**FLIGHT RESERVATION SYSTEM**

MINI PROJECT

**(Software Development Fundamentals-2)**

**Submitted by:**

Devang Sharma 21102172

Amritanshu 21802002

Radhika Jangra(21102171)

Srishti Sinha(21102174)



Department of ECE

Jaypee Institute of Information Technology, Noida

May 2022

ACKNOWLEDGEMENT

I am thankful to all the faculty members, providing their valuable time and guidance in elaborating view of studying the project details and getting the right vision for its implementation.

I am also thankful to Ms. Nishtha Ahuja who is having a vast knowledge of C++ Programming language which are the building block of project.

DATE:

27-5-2022 Devang Sharma 21102172

Amritanshu 21802002

Radhika Jangra(21102171)

Srishti Sinha(21102174)

**Table of Contents**

1. **Introduction of the Project**
   1. Project Requirement and Overview
   2. Project Objective
   3. Project Benefit
   4. Project Scope
2. **Synopsis and Flow Chart**
3. **Implementation**
4. **Output**
5. **Conclusion**
6. **Introduction of the Project**

**1.1 Project Requirement and Overview**

Requirement’s definition and management is recognized as a necessary step in the delivery of successful systems and software projects, discipline is also required by standards, regulations, and quality improvement initiatives. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization needs to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirement’s definition and management is an activity that can deliver a high, fast return on investment.

The FLIGHT RESERVATION SYSTEM undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for flight booking. This project is to develop software for booking flights for international and domestic passengers. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using C++ language and. Hence it provides the complete solution for the current booking system.

**1.2. Project Objective**

* To search for domestic and international flights available.
* To book multiple tickets for domestic flight.
* To book multiple tickets for international flight.
* To provide boarding pass for the flight.
* To store ticket details permanently.
* To access ticket details anytime.
* To cancel ticket and delete data from database.
* To make payment through UPI or Bank Transfer.
* Provide greater speed & reduced time consumption.

**1.3. Project Benefits**

Some benefits are:

* Access to privilege booking system.
* Advanced system to regularly check the validity of entered details.
* Data gets stored and can be accessed easily.
* The flight booking system is easy to modify to add the additional features you wished for.
  1. **Project Scope**
  + In the hilly or flooded or other inaccessible areas where water transport or land transportation cannot be developed, passengers and goods can be sent by small planes or helicopters. Air transport is the most useful for relief work in natural disasters such as flood, earthquake, and war etc.
  + Efficient administration and management of the airways system helps in the economic Growth of the nation. Airways is useful to trade and commerce.

1. **Synopsis**

**Classes used-**

* “Search” – Class to show Available Boarding Locations, Destination and Routes.
* Member-Functions: -

1. L2I(void): -

* Return type void.
* Prints Local to International Flight routes.

1. internsearch(void): -

* Return type void.
* Prints International to International Flight routes.

1. localsearch(void): -

* Return type void.
* Prints Local to Local Flight routes.
* “payment” - Class to calculate Flight Bill.
* Member-Functions:-

1. Calc\_Bill(char [],char[],char[]): -

* Return type void.
* calculates amount of ticket with the help of Boarding location, Destination and Class(Business/Economy).

1. waystopay(void): -

* Return type void.
* Provides interface to ask user to pay via UPI or Bank Transfer.

1. Pay\_UPI(void): -

* Asks user for Mobile no.
* Verify Payment through OTP and CAPTCHA Verification.

1. Pay\_Bank(void): -

* Asks user for Bank acc. no., acc. name, IFSC Code and IPIN.
* “mytime” –
* Member-Functions: -

1. getdata(void): -

* Asks User for flight time.
* Return type void.

1. show(void): -

* Displays Flight time on screen.
* Return type void.
* “passenger” – It stores Users' Personal Information and Ticket Details
* Member-Functions: -

1. show\_logs(void): -

* Prints single ticket.
* Return type void.

1. getdata(void): -Virtual Function to be defined in sub-class.
2. show(void): -Virtual Function to be defined in sub-class.

* “booking”- Class booking inherits passenger publicly and defines getdata() and show() to take input() for personal info and print Ticket respectively.
* Member-Functions: -

1. getdata(int): -

* Asks user for:
* First name
* Last name
* Address
* Passport No.
* Phone No.
* Gmail
* Boarding Location
* Destination
* Flight Date
* Return type void.

1. show(void): -

* Return type void.
* Write Passenger class instance in Data File.
* Shows the ticket on screen.
* Generates and print Ticket no. and Reference no.
* “local”- Class local inherits booking Class and redefines getdata() to book tickets for Local To Local Routes
* Member-Functions: -

1. getdata(void): -

* Return type void.
* Asks user for Airline, Class(Business, Economy).
* Calls mytime::getdata()
* Calls booking::getdata(int)
* Calls booking::show(int)
* “international”- Class international inherits booking Class and redefines getdata() to book tickets for Local To International & International To International Routes
* Member-Functions: -

1. getdata(void): -

* Return type void.
* Asks user for Airline, Class(Business, Economy).
* Calls mytime::getdata()
* Calls booking::getdata(int)
* Calls booking::show(int)
* “main1” - Class main1 is Mother of all Classes and Functions.
* Member-Functions: -

