JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

B. TECH II SEMESTER

PROJECT SYNOPSIS

Session : 2023-24

TITLE OF PROJECT

DIGITAL DOCUMENT LOCKER

Supervision of:

Mrs. ARCHANA PURWAR

Dr. SHWETA RANI

Mrs. K RAJALAKSHMI

Submitted by:

Siddhant Saxena(23103290)

Divyansh Rastogi(23103296)

Bikram Mistry(23103276)

ABSTRACT OF THE PROJECT:

Upon initiation, the user will be presented with a window prompting them to either create a new account or log in. Subsequently, the user will input their details, including a username, phone number, email ID, and a password. The password strength will be checked, requiring a combination of eight alphabetical letters, one special character, one uppercase letter, one lowercase letter, and one digit.

Once the account creation process is successfully completed, the user can log in using their username and password. Following login, the system will prompt the user to explore available documents, such as Aadhar Card, 10th and 12th Marksheet, PAN card, Driving License, etc.

Upon selection, the chosen document will be displayed on the screen, ensuring a seamless and secure document retrieval experience

Project Objectives:

- **The Digital Locker system aims to enhance document security by implementing robust password protection measures.**
- **It seeks to provide effortless accessibility to documents, promoting user convenience and efficient retrieval.**

Features of the Project:

- **LOGIN or SIGNUP WINDOW** : User will prompted with window to sign up or sign in with username and password .

- **DATA SECURITY** : In this digital un-ethical hacking world of capturing other persons documents , we are hiding our documents from them by providing password security .

- **EASY ACCESS OF DOCUMENTS** : Documents are now easily accessible wherever we want and whatever we want just by simple login as these are present in online mode.

TOPICS OF SDF USED :

- ***OBJECT ORIENTED PROGRAMMING CONCEPT***

1. CLASSES
2. OBJECT
3. INHERITANCE

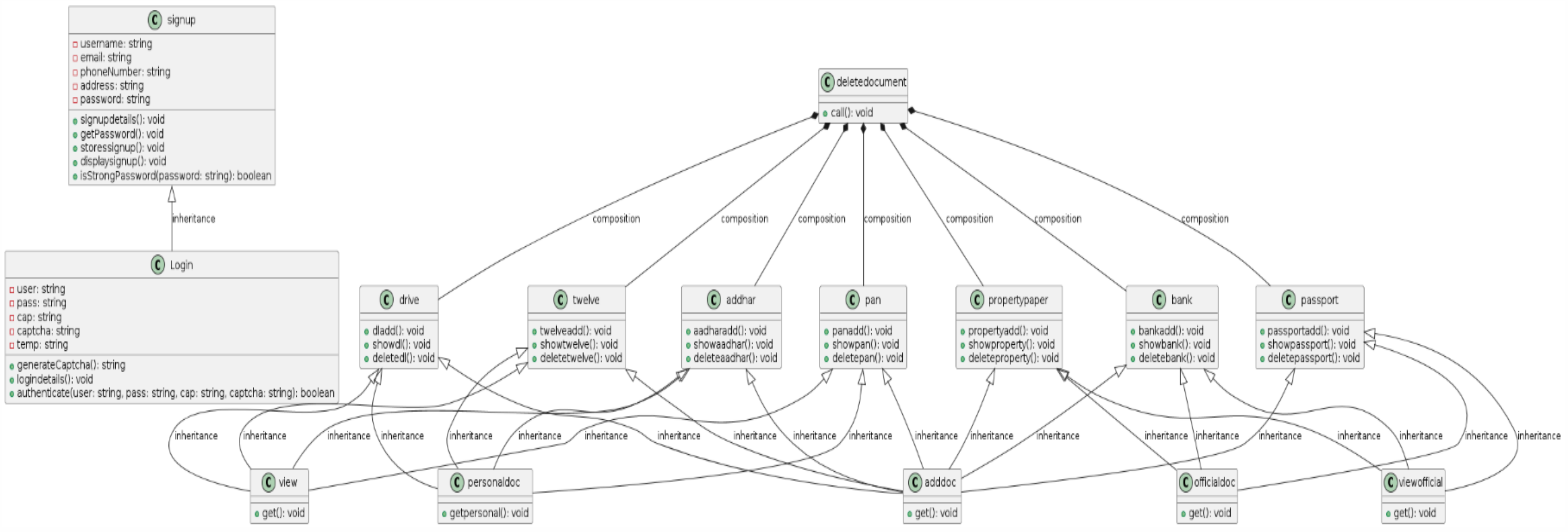
- ***OTHER CONCEPTS:***

1. STRINGS OPERATION
2. FUNCTION
3. FILE HANDLING CONCEPT
4. CONTROL FLOW
5. SRAND AND RAND FUNCTION (CAPTCHA)
6. EXCEPTION HANDLING CONCEPT

