

graphicx

Ishaan Dasgupta

0801CS211046

PP-final project

Casino Game

1 Aim

The Aim of this mini project program is to create a login system and a leader-board in a casino game.

2 Working

The use can either register a new profile or sign in to an already created profile, the user will then be able to play the game and his total coin count will be stored, through this he can also get his ranking the the leader-board or just view the top 3 players.

3 Following are the functions used in this project

3.1 operationSelector

This function makes a interface in which user the chose from 4 options i.e Register , Sign-In , view leader-board and view rules.

3.2 registerUser

This function is used to register user by using their email ID and password.

3.3 signInUser

This function is used to verify is the user is present in the system and if true it validated the email ID and password and then logs them into the system.

3.4 printHighScore

This function is used to get the top 3 player's coins from the system and them log them into the terminal.

3.5 gameStart

This function is used for the actual game logic. Betting, adding coins and viewing ones ranking is all done in this function.

3.6 getYourRanking

The function contains the logic to get ones ranking among all the registred users.

3.7 getRules

This function is used to display the rules of the game.

3.8 emailValidator

This function is used to validated email address entered by the user. It reatuns true if there are no issues with the email otherwise it return false.

3.9 addCoins

This function adds the entered ammount into the users account.

4 Code In C++ language:

```
[style=chstyle,language=C++]
#include <bits/stdc++.h>
using namespace std;

map<string, string> getPassword;
map<string, int> getCoins;
bool gameOn = true;
string currentEmail;

void operationSelector();

void addCoins(string email)
{
    cout << "Enter the ammount you wish to add\n";
    int ammount;
    cin >> ammount;
    getCoins[currentEmail] += ammount;
}

bool emailValidator(string email)
{
    string allLowercase = email;
    for (int i = 0; i < email.size(); i++)
    {
```

```
        if (allLowercase[i] >= 65 and allLowercase[i] <= 90)
        {
            allLowercase[i] += 32;
        }
    }
    if (allLowercase != email)
    {
        return false;
    }

    bool atPresent = false;
    for (int i = 0; i < email.size(); i++)
    {
        if (email[i] == '@')
        {
            atPresent = true;
        }
    }

    if (!atPresent)
    {
        return false;
    }

    if (email.size() < 5)
    {
        return false;
    }

    int n = email.size();
    bool good = false;
    if (email[n - 3] == '.' and email[n - 2] == 'i' and email[n - 1] == 'n')
    {
        good = true;
    }

    if (email[n - 4] == '.' and email[n - 3] == 'c' and email[n - 2] == 'o' and email[n - 1] == 'm')
    {
        good = true;
    }

    return good;
}

void getRules()
{
    cout << "-----\n";
    cout << "RULES\n";
}
```

```
    cout << "This is a game in which you are supposed to guess the color of the casion chip and be
    cout << "To play this game you must login with a already registered email ID or register a new
    cout << "This game also have a leaderboard in which you can see the coins of top players\n";
    cout << "-----\n";

    return;
}

int getYourRanking(string email)
{
    int yourScore = getCoins[email];
    int ranking = 1;
    for (auto i : getCoins)
    {
        if (yourScore < i.second)
        {
            ranking++;
        }
    }

    return ranking;
}

void gameStart()
{
    bool stillPlaying = true;
    if (stillPlaying)
    {
        cout << "Guess the color of the casino chip (Red or Black)\n";
        int randomNum = rand();
        randomNum %= 2;
        string userString;
        cin >> userString;
        int userInput = 0;
        if (userString == "Red")
        {
            userInput = 1;
        }

        cout << "Enter the ammount you want to bet\n";
        int ammount;
        cin >> ammount;

        if (getCoins[currentEmail] == 0)
        {
            cout << "You dont have enough balance to bet\n";
        }
        else
```

```

{
    if (ammount >= getCoins[currentEmail])
    {
        ammount = getCoins[currentEmail];
        cout << "Betting all coins\n";
    }

    if (randomNum == userInput)
    {
        cout << "You gussed correctly\n";
        cout << "You won " << ammount << " coins\n";
        getCoins[currentEmail] += ammount;
        cout << "Your current coins are : " << getCoins[currentEmail] << "\n";
    }
    else
    {
        cout << "Your Guess was wrong\n";
        cout << "You lost " << ammount << " coins\n";
        getCoins[currentEmail] -= ammount;
    }
}

cout << "-----\n";
cout << "1 to play again\n";
cout << "2 to add balance\n";
cout << "3 to signout and quit\n";
cout << "4 to get your ranking\n";
cout << "-----\n";

int operation;
cin >> operation;
if (operation == 2)
{
    addCoins(currentEmail);
}
if (operation == 3)
{
    currentEmail = "";
    operationSelector();
}

if (operation == 4)
{
    cout << "Your current rank is : " << getYourRanking(currentEmail) << " with total coin
}

if (stillPlaying)
{
    gameStart();
}