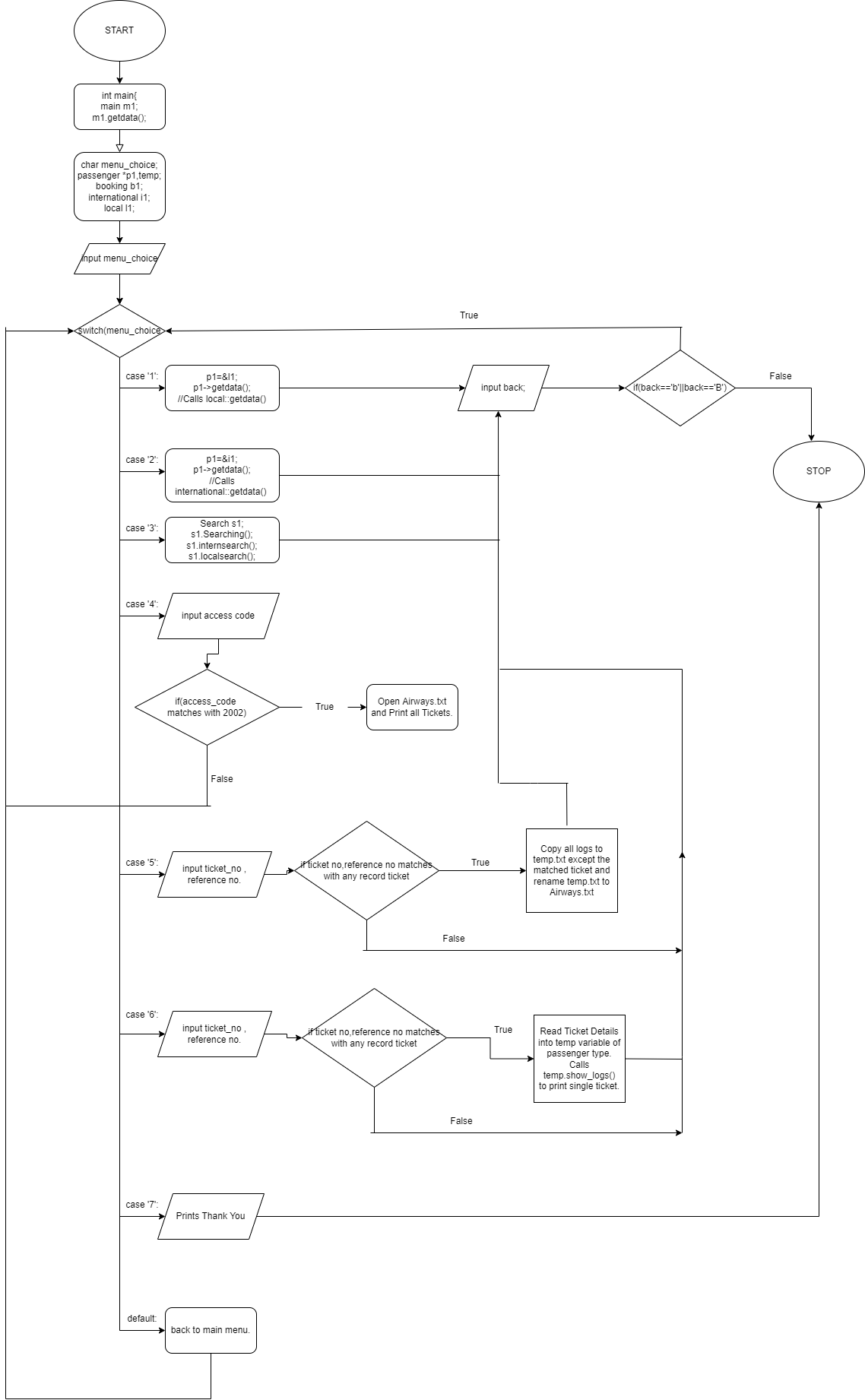
1. getdata(void):- main1::getdata() Provides User-Interface to access all Functionalities of the Program.

* Return type void.
* Provides following choices:
* To book local to local flight.
* To book local to international or international to international flight.
* To Search for Flights.
* To check logs.
* To Cancel Ticket.
* To Print Personal Ticket.
* To Exit Program.

**Functions used-**

* checkCaptcha(string &captcha, string &user\_captcha)
* return 1 if Captcha entered by user is equal to Captcha generated by generateCaptcha().
* return 0 if Captcha entered by user is not equal to Captcha generated by generateCaptcha().
* generateCaptcha(void)
* returns Captcha string generated using <ctime.h>,random function and push\_back function.
* captcha\_validation(void)
* Return type void.
* Calls generateCaptcha() and checkCaptcha() to generate and validate Captcha.
* Terminates program if Captcha is entered wrongly thrice.
* otp\_validation(void)
* Return type void.
* Generates OTP and validates it.
* Terminates Program if user enters wrong OTP thrice.

**Flowchart**



1. **Implementation**

#include <iostream>

#include <fstream>

#include <string.h>

#include <iomanip>

#include <ctime>

#include <conio.h>

#include <stdio.h>

#include <stdlib.h>

#include <windows.h>

using *namespace* std;

*bool* checkCaptcha(string &*captcha*, string &*user\_captcha*);

string generateCaptcha();

*void* captcha\_validation();

*void* otp\_validation();

*char* l\_destination[7][20] = {"DELHI", "MUMBAI", "AYODHYA", "SIACHEN", "BANGLORE", "A", "B"};

*char* i\_destination[14][20] = {"BERLIN", "STOCKHOLM", "RIO", "DENVER", "HELSINKI", "NAIROBI", "BOGOTA", "TOKYO", "MARSELLA", "MOSCOW", "MUMBAI", "DELHI", "C", "D"};

// Class Search for Showing Locations and Flight Routes...

*class* Search{

*public:*

    Search()

    {

        cout << "::FOLLOWING ARE THE FLIGHTS AVAILABLE::\n";

    }

*void* L2I();

*void* internsearch();

*void* localsearch();

};

// Class Payment is used to provide payment options to the user for the flight ticket...

*class* payment

{

*protected:*

*float* amount;

*char* UPI[20];

*long* *long* Bank\_acc;

*char* PIN[7];

*int* i;

*int* j;

*public:*

    // Calc\_Bill() calculates amount of ticket with the help of Boarding location,Destination and Class(Business/Economy)

*void* Calc\_Bill(*char* *a*[], *char* *b*[], *char* *c*[])

    {

*time\_t* t;

        srand((*unsigned*)time(&t));

*bool* temp1, temp2, temp3, temp4, temp5;

        for (i = 0; i < 7; i++)

        {

            for (j = 0; j < 14; j++)

            {

                temp1 = (!strcmp(a, l\_destination[i]));

                temp2 = (!strcmp(b, i\_destination[j]));

                temp5 = (!strcmp(a, i\_destination[j]));

                temp3 = (!strcmp(c, "BUSINESS"));

                temp4 = (!strcmp(c, "ECONOMY"));

                if (((!strcmp(a, "DELHI") && !(strcmp(b, "MUMBAI"))) || (!strcmp(b, "DELHI") && !(strcmp(a, "MUMBAI")))) && temp4)

                {

*int* range = 6000 - 1000 + 1;

                    amount = rand() % range + 1000;

                    goto done;

                }

                else if (((!strcmp(a, "DELHI") && !(strcmp(b, "MUMBAI"))) || (!strcmp(b, "DELHI") && !(strcmp(a, "MUMBAI")))) && temp3)