- ***HEADER FILES USED:***

1. iostream
2. bits/stdc++.h
3. string
4. unistd.h
5. process.h

UML CLASS DIAGRAM



CODE EXPLANATION

1.signup class:

This class is used to create a new user account. It has a signupdetails() function that prompts the user to enter their personal details like username, email, phone number, address, and password. It also has a getPassword() function that checks the strength of the password and stores the user's details in binary files.

2.Login class:

This class is used to log in to an existing user account. It inherits from the signup class and has a logindetails() function that prompts the user to enter their username, password, and a captcha. It also has an authenticate() function that checks if the user's credentials are correct and grants access to the digital document locker.

3.**twelve, pan, drive, addhar, propertypaper, passport, bank classes:**

These classes are used to handle different types of documents like 12th marksheet, PAN card, driving license, Aadhaar card, property paper, passport, and bank details. Each class has functions to add, delete, and view the respective document.

4.view, viewofficial classes:

These classes are used to handle the document viewing functionality. The view class is used to view personal documents like Aadhaar card, 12th marksheet, PAN card, and driving license. The viewofficial class is used to view official documents like property paper, bank details, and passport.

5.personaldoc, officialdoc, adddoc classes:

These classes are used to handle the document addition functionality. The personaldoc class is used to add personal documents like Aadhaar card, 12th marksheet, PAN card, and driving license. The officialdoc class is used to add official documents like property paper, bank details, and passport. The adddoc class is used to add both personal and official documents.

6.deletedocument class:

This class is used to handle the document deletion functionality. It has a call() function that prompts the user to choose the document they want to delete and deletes the corresponding file.

7.document() and start() functions:

These functions are used to handle the main menu of the digital document locker. The document() function displays the main menu and prompts the user to choose an option. The start() function displays a welcome message and clears the console.

8.main() function:

This is the entry point of the program. It initializes the console, displays the welcome message, and enters an infinite loop that displays the main menu and handles user input.

CODE :

```
#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;
```

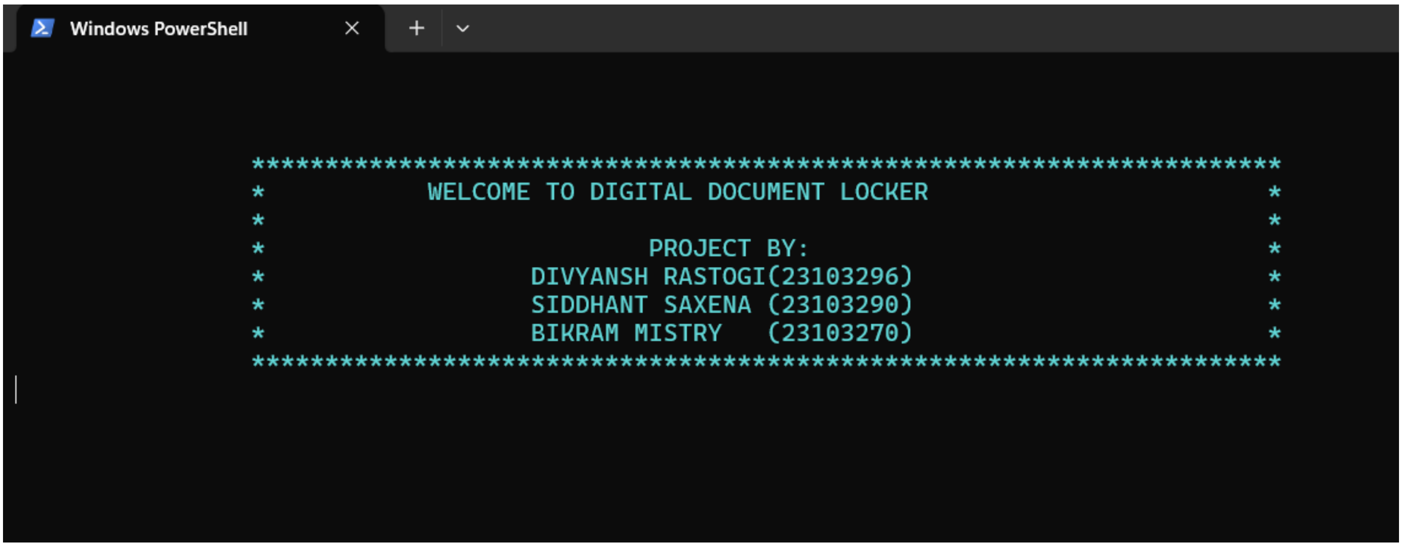
```
                fout << "*  D.O.B: " << left << setw(12) << dob <<
"                      Blood group:AB+          *" << endl;
            }
            else
            {
                fout << line << endl;
            }
        }
        fin.close();
        fout.close();
    }
    void showdl()
    {
        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 <<
"*                                                                                     Cer
tificate 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";
```

