















#include <unistd.h>

#include <bits/stdc++.h>

#include <string>

#include <process.h>

using namespace std;

char ch = 'y';

class personaldoc;

class officialdoc;

class signup

{

protected:

    string username;

    string email;

    string phoneNumber;

    string address;

    string password;

public:

    void signupdetails()

    {

        system("cls");

        cout << "\t\t SIGN UP WINDOW " << endl

             << endl;

        cout << "Enter Username: ";

        getline(cin >> ws, username);

        cout << "Enter Email ID: ";

        getline(cin >> ws, email);

        cout << "Enter Phone Number: ";

        getline(cin >> ws, phoneNumber);

        cout << "Enter Address: ";

        getline(cin >> ws, address);

        getPassword();

    }

protected:

    void getPassword()

    {

        cout << "Enter Password: ";

        getline(cin >> ws, password);

        while (!isStrongPassword(password))

        {

            cout << "Password is not strong enough. Please try again." << endl;

            cout << "Password must have at least 8 characters, including uppercase, lowercase, digits, and special characters." << endl;

            cout << "Enter Password: ";

            getline(cin >> ws, password);

        }

        storessignup();

        system("cls");

        cout << "Signed up successfully!" << endl

             << endl;

        displaysignup();

        sleep(3);

        system("cls");

    }

    void storessignup()

    {

        ofstream fout;

        fout.open("username.txt", ios::binary | ios::out);

        if (!fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        fout << username;

        fout.close();

        fout.open("password.txt", ios::binary | ios::out);

        if (!fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        fout << password;

        fout.close();

        fout.open("SignupDetails.txt", ios::binary | ios::out);

        if (!fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        fout << "\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        fout << "\t\*            Your SignUp Details                 \*" << endl;

        fout << "\t\* User Name = " << left << setw(35) << username << "\*" << endl;

        fout << "\t\* Password = " << left << setw(36) << password << "\*" << endl;

        fout << "\t\* Email id = " << left << setw(36) << email << "\*" << endl;

        fout << "\t\* Phone Number = " << left << setw(32) << phoneNumber << "\*" << endl;

        fout << "\t\* Address= " << left << setw(38) << address << "\*" << endl;

        fout << "\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        fout.close();

    }

    void displaysignup()

    {

        ifstream fin("SignupDetails.txt");

        char ch;

        while (fin.get(ch))

        {

            cout << ch;

        }

        fin.close();

    }

    bool isStrongPassword(string &password)

    {

        if (password.length() < 8)

            return false;

        bool hasUpper = false, hasLower = false, hasDigit = false, hasSpecial = false;

        for (int i = 0; i < password.length(); ++i)

        {

            char c = password[i];

            if (isupper(c))

                hasUpper = true;

            if (islower(c))

                hasLower = true;

            if (isdigit(c))

                hasDigit = true;

            if (!isalnum(c))

                hasSpecial = true;

        }

        return hasUpper && hasLower && hasDigit && hasSpecial;

    }

};

class Login : public signup

{

protected:

    string user;

    string pass;

    string cap;

    string captcha;

    string temp;

public:

    string generateCaptcha()

    {

        captcha = "";

        srand(time(0));

        captcha += to\_string(rand() % 10);

        for (int i = 0; i < 3; ++i)

        {

            captcha += 'A' + rand() % 26;

        }

        captcha += to\_string(rand() % 10);

        return captcha;

    }

    void logindetails()

    {

        cout << "\t\tLOGIN WINDOW\n\n";

        cout << "enter username = ";

        getline(cin >> ws, user);

        cout << "enter password = ";

        getline(cin >> ws, pass);

        captcha = generateCaptcha();

        cout << "captcha = " << captcha << endl;

        cout << "enter above captcha = ";

        getline(cin >> ws, cap);

        while (!authenticate(user, pass, cap, captcha))

        {

            cout << "incorrect login details...enter correctly" << endl;

            cout << "enter username = ";

            getline(cin >> ws, user);

            cout << "enter password = ";

            getline(cin >> ws, pass);

            captcha = generateCaptcha();

            cout << "captcha = " << captcha << endl;

            cout << "enter above captcha = ";

            getline(cin >> ws, cap);

        }

        system("cls");

        cout << "LOGGEDIN  SUCCESSFULLY..." << endl

             << endl

             << endl;

    }

    bool authenticate(string user, string pass, string cap, string temp)

    {

        bool flag1 = false;

        bool flag2 = false;

        bool flag3 = false;

        ifstream fin;

        string storedUser, storedPass;

        fin.open("username.txt");

        if (fin.is\_open())