                {

*int* range = 22000 - 1000 + 1;

                    amount = rand() % range + 2000;

                    goto done;

                }

                else if ((!strcmp(a, "DELHI") || !(strcmp(a, "MUMBAI"))) && temp2 && temp4)

                {

*int* range = 10000 - 5000 + 1;

                    amount = rand() % range;

                    goto done;

                }

                else if ((!(strcmp(a, "DELHI")) || !(strcmp(a, "MUMBAI"))) && temp2 && temp3)

                {

*int* range = 10000 - 5000 + 1;

                    amount = rand() % range + 3000;

                    goto done;

                }

                else if (temp1 && temp3)

                {

*int* range = 22000 - 1000 + 1;

                    amount = rand() % range + 3000;

                    goto done;

                }

                else if (temp1 && temp4)

                {

*int* range = 6000 - 1000 + 1;

                    amount = rand() % range + 1000;

                    goto done;

                }

                else if (temp5 && temp4)

                {

*int* range = 70000 - 1000 + 1;

                    amount = rand() % range;

                    goto done;

                }

                else if (temp5 && temp3)

                {

*int* range = 180000 - 1000 + 1;

                    amount = rand() % range + 50000;

                    goto done;

                }

            }

        }

    done:

        cout << "";

    }

    // waystopay() prints amount of ticket price and asks user to select from payment through UPI or Bank Transfer...

*int* waystopay()

    {

        system("CLS");

        cout << "Total Amount: Rs." << amount << endl;

        cout << "::Please Pay before Printing your Final Ticket::";

        cout << endl

             << endl;

    bb:

        cout << "Please select one of the method to pay:" << endl;

        cout << " 1.UPI" << endl;

        cout << " 2.Bank Transfer" << endl;

*int* ch;

        cin >> ch;

        switch (ch)

        {

        case 1:

            Pay\_UPI();

            break;

        case 2:

            Pay\_Bank();

            break;

        default:

            cout << "Wrong Input...Enter Again" << endl;

            system("pause");

            system("CLS");

            goto bb;

        }

    }

    // Pay\_UPI() asks for UPI Id,Mobile No.,Verify OTPand calls otp\_validation() to make payment...

*void* Pay\_UPI()

    {

        long long mobile;

        cout << "Enter Your UPI ID:" << endl;

        cin >> UPI;

        captcha\_validation();

        cout << "CAPTCHA Verified";

        for (int i = 0; i < 3; i++)

        {

            Sleep(0.4 \* 1000);

            cout << ".";

        }

    phone:

        system("CLS");

        cout << "Enter Your Mobile No.:";

        cin >> mobile;

        long long temp;

        temp = mobile;

        int i = 0;

        while (temp)

        {

            temp /= 10;

            i++;

        }

        if (i != 10)

        {

            cout << "Invalid Number Entered...Enter Again";

            cout << endl

                 << endl;

            system("pause");

            cout << endl

                 << endl;

            goto phone;

        }

        otp\_validation();

        cout << "OTP Verified";

        for (int i = 0; i < 3; i++)

        {

            Sleep(0.4 \* 1000);

            cout << ".";

        }

        cout << endl;

        cout << "Payment Processing...Please wait few seconds";

        for (int i = 0; i < 13; i++)

        {

            Sleep(0.4 \* 1000);

            cout << ".";

        }

        system("CLS");

        cout << "Payment Received" << endl;

        cout << "Redirecting Page back to Airline";

        for (int i = 0; i < 10; i++)

        {

            Sleep(0.4 \* 1000);

            cout << ".";

        }

        system("CLS");

    }

    // Pay\_Bank() asks for Account Name,No.,IFSC code,IPIN and calls captcha\_validation() to make payment...

    void Pay\_Bank()

    {

        long long mobile;

        char name[90];

        long long acc;

        char IFSC[12];

        fflush(stdin);

        cout << "Enter Your Bank Account Name: " << endl;

        gets(name);

    ee:

        cout << "\n\nEnter Your Bank Account No.: " << endl;

        cin >> Bank\_acc;

        cout << "\n\nRe-enter Your Bank Account No.: " << endl;

        cin >> acc;

        if (Bank\_acc != acc)

        {

            cout << "Enter Your Bank Account No. Again";

            for (int i = 0; i < 5; i++)

            {

                Sleep(0.4 \* 1000);

                cout << ".";

            }

            system("CLS");

            goto ee;

        }

        cout << "\n\nEnter Your Bank Account IFSC Code" << endl;

        fflush(stdin);

        gets(IFSC);

        cout << endl

             << endl;

        cout << "Enter Your IPIN" << endl;

        for (int i = 0; i < 6; i++)

        {

            PIN[i] = getch();

            cout << "\*";

        }

        Sleep(0.4 \* 1000);

        cout << endl;

        system("CLS");

        cout << "Please Wait For Bank Confirmation";

        for (int i = 0; i < 10; i++)

        {

            Sleep(0.4 \* 1000);

            cout << ".";

        }

        Sleep(0.4 \* 1000);

        cout << "." << endl;

        cout << "Account Verified..." << endl;

        Sleep(2 \* 1000);

        captcha\_validation();

        cout << "Payment Processing...Please wait few seconds";

        for (int i = 0; i < 15; i++)

        {

            Sleep(0.4 \* 1000);

            cout << ".";

        }

        system("CLS");

        cout << "Payment Received" << endl;

        cout << "Redirecting Page back to Airline";

        for (int i = 0; i < 10; i++)

        {

            Sleep(0.4 \* 1000);

            cout << ".";

        }

        system("CLS");

    }

};

// checkCaptcha() return 1 if Captcha  entered by user is equal to Captcha generated by generateCaptcha()...

bool checkCaptcha(string &captcha, string &user\_captcha)

{

    if (captcha == user\_captcha)

        return true;

    else

        return false;

}

// generateCaptcha() returns Captcha generated using <ctime.h>,random function and push\_back function...

string generateCaptcha()

{

    time\_t t;

    int n = 6;

    srand((unsigned)time(&t));

    char \*chr = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

    string captcha = "";

    while (n--)

        captcha.push\_back(chr[rand() % 62]);

    return captcha;

}

// captcha\_validation() calls generateCaptcha() and checkCaptcha() to generate and validate Captcha...

