**INTRODUCTION**

The project IPL Tickets Booking System in C++ makes use of the concepts of Object Oriented Programming (OOPs) and file handling.

The concept of this system is to make the transaction of booking the tickets for the upcoming IPL matches very convenient for the user and also to satisfy the customer’s needs.

In this project we have made and used 5 data files- schedule.txt**,** Wankhede Stadium.txt,Arun Jaitley.txt,data.txt and accounts.txt. These files are displayed and appended accordingly.

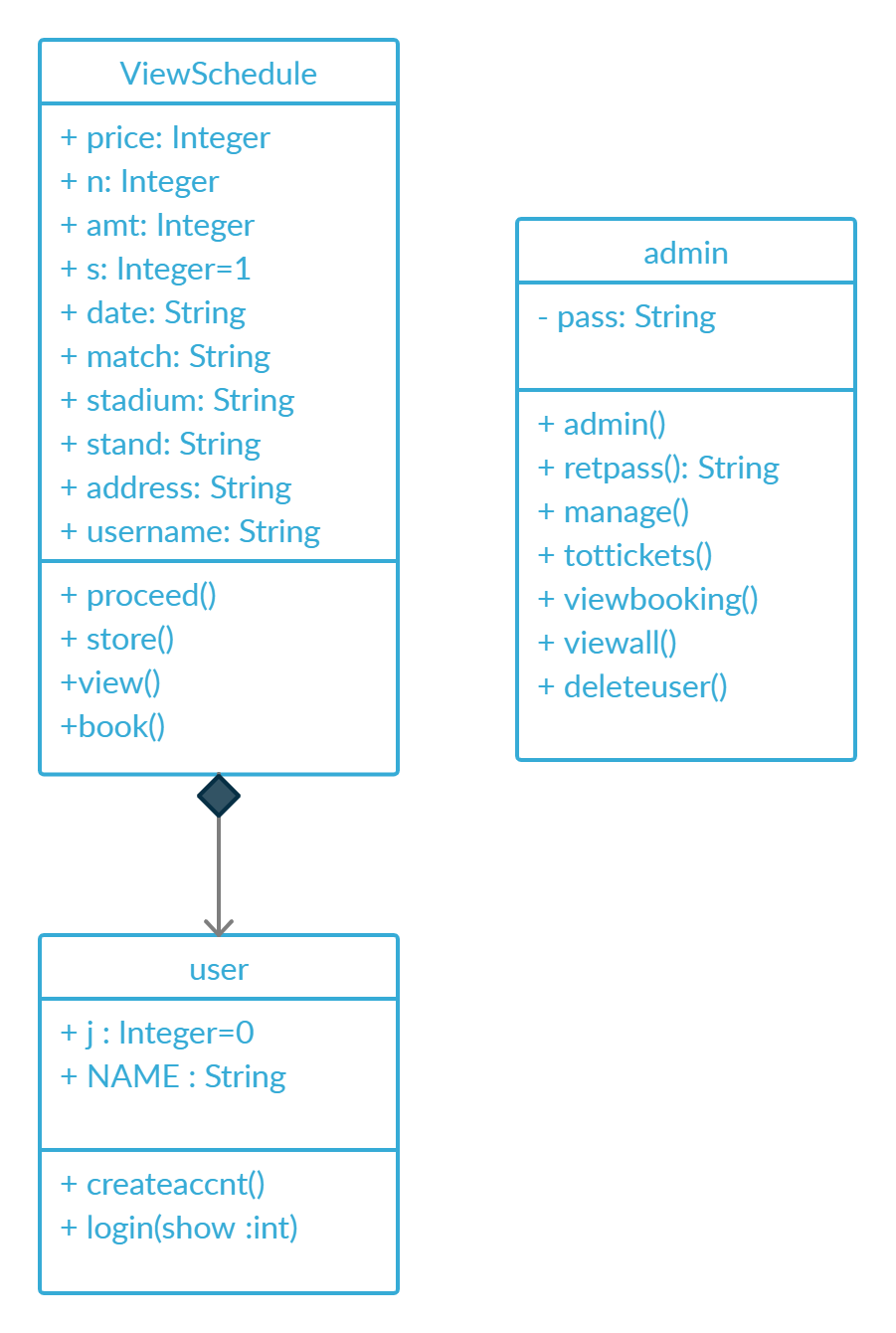
The project features two modes i.e., Admin mode and User mode. This project allows user registration in user mode and to book tickets the user will then have to login to User mode. Admin mode is also used to display total number of seats booked and revenue generated etc. You can choose the match, location of the seats as well as the total no of tickets you want. After that, you proceed towards the payment and confirm the transaction. To access the administrator mode, you first have to enter the correct password. Afterwards, you can either edit the detailed database or view the ticket booking database. Users can get both match details and ticket booking details. The details provided for ticket booking are date, time, venue, no of seats in each wing and fare per ticket. You can choose the tickets accordingly.  And in the case of ticket booking database, the information displayed is username, match, date, stadium, price, no of tickets and amount paid.