        {

            getline(fin, storedUser);

            fin.close();

        }

        else

        {

            cout << "Error opening username file!" << endl;

            return false;

        }

        fin.open("password.txt");

        if (fin.is\_open())

        {

            getline(fin, storedPass);

            fin.close();

        }

        else

        {

            cout << "Error opening password file!" << endl;

            return false;

        }

        if (storedUser == user)

        {

            flag1 = true;

        }

        if (storedPass == pass)

        {

            flag2 = true;

        }

        if (temp == cap)

        {

            flag3 = true;

        }

        return (flag1 && flag2 && flag3);

    }

};

class twelve

{

public:

    void twelveadd()

    {

        string name;

        string fathername;

        string mothername;

        string roll;

        string school;

        cout << "Enter name = ";

        getline(cin >> ws, name);

        cout << "Enter Father Name  = ";

        getline(cin >> ws, fathername);

        cout << "Enter Mother Name= ";

        getline(cin >> ws, mothername);

        cout << "Enter ROLL NUMBER  = ";

        getline(cin >> ws, roll);

        cout << "Enter SCHOOL NAME  = ";

        getline(cin >> ws, school);

        ifstream fin("twelveadd.txt");

        ofstream fout("twelve\_temp.txt");

        if (!fin.is\_open() || !fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        string line;

        while (getline(fin, line))

        {

            size\_t pos = line.find("#");

            if (pos != string::npos)

            {

                fout << "\* NAME:" << left << setw(12) << name << "                                                 \*" << endl;

                fout << "\* ROLL NO." << left << setw(12) << roll << "                                              \*" << endl;

                fout << "\* MOTHERNAME:" << left << setw(12) << mothername << "                                           \*" << endl;

                fout << "\* FATHERNAME:" << left << setw(12) << fathername << "                                           \*" << endl;

                fout << "\* SCHOOL NAME:" << left << setw(12) << school << "                                          \*" << endl;

            }

            else

            {

                fout << line << endl;

            }

        }

        fin.close();

        fout.close();

    }

    void showtwelve()

    {

        ifstream fin("twelve\_temp.txt");

        char ch;

        if (!fin.is\_open() )

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        system("type twelvehadd.txt");

        while (fin.get(ch))

        {

            cout << ch;

        }

        fin.close();

    }

    void deletetwelve()

    {

        remove("twelve\_temp.txt");

        cout << "\tDATA DELETED FROM TABLE" << endl;

        sleep(3);

        system("cls");

    }

};

class pan

{

public:

    void panadd()

    {

        string pan;

        string name;

        string fathername;

        string dob;

        cout << "Enter name = ";

        getline(cin >> ws, name);

        cout << "Enter Father Name  = ";

        getline(cin >> ws, fathername);

        cout << "Enter date of brith = ";

        getline(cin >> ws, dob);

        cout << "Enter Permanent Account Number = ";

        getline(cin >> ws, pan);

        ifstream fin("panadd.txt");    // take content from file or read from file

        ofstream fout("pan\_temp.txt"); // write the content to file

        if (!fin.is\_open() || !fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        string line;

        while (getline(fin, line))

        {

            size\_t pos = line.find("#");

            if (pos != string::npos)

            {

                fout << "\*                                   " << left << setw(12) << pan << "                    \*" << endl;

                fout << "\* Name : " << left << setw(12) << name << "                                               \*" << endl;

                fout << "\* Father Name : " << left << setw(12) << fathername << "                                        \*" << endl;

                fout << "\* D.O.B : " << left << setw(12) << dob << "                                              \*" << endl;

            }

            else

            {

                fout << line << endl;

            }

        }

        fin.close();

        fout.close();

    }

    void showpan()

    {

        ifstream fin("pan\_temp.txt");

        char ch;

        if (!fin.is\_open() )

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        system("type panhadd.txt");

        while (fin.get(ch))

        {

            cout << ch;

        }

        fin.close();

    }

    void deletepan()

    {

        remove("pan\_temp.txt");

        cout << "\tDATA DELETED FROM TABLE" << endl;

        sleep(3);

        system("cls");

    }

};

class drive

{

public:

    void dladd()

    {

        string name;

        string dob;

        string dateofissue;

        string refname;

        string validity;

        cout << "Enter Date of Brith  = ";

        getline(cin >> ws, dob);

        cout << "Enter Date of First issue of DL = ";

        getline(cin >> ws, dateofissue);

        cout << "Enter Validity Date = ";

        getline(cin >> ws, validity);

        ifstream fin("dladd.txt");    // take content from file or read from file

        ofstream fout("dl\_temp.txt"); // write the content to file

        if (!fin.is\_open() || !fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        string line;

        while (getline(fin, line))