```

```
        }
    }
    return;
}

void printHighScore()
{
    vector<int> coins;
    for (auto i : getCoins)
    {
        coins.push_back(i.second);
    }

    sort(coins.begin(), coins.end());
    int first = 0, second = 0, third = 0;

    for (int i = coins.size() - 1; i >= max(0, (int)coins.size() - 3); i--)
    {
        if (i == coins.size() - 1)
        {
            first = coins[i];
        }
        if (i == coins.size() - 2)
        {
            second = coins[i];
        }
        if (i == coins.size() - 3)
        {
            third = coins[i];
        }
    }

    cout << "1st place is at " << first << "\n";
    cout << "2nd place is at " << second << "\n";
    cout << "3rd place is at " << third << "\n";
    return;
}

void registerUser()
{
    bool registered = false;
    while (!registered)
    {
        cout << "Enter email address or input X to exit\n";
        string enteredEmail;
        cin >> enteredEmail;
        if (enteredEmail == "X")
        {

```

```

        return;
    }
    while (!emailValidator(enteredEmail))
    {
        cout << "The previously entered email is invalid please enter a correct email or input X\n";
        string enteredEmail;
        cin >> enteredEmail;
        cout << enteredEmail << emailValidator(enteredEmail) << "\n";
        if (enteredEmail == "X")
        {
            return;
        }
    }
    cout << "Enter password\n";
    string enteredPassword;
    cin >> enteredPassword;
    if (getPassword[enteredEmail] == "")
    {
        cout << "Registration Success\n";
        getPassword[enteredEmail] = enteredPassword;
        getCoins[enteredEmail] = 0;
        registered = true;
        currentEmail = enteredEmail;
        break;
    }
    cout << "This email is already under use\n";
    cout << "Please enter another username\n";
}
gameStart();
return;
}

void signInUser()
{
    bool loggedIn = false;
    while (!loggedIn)
    {
        cout << "Enter email address or input X to exit\n";
        string enteredEmail;
        cin >> enteredEmail;
        if (enteredEmail == "X")
        {
            return;
        }
    }
    while (!emailValidator(enteredEmail))
    {
        cout << "The previously entered email is invalid please enter a correct email or input X\n";
        string enteredEmail;

```

```
        cin >> enteredEmail;
        if (enteredEmail == "X")
        {
            return;
        }
    }

    cout << "Enter password\n";
    string enteredPassword;
    cin >> enteredPassword;

    if (getPassword[enteredEmail] == enteredPassword)
    {
        cout << "Login Success\n";
        currentEmail = enteredEmail;
        loggedIn = true;
        break;
    }
    cout << "The entered email and password does not match\n";
}

gameStart();
return;
}

void operationSelector()
{
    cout << "-----\n";
    cout << "1 to Register\n";
    cout << "2 to Sign in\n";
    cout << "3 to view Leaderboard\n";
    cout << "4 to view Rules\n";
    cout << "-----\n";

    int operation;
    cin >> operation;
    switch (operation)
    {
        case 1:
            registerUser();
            break;

        case 2:
            signInUser();
            break;

        case 3:
            printHighScore();
```



```
        break;

    case 4:
        getRules();
        break;
    }

    if (gameOn)
    {
        operationSelector();
    }
};

int main()
{
    {
        operationSelector();
    }
    return 0;
}
```

5 Execution and Compilation

The top screenshot shows the initial execution of the 'Competitive Coding' project. The terminal output is as follows:

```
PS C:\Coding\Competitive Coding> cd "C:\Coding\Competitive Coding\" ; if ($?) { g++ project.cpp -o project } ; if ($?) { .\project }
-----
1 to Register
2 to Sign in
3 to view Leaderboard
4 to view Rules
-----
1
Enter email address or input X to exit
ishaan
The previously entered email is invalid please enter a correct email or input X to exit
ishaan@gmail.com
Enter password
123
Registration Success
Guess the color of the casino chip (Red or Black)
Red
Enter the amount you want to bet
0
You don't have enough balance to bet
-----
1 to play again
2 to add balance
3 to signout and quit
4 to get your ranking
-----
4
Your current rank is : 1 with total coins : 1500
Guess the color of the casino chip (Red or Black)
```