// captcha\_validation() also terminates program if Captcha is entered wrongly thrice...

void captcha\_validation()

{

    int i = 0;

again:

    system("CLS");

    string captcha = generateCaptcha();

    fflush(stdin);

    cout << captcha;

    fflush(stdin);

    string usr\_captcha;

    fflush(stdin);

    cout << "\nEnter above CAPTCHA: ";

    cin >> usr\_captcha;

    if (checkCaptcha(captcha, usr\_captcha))

        system("PAUSE");

    else

    {

        cout << "Wrong CAPTCHA!! Enter again.";

        i++;

        if (i == 3)

        {

            system("CLS");

            Sleep(1 \* 1000);

            cout << "Payment Failed..." << endl;

            cout << "Please try again later..." << endl;

            Sleep(2 \* 1000);

            cout << "Program Terminating";

            for (int i = 0; i < 5; i++)

            {

                Sleep(0.4 \* 1000);

                cout << "!";

            }

            Sleep(1 \* 1000);

            exit(0);

        }

        cout << endl;

        system("PAUSE");

        goto again;

    }

}

// Class mytime provides function to choose Flight Time and print it in while printing final ticket...

class mytime

{

protected:

    int choose;

public:

    // mytime::getdata() asks user to choose Flight Time...

    // mytime::getdata() also checks for wrong input and return to ask for Flight Time again...

    void getdata()

    {

    undo:

        cout << "ON WHICH TIME YOU WANT TO TRAVEL" << endl;

        cout << "PRESS 1 FOR 11 AM" << endl;

        cout << "PRESS 2 FOR 6 PM" << endl;

        cout << "PRESS 3 FOR 9 PM" << endl;

        cin >> choose;

        if (choose < 1 || choose > 3)

        {

            cout << "Wrong choice Entered..." << endl;

            cout << "Enter Again" << endl;

            system("pause");

            system("CLS");

            goto undo;

        }

    }

    // mytime::show() displays Flight Time while printing Final Ticket...

    void show()

    {

        if (choose == 1)

        {

            cout << "|Flight Time 11 AM                                                        " << endl;

            cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

            cout << "                  GOOD BYE AND HAVE A SAFE JOURNEY                        " << endl;

        }

        if (choose == 2)

        {

            cout << "|Flight Time 6 PM                                                        " << endl;

            cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

            cout << "                  GOOD BYE AND HAVE A SAFE JOURNEY                        " << endl;

        }

        if (choose == 3)

        {

            cout << "|Flight Time 9 PM                                                        " << endl;

            cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

            cout << "                  GOOD BYE AND HAVE A SAFE JOURNEY                        " << endl;

        }

    }

};

// Class passenger is the main class of the program...

// It stores Users' Personal Information and Ticket Details...

class passenger

{

public:

    int ticketno;

    int referenceno;

    char passport[9];

    char lastname[60];

    char firstname[60];

    char address[60];

    char c\_loc[30];

    char dest[30];

    char airclass[9];

    char date[11];

    long long phoneno;

    char gmail[20];

public:

    // show\_logs() is used to print Tickets in Logs...

    void show\_logs()

    {

        cout << "                           AIRLINE TICKET                                 " << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|Ticket No." << ticketno << endl;

        cout << "|Reference No." << referenceno << endl;

        cout << "|Passport No." << passport << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|Passenger Information :                                                  " << endl;

        cout << "|Name :" << lastname << "/" << firstname << "                                     " << endl;

        cout << "|Address :" << address << "                                                   " << endl;

        cout << "|Phone No. :" << phoneno << endl;

        cout << "|Gmail id :" << gmail << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|Flight Information :                                                     " << endl;

        cout << "|Date: " << date << "                                                              " << endl;

        cout << "|BOING 747                                                                " << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|Depart :" << c\_loc << "                                                      " << endl;

        cout << "|Arrive :" << dest << "                                                       " << endl;

        cout << "|\*BAGGAGE allowed 40 Kilos                                                " << endl;

        cout << "|\*contact Airline to Confirm baggage allowance                            " << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << endl;

    }

    // These functions will be defined in Sub-class Booking,Local and International...

    virtual void getdata()

    {

    }

    virtual void show()

    {

    }

};

// otp\_validation() is used to generate OTP and validate it...

// otp\_validation() also terminates Program if user enters wrong OTP thrice...

void otp\_validation()

{

    int i = 0;

o:

    time\_t t;

    int n = 4;

    srand((unsigned)time(&t));

    char \*num = "0123456789";

    string otp = "";

    while (n--)

        otp.push\_back(num[rand() % 10]);

    fflush(stdin);

    cout << "..." << otp << "...";

    string uotp;

    fflush(stdin);

    cout << "\nEnter OTP...: ";

    cin >> uotp;

    fflush(stdin);

    if (uotp != otp)

    {

        cout << "\nWrong OTP...";

        i++;

        if (i == 3)

        // Code to Terminate Program if Wrong OTP is entered Thrice...

        {

            system("CLS");

            Sleep(1 \* 1000);

            cout << "Sorry We cannot verify your Number at the moment..." << endl;

            cout << "Please try again later..." << endl;

            Sleep(2 \* 1000);

            cout << "Program Terminating";

            for (int i = 0; i < 10; i++)

            {

                Sleep(0.4 \* 1000);

                cout << "!";

            }

            exit(0);

        }

        cout << endl;

        system("PAUSE");

        cout << endl;

        goto o;

    }

}

// Class booking inherits passenger publicly and defines getdata() and show() to take input() for personal info and print Ticket respectively...

class booking : public passenger

{

protected:

    int num;

    char ch;

    int option;

public:

    void getdata(int check)

    {

        cout << setw(40) << "\n\n\n           ::please enter the required information for Passenger number::" << endl;

        cout << "\n\n\nEnter the first name of passenger: ";

        fflush(stdin);

        gets(firstname);

        Sleep(0.5 \* 1000);

        cout << endl

             << endl;

        cout << "Enter the last name of the passenger: ";

        gets(lastname);

        Sleep(0.5 \* 1000);

        cout << endl

             << endl;

        cout << "Enter address of passenger:  ";

        gets(address);

        Sleep(0.5 \* 1000);

        cout << endl

             << endl;

    p:

        cout << "Enter your passport number:  ";

        gets(passport);

        int pass = 8;

        // This Snippet of code checks Passport Length and redirect program to input Passport again if Invalid Passport Length is entered by the User...

        if (strlen(passport) != pass)

        {

            cout << "\nEnter your passport number correctly!!";

            cout << endl

                 << endl;

            system("pause");

            system("CLS");

            goto p;