        {

            size\_t pos = line.find("#");

            if (pos != string::npos)

            {

                fout << "\*  Date of Issue: " << left << setw(12) << dateofissue << "             Validity : " << left << setw(10) << validity << "    \*" << endl;

                fout << "\*  D.O.B: " << left << setw(12) << dob << "                     Blood group:AB+          \*" << endl;

            }

            else

            {

                fout << line << endl;

            }

        }

        fin.close();

        fout.close();

    }

    void showdl()

    {

        ifstream fin("dl\_temp.txt");

        char ch;

        if (!fin.is\_open() )

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        while (fin.get(ch))

        {

            cout << ch;

        }

        system("type dlhadd.txt");

        fin.close();

    }

    void deletedl()

    {

        remove("dl\_temp.txt");

        cout << "\tDATA DELETED FROM TABLE" << endl;

        sleep(3);

        system("cls");

    }

};

class addhar

{

public:

    void aadharadd()

    {

        string name;

        string dob;

        string gen;

        string no;

        cout << "Enter name = ";

        getline(cin >> ws, name);

        cout << "Enter dob = ";

        getline(cin >> ws, dob);

        cout << "Enter gender = ";

        getline(cin >> ws, gen);

        cout << "Enter aadhar no = ";

        getline(cin >> ws, no);

        ifstream fin("aadharadd.txt");

        ofstream fout("aadharcard\_temp.txt");

        if (!fin.is\_open() || !fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        string line;

        while (getline(fin, line))

        {

            size\_t pos = line.find("#");

            if (pos != string::npos)

            {

                fout << "\* Name:" << left << setw(12) << name << "                                     \*" << endl;

                fout << "\* D.O.B:" << left << setw(12) << dob << "                                    \*" << endl;

                fout << "\* Gender:" << left << setw(12) << gen << "                                   \*" << endl;

                fout << "\* AADHAR NO." << left << setw(12) << no << "                                \*" << endl;

            }

            else

            {

                fout << line << endl;

            }

        }

        fin.close();

        fout.close();

    }

    void showaadhar()

    {

        ifstream fin("aadharcard\_temp.txt");

        char ch;

        if (!fin.is\_open() )

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        system("type ind.txt");

        while (fin.get(ch))

        {

            cout << ch;

        }

        system("type last.txt");

        fin.close();

    }

    void deleteaadhar()

    {

        remove("aadharcard\_temp.txt");

        cout << "\tDATA DELETED FROM TABLE" << endl;

        sleep(3);

        system("cls");

    }

};

class propertypaper

{

public:

    void propertyadd()

    {

        string scheme;

        string plot;

        string code;

        cout << "Enter Certificate Number(10 digit) = ";

        getline(cin >> ws, code);

        cout << "Enter Plot Number(3 digit) = ";

        getline(cin >> ws, plot);

        cout << "Enter Location  = ";

        getline(cin >> ws, scheme);

        ifstream fin("propertyadd.txt");

        ofstream fout("property\_temp.txt");

        if (!fin.is\_open() || !fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        string line;

        while (getline(fin, line))

        {

            size\_t pos = line.find("#");

            if (pos != string::npos)

            {

                fout << "\*                                                                                     Certificate No." << left << setw(10) << code << " \*" << endl;

                fout << "\*  PLot no. :" << left << setw(10) << plot << "                                                                                         \*" << endl;

                fout << "\*  Location :" << left << setw(20) << scheme << "                                                                               \*" << endl;

            }

            else

            {

                fout << line << endl;

            }

        }

        fin.close();

        fout.close();

    }

    void showproperty()

    {

        ifstream fin("property\_temp.txt");

        char ch;

        if (!fin.is\_open() )

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        system("type propertyhadd.txt");

        while (fin.get(ch))

        {

            cout << ch;

        }

        fin.close();

    }

    void deleteproperty()

    {

        remove("property\_temp.txt");

        cout << "\tDATA DELETED FROM TABLE" << endl;

        sleep(3);

        system("cls");

    }

};

class passport

{

public:

    void passportadd()

    {

        string number;

        cout << "Enter passport number (8 digits) = ";

        getline(cin >> ws, number);

        ifstream fin("passportadd.txt");

        ofstream fout("passport\_temp.txt");

        if (!fin.is\_open() || !fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        string line;

        while (getline(fin, line))

        {

            size\_t pos = line.find("#");

            if (pos != string::npos)

            {

                fout << "\t\* " << left << setw(10) << number << "                                            \*" << endl;