**OBJECTIVE**

The main objective of the C++ Project on IPL Ticket Booking System is to manage the details of Booking seats, User and schedule of the matches. It manages all the information about ticket-booking, Payment and Schedule. The project IPL Tickets Booking System in C++ makes use of the concepts of Object Oriented Programming (OOPS),vectors and file handling. The purpose of the project is to build an application program to reduce the manual work for managing the ticket-booking, Seats, Payment and User. It forms a database of the information and keeps the track of all the details about the User, ticket-booking and schedule of the matches. The project will allow the user to book tickets for upcoming IPL matches. we have made and used 5 data files- schedule.txt, Wankhede Stadium.txt, Arun Jaitley.txt,data.txt and accounts.txt. These files are displayed and appended accordingly.

* User can book tickets from any of the listed upcoming IPL matches.
* It tracks all the information of seats, payment and user.
* Manage the information of seats.
* Integration of all records of users.
* View and manage the ticket-booking details.
* To add and update the schedule of the matches.
* Manage user accounts
* Displays the total revenue generated per stadium
* Also displays the total number of seats booked per stadium
* The password of Admin Mode is: PROJECT

**CLASS DIAGRAM**



**DISCUSSION**

The main menu features three options: Admin mode, User mode and Exit. You can login into user mode to book tickets. Only admins can login to admin mode.

In the user mode, if you are a new user you will have to create an account or you can directly login to book tickets. Once you create a new account, login with the same account so that you can book the tickets. The schedule of upcoming IPL matches is displayed along with the date and stadium of each match. User will have to enter the serial number of the match of his/her own choice. After this, a list of tickets in different stands is displayed with their prices, so that you can choose the stand and the number of tickets according to your budget and convenience. After choosing the above-mentioned details, the user is asked to login once again for security purposes. After logging in successfully, enter the amount and the residential address and thus your ticket is booked. After that the user can log out safely.

While in the admin mode, only the admin can login (password: PROJECT). It displays 4 options: to find total number of seats booked and revenue generated, to view booking details, to manage user accounts and lastly, to return to Main Menu. For finding out the number of seats booked and revenue generated the admin has to enter the correct stadium name (either Arun Jaitley or Wankhede). In the manage user account option, the admin can either view all the user accounts or can delete a user account.

We have used data files such as schedule.txt that displays the schedule of upcoming IPL matches while Wankhede Stadium.txt and Arun Jaitley.txt displays the price of tickets in each stand. Also, data.txt and accounts.txt are the data files that are appended and displayed.

**CODE**

#include<iostream>

#include<fstream>

#include<stdio.h>

#include<string.h>

#include<sstream>

#include<vector>

#include<algorithm>

using namespace std;

string a;

int xyz=1, time=0;

void userlogin();

void choice();

void booking();

void adminlogin(int);

vector<string> tokenize(string s, string del = " ")

{

vector<string> store;

int start = 0;

int end = s.find(del);

while (end != -1)

{

store.push\_back(s.substr(start, end - start));

start = end + del.size();

end = s.find(del, start);

}

store.push\_back(s.substr(start, end - start));

return store;

}

class user

{

public:

int j = 0;

string NAME;

void createaccnt()

{

ifstream inputFile("accounts.txt");

string line, Name, pass, name;

j = 0;

getchar();

cout<<"\n CREATE YOUR ACCOUNT \n";

k:

cout << "\nEnter Username: ";

getline(cin, Name);

if ((Name=="")&&(Name!="Username"))

{

cout<<"\nPlease enter a valid username\n";

goto k;

}

while (getline(inputFile, line))

{

istringstream ss(line);

string name, pass;

ss >> name ;

if (name==Name)

{

j = 1;

time++;

cout<<"\nUSERNAME ALREADY EXISTS. TRY USING A DIFFERENT NAME";

break;

}

}

if (j!=1)

{

p:

cout << "SET a Password: ";

getline(cin, pass);

if (pass.size()<6)

{

cout<<"\nPlease enter a 6-8 character long password\n";

goto p;

}

std::ofstream outfile;

outfile.open("accounts.txt", std::ios\_base::app);

outfile <<"\n\n\t"<<Name<<"\t\t\t\t"<<pass;

cout<<"\nAccount created Successfully!!\n";

outfile.close();

}

}

void login(int show)

{

ifstream inputFile("accounts.txt");

string line, Name, Pass;

int i=0;

cout<<"\n LOGIN TO YOUR ACCOUNT \n\n";

if(show==1)

{

getchar();

}

k:

cout<<"Enter your Username: ";

getline(cin, Name);

if ((Name=="")&&(Name!="Username"))

{

cout<<"\nPlease enter a valid username\n";

goto k;

}

if ((xyz==2)&&(Name!=a))

{

j=0;

time++;

if(time>=3)

{

return;

}

cout<<"\nPLEASE USE THE SAME ACCOUNT FROM WHICH YOU HAVE LOGGED IN BEFORE!!\n";

goto k;

}

while (getline(inputFile, line))

{

istringstream ss(line);

string name, pass;

ss >> name ;

if (name==Name)

{

i=1;

ss >> pass;

cout<<"Enter Password: ";

getline(cin, Pass);

if (Pass==pass)

{

cout<<"\nLOGIN SUCCESSFULL\n";

NAME=name;

j=1;

}

else

{

cout<<"\nINCORRECT PASSWORD\n";

}

break;

}

}

if (i==0)

{

cout<<"\nACCOUNT NOT FOUND\n";

}

}

};

class admin

{

string pass;

public:

admin()

{

pass="PROJECT";

}

string retpass()

{

return pass;