        }

    phone:

        Sleep(0.5 \* 1000);

        system("CLS");

        cout << "Enter your Mobile No.: ";

        cin >> phoneno;

        Sleep(0.5 \* 1000);

        system("CLS");

        long long temp;

        temp = phoneno;

        int i = 0;

        // This Snippet of code checks Mobile No. Length and redirect program to input Mobile No. again if Invalid Mobile NO. Length is entered by the User...

        while (temp)

        {

            temp /= 10;

            i++;

        }

        if (i != 10)

        {

            cout << "Invalid Number Entered...";

            cout << endl

                 << endl;

            system("pause");

            goto phone;

        }

        otp\_validation();

        Sleep(0.5 \* 1000);

        system("CLS");

        cout << "Enter your Gmail ID: ";

        gets(gmail);

        Sleep(0.5 \* 1000);

        system("CLS");

        Search s1;

        // Calls Search::localsearch() to show Local to Local Flight Routes...

        if (check == 1)

        {

            s1.localsearch();

        }

        // Calls Search::internsearch() & Search::L2I() to show International to International and Local to International Flight Routes respectively...

        if (check == 0)

        {

            s1.L2I();

            s1.internsearch();

        }

        cout << endl

             << endl;

        cout << "Enter the Boarding location of passenger(IN CAPITAL LETTERS ONLY):  ";

        gets(c\_loc);

        Sleep(0.5 \* 1000);

        cout << endl

             << endl;

        cout << "Enter the Destination of passenger(IN CAPITAL LETTERS ONLY):  ";

        gets(dest);

    dat:

        Sleep(0.5 \* 1000);

        cout << endl

             << endl;

        cout << "\*\* DECLAIMER:-FLIGHTS ARE AVAILABLE FROM JUNE 2022 TO DECEMBER 2022 \*\*\n\n";

        cout << "Enter Flight Date in DD/MM/YYYY Format:  ";

        gets(date);

        date[2] = '/';

        date[5] = '/';

        // This ladder of ifs check if Entered Date is in Correct Format and logically Correct...

        if ((date[0] == '0' || date[0] == '1' || date[0] == '2' && date[1] >= '1' && date[1] <= '9') || (date[0] == '3' && (date[1] == '1' || date[1] == '0')))

        {

            if ((date[3] == '0' && date[4] >= '6' && date[4] <= '9') || (date[3] == '1' && (date[4] == '1' || date[4] == '2')))

            {

                if (date[6] == '2' && date[7] == '0' && date[8] == '2' && date[9] == '2')

                {

                    cout << "Validating Your Details";

                    for (int i = 0; i < 7; i++)

                    {

                        Sleep(0.4 \* 1000);

                        cout << ".";

                    }

                    cout << endl

                         << endl;

                    cout << "Valid Details Entered";

                }

                else

                {

                    goto err1;

                }

            }

            else

            {

                goto err;

            }

        }

        else

        {

        err:

        err1:

            cout << "Enter Valid Date of Current Year only...";

            cout << endl

                 << endl;

            system("pause");

            system("CLS");

            goto dat;

        }

        for (int i = 0; i < 8; i++)

        {

            Sleep(0.4 \* 1000);

            cout << ".";

        }

        system("CLS");

    }

    // Booking::show() generates Ticket No. ,Reference No. and prints Ticket...

    void show()

    {

        srand(time(0));

        int t = rand();

        ticketno = t;

        int r = rand();

        referenceno = r;

        passenger temp;

        temp.ticketno = this->ticketno;

        temp.referenceno = this->referenceno;

        strcpy(temp.lastname, this->lastname);

        strcpy(temp.firstname, this->firstname);

        strcpy(temp.address, this->address);

        strcpy(temp.c\_loc, this->c\_loc);

        strcpy(temp.dest, this->dest);

        strcpy(temp.airclass, this->airclass);

        strcpy(temp.passport, this->passport);

        strcpy(temp.date, this->date);

        temp.phoneno = this->phoneno;

        strcpy(temp.gmail, this->gmail);

        payment pay;

        pay.Calc\_Bill(temp.c\_loc, temp.dest, temp.airclass);

        pay.waystopay();

        ofstream fillin;

        fillin.open("Airways.txt", ios::app | ios::binary);

        if (!fillin)

        {

            cout << "Cannot open File!!";

            cout << endl;

        }

        fillin.write((char \*)&temp, sizeof(temp));

        fillin.close();

        cout << setw(40) << "Your E-Ticket is :" << endl;

        cout << "                           AIRLINE TICKET                                 " << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|Ticket No." << t << endl;

        cout << "|Reference No." << r << endl;

        cout << "|Passport No." << passport << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|Passenger Information :                                                  " << endl;

        cout << "|Name :" << lastname << "/" << firstname << "                                     " << endl;

        cout << "|Address :" << address << "                                                   " << endl;

        cout << "|Phone No. :" << phoneno << endl;

        cout << "|Gmail id :" << gmail << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|flight Information :                                                     " << endl;

        cout << "|Date: " << date << "                                                              " << endl;

        cout << "|BOEING 747                                                                " << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|Depart :" << c\_loc << "                                                      " << endl;

        cout << "|Arrive :" << dest << "                                                       " << endl;

        cout << "|\*BAGGAGE allowed 40 Kilos                                                " << endl;

        cout << "|\*contact Airline to Confirm baggage allowance                            " << endl;

        cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

    }

};

// Search::L2I() prints LOCAL TO INTERNATIONAL FLIGHT ROUTES...

void Search::L2I()

{

    cout << "---------------------------------------\n";

    cout << "        :LOCAL <-> INTERNATIONAL:\n";

    cout << "---------------------------------------\n";

    for (int i = 0; i < 2; i++)

    {

        for (int j = 0; j < 10; j++)

        {

            switch (i)

            {

            case 0:

                cout << setw(14) << l\_destination[i] << setw(5) << "<-> " << setw(3) << i\_destination[j];

                Sleep(0.3 \* 1000);

                break;

            case 1:

                cout << setw(14) << l\_destination[i] << setw(4) << " <-> " << setw(3) << i\_destination[j];

                Sleep(0.3 \* 1000);

                break;

            default:

                cout << setw(17) << l\_destination[i] << setw(4) << "     <-> " << setw(3) << i\_destination[j];

                Sleep(0.3 \* 1000);

            }

            cout << "\n";

