Bus Reservation

Abstract:

Are you looking for an online bus booking system project? We are here to help you. You can contact us. Online Bus Booking System cloud-based online software. This system would help customers to book a seat for their journey, book a bus. This system would also help the owner to manage the coaches, employees, clients, services, etc.

Bus Reservation System will increase the booking process faster, convenient, and comfortable. Customers can book their desired seats. They can check the availability of posts on a specific date. The customer can check availability, book a ticket, or cancel a ticket 24X7. The online system is available to use anytime. The user doesn't require to visit any office.

Problem statement:

when they buy the bus ticket and sometimes needs to queue up long time to get the

bus ticket. Besides that, passengers also not allowed to buy bus ticket through online system. Besides there is no any online system for ticket booking.

Solution:

The method to solve this problem is to create an online booking bus ticket system. Customer can buy the book ticket over the Internet, 16 hours a day, 5 days a week and the bus ticket can't be lost, stolen or left behind. In addition, the online system lets the passengers check the availability of the bus ticket before they buy bus ticket. Furthermore, passengers no need to pay cash to buy bus ticket on counter because they can pay the bus ticket when ticket purchasing on online.

Project overview:

```
// 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 passenger information
struct Passenger {
    char name[50];
    int age;
```

```
    int seatNumber;

    int busNumber;
};



// 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=== Bus Reservation System ===\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 a ticket

void bookTicket(struct Bus buses[], int numBuses,

                struct Passenger passengers[],

                int* numPassengers, int userId)
{

    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 if (buses[busIndex].availableSeats == 0) {

        printf("Sorry, the bus is fully booked.\n");

    }

    else {
```

```
printf("Enter Passenger Name: ");

scanf("%s", passengers[*numPassengers].name);



printf("Enter Passenger Age: ");

scanf("%d", &passengers[*numPassengers].age);



// Assign a seat number to the passenger

passengers[*numPassengers].seatNumber

   = buses[busIndex].totalSeats

    - buses[busIndex].availableSeats + 1;



// Update the passenger's bus number
```

```
        passengers[*numPassengers].busNumber = busNumber;




        // Update available seats

        buses[busIndex].availableSeats--;




        printf("Ticket booked successfully!\n");

        (*numPassengers)++;

    }
}


// Function to cancel a ticket

void cancelTicket(struct Bus buses[], int numBuses,

                struct Passenger passengers[],

                int* numPassengers, int userId)
```

```
{
    printf("\nEnter Passenger Name: ");
    char name[50];
    scanf("%s", name);

    int found = 0;
    for (int i = 0; i < *numPassengers; i++) {
        if (strcmp(passengers[i].name, name) == 0
            && passengers[i].busNumber
                == buses[userId].busNumber) {
            // Increase available seats
            int busIndex = -1;
            for (int j = 0; j < numBuses; j++) {
```

```
    if (buses[j].busNumber

        == passengers[i].busNumber) {

        busIndex = j;

        break;

    }

}

buses[busIndex].availableSeats++;



// Remove the passenger entry

for (int j = i; j < (*numPassengers) - 1; j++) {

    passengers[j] = passengers[j + 1];

}
```

```
            (*numPassengers)--;

            found = 1;

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

            break;

        }

    }

    if (!found) {

        printf("Passenger with name %s not found on this "

               "bus.\n",

               name);

    }
}


// Function to check bus status
```

```
void checkBusStatus(struct Bus buses[], int numBuses,

                    int userId)
{

    printf("\nBus Number: %d\n", buses[userId].busNumber);

    printf("Source: %s\n", buses[userId].source);

    printf("Destination: %s\n", buses[userId].destination);

    printf("Total Seats: %d\n", buses[userId].totalSeats);

    printf("Available Seats: %d\n",

          buses[userId].availableSeats);

    printf("Fare: %.2f\n", buses[userId].fare);
}




int main()
{
```

```
// Initialize user data

struct User users[5] = {

    { "user1", "password1" }, { "user2", "password2" },

    { "user3", "password3" }, { "user4", "password4" },

    { "user5", "password5" },

};

int numUsers = 5;




// Initialize bus data

struct Bus buses[3] = {

    { 101, "City A", "City B", 50, 50, 25.0 },

    { 102, "City C", "City D", 40, 40, 20.0 },
```

```
    { 103, "City E", "City F", 30, 30, 15.0 },

};

int numBuses = 3;




struct Passenger

    passengers[500]; // Array to store passenger

                    // information

int numPassengers = 0; // Number of passengers




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, passengers,

            &numPassengers, loggedInUserId);

   break;

case 2:

   cancelTicket(buses, numBuses, passengers,

             &numPassengers,

             loggedInUserId);

   break;

case 3:
```

```
            checkBusStatus(buses, numBuses,

                    loggedInUserId);

            break;

        case 4:

            printf("Logging out.\n");

            loggedInUserId = -1;

            break;

        default:

            printf(

                "Invalid choice. Please try again.\n");

    }

}
```

```
    }


    return 0;
}
```

Modeling & Results:

```
 === Bus Reservation System ===
1. Login
2. Exit
Enter your choice: 1
Enter Username: kesava
Enter Password: 123
Login failed. Please check your username and password.
=== Bus Reservation System ===
1. Login
2. Exit
Enter your choice: 1
Enter Username: user1
Enter Password: password1
Login successful. Welcome, user1!
=== User Menu ===
1. Book a Ticket
2. Cancel a Ticket
3. Check Bus Status
```

4. Logout

Enter your choice: 1

Enter Bus Number: 101

Enter Passenger Name: kesav

Enter Passenger Age: 20

Ticket booked successfully!

=== User Menu ===

1. Book a Ticket

2. Cancel a Ticket

3. Check Bus Status

4. Logout

2. Cancel a Ticket

3. Check Bus Status

4. Logout

Enter your choice: 3

Bus Number: 101

Source: City A

Destination: City B

Total Seats: 50

Available Seats: 49

Fare: 25.00

=== User Menu ===

1. Book a Ticket

2. Cancel a Ticket

3. Check Bus Status

4. Logout

Enter your choice: 2

Enter Passenger Name: kesav

Ticket canceled successfully!

=== User Menu ===

1. Book a Ticket

2. Cancel a Ticket

3. Check Bus Status

4. Logout

Enter your choice: 4

Logging out.

=== Bus Reservation System ===

1. Login

2. Exit

• Enter your choice:

Screen shot:

Future enhancement:

The expense of long-distance travel varies significantly depending on the type of business

you operate.

Our research has shown the diversification of operations by international, lengthy, and shuttle services, particularly by those market leaders in their respective regions.

Such operators' reservation systems need to provide single-price models which comply with their operational requirements.

To provide the operator with a full passage experience throughout his operations, a modern bus reservation system must be able to handle scenarios such as dynamic prices in competitive intercity operations or serial travel passes for commuter/shuttle services.

Conclusion:

It can be seen that computer applications are very important in every field of human endeavor. Here with this

new system all the information about the customer making a reservation can be obtained by clicking a button,

removing some of the difficulties that come with the manual system. This will reduce the workload of the

employees, reduce the time taken to make reservations at the bus terminal and will also increase the efficiency.

The application also has the ability to automatically update records in various files thereby relieving the

employees of the stress of working with the file security of the data. This project will, overall, give a new path to

the bus reservation and ticketing processes. Automation and management of seats and reservations will be

done online. However, the project does not limit walk-in passengers who are passengers at the company's

counter as it caters for them as well. It also reduces the use of paper like the traditional way of ticketing.

Thanking you