The bottom screenshot shows the continuation of the game. The terminal output is as follows:

```
4 to get your ranking
-----
4
Your current rank is : 1 with total coins : 1500
Guess the color of the casino chip (Red or Black)
Red
Enter the amount you want to bet
50
Your guess was wrong
You lost 50 coins
-----
1 to play again
2 to add balance
3 to signout and quit
4 to get your ranking
-----
4
-----
3
-----
1st place is at 1450
2nd place is at 0
3rd place is at 0
-----
1 to Register
2 to Sign in
3 to view Leaderboard
4 to view Rules
-----
3
-----
1 to Register
2 to Sign in
3 to view Leaderboard
4 to view Rules
-----
4
-----
RULES
This is a game in which you are supposed to guess the color of the casino chip and bet a certain amount.
To play this game you must login with a already registered email ID or register a new email ID
This game also have a leaderboard in which you can see the coins of top players
-----
1 to Register
2 to Sign in
3 to view Leaderboard
4 to view Rules
```

6 Code In Python language:

```
[style=chstyle,language=Python]
```

```
import random
```

```
getPassword = {};
```

```
getCoins = {};  
gameOn = True;  
currentEmail = "";
```

```
def addCoins(email):
```

```
    global currentEmail;  
    global getPassword;  
    global getCoins;  
    global gameOn;  
  
    print("Enter the ammount you wish to add");  
    ammount = int(input());  
    getCoins[currentEmail] += ammount;
```

```
def emailValidator(email):
```

```
    global currentEmail;  
    global getPassword;  
    global getCoins;  
    global gameOn;  
    if (email.lower() != email):  
        return False;
```

```
    atPresent = False;  
    for i in email:
```

```
        if (i == '@'):  
            atPresent = True;
```

```
    if (atPresent == False):  
        return False;
```

```
    if (len(email) < 5):  
        return False;
```

```
    n = len(email);  
    good = False;  
    if (email[n - 3] == '.' and email[n - 2] == 'i' and email[n - 1] == 'n'):  
        good = True;
```

```
    if (email[n - 4] == '.' and email[n - 3] == 'c' and email[n - 2] == 'o' and email[n - 1] == 'm'):  
        good = True;
```

```
    return good;
```

```
def getRules():
    print("-----");
    print("RULES");
    print("This is a game in which you are supposed to guess the color of the casion chip and bet");
    print("To play this game you must login with a already registered email ID or register a new email");
    print("This game also have a leaderboard in which you can see the coins of top players");
    print("-----");
    return;

def getYourRanking(email):
    global currentEmail;
    global getPassword;
    global getCoins;
    global gameOn;
    yourScore = getCoins[email];
    ranking = 1;

    keys = getCoins.keys();

    for key in keys:
        if (yourScore < getCoins[key]):
            ranking += 1;

    return ranking;

def gameStart():
    global currentEmail;
    global getPassword;
    global getCoins;
    global gameOn;
    stillPlaying = True;
    if (stillPlaying):
        print("Guess the color of the casino chip (Red or Black)");
        colourList = ["Red", "Black"]
        randomColor = random.choice(colourList);
        userString = input();

        print("Enter the ammount you want to bet");
        ammount = int(input());

        if (getCoins[currentEmail] == 0):
            print("You dont have enough balance to bet");
        else:
            if (ammount >= getCoins[currentEmail]):
                ammount = getCoins[currentEmail];
                print("Betting all coins");

            if (userString == randomColor):
```

```

        print("You gussed correctly");
        print("You won " , ammount , " coins");
        getCoins[currentEmail] += ammount;
        print("Your current coins are : " , getCoins[currentEmail]);
    else:
        print("Your Guess was wrong");
        print("You lost " , ammount , " coins");
        getCoins[currentEmail] -= ammount;

    print("-----");
    print("1 to play again");
    print("2 to add balance");
    print("3 to signout and quit");
    print("4 to get your ranking");
    print("-----");

    operation = int(input());
    if (operation == 2):
        addCoins(currentEmail);
    if (operation == 3):
        currentEmail = "";
        operationSelector();

    if operation == 4:
        if (len(currentEmail) == 0):
            print("Please sign in to get your leaderboard ranking");
        else:
            print("Youre current rank is : " , getYourRanking(currentEmail) , " with total coins : " , getCoins[currentEmail]);

    gameStart();

    if (stillPlaying):
        gameStart();

    return;

def printHighScore():
    global currentEmail;
    global getPassword;
    global getCoins;
    global gameOn;
    coins = [];

    keys = getCoins.keys();

    for key in keys:
        coins.append(getCoins[key]);