        }

    }

}

// Search::internsearch() prints INTERNATIONAL TO INTERNATIONAL FLIGHT ROUTES...

void Search::internsearch()

{

    cout << "---------------------------------------\n";

    cout << ":INTERNATIONAL <-> INTERNATIONAL:\n";

    cout << "---------------------------------------\n";

    for (int j = 1; j < 10; j++)

    {

        cout << setw(14) << i\_destination[0] << setw(4) << " <-> " << setw(3) << i\_destination[j] << endl;

        Sleep(0.3 \* 1000);

    }

}

// Search::localsearch() prints LOCAL TO LOCAL FLIGHT ROUTES...

void Search::localsearch()

{

    cout << "---------------------------------------\n";

    cout << "        :LOCAL <-> LOCAL:\n";

    cout << "---------------------------------------\n";

    for (int j = 1; j < 5; j++)

    {

        cout << setw(14) << l\_destination[0] << setw(4) << " <-> " << setw(5) << l\_destination[j] << endl;

        Sleep(3);

    }

}

// Class local inherits booking Class and redefines getdata() to book tickets for Local To Local Routes...

class local : public booking

{

protected:

    int b\_seat;

    int e\_seat;

    int ch;

    int k;

    int p;

    int option;

    mytime t1;

public:

    void getdata();

};

// local::getdata() asks User for Airline , Class(Business/Economy) , calls mytime::getdata() , booking::getdata(int) and booking::show() in the end...

void local::getdata()

{

    k = 0;

    p = 0;

    int checkk = 1;

    booking::getdata(checkk);

undo1:

    cout << "In which AIRLINE you want to travel" << endl

         << endl;

    cout << "1)INDIGO" << endl;

    cout << "2)AIR BLUE" << endl;

    cout << "3)ARAB EMIRATES" << endl;

    cout << "4)QATAR AIRWAYS" << endl;

    cout << "enter your choice" << endl;

    ;

    cin >> option;

    if (option < 1 || option > 4)

    {

        cout << "Wrong choice Entered..." << endl;

        cout << "Enter Again" << endl;

        system("pause");

        system("CLS");

        goto undo1;

    }

    system("pause");

    system("CLS");

    switch (option)

    {

    case 1:

        if (option == 1)

            cout << setw(40) << "\n\n\n                                WELCOME TO INDIGO" << endl;

        break;

    case 2:

        if (option == 2)

            cout << setw(40) << "\n\n\n                                WELCOME TO AIRBLUE" << endl;

        break;

    case 3:

        if (option == 3)

            cout << setw(40) << "\n\n\n                                WELCOME TO ARAB EMIRATES" << endl;

        break;

    case 4:

        if (option == 4)

            cout << setw(40) << "\n\n\n                                WELCOME TO QATAR AIRWAYS" << endl;

        break;

    }

    // Loops to check if Entered BOARDING LOCATION & DESTINATION is Available or Not...

    for (int j = 0; j < 7; j++)

    {

        if (strcmp(l\_destination[j], dest) == 0)

        {

            k++;

        }

    }

    for (int i = 0; i < 7; i++)

    {

        if (strcmp(l\_destination[i], c\_loc) == 0)

        {

            p++;

        }

    }

    int check = strcmp(dest, c\_loc);

    if (k == 1 && p == 1 && check != 0)

    {

        cout << " FLIGHT AVAILABLE ON ROUTE " << c\_loc << " -> " << dest << endl;

        b\_seat = 30;

        e\_seat = 40;

    undo2:

        cout << "In which class you want to travel?" << endl;

        cout << "Make your choice" << endl

             << endl;

        cout << "\n\n\n\n\n\n";

        cout << setw(60) << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cout << setw(60) << "\*     PRESS 1 FOR BUSINESS CLASS    \*" << endl;

        cout << setw(60) << "\*     PRESS 2 FOR ECONOMY CLASS     \*" << endl;

        cout << setw(60) << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl

             << endl

             << endl

             << endl;

        cout << "Enter your choice" << endl;

        cin >> ch;

        if (ch < 1 || ch > 2)

        {

            cout << "Wrong choice Entered..." << endl;

            cout << "Enter Again" << endl;

            system("pause");

            system("CLS");

            goto undo2;

        }

        if (ch == 1)

            strcpy(airclass, "BUSINESS");

        if (ch == 2)

            strcpy(airclass, "ECONOMY");

        cout << endl;

        t1.getdata();

        system("CLS");

        booking::show();

        if (ch == 1)

        {

            cout << "|Ticket Information:                                                      " << endl;

            cout << "|BUISNESS CLASS                                                           " << endl;

        }

        if (ch == 2)

        {

            cout << "|Ticket Information:                                                      " << endl;

            cout << "|ECONOMY CLASS                                                            " << endl;

        }

        cout << "|Status: confirmed                                                        " << endl;

        t1.show();

    }

    else

    {

        cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|     SORRY NO FLIGHT AVAILABLE ON THIS ROUTE RIGHT NOW                   " << endl;

        cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

    }

}

// Class international inherits booking Class and redefines getdata() to book tickets for Local To International & International To International Routes...

class international : public booking

{

protected:

    int b1\_seat;

    int e1\_seat;

    int choice;

    int c;

    int b;

    int v;

    char g;

    mytime t1;

public:

    void getdata();

};

// international::getdata() asks User for Airline , Class(Business/Economy) , calls mytime::getdata() , booking::getdata(int) and booking::show(int) in the end...

void international::getdata()

{

    b = 0;

    v = 0;

    int checkk = 0;

    booking::getdata(checkk);

undo3:

    cout << "In which AIRLINE you want to travel" << endl

         << endl;

    cout << "1)INDIGO" << endl;

    cout << "2)AIR BLUE" << endl;

    cout << "3)ARAB EMIRATES" << endl;

    cout << "4)QATAR AIRWAYS" << endl;

    cout << "enter your choice" << endl;

    ;

    cin >> option;

    if (option < 1 || option > 4)

    {

        cout << "\a";

        cout << "Wrong choice Entered..." << endl;

        cout << "Enter Again" << endl;

        system("pause");

        system("CLS");

        goto undo3;

    }

    system("pause");

    system("CLS");

    switch (option)

