**Jaypee institute of information technology**

(Deemed to be university under section 3 of

UGC Act 1956) Sector 62, Noida



**Software Development Fundamentals Lab-1**

**(24B15CS111)**

**Airplane Ticket Booking System - Project Synopsis**

**TEAM MEMBERS : BATCH ENROLLMENT NO**

1. ADITYA VERMA F7 992501030409

2. SHIV SHARMA F8 992501030450

3. SONU RAI F7 992501030408

4. SUMIT SHARMA F7 992501030402

**Table of Contents**

|  |  |
| --- | --- |
| S NO. | NAME |
| 1 | Title page |
| 2 | Brief overview of project |
| 3 | Code |
| 4 | Output |
| 5 | Design and implementation |
| 6 | Conclusion |
| 7 | References |

7

**Brief overview of project**

This project simulates an Airplane Ticket Booking System, similar in concept to online platforms like IRCTC or

MakeMyTrip. It allows users to search for flights between two destinations and displays details such as flight

number, airline, time, and price. The system checks availability of flights and provides external links to the

corresponding airline website based on user input.

This project demonstrates the use of C language for logic and HTML/CSS for front-end visualization**.** Future

enhancements may include API integration or database support to fetch real-time flight data dynamically

Code

#include <stdio.h>

#include <string.h>

#define TOTAL\_FLIGHTS 10

#define NAME\_LENGTH 50

struct Flight {

int id;

char source[30];

char destination[30];

int seats;

float price;

};

struct Booking {

int flightId;

char passenger[NAME\_LENGTH];

};

struct Flight flights[TOTAL\_FLIGHTS] = {

{1, "New Delhi", "Kashmir ", 80, 750},

{2, "Punjab", " Haryana ", 70, 1200},

{3, "Chennai", " Maharashtra", 90, 980},

{4, "Kolkata", " Bangalore", 80, 680},

{5, "Mumbai", " Pune ", 70, 450},

{6, "Gujarat", " Rajasthan", 85, 820},

{7, "Assam", " West Bengal", 75, 900},

{8, "Kerala", " Tamil Nadu", 88, 760},

{9, "ranchi ","Telangana", 95, 650},

{10, "Odisha", " Jharkhand", 78, 880}

};

struct Booking bookings[100];

int bookingCount = 0;

void displayFlights() {

printf("\nAvailable Flights:\n");

printf("ID\tFrom\t\tTo\t\tSeats\tPrice\n");

printf("-------------------------------------------------------\n");

for (int i = 0; i < TOTAL\_FLIGHTS; i++) {

printf("%d\t%s\t%s\t%d\t%.2f\n",

flights[i].id, flights[i].source,

flights[i].destination, flights[i].seats, flights[i].price);

}

}

void bookTicket() {

int id;

char name[NAME\_LENGTH];

displayFlights();

printf("\nEnter Flight ID to book: ");

scanf("%d", &id);

int found = 0;

for (int i = 0; i < TOTAL\_FLIGHTS; i++) {

if (flights[i].id == id) {

found = 1;

if (flights[i].seats > 0) {

printf("Enter passenger name: ");

scanf("%s", name);

strcpy(bookings[bookingCount].passenger, name);

bookings[bookingCount].flightId = id;

flights[i].seats--;

bookingCount++;

printf("\nTicket booked successfully for %s on Flight %d!\n", name, id);

} else {

printf("\nSorry! No seats available on this flight.\n");

}

break;

}

}

if (found==0)

printf("\nInvalid Flight ID!\n");

}

void viewBookings() {

if (bookingCount == 0) {

printf("\nNo bookings yet!\n");

return;

}

printf("\nCurrent Bookings:\n");

printf("Passenger\tFlight ID\n");

printf("-----------------------------\n");

for (int i = 0; i < bookingCount; i++) {

printf("%s\t\t%d\n", bookings[i].passenger, bookings[i].flightId);

}

}

void cancelTicket() {

char name[NAME\_LENGTH];

printf("\nEnter passenger name to cancel booking: ");

scanf("%s", name);

int found = 0;

for (int i = 0; i < bookingCount; i++) {

if (strcmp(bookings[i].passenger, name) == 0) {

found = 1;

for (int j = 0; j < TOTAL\_FLIGHTS; j++) {

if (flights[j].id == bookings[i].flightId) {

flights[j].seats++;

}

}

for (int k = i; k < bookingCount - 1; k++) {

bookings[k] = bookings[k + 1];

}

bookingCount--;

printf("\nBooking for %s has been canceled!\n", name);

break;

}

}

if (found==0)

printf("\nBooking not found!\n");

}

int main() {

int choice;

while (1) {

printf("\n===== AIRPLANE TICKET BOOKING SYSTEM =====\n");

printf("1. View Flights\n");

printf("2. Book Ticket\n");

printf("3. View Bookings\n");

printf("4. Cancel Ticket\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1: displayFlights(); break;

case 2: bookTicket(); break;

case 3: viewBookings(); break;

case 4: cancelTicket(); break;

case 5:

printf("\nThank you! Goodbye.\n");

return 0;