}

void manage();

void tottickets()

{

string stad,line;

ifstream fin;

vector<string> accpt;

int tick,amt;

int tt=0;

int fa=0;

getchar();

cout<<"\nEnter name of any stadium: ";

getline(cin,stad);

int totaltickets=0;

fin.open("data.txt");

while(getline(fin,line))

{

if(line.find(stad)!=string::npos)

{

accpt=tokenize(line,"\t");

string s1,s2,temp;

temp=stad;

transform(temp.begin(),temp.end(),temp.begin(),::toupper);

if(temp=="WANKHEDE" || temp=="WANKHEDE STADIUM")

{

s1+=accpt[8];

s2+=accpt[10];

}

else

{

s1+=accpt[9];

s2+=accpt[11];

}

tick=stoi(s1);

amt=stoi(s2);

tt+=tick;

fa+=amt;

}

}

cout<<"\nTotal tickets booked in the stadium: "<<tt<<"\nTotal revenue generated: Rs. "<<fa<<endl;

cout<<"\nPress any key to continue ";

getchar();

adminlogin(0);

}

void viewbooking()

{

ifstream fin;

fin.open("data.txt");

string read;

while(getline(fin,read))

{

string data;

istringstream ss(read);

data=read;

cout<<data<<endl;

}

fin.close();

cout<<"\nPress any key to continue ";

getchar();

getchar();

adminlogin(0);

}

void viewall()

{

ifstream fin;

fin.open("accounts.txt");

string read;

while(getline(fin,read))

{

string data;

istringstream ss(read);

data=read;

cout<<data<<endl;

}

fin.close();

cout<<"\nPress any key to continue ";

getchar();

getchar();

adminlogin(0);

}

void deleteuser()

{

string deluser;

string inp;

ifstream fin;

ofstream ftemp;

fin.open("accounts.txt");

cout<<"\n\t\*\*\*\*\*\*\*\*\*\*\*\* Active User Accounts \*\*\*\*\*\*\*\*\*\*\*\*\*\* \n\n";

while(getline(fin,inp))

{

/\* istringstream ss(inp);

string user;

ss>>user;

if(user!="" && user!="Username") \*/

cout<<inp<<endl;

}

fin.close();

del:

int found=0;

ftemp.open("temp.txt");

fin.open("accounts.txt");

cout<<"\nEnter user name you want to delete: ";

getchar();

cin>>deluser;

while(getline(fin,inp))

{

string user;

istringstream ss(inp);

ss>>user;

if(user!="" && user!=deluser)

{

ftemp<<"\n\n"<<inp;

}

else if(user==deluser)

found = 1;

}

fin.close();

ftemp.close();

if(found==1)

{

remove("accounts.txt");

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

cout<<"\nUser deleted successfully"<<endl;

}

else

{

cout<<"\nInvalid User name. Please enter again!\n";

goto del;

}

cout<<"\nPress any key to continue ";

getchar();

getchar();

adminlogin(0);

}

};

class ViewSchedule

{

public:

int price, n, amt,s=1;

string date, match, stadium, stand, address,username;

void proceed();

void store();

void view()

{

char ch;

fstream fp;

fp.open("schedule.txt", fstream::in);

if(!fp)

{

cout<<"\nError Occurred!";

}

while(fp>>noskipws>>ch)

cout<<ch;

fp.close();

}

void book()

{

ifstream inputFile("schedule.txt");

string line;

int x, i=0;

getchar();

cout<<"\nEnter your choice: ";

cin>>x;

while (getline(inputFile, line))

{

istringstream ss(line);

int sno;

ss >> sno ;

if (sno==x)

{

i=1;

ss >> date >> match >> stadium;

if (stadium == "Arun")

stadium = "Arun Jaitley";

else if (stadium == "Wankhede")

stadium = "Wankhede Stadium";

cout<<"\nDATE: "<<date<<"\nTIME: 7:30 pm (IST)"<<"\nMATCH: "<<match<<"\nVenue: "<<stadium;

cout<<"\n\nPress any key to continue ";

break;

}

}

if (i==0)

{

cout<<"\nWRONG INPUT\n";

booking();

}

}

};

void ViewSchedule:: proceed()

{

char ch;

string f, line;

fstream fp;

if (stadium=="Wankhede Stadium")

{

fp.open("Wankhede Stadium.txt", fstream::in);

f = "Wankhede Stadium.txt";

}

else

{

fp.open("Arun Jaitley.txt", fstream::in);

f = "Arun Jaitley.txt";

}

getchar();

getchar();

while(fp>>noskipws>>ch)

cout<<ch;

fp.close();

int x, i=0;

ifstream inputFile(f);

cout<<"\nEnter your choice: ";

cin>>x;

while (getline(inputFile, line))

{

istringstream ss(line);

int sno;

ss >> sno ;

if (sno==x)

{

i=1;

ss >> stand >> price ;

break;

}

}

if (i==0)

{

cout<<"\nWRONG INPUT\n";

booking();

}

g:

cout<<"\n"<<stand<<"\t@ "<<price<<" Rs per ticket\n";

cout<<"\nEnter 0 to cancel booking.\nOR Enter the number of tickets: ";

cin>>n;

if (n<=0)

{

cout<<"\nBooking Cancelled\n";

booking();

}

if (n>50)