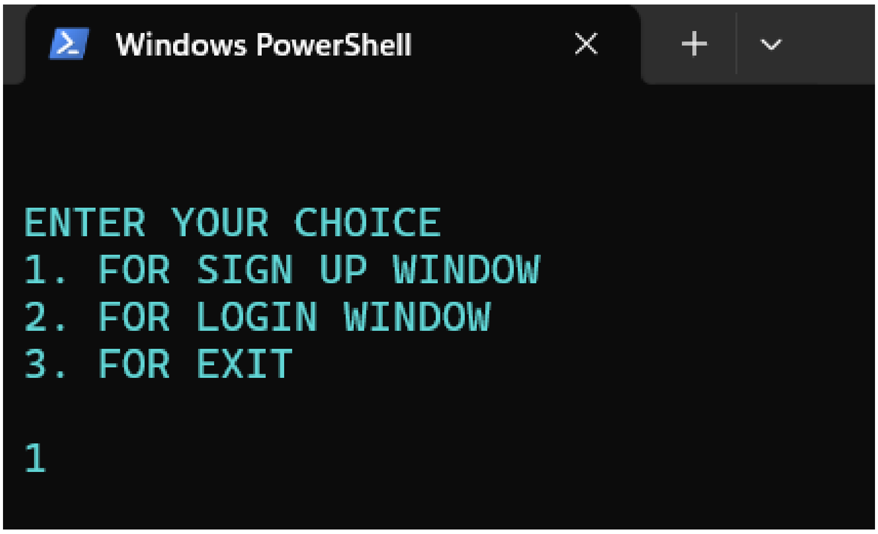
```
        cin >> choicetype;
        switch (choicetype)
        {
        case 1:
            a.aadharadd();
            break;
        case 2:
            t.twelveadd();
            break;
        case 3:
            p.panadd();
            break;
        case 4:
            d.dladd();
            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");
}
```

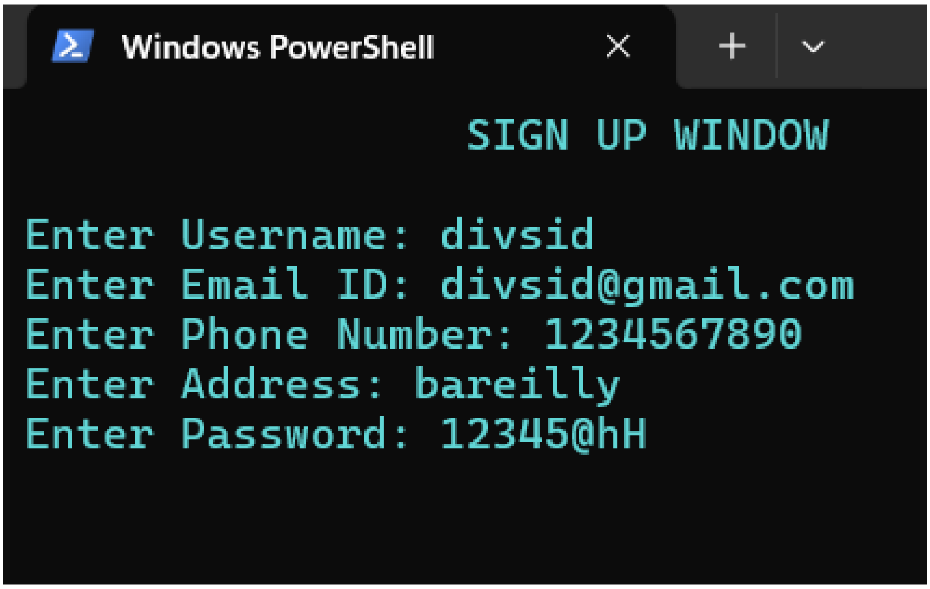
OUTPUT:



