNo` → Stores the bus number

`route` → Stores the route (e.g., CityA - CityB)

`totalSeats` → Total number of seats in the bus.

`bookedSeats` → Number of seats already booked

# *PROGRAM*

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

struct Bus {
    int busNo;
    char route[100];
    int totalSeats;
    int bookedSeats;
};

/* Function declarations */
void addBus();
void displayBuses();
void searchBus();
void bookTicket();
void cancelBooking();
void viewBookings();

/* Main Function */
int main() {
    int choice;

    do {
        printf("\n----------------------------------------\n");
        printf("   BUS TICKET RESERVATION SYSTEM\n");
        printf("----------------------------------------\n");
        printf("1. Add Bus\n");
        printf("2. Display All Buses\n");
        printf("3. Search Bus\n");
        printf("4. Book Ticket\n");
        printf("5. Cancel Booking\n");
        printf("6. View Bookings\n");
        printf("0. Exit\n");
        printf("----------------------------------------\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch(choice) {
            case 1: addBus(); break;
            case 2: displayBuses(); break;
            case 3: searchBus(); break;
            case 4: bookTicket(); break;
            case 5: cancelBooking(); break;
            case 6: viewBookings(); break;
            case 0: printf("Exiting program...\n"); break;
            default: printf("Invalid choice!\n");
        }
    } while(choice != 0);

    return 0;
}

/* Add Bus */
void addBus() {
    struct Bus b;
    FILE *fp = fopen("buses.txt", "ab");

    if (fp == NULL) {
        printf("File error!\n");
        return;
    }

    printf("Enter Bus Number: ");
    scanf("%d", &b.busNo);
    getchar();  // clear newline

    printf("Enter Route (CityA - CityB): ");
    fgets(b.route, sizeof(b.route), stdin);
    b.route[strcspn(b.route, "\n")] = '\0';

    printf("Enter Total Seats: ");
    scanf("%d", &b.totalSeats);

    b.bookedSeats = 0;

    fwrite(&b, sizeof(b), 1, fp);
    fclose(fp);

    printf("Bus added successfully!\n");
}

/* Display All Buses */
void displayBuses() {
    struct Bus b;
    FILE *fp = fopen("buses.txt", "rb");

    if (fp == NULL) {
        printf("No buses available.\n");
        return;
    }

    printf("\nBusNo  Route                Total  Booked  Available\n");
    printf("----------------------------------------------------\n");

    while (fread(&b, sizeof(b), 1, fp)) {
        printf("%-6d %-20s %-6d %-7d %-6d\n",
                b.busNo, b.route, b.totalSeats,
                b.bookedSeats, b.totalSeats - b.bookedSeats);
    }

    fclose(fp);
}

/* Search Bus */
void searchBus() {
    struct Bus b;
    FILE *fp = fopen("buses.txt", "rb");
    int busNo, found = 0;

    if (fp == NULL) {
        printf("No data found.\n");
        return;
    }

    printf("Enter Bus Number to search: ");
    scanf("%d", &busNo);

    while (fread(&b, sizeof(b), 1, fp)) {
        if (b.busNo == busNo) {
            printf("\nBus Found!\n");
            printf("Bus No: %d\n", b.busNo);
            printf("Route: %s\n", b.route);
            printf("Total Seats: %d\n", b.totalSeats);
            printf("Booked Seats: %d\n", b.bookedSeats);
            printf("Available Seats: %d\n",
                   b.totalSeats - b.bookedSeats);
            found = 1;
            break;
        }
    }

    if (!found)
        printf("Bus not found!\n");

    fclose(fp);
}

/* Book Ticket */
void bookTicket() {
    struct Bus b;
    FILE *fp = fopen("buses.txt", "rb");
    FILE *temp = fopen("temp.txt", "wb");
    int busNo, seats, found = 0;

    if (fp == NULL) {
        printf("No buses available.\n");
        return;
    }

    printf("Enter Bus Number: ");
    scanf("%d", &busNo);
    printf("Enter seats to book: ");
    scanf("%d", &seats);

    while (fread(&b, sizeof(b), 1, fp)) {
        if (b.busNo == busNo && (b.totalSeats - b.bookedSeats) >= seats) {
            b.bookedSeats += seats;
            found = 1;
            printf("Booking successful! Seats left: %d\n",
                   b.totalSeats - b.bookedSeats);
        }
        fwrite(&b, sizeof(b), 1, temp);
    }

    if (!found)
        printf("Booking failed or bus not found.\n");

    fclose(fp);
    fclose(temp);

    remove("buses.txt");
    rename("temp.txt", "buses.txt");
}

/* Cancel Booking */
void cancelBooking() {
    struct Bus b;
    FILE *fp = fopen("buses.txt", "rb");
    FILE *temp = fopen("temp.txt", "wb");
    int busNo, seats, found = 0;

    if (fp == NULL) {
        printf("No buses available.\n");
        return;
    }

    printf("Enter Bus Number: ");
    scanf("%d", &busNo);
    printf("Enter seats to cancel: ");
    scanf("%d", &seats);

    while (fread(&b, sizeof(b), 1, fp)) {
        if (b.busNo == busNo && b.bookedSeats >= seats) {
            b.bookedSeats -= seats;
            found = 1;
            printf("Cancellation successful!\n");
        }
        fwrite(&b, sizeof(b), 1, temp);
    }

    if (!found)
        printf("Cancellation failed!\n");

    fclose(fp);
    fclose(temp);

    remove("buses.txt");
    rename("temp.txt", "buses.txt");
}

/* View Bookings */
void viewBookings() {
    struct Bus b;
    FILE *fp = fopen("buses.txt", "rb");

    if (fp == NULL) {
        printf("No bookings found.\n");
        return;
    }

    printf("\nBusNo  Route                Total  Booked  Available\n");
    printf("-----------------------------------------------------\n");

    while (fread(&b, sizeof(b), 1, fp)) {
        printf("%-6d %-20s %-6d %-7d %-6d\n",
               b.busNo, b.route, b.totalSeats,
               b.bookedSeats, b.totalSeats - b.bookedSeats);
    }

    fclose(fp);
}
```

## *OUTPUT*

```
----------------------------------------
  BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
```

```
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 1
File error!

----------------------------------------
   BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 2
No buses available.

----------------------------------------
   BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 3
No data found.

----------------------------------------
   BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 4
No buses available.

----------------------------------------
   BUS TICKET RESERVATION SYSTEM
```

```
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 5
No buses available.

----------------------------------------
   BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 6
No bookings found.

----------------------------------------
   BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 7
Invalid choice!

----------------------------------------
   BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
```

```
----------------------------------------
Enter your choice: 8
Invalid choice!

----------------------------------------
  BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 9
Invalid choice!

----------------------------------------
  BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 10
Invalid choice!

----------------------------------------
  BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice: 11
Invalid choice!

----------------------------------------
  BUS TICKET RESERVATION SYSTEM
----------------------------------------
1. Add Bus
```

```
2. Display All Buses
3. Search Bus
4. Book Ticket
5. Cancel Booking
6. View Bookings
0. Exit
----------------------------------------
Enter your choice:
```

## CONCLUSION

The **Bus Ticket Reservation System** successfully demonstrates how **C programming** can be used to build a real-world application using **structures, file handling, and modular functions**.
This project provides a strong foundation for understanding **data storage, record management, and menu-driven programs**.