    {

    case 1:

        if (option == 1)

            cout << setw(40) << "\n\n\n                                WELCOME TO INDIGO" << endl;

        break;

    case 2:

        if (option == 2)

            cout << setw(40) << "\n\n\n                                WELCOME TO AIRBLUE" << endl;

        break;

    case 3:

        if (option == 3)

            cout << setw(40) << "\n\n\n                                WELCOME TO ARAB EMIRATES" << endl;

        break;

    case 4:

        if (option == 4)

            cout << setw(40) << "\n\n\n                                WELCOME TO QATAR AIRWAYS" << endl;

        break;

    }

    // Loops to check if Entered BOARDING LOCATION & DESTINATION is Available or Not...

    for (int j = 0; j < 14; j++)

    {

        if (strcmp(i\_destination[j], dest) == 0)

        {

            v++;

        }

        if (strcmp(i\_destination[j], c\_loc) == 0)

        {

            b++;

        }

    }

    int check = strcmp(dest, c\_loc);

    if (b == 1 && v == 1 && check != 0)

    {

        cout << " FLIGHT AVAILABLE ON ROUTE " << c\_loc << " -> " << dest << endl;

        b1\_seat = 30;

        e1\_seat = 40;

    undo4:

        cout << "In which class you want to travel?" << endl;

        cout << "Make your choice" << endl

             << endl;

        cout << "\n\n\n\n\n\n";

        cout << setw(60) << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cout << setw(60) << "\*     PRESS 1 FOR BUSINESS CLASS    \*" << endl;

        cout << setw(60) << "\*     PRESS 2 FOR ECONOMY CLASS     \*" << endl;

        cout << setw(60) << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl

             << endl

             << endl

             << endl;

        cout << "Enter your choice" << endl;

        cin >> choice;

        if (choice < 1 || choice > 2)

        {

            cout << "Wrong choice Entered..." << endl;

            cout << "Enter Again" << endl;

            system("pause");

            system("CLS");

            goto undo4;

        }

        if (choice == 1)

            strcpy(airclass, "BUSINESS");

        if (choice == 2)

            strcpy(airclass, "ECONOMY");

        cout << endl;

        t1.getdata();

        system("pause");

        system("CLS");

        booking::show();

        if (choice == 1)

        {

            cout << "|Ticket Information:                                                      " << endl;

            cout << "|BUISNESS CLASS                                                           " << endl;

        }

        if (choice == 2)

        {

            cout << "|Ticket Information:                                                      " << endl;

            cout << "|ECONOMY CLASS                                                           " << endl;

        }

        cout << "|Status: confirmed                                                        " << endl;

        t1.show();

    }

    else

    {

        cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

        cout << "|     SORRY NO FLIGHT AVAILABLE ON THIS ROUTE RIGHT NOW                   " << endl;

        cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

    }

}

// Class main1 is Mother of all Classes and Functions...

class main1

{

private:

    char a, f;

    char menu\_choice;

public:

    void getdata();

};

// main1::getdata() Provides User-Interface to access all Functionalities of the Program...

void main1::getdata()

{

backk:

m:

    cout << "\n\n\n\n\n\n";

    cout << setw(60) << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    cout << setw(60) << "\*        AIRWAYS RESERVATION        \*" << endl;

    cout << setw(60) << "\*             MAIN MENU             \*" << endl;

    cout << setw(60) << "\*        ENTER YOUR CHOICE          \*" << endl;

    cout << setw(60) << "\*     PRESS 1 FOR LOCAL BOOKING     \*" << endl;

    cout << setw(60) << "\* PRESS 2 FOR INTERNATIONAL BOOKING \*" << endl;

    cout << setw(60) << "\*   PRESS 3 FOR SEARCH OF FLIGHTS   \*" << endl;

    cout << setw(60) << "\*       PRESS 4 TO CHECK LOGS       \*" << endl;

    cout << setw(60) << "\*  PRESS 5 FOR TICKET CANCELLATION  \*" << endl;

    cout << setw(60) << "\*    PRESS 6 TO PRINT YOUR TICKET   \*" << endl;

    cout << setw(60) << "\*   PRESS 7 FOR EXIT THE PROGRAM    \*" << endl;

    cout << setw(60) << "\*       NOW ENTER YOUR CHOICE:      \*" << endl;

    cout << setw(60) << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    cin >> menu\_choice;

    system("CLS");

    passenger \*p1;

    passenger temp;

    booking b1;

    international i1;

    local l1;

    char back;

    switch (menu\_choice)

    {

    // Case for Local to Local Flight Booking...

    case '1':

    {

        p1 = &l1;

        p1->getdata();

        cout << "To Go To Main Menu Press b/B" << endl;

        cin >> back;

        system("CLS");

        if (back == 'b' || back == 'B')

        {

            goto m;

        }

        else

        {

            cout << "\n\n\n\n\n\n";

            cout << setw(40) << "PROGRAM TERMINATING";

            for (int i = 0; i < 3; i++)

            {

                Sleep(0.4 \* 1000);

                cout << "!";

            }

        }

        break;

    }

        // Case for Local to International & International to International Flight Booking...

    case '2':

    {

        p1 = &i1;

        p1->getdata();

        cout << "To Go To Main Menu Press b/B" << endl;

        cin >> back;

        system("CLS");

        if (back == 'b' || back == 'B')

        {

            goto m;

        }

        else

        {

            cout << "\n\n\n\n\n\n";

            cout << setw(40) << "PROGRAM TERMINATING";

            for (int i = 0; i < 3; i++)

            {

                Sleep(0.4 \* 1000);

                cout << "!";

            }

        }

        break;

    }

        // Case to Print All of the Available Flight Routes....

    case '3':

    {

        Search s1;

        s1.L2I();

        s1.internsearch();

        s1.localsearch();

        cout << "To Go To Main Menu Press b/B" << endl;

        cin >> back;

        system("CLS");

        if (back == 'b' || back == 'B')

        {

            goto m;

        }

        else

        {

            cout << "\n\n\n\n\n\n";

            cout << setw(40) << "PROGRAM TERMINATING";

            for (int i = 0; i < 3; i++)

            {

                Sleep(0.4 \* 1000);

                cout << "!";

            }

        }

        break;

    }

        // Case to access Data File and print all of the Old Tickets

    case '4':

    {

        int check = 2002, access;

        cout << "Enter Access Code to Access Logs...";

        cin >> access;

        if (check == access)

        {

            cout << endl

                 << "Access Granted..." << endl;

            cout << "Redirecting To Ticket Logs.";

            for (int i = 0; i < 3; i++)