{

cout<<"\nYou can not book more than 50 TICKETS at a time\n";

goto g;

}

getchar();

cout<<"\nTOTAL AMOUNT = "<<n\*price;

cout<<"\n\nLogin again(For security reasons) to proceed for payment: \n";

user u;

t:

xyz=2;

u.login(0);

if (u.j == 0)

{

time++;

if (time>=3)

{

cout<<"\nSORRY, ACCOUNT COULD NOT BE CONFIMED.\nTRY AGAIN AFTER SOME TIME\n";

userlogin();

}

else

{

cout<<"\nTRY AGAIN\n";

goto t;

}

}

cout<<"\nPayment Amount: ";

cin>>amt;

if (amt!=price\*n)

{

cout<<"\nAmount entered does not match with the Total Amount\n\tTRANSACTION CANCELLED\n";

}

else

{

getchar();

cout<<"\nEnter your residential address: ";

getline(cin,address);

cout<<"\n\t\tTRANSACTION SUCCESSFULL!!\n";

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

cout<<"\n\nMATCH: "<<match<<"\n\nDATE- "<<date<<"\n\nTIME- 7:30 pm (IST)"<<"\n\nVenue: "<<stadium<<"\n\nSeat Location: "<<stand;

cout<<"\n\nNo. of TICKETS purchased: "<<n<<"\n\nAmount Paid: "<<amt<<" @ "<<price<<" Rs per ticket";

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

cout<<"\n\nThe Tickets will be delivered at your doorstep SOON!!\n\tThankss for purchasing.\n\n";

store();

cout<<"\nPress any key to continue ";

getchar();

}

booking();

}

void ViewSchedule:: store()

{

ifstream inputFile("data.txt");

string line;

int sno;

while (getline(inputFile, line))

{

istringstream ss(line);

ss >> sno ;

}

s=sno+1;

std::ofstream outfile;

outfile.open("data.txt", std::ios\_base::app);

outfile <<"\n\n"<<s<<"\t"<<username<<"\t\t"<<match<<"\t"<<date<<"\t"<<stadium;

if(stadium=="Arun Jaitley")

{

outfile<<"\t\t"<<price<<"\t\t"<<n<<"\t\t"<<amt<<"\t\t"<<stand;

}

else

{

outfile<<"\t"<<price<<"\t\t"<<n<<"\t\t"<<amt<<"\t\t"<<stand;

}

outfile.close();

}

int main()

{

choice();

}

void choice()

{

cout<<"\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* IPL TICKETS RESERVATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

int ch;

cout<<"\n\n1.Admin Mode \n2.User Mode \n3.Exit";

cout<<"\nIn which mode do you want to login? ";

cin>>ch;

switch(ch)

{

case 1:

adminlogin(1);

break;

case 2:

userlogin();

break;

case 3:

exit(0);

break;

default:

cout<<"\nWRONG INPUT";

choice();

}

}

void userlogin()

{

user u1;

int ch;

label:

cout<<"\n\n # # # # # # # # # # # # USER - MODE # # # # # # # # # # # #\n";

cout<<"\n1.LOGIN \n2.Create an Account \n3.Return to main menu";

cout<<"\nEnter choice: ";

cin>>ch;

switch(ch)

{

case 1:

u1.login(1);

if (u1.j==0)

{

goto label;

}

a=u1.NAME;

booking();

break;

case 2:

u1.createaccnt();

goto label;

break;

case 3:

choice();

break;

default:

cout<<"\nWRONG INPUT.. Try Again";

goto label;

}

}

void adminlogin(int x)

{

admin a1;

string s;

if(x==1)

{

cout<<"\nEnter password: ";

cin>>s;

while(s!=a1.retpass())

{

cout<<"Wrong Password, Try again ";

cin>>s;

}

}

cout<<"\n\n # # # # # # # # # # # # ADMIN - MODE # # # # # # # # # # # #\n";

int ch;

label2:

cout<<"\n1. Find total number of seats booked and revenue generated"<<endl;

cout<<"2. View booking details"<<endl;

cout<<"3. Manage user accounts"<<endl;

cout<<"4. Return to Main Menu "<<endl;

cout<<"Enter your choice: ";

cin>>ch;

switch(ch)

{

case 1:

{

a1.tottickets();

break;

}

case 2:

a1.viewbooking();

break;

case 3:

{

a1.manage();

break;

}

case 4:

choice();

break;

default:

cout<<"\nError: Wrong choice!"<<endl<<"Enter choice again.";

goto label2;

break;

}

}

void booking()

{

ViewSchedule c;

c.username=a;

int y;

l:

cout<<"\n............. "<<c.username<<"'s ACCOUNT .................\n\n";

cout<<"\n1. BOOK TICKETS \n2. LOGOUT\n";

cout<<"Enter your choice: ";

cin>>y;

switch(y)

{

case 1:

cout<<"\n\n======================== IPL TICKET BOOKING ========================\n";

c.view();

c.book();

c.proceed();

break;

case 2:

userlogin();

break;

default:

cout<<"\nWrong Input.. TRY AGAIN\n";

goto l;

}

};

void admin::manage()

{

int ch;

admin a;

cout<<"\n1. View all user accounts \n2. Delete a user account";

cout<<"\nEnter your choice: ";

cin>>ch;

switch(ch)