            }

            else

            {

                fout << line << endl;

            }

        }

        fin.close();

        fout.close();

    }

    void showpassport()

    {

        ifstream fin("passport\_temp.txt");

        char ch;

        if (!fin.is\_open() )

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        system("type passporthadd.txt");

        while (fin.get(ch))

        {

            cout << ch;

        }

        fin.close();

    }

    void deletepassport()

    {

        remove("passport\_temp.txt");

        cout << "\tDATA DELETED FROM TABLE" << endl;

        sleep(3);

        system("cls");

    }

};

class bank

{

public:

    void bankadd()

    {

        string cif;

        string name;

        string phone;

        string nominee;

        string address;

        string accno;

        cout << "Enter Account Holder name = ";

        getline(cin >> ws, name);

        cout << "Enter CIF Number = ";

        getline(cin >> ws, cif);

        cout << "Enter Account Number = ";

        getline(cin >> ws, accno);

        cout << "Enter  Phone Number = ";

        getline(cin >> ws, phone);

        cout << "Enter  Nominee Name = ";

        getline(cin >> ws, nominee);

        cout << "Enter  Address = ";

        getline(cin >> ws, address);

        ifstream fin("bankadd.txt");

        ofstream fout("bank\_temp.txt");

        if (!fin.is\_open() || !fout.is\_open())

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        string line;

        while (getline(fin, line))

        {

            size\_t pos = line.find("#");

            if (pos != string::npos)

            {

                fout << "\* CIF no : " << left << setw(10) << cif << "                                                        \*" << endl;

                fout

                    << "\* Account no : " << left << setw(10) << accno << "                                                    \*" << endl;

                fout

                    << "\* Account Name : " << left << setw(10) << name << "                                                  \*" << endl;

                fout

                    << "\* Phone Number : " << left << setw(10) << phone << "                                                  \*" << endl;

                fout

                    << "\* Nominee Name : " << left << setw(10) << nominee << "                                                  \*" << endl;

                fout

                    << "\* Address : " << left << setw(10) << address << "                                                       \*" << endl;

            }

            else

            {

                fout << line << endl;

            }

        }

        fin.close();

        fout.close();

    }

    void showbank()

    {

        ifstream fin("bank\_temp.txt");

        char ch;

        if (!fin.is\_open() )

        {

            cerr << "Error opening the file!" << endl;

            return;

        }

        system("type bankhadd.txt");

        while (fin.get(ch))

        {

            cout << ch;

        }

        fin.close();

    }

    void deletebank()

    {

        remove("bank\_temp.txt");

        cout << "\tDATA DELETED FROM TABLE" << endl;

        sleep(3);

        system("cls");

    }

};

class viewofficial

{

public:

    propertypaper p;

    bank b;

    int chch;

    passport pass;

    void get()

    {

        cout << "ENTER CHOICE FOR VIEWING DOCUMENT \n1.Property Paper\n2.Bank Details \n3.Passport\n\n";

        cin >> chch;

        switch (chch)

        {

        case 1:

            system("cls");

            p.showproperty();

            break;

        case 2:

            system("cls");

            b.showbank();

            break;

        case 3:

            system("cls");

            pass.showpassport();

            break;

        }

    }

};

class view

{

public:

    drive d;

    int ch;

    addhar a;

    pan p;

    twelve t;

    void get()

    {

        cout << "ENTER CHOICE FOR VIEWING DOCUMENT \n1.Aadhar Card\n2.12th Marksheet\n3.PAN card\n4.Driving LICENCE\n\n";

        cin >> ch;

        switch (ch)

        {

        case 1:

            system("cls");

            a.showaadhar();

            break;

        case 2:

            system("cls");

            t.showtwelve();

            break;

        case 3:

            system("cls");

            p.showpan();

            break;

        case 4:

            system("cls");

            d.showdl();

            break;

        }

    }

};

class personaldoc

{

public:

    drive d;

    addhar a;

    pan p;

    twelve t;

    int choicetype;

    void getpersonal()

    {

        cout << "WHICH DOCUMENT DO YOU WANT TO ADD\nENTER \n1.AADHAR CARD DETAIL\n2.12th MARKSHEET\n3.PAN CARD DETAIL\n4.DRIVING LICENSE\n\n";

        cin >> choicetype;

        switch (choicetype)

        {

        case 1:

            a.aadharadd();

            break;

        case 2:

            t.twelveadd();

            break;

        case 3:

            p.panadd();

            break;

        case 4:

            d.dladd();

            break;

        }

    }

};

class officialdoc

{

public:

    propertypaper p;

    bank b;

    passport pass;

    int choicetype;

    void get()

