**FLIGHT MANAGEMENT SYSTEM**

#include<iostream>

#include<bits/stdc++.h>

using namespace std;

class Flight{

public:

string flightname;

string flightnumber;

string source;

string destination;

int seats;

int seatallotment=0;

string dept\_time;

string day;

Flight(string fnum, string fname, string sour, string dest, int seat,string tm,string d) {

flightnumber=fnum;

flightname=fname;

source=sour;

destination=dest;

seats=seat;

dept\_time=tm;

day=d; }

void displayflights() const{

cout<<"Flight number:"<<flightnumber<<endl;

cout<<"Flight name:"<<flightname<<endl;

cout<<"Flight source:"<<source<<endl;

cout<<"Flight destination:"<<destination<<endl;

cout<<flightname<<" "<<"seats:"<<seats<<endl;

cout<<"Departure time:"<<dept\_time<<endl;

cout<<"Days:"<<day<<endl; }

void update(string new\_source,string new\_destination,int new\_seats,string new\_time,string new\_day){

source=new\_source;

destination=new\_destination;

seats=new\_seats;

dept\_time=new\_time;

day=new\_day; }

int bookseat(){

if(seats!=0){

seats--;

seatallotment++;

return 1; }

else

return 0; } };

class FlightManagement{

private:

vector<Flight>flights;

public:

void addflight(const Flight&flight){

flights.push\_back(flight); }

void displayallflights() const{

if(flights.empty()){

cout<<"No flights available"<<endl;

return; }

for (const auto& flight:flights){

flight.displayflights();

cout<<"xxxxxxxxxxxxxxxxxxxxxxx"<<endl;} }

void updateflight(string flightnumber){

for( auto flight :flights){

if(flight.flightnumber==flightnumber){

cout<<"Enter new source:"<<endl;

string new\_source;

cin>>new\_source;

cout<<"Enter new destination:"<<endl;

string new\_destination;

cin>>new\_destination;

cout<<"Enter new available seats:"<<endl;

int new\_seats;

cin>>new\_seats;

cin.ignore();

cout<<"Enter new time:"<<endl;

string new\_time;

cin>>new\_time;

cout<<"Enter new days:"<<endl;

string new\_day;

cin>>new\_day;

flight.update(new\_source,new\_destination,new\_seats,new\_time,new\_day);

cout<<"Flight details is updated"<<endl;

cout<<"xxxxxxxxxxxxxxxxxxxxxxxxxxxxx"<<endl;

return;} }

cout<<"Flight is not found"<<endl;

cout<<"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"<<endl;}

void display\_flightssour(string source,string destination){

int section=1;

cout<<"The list of flights are:"<<endl;

for(auto& flight:flights){

if(flight.source==source&&flight.destination==destination){

cout<<section<<". "<<endl;

cout<<"Flight number:"<<flight.flightnumber<<endl;

cout<<"Flight name:"<<flight.flightname<<endl;

cout<<"Flight seats:"<<flight.seats<<endl;

cout<<"Departure Time:"<<flight.dept\_time<<endl;

cout<<"Day:"<<flight.day<<endl;

cout<<"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"<<endl;

section++; }

} }

void bookticket(string source,string destination,string flightnumber){

for(auto flight:flights){

if(flight.source==source&&flight.destination==destination&&flight.flightnumber==flightnumber){

if(flight.bookseat()){

cout<<"Flight is booked"<<endl;

cout<<"Flight number is:"<<flight.flightnumber<<endl;

cout<<"Seat number is:"<<flight.seatallotment<<endl;}

else

cout<<"Seats are not available"<<endl;

return; } }

cout<<"Flight is not found"<<endl;

}void seatavailable(string flightnumber){

for(auto flight:flights){

if(flight.flightnumber==flightnumber)

cout<<flight.seats<<endl; } } };

int main(){

FlightManagement fm;

while(1){

cout<<"Welcome to Flight Management System!"<<endl;

cout<<"1.Display all the fights"<<endl;

cout<<"2.Add new flight"<<endl;

cout<<"3.Update the existing flight"<<endl;

cout<<"4.Display the Seat availability"<<endl;

cout<<"5.Book flight"<<endl;

cout<<"6.Exit"<<endl;

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

int query;

cin>>query;

cin.ignore();

switch(query){

case 1: { fm.displayallflights();

break; }

case 2: { cout<<"Enter flight Number:"<<endl;

string flightnumber;

cin>>flightnumber;

cout<<"Enter flight Name:"<<endl;

string flightname;

cin>>flightname;

cout<<"Enter flight Source:"<<endl;

string source;

cin>>source;

cout<<"Enter flight destination:"<<endl;

string destination;

cin>>destination;

cout<<"Enter Seat Availability:"<<endl;

int seats;

cin>>seats;

cin.ignore();

cout<<"Enter Departure Time:"<<endl;

string dept\_time;

getline(cin,dept\_time);

cout<<"Enter day:"<<endl;

string day;

getline(cin,day);

fm.addflight(Flight(flightnumber,flightname,source,destination,seats,dept\_time,day));

cout<<"Flight is added"<<endl;

break; }

case 3:{ cout<<"Enter Flight number to update:"<<endl;

string flightnumber;

cin>>flightnumber;

fm.updateflight(flightnumber);

break; }

case 4:{ cout<<"Enter flight number to check seats:"<<endl;

string flightnumber;

cin>>flightnumber;

fm.seatavailable(flightnumber);

break; }

case 5:{ cout<<"Enter the source and destination to book"<<endl;

string source,destination;

cin>>source;

cout<<"<->";

cin>>destination;

fm.display\_flightssour(source,destination);

cout<<"Enter Flight number:"<<endl;

string flightnumber;

cin>>flightnumber;

cin.ignore();

fm.bookticket(source,destination,flightnumber);

break; }

case 6:{ cout<<"Flight Management System is exited."<<endl;

break; }

default:{ cout<<"Invalid query.Try another queries"<<endl;

break; } } } }