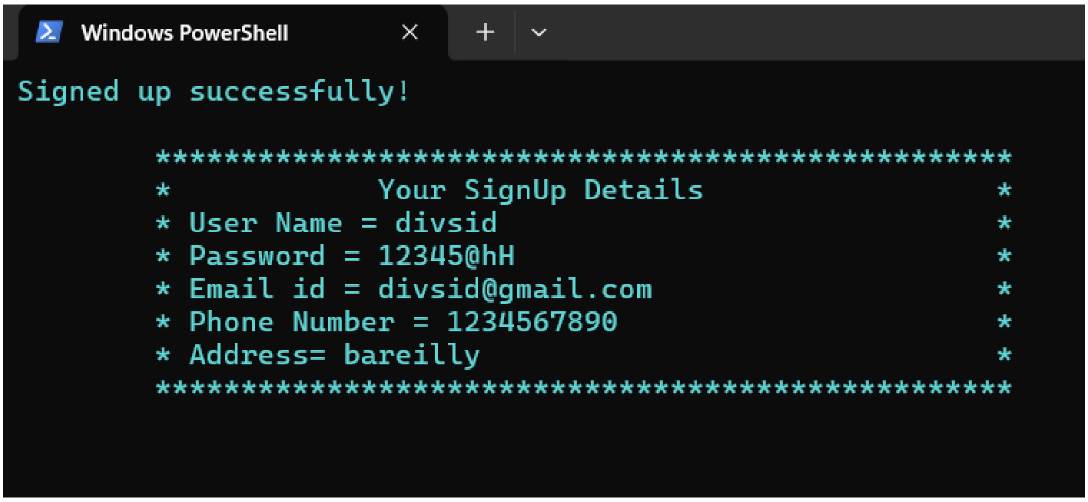
```
**********************************************************************
*          WELCOME TO DIGITAL DOCUMENT LOCKER                        *
*                                                                    *
*                       PROJECT BY:                                  *
*                 DIVYANSH RASTOGI(23103296)                         *
*                 SIDDHANT SAXENA (23103290)                         *
*                 BIKRAM MISTRY   (23103270)                         *
**********************************************************************
```



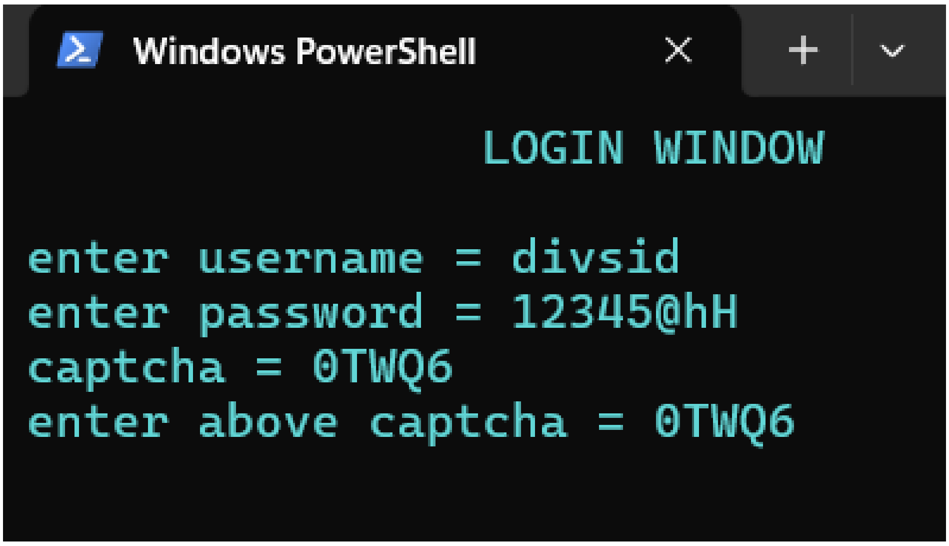
```
ENTER YOUR CHOICE
1. FOR SIGN UP WINDOW
2. FOR LOGIN WINDOW
3. FOR EXIT

1
```