    {

        cout << "WHICH DOCUMENT DO YOU WANT TO ADD\nENTER \n1.PROPERTY PAPER\n2.BANK DETAILS\n3.PASSPORT\n\n";

        cin >> choicetype;

        switch (choicetype)

        {

        case 1:

            p.propertyadd();

            break;

        case 2:

            b.bankadd();

            break;

        case 3:

            pass.passportadd();

            break;

        }

    }

};

class adddoc : public personaldoc, public officialdoc

{

    int ch;

public:

    void get()

    {

        cout << "\nDO YOU WANT TO ADD PERSONAL DOCUMENT OR OFFICIAL DOCUMENT \nPRESS\n1.PERSONAL DOCUMENT \n2.PROFESSIONAL DOCUMENT \n\n";

        cin >> ch;

        system("cls");

        personaldoc per;

        officialdoc official;

        switch (ch)

        {

        case 1:

            per.getpersonal();

            break;

        case 2:

            official.get();

            break;

        }

    }

};

class deletedocument

{

public:

    twelve t;

    pan p;

    drive d;

    addhar a;

    propertypaper prop;

    bank b;

    passport pp;

    int choice;

    void call()

    {

        cout << "WHICH DOCUMENT DO YOU WANT TO DELETE \nPRESS\n1.DELETE AADHAR CARD\n2.DELETE 12th MARKSHEET\n3.DELETE PAN CARD\n4.DELETE DRIVING LICENCE\n5.DELETE PROPERTY DETAIL\n6.DELETE BANK DETIAL\n7.DELETE PASSPORT\n\n";

        cin >> choice;

        system("cls");

        switch (choice)

        {

        case 1:

            a.deleteaadhar();

            break;

        case 2:

            t.deletetwelve();

            break;

        case 3:

            p.deletepan();

            break;

        case 4:

            d.deletedl();

            break;

        case 5:

            prop.deleteproperty();

            break;

        case 6:

            b.deletebank();

            break;

        case 7:

            pp.deletepassport();

            break;

        }

    }

};

void document();

void start(void);

int main()

{

    system("cls");

    start();

    signup s;

    Login l;

    while (1)

    {

        int choice;

        cout << "\n\nENTER YOUR CHOICE \n1. FOR SIGN UP WINDOW\n2. FOR LOGIN WINDOW\n3. FOR EXIT \n\n";

        cin >> choice;

        system("cls");

        switch (choice)

        {

        case 1:

            s.signupdetails();

            break;

        case 2:

            l.logindetails();

            char option;

            cout << "DO YOU WANT TO PROCEED OR NOT. \nPRESS 'Y' FOR YES ,'N' FOR NOT\n\n";

            cin >> option;

            system("cls");

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

                document();

            else

            {

                cout << "WRONG CHOICE ENTERED \n";

                exit(0);

            }

            break;

        case 3:

            exit(0);

        }

    }

    return 0;

}

void document()

{

    int choicedoc;

    int choicestage;

    cout << "ENTER YOUR CHOICE FOR DOCUMENT REQUIRED \nPRESS\n1.ADD DOCUMENT \n2.DELETE DOCUMENT \n3.VIEW DOCUMENT \n4.EXIT\n"

         << endl;

    cin >> choicedoc;

    system("cls");

    switch (choicedoc)

    {

    case 1:

        class adddoc add;

        add.get();

        break;

    case 2:

        class deletedocument d;

        d.call();

        break;

    case 3:

        class view v;

        int ch;

        class viewofficial vv;

        cout << "DO YOU WANT TO VIEW PERSONAL DOCUMENTS OR OFFICIAL DOCUMENTS\n PRESS\n1.PERSONAL DOCUMENT\n2.OFFICIAL DOCUMENT\n\n";

        cin >> ch;

        system("cls");

        if (ch == 1)

            v.get();

        else if (ch == 2)

            vv.get();

        break;

    case 4:

        exit(0);

    }

}

void start(void)

{

    system("color 0B");

    cout << "\n\n\n";

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

    cout << "\t\t\*           WELCOME TO DIGITAL DOCUMENT LOCKER                       \*\n";

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

    cout << "\t\t\*                          PROJECT BY:                               \*\n";

    cout << "\t\t\*                  DIVYANSH RASTOGI(23103296)                        \*\n";

    cout << "\t\t\*                  SIDDHANT SAXENA (23103290)                        \*\n";

    cout << "\t\t\*                  BIKRAM MISTRY   (23103270)                        \*\n";

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

    sleep(3);

    system("cls");

}