{

case 1:

a.viewall();

break;

case 2:

a.deleteuser();

break;

default:

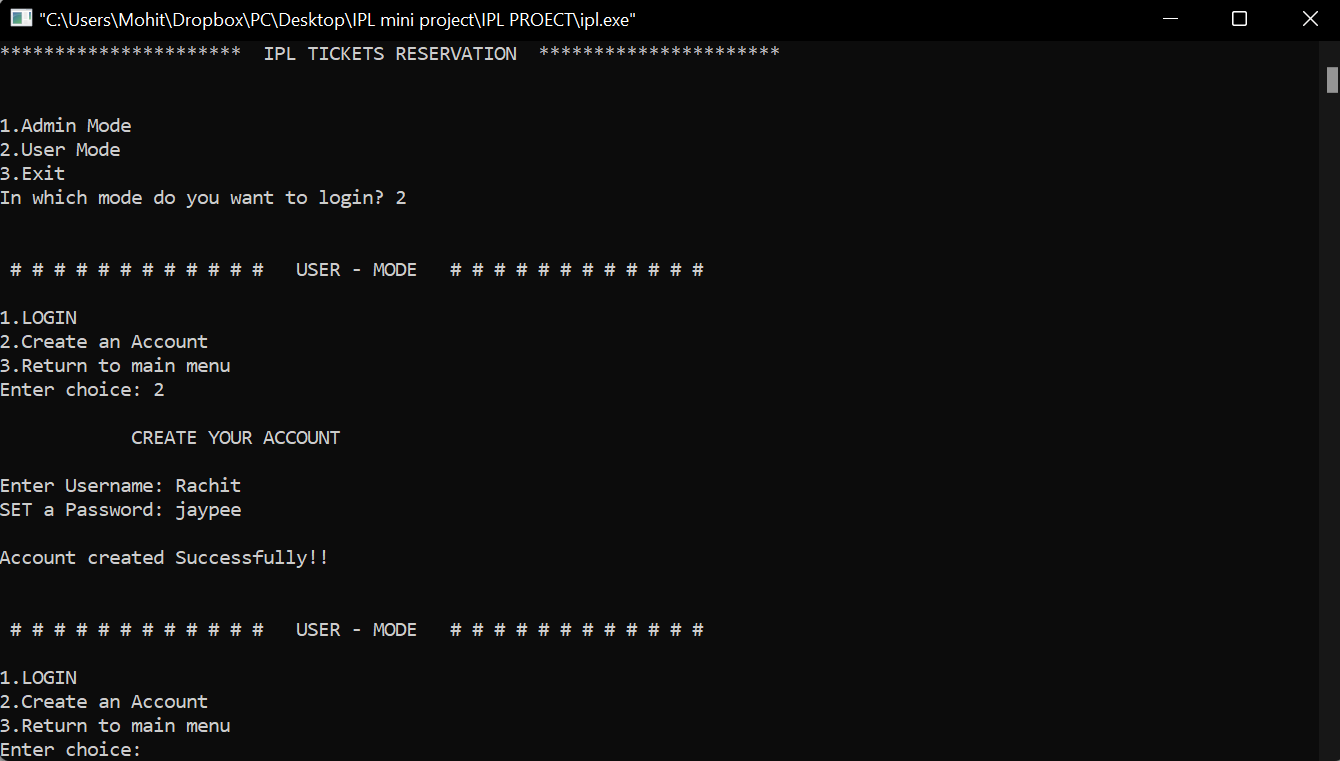
cout<<"\nINVALID INPUT";

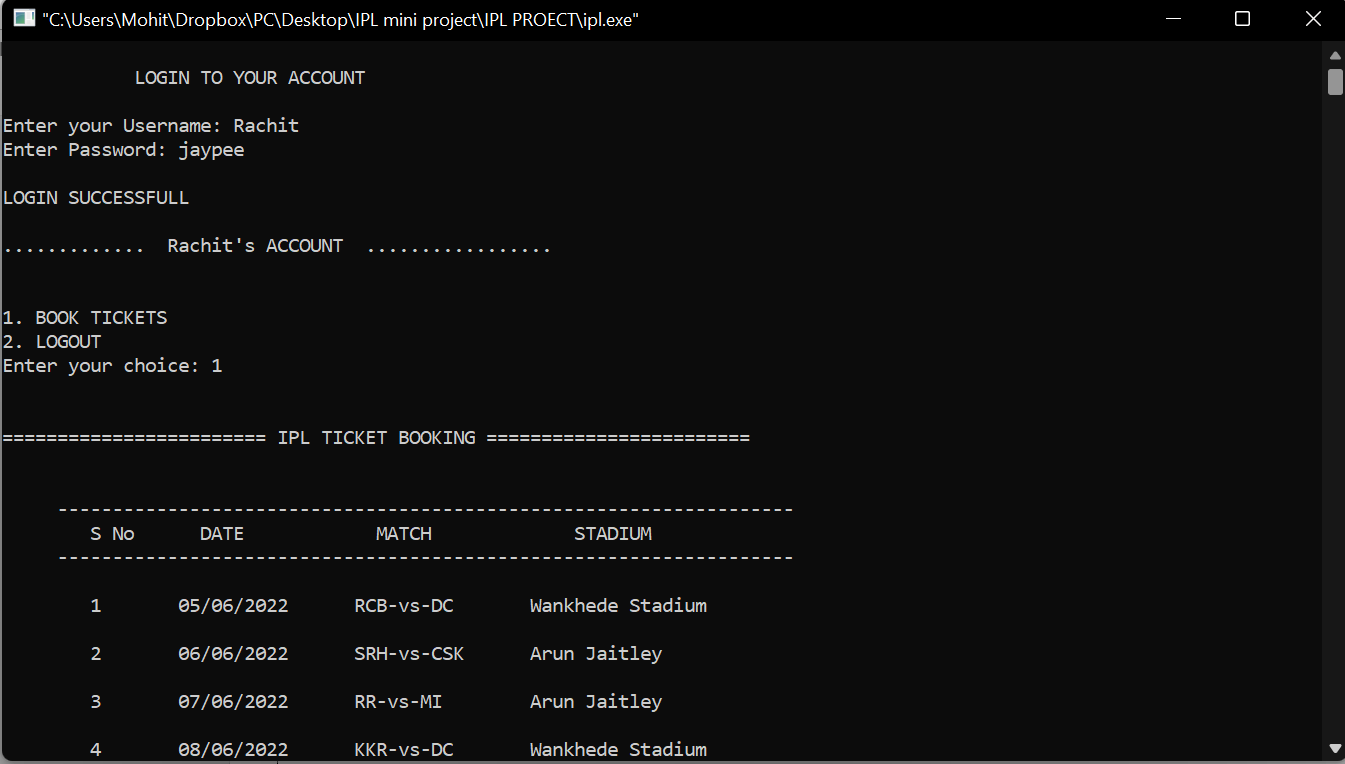
adminlogin(0);

