BUS RESERVATION SYSYTEM

```
// C Program to implement Bus Reservation System
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Define a structure to store bus information
struct Bus{
   int busNumber;
   char source[50];
   char destination[50];
   int totalSeats;
   int availableSeats;
   float fare;
};

// Define a structure to store user login information
struct User{
   char username[50];
   char password[50];
};

// Function to display the main menu
void displayMainMenu(){
   printf("\n=== Main Menu ===\n");
   printf("1. Login\n");
   printf("2. Exit\n");
   printf("Enter your choice: ");
}
```

```
// Function to display the user menu
void displayUserMenu(){
    printf("\n=== User Menu ===\n");
    printf("1. Book a Ticket\n");
    printf("2. Cancel a Ticket\n");
    printf("3. Check Bus Status\n");
    printf("4. Logout\n");
    printf("Enter your choice: ");
}

// Function to perform user login
int loginUser(struct User users[], int numUsers, char username[], char password[]){
    for (int i = 0; i < numUsers; i++){
        if (strcmp(users[i].username, username) == 0 && strcmp(users[i].password, password) == 0){
            return i; // Return the index of the logged-in user
        }
    }
    return -1; // Return -1 if login fails
}

// Function to book tickets
void bookTicket(struct Bus buses[], int numBuses){
    printf("\nEnter Bus Number: ");
    int busNumber;
    scanf("%d", &busNumber);

    // Find the bus with the given busNumber
    int busIndex = -1;
    for (int i = 0; i < numBuses; i++){
        if (buses[i].busNumber == busNumber){
            busIndex = i;
            break;
        }
    }

    if (busIndex == -1){
        printf("Bus with Bus Number %d not found.\n", busNumber);
    }
    else{
        printf("Enter Number of Seats: ");
        int seatsToBook;
        scanf("%d", &seatsToBook);

        if (buses[busIndex].availableSeats < seatsToBook){
            printf("Sorry, only %d seats are available.\n", buses[busIndex].availableSeats);
        }
        else{
            buses[busIndex].availableSeats -= seatsToBook;
            printf("Booking successful! %d seats booked on Bus Number %d.\n", seatsToBook,
busNumber);
        }
    }
}

// Function to cancel tickets
void cancelTicket(struct Bus buses[], int numBuses){
    printf("\nEnter Bus Number: ");
    int busNumber;
    scanf("%d", &busNumber);

    // Find the bus with the given busNumber
    int busIndex = -1;
    for (int i = 0; i < numBuses; i++){
        if (buses[i].busNumber == busNumber){
            busIndex = i;
            break;
        }
    }

    if (busIndex == -1){
        printf("Bus with Bus Number %d not found.\n", busNumber);
    }
    else{
        printf("Enter Number of Seats to Cancel: ");
        int seatsToCancel;
        scanf("%d", &seatsToCancel);

        if (seatsToCancel > (buses[busIndex].totalSeats - buses[busIndex].availableSeats)){
            printf("Error: You can't cancel more seats than were booked.\n");
        }
        else{
            buses[busIndex].availableSeats += seatsToCancel;
            printf("Cancellation successful! %d seats canceled on Bus Number %d.\n", seatsToCancel,
                   busNumber);
        }
    }
}

// Function to check bus status
void checkBusStatus(struct Bus buses[], int numBuses){
    printf("\nEnter Bus Number: ");
    int busNumber;
    scanf("%d", &busNumber);
    // Find the bus with the given busNumber
    int busIndex = -1;
    for (int i = 0; i < numBuses; i++){
        if (buses[i].busNumber == busNumber){
            busIndex = i;
            break;
        }
    }

    if (busIndex != -1){
        printf("\nBus Number: %d\n", buses[busIndex].busNumber);
        printf("Source: %s\n", buses[busIndex].source);
        printf("Destination: %s\n", buses[busIndex].destination);
        printf("Total Seats: %d\n", buses[busIndex].totalSeats);
        printf("Available Seats: %d\n", buses[busIndex].availableSeats);
        printf("Fare: %.2f\n", buses[busIndex].fare);
    }
    else{
        printf("Bus with Bus Number %d not found.\n", busNumber);
    }
}

int main(){
    // Initialize user data
    struct User users[5] = {
        {"user1", "pass1"}, {"user2", "pass2"}, {"user3", "pass3"}, {"user4", "pass4"}, {"user5", "pass5"},
    };
    int numUsers = 5;

    // Initialize bus data
```

```
struct Bus buses[3] = {
    {101, "City A", "City B", 50, 50, 500.0},
    {102, "City C", "City D", 40, 40, 400.0},
    {103, "City E", "City F", 30, 30, 300.0},
};
int numBuses = 3;

int loggedInUserId = -1; // Index of the logged-in user

while (1){
    if (loggedInUserId == -1){
        displayMainMenu();
        int choice;
        scanf("%d", &choice);

        if (choice == 1){
            char username[50];
            char password[50];

            printf("Enter Username: ");
            scanf("%s", username);
            printf("Enter Password: ");
            scanf("%s", password);

            loggedInUserId = loginUser(users, numUsers, username, password);
            if (loggedInUserId == -1){
                printf("Login failed. Please check your username and password.\n");
            }
            else{
                printf("Login successful. Welcome, %s!\n", username);
            }
        }
        else if (choice == 2){
            printf("Exiting the program.\n");
            break;
        }
        else{
            printf("Invalid choice. Please try again.\n");
        }
    }
    else{
        displayUserMenu();
        int userChoice;
        scanf("%d", &userChoice);

        switch (userChoice){
        case 1:
            bookTicket(buses, numBuses);
            break;
        case 2:
            cancelTicket(buses, numBuses);
            break;
        case 3:
            checkBusStatus(buses, numBuses);
            break;
        case 4:
            printf("Logging out.\n");
            loggedInUserId = -1;
            break;
        default:
            printf("Invalid choice. Please try again.\n");
        }
        }
    }

    return 0;
}
```