```

```
coins.sort();
first = 0;
second = 0;
third = 0;

i = len(coins) - 1;
while i>=0:
    if (i == len(coins) - 1):
        first = coins[i];

    if (i == len(coins) - 2):
        second = coins[i];

    if (i == len(coins) - 3):
        third = coins[i];
        break;

    i -= 1;

print("1st place is at " , first );
print("2nd place is at " , second );
print("3rd place is at " , third );
return;

def registerUser():
    global currentEmail;
    global getPassword;
    global getCoins;
    global gameOn;
    registered = False;
    while (registered == False):
        print("Enter email address or input X to exit");
        enteredEmail = input();
        if (enteredEmail == "X"):
            return;
        while (emailValidator(enteredEmail) == False):
            print("The previously entered email is invalid please enter a correct email or input X ");
            enteredEmail = input();
            if (enteredEmail == "X"):
                return;

        print("Enter password");
        enteredPassword = input();
        if (getPassword.get(enteredEmail) == None):
            print("Registration Success");
```

```
        getPassword[enteredEmail] = enteredPassword;
        getCoins[enteredEmail] = 0;
        currentEmail = enteredEmail;
        break;

    print("This email is already under use");
    print("Please enter another username");

gameStart();
return;

def signInUser():
    global currentEmail;
    global getPassword;
    global getCoins;
    global gameOn;
    loggedIn = False;
    while (loggedIn == False):
        print("Enter email address or input X to exit");
        enteredEmail = input();
        if (enteredEmail == "X"):
            return;

        while (emailValidator(enteredEmail) == False):
            print("The previously entered email is invalid please enter a correct email or input X");
            enteredEmail = input();
            if (enteredEmail == "X"):
                return;

        print("Enter password");
        enteredPassword = input();

        if (getPassword[enteredEmail] == enteredPassword):
            print("Login Success");
            currentEmail = enteredEmail;
            loggedIn = True;
            break;

        print("The entered email and password does not match");

gameStart();
return;

def operationSelector():
    global currentEmail;
    global getPassword;
    global getCoins;
```

```
global gameOn;
print("-----");
print("1 to Register");
print("2 to Sign in");
print("3 to view Leaderboard");
print("4 to view Rules");
print("5 to get your ranking");
print("-----");

operation = int(input());

if operation == 1:
    registerUser();

if operation == 2:
    signInUser();

if operation == 3:
    printHighScore();

if operation == 4:
    getRules();

if (gameOn):
    operationSelector();

if (gameOn):
    operationSelector();
```


9 Debugging

```
File Edit Selection View Go Run Terminal Help
project.cpp - Competitive Coding - Visual Studio Code

EXPLORER
COMPETITIVE CODING
> srccode
  a.exe
  a.out
  a.output
  book.class
  exoutput.txt
  gmon.out
  input.txt
  kickstart.cpp
  kickstart.exe
  leetcode.cpp
  m.txt
  Main.class
  main.cpp
  main.exe
  Main.java
  output.txt
  outputInCpp.txt
  project.cpp
  project.exe
  project.py
  publication.class
  tempCodeRun...
  tempCodeRun...

PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

project.cpp: In function 'int main()':
project.cpp:327:20: error: expected ';' before 'if'
327 |     cout << "hello" if (gameOn)
    |                    ^
    |
    |
PS C:\Coding\Competitive Coding> g++ -g project.cpp -o a
project.cpp: In function 'int main()':
project.cpp:327:20: error: expected ';' before 'if'
327 |     cout << "hello" if (gameOn)
    |                    ^
    |
    |
PS C:\Coding\Competitive Coding> g++ -g project.cpp -o a
C:\MinGW\bin\gdb.exe: warning: couldn't determine a path for the index cache directory.
GNU gdb (GDB) 8.3.1
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
--Type <RET> for more, q to quit, c to continue without paging--

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a...
(gdb) b 25
Breakpoint 1 at 0x402700: file ../../src/mingw-w64-crt/crt/gcmain.c, line 25.
(gdb) run
Starting program: C:\Coding\Competitive Coding\la.exe
[New Thread 24868.0x0054]
warning: Can not parse XML library list; XML support was disabled at compile time
[New Thread 24868.0x7cb4]
-----
1 to Register
2 to Sign in

Ln 27, Col 10 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 Prettier 08:46 23-11-2022
```