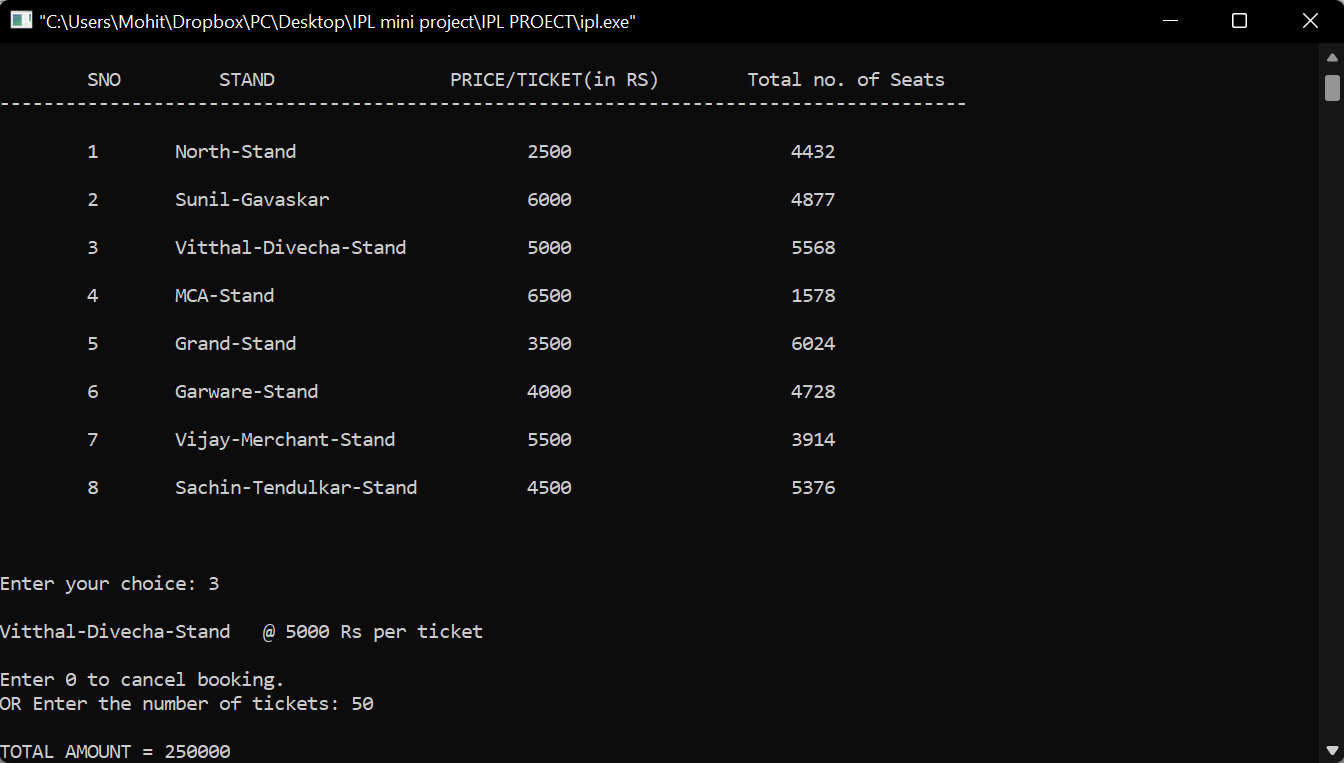
}

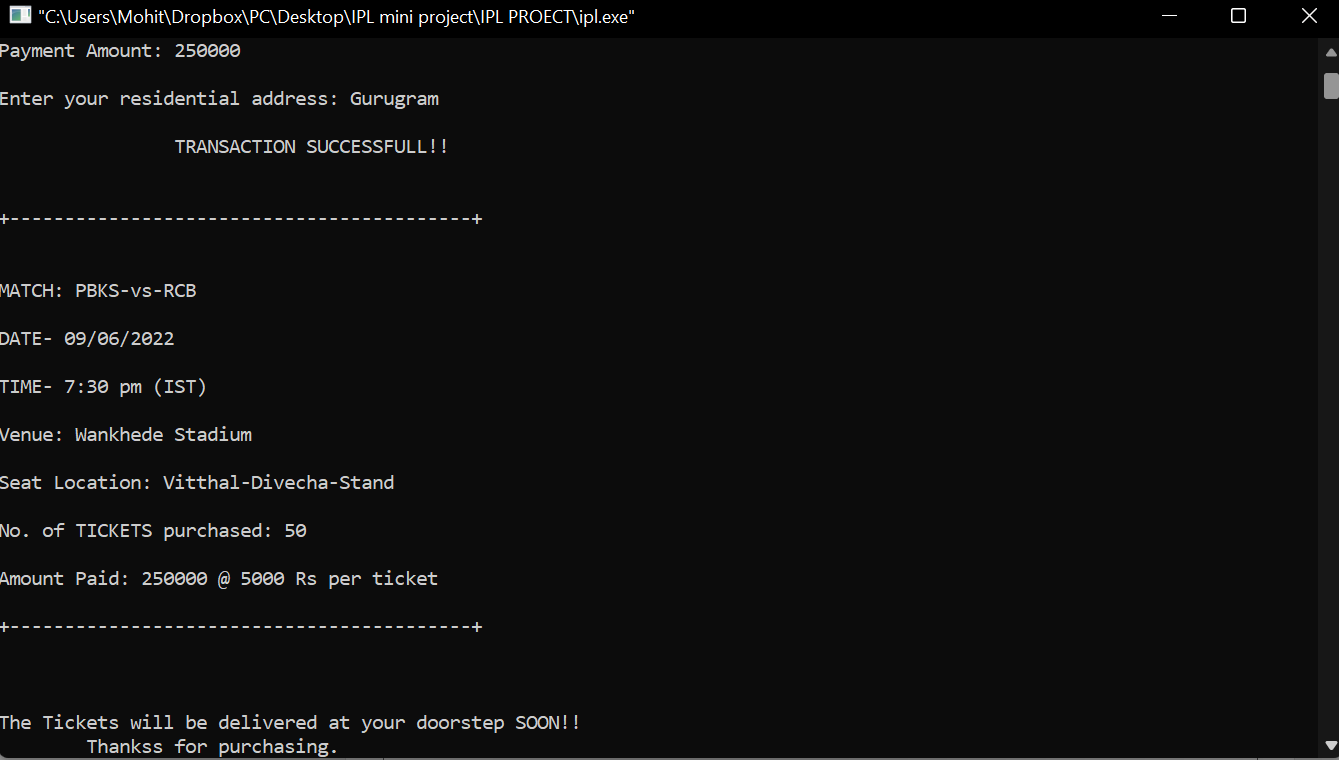
}

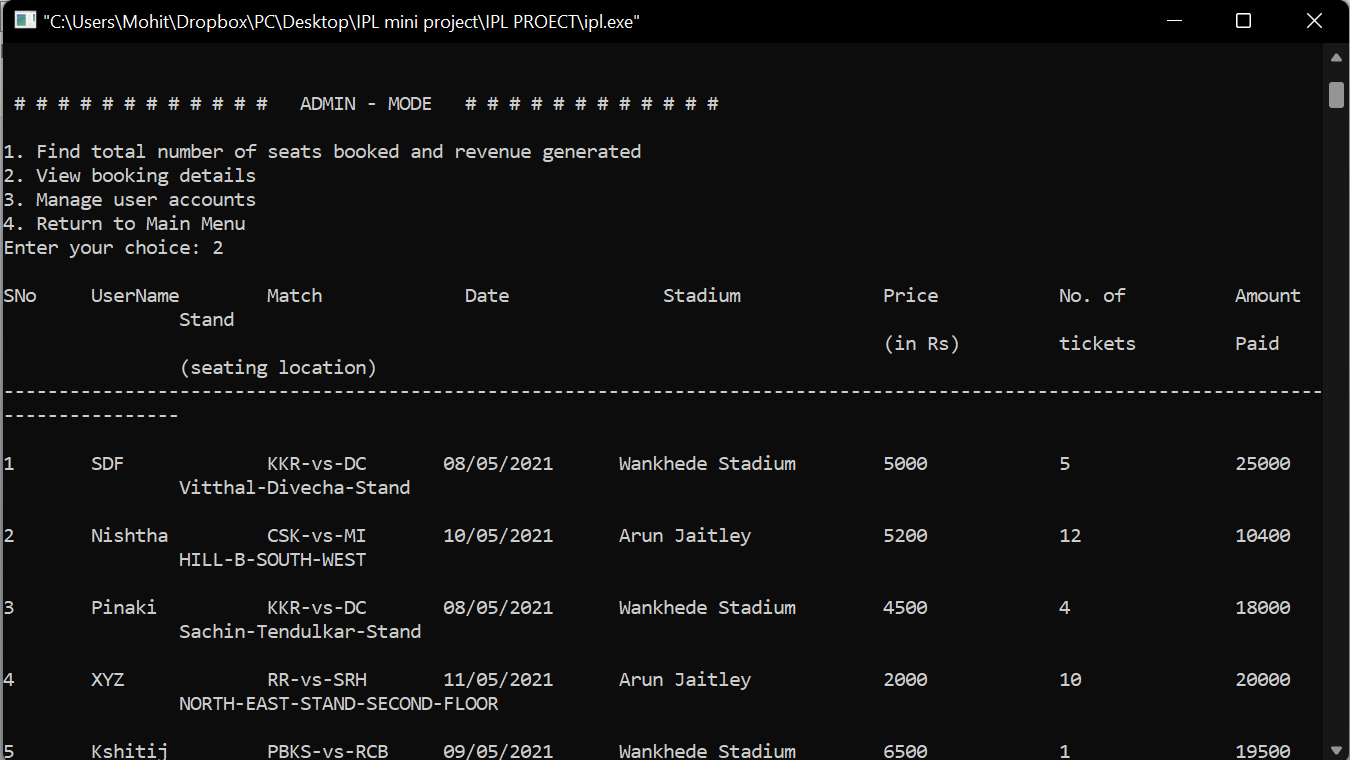
**OUTPUT**

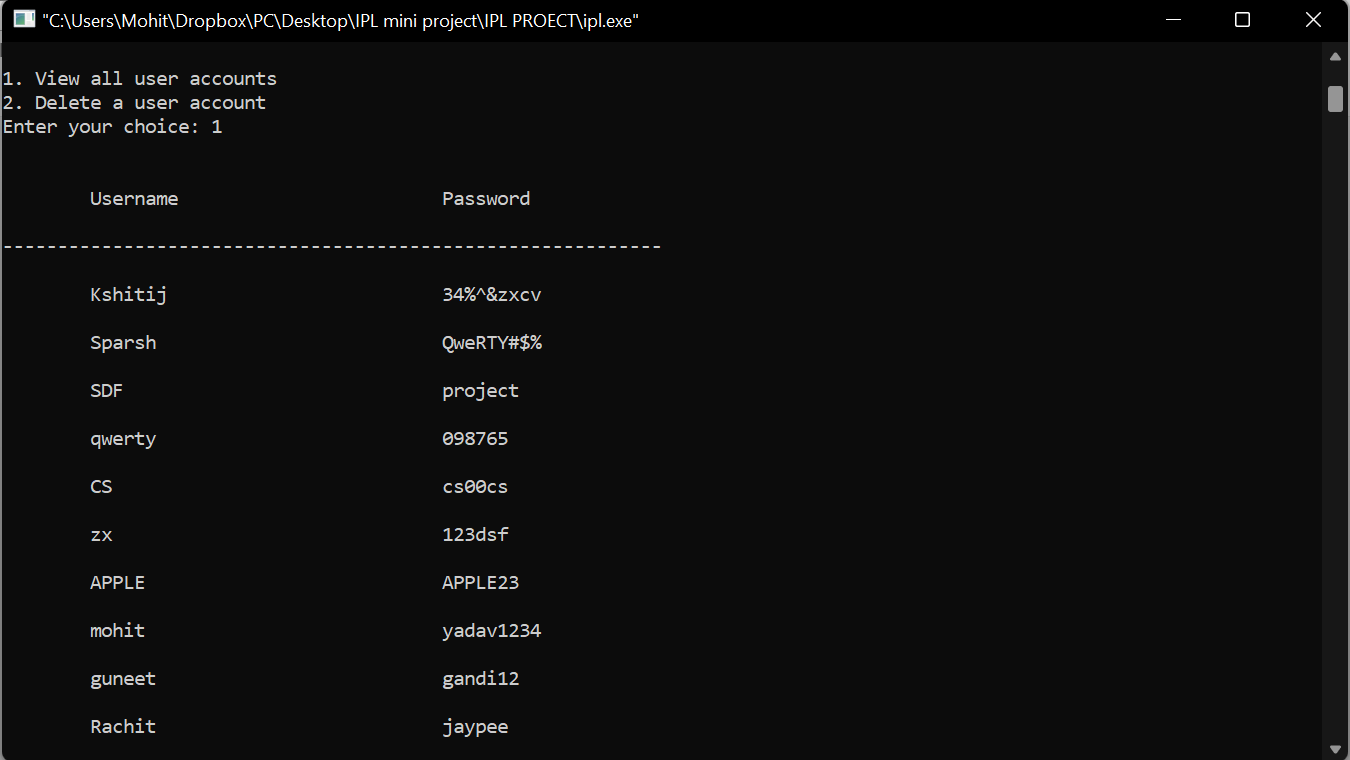












**CONCLUSION**

The purpose of the project is to build an application program to reduce the manual work for managing the ticket-booking, Seats, Payment and User. It forms a database of the information and keeps the track of all the details about the User, ticket-booking and schedule of the matches. The project will allow the user to book tickets for upcoming IPL matches. This makes use of the concepts of Object Oriented Programming (OOPS),vectors and file handling.

**REFERENCES**

1. <https://stackoverflow.com/>
2. <https://www.geeksforgeeks.org/>
3. https://www.cplusplus.com/
4. C++ The complete reference (By Herbert Schildt)