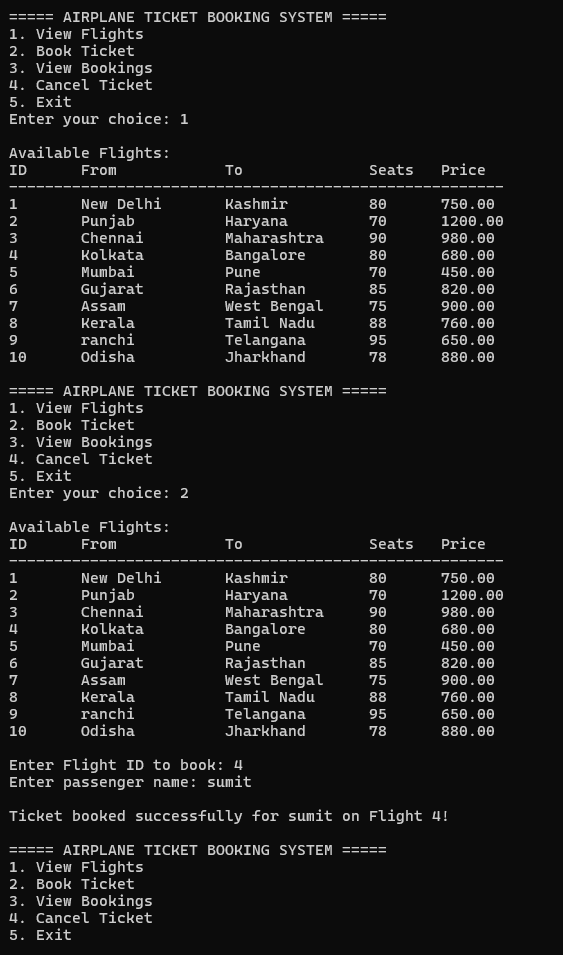
default:

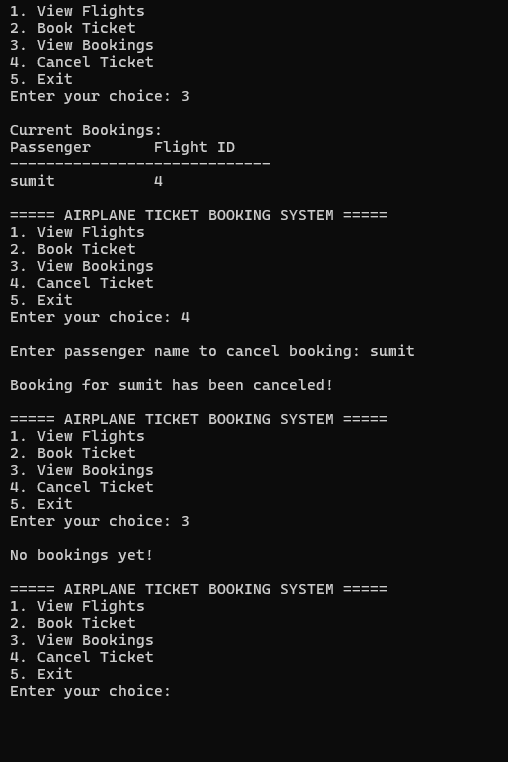
printf("\nInvalid choice. Try again.\n");

}

}

}

**Output**  
  




**DESIGN AND IMPLIMENTATION**

SRS Document

What the Flight Information System is meant to do and how it functions.

• Purpose: To show flight details such as airline name, timing, and seat availability.

• Inputs: Source, destination, and travel date given by the user.

• Outputs: List of flights with details and links for booking externally.

• Features: Easy search system, clear result display, and simple user interface.

Flow Chart

The working of the system can be understood with this simple flow:

Start → Enter Source, Destination, and Date → Search Flights → Show Flight Details → Provide Booking Link → End

**Conclusion**

The Flight Information System makes it easy for users to find flight details by entering their source, destination, and travel date. It helps people check flight availability, airline names, and timings quickly, without needing to book tickets. The project shows how a simple program can help users get useful travel information easily.

**Summary & Achievement of Objectives:**

This project successfully meets its main goals:

• It allows users to search flights by entering simple details.

• It shows flight name, time, and availability clearly.

• It gives links for further booking if needed.

• It uses basic programming logic and simple design to display information correctly.

**Future Work and Recommendations:**

• A ticket buying system can also be integrated.

• A login system can be added for personalized flight searches.

• A mobile version or app can make it more accessible.

• Future updates can include features like price comparison, seat selection, and notifications for

flight delays or offers.

**References**

List of all sources cited in the report:

• Official Airline Websites - ( for flight route and timing information.)

• https://www.irctc.co.in - (reference for interface and travel information design.)

• https://www.makemytrip.com - ( for understanding flight search layouts.)

• https://www.cleartrip.com

• GeeksforGeeks and TutorialsPoint (C language basics)