            {

                Sleep(0.4 \* 1000);

                cout << ".";

            }

            system("CLS");

        }

        else

        {

            cout << endl

                 << "Access Denied..." << endl;

            cout << "Redirecting To Main Menu.";

            for (int i = 0; i < 3; i++)

            {

                Sleep(0.4 \* 1000);

                cout << ".";

            }

            system("CLS");

            goto backk;

        }

        passenger temp1;

        ifstream fillin;

        fillin.open("Airways.txt", ios::out | ios::binary);

        if (!fillin)

        {

            cout << "Cannot open File!!";

            cout << endl;

        }

        while (fillin.read((char \*)&temp1, sizeof(passenger)))

        {

            temp1.show\_logs();

        }

        fillin.close();

        cout << "To Go To Main Menu Press b/B" << endl;

        cin >> back;

        system("CLS");

        if (back == 'b' || back == 'B')

        {

            goto m;

        }

        else

        {

            cout << "\n\n\n\n\n\n";

            cout << setw(40) << "PROGRAM TERMINATING";

            for (int i = 0; i < 3; i++)

            {

                Sleep(0.4 \* 1000);

                cout << "!";

            }

        }

        break;

    }

        // Case to Cancel Flight Ticket and remove it from the Data File...

    case '5':

    {

        int check = 0;

        passenger temp;

        int tick;

        int reff;

        char confirm;

        cout << "Enter your Ticket No.: ";

        cin >> tick;

        cout << "Enter your Reference No.: ";

        cin >> reff;

        passenger temp1;

        ofstream fillout;

        ifstream fillin;

        fillin.open("Airways.txt", ios::in | ios::binary);

        fillout.open("temp.txt", ios::out | ios::binary);

        if (!fillout)

        {

            cout << "Cannot open File!!";

            cout << endl;

        }

        if (!fillin)

        {

            cout << "Cannot open File!!";

            cout << endl;

        }

        while (fillin.read((char \*)&temp, sizeof(passenger)))

        {

            if (temp.ticketno == tick && temp.referenceno == reff)

            {

                cout << "Ticket Found..." << endl;

                Sleep(1 \* 1000);

                cout << "Confirm y/Y to Proceed Cancellation: ";

                cin >> confirm;

                if (confirm == 'y' || confirm == 'Y')

                {

                    cout << "Ticket Canceled...Thank You" << endl

                         << endl;

                    check++;

                }

            }

            else

            {

                fillout.write((char \*)&temp, sizeof(passenger));

            }

        }

        fillout.close();

        fillin.close();

        remove("Airways.txt");

        rename("temp.txt", "Airways.txt");

        if (check == 0)

        {

            cout << "No Ticket Found..." << endl

                 << endl;

            Sleep(1 \* 1000);

        }

        cout << "To Go To Main Menu Press b/B" << endl;

        cin >> back;

        system("CLS");

        if (back == 'b' || back == 'B')

        {

            goto m;

        }

        else

        {

            cout << "\n\n\n\n\n\n";

            cout << setw(40) << "PROGRAM TERMINATING";

            for (int i = 0; i < 3; i++)

            {

                Sleep(0.4 \* 1000);

                cout << "!";

            }

        }

    }

        // Case to print your Ticket...

    case '6':

    {

        int checkk = 0;

        passenger temp;

        int tick;

        int reff;

        char confirm;

        cout << "Enter your Ticket No.: ";

        cin >> tick;

        cout << "Enter your Reference No.: ";

        cin >> reff;

        ifstream fillin;

        fillin.open("Airways.txt", ios::in | ios::binary);

        if (!fillin)

        {

            cout << "Cannot open File!!";

            cout << endl;

        }

        while (fillin.read((char \*)&temp, sizeof(passenger)))

        {

            if (temp.ticketno == tick && temp.referenceno == reff)

            {

                cout << "Ticket Found..." << endl;

                Sleep(1 \* 1000);

                system("CLS");

                temp.show\_logs();

                checkk = 1;

            }

        }

        if (checkk != 1)

        {

            cout << "\nTicket Not Found...\n";

        }

        fillin.close();

        cout << "To Go To Main Menu Press b/B" << endl;

        cin >> back;

        system("CLS");

        if (back == 'b' || back == 'B')

        {

            goto m;

        }

        else

        {

            cout << "\n\n\n\n\n\n";

            cout << setw(40) << "PROGRAM TERMINATING";

            for (int i = 0; i < 3; i++)

            {

                Sleep(0.4 \* 1000);

                cout << "!";

            }

        }

    }

        // Program Terminating Case...

    case '7':

        cout << "\n\n\n\n\n";

        Sleep(0.5 \* 1000);

        cout << "\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cout << "\t\t\*                     THANK YOU FOR USING OUR PROJECT                            \*" << endl;

        cout << "\t\t\* WE CORDIALLY ACKNOWLEDGE THE INVOLVEMENT AND INTERACTION OF FOLLOWING NOTABLES \*" << endl;

        cout << "\t\t\*                          IN COMPLETING THIS PROJECT                            \*" << endl;

        cout << "\t\t\*:-OUR FELLOW CLASSMATES                                                         \*" << endl;

        cout << "\t\t\*:-GOOGLE, GEEKS FOR GEEKS AND OTHER CODING PLATFORMS                            \*" << endl;

        cout << "\t\t\*:-MS. Nisthha Singh Ma'am                                                       \*" << endl;

        cout << "\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        Sleep(1 \* 1000);

        getchar();

        exit(0);

    default:

        goto m;

    }

};

// main() makes class main1 instance and calls main1::getdata();

*int* main()

{

    system("CLS");

    system("Color 0A");

    cout << "\n\n\n\n\n\n\n\n\n\n\t\t    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

    cout << "                  \n\t\t    \* WELCOME TO FLIGHT RESERVATION SYSTEM \*";

    cout << "                  \n\t\t    \*            Developed by:             \*";

    cout << "                  \n\t\t    \*            Devang Sharma             \*";

    cout << "                  \n\t\t    \*            Amritanshu                \*";

    cout << "                  \n\t\t    \*            Radhika Jangra            \*";

    cout << "                  \n\t\t    \*            Srishti Sinha             \*";

    cout << "                  \n\t\t    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

    cout << "                  \n\n\t\t    ";

    system("pause");

    system("CLS");

    main1 m1;

    m1.getdata();

    return 0;

}

1. **Output**