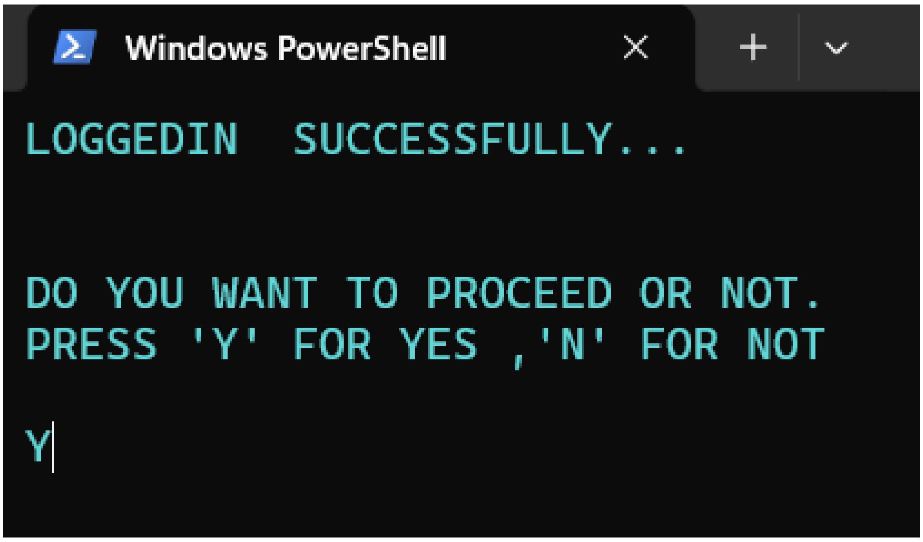
```
                   SIGN UP WINDOW

Enter Username: divsid
Enter Email ID: divsid@gmail.com
Enter Phone Number: 1234567890
Enter Address: bareilly
Enter Password: 12345@hH
```



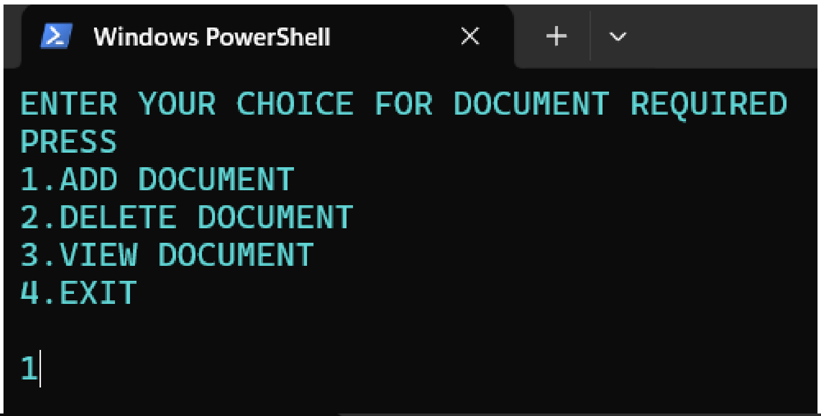
```
Signed up successfully!

        **************************************************
        *             Your SignUp Details                *
        * User Name = divsid                             *
        * Password = 12345@hH                            *
        * Email id = divsid@gmail.com                    *
        * Phone Number = 1234567890                      *
        * Address= bareilly                              *
        **************************************************
```



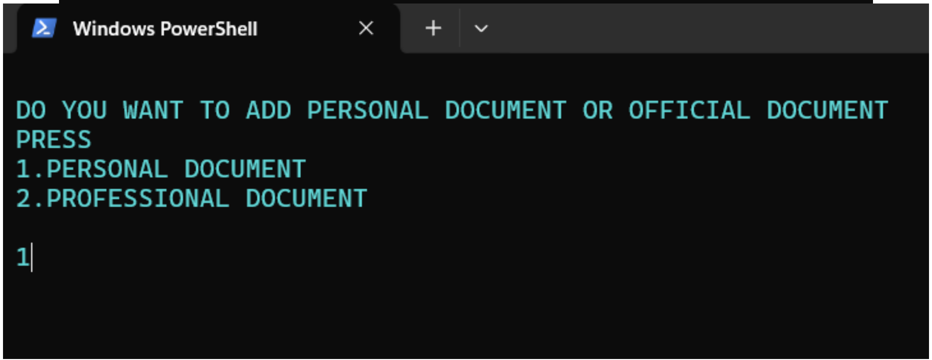
```
                LOGIN WINDOW

enter username = divsid
enter password = 12345@hH
captcha = 0TWQ6
enter above captcha = 0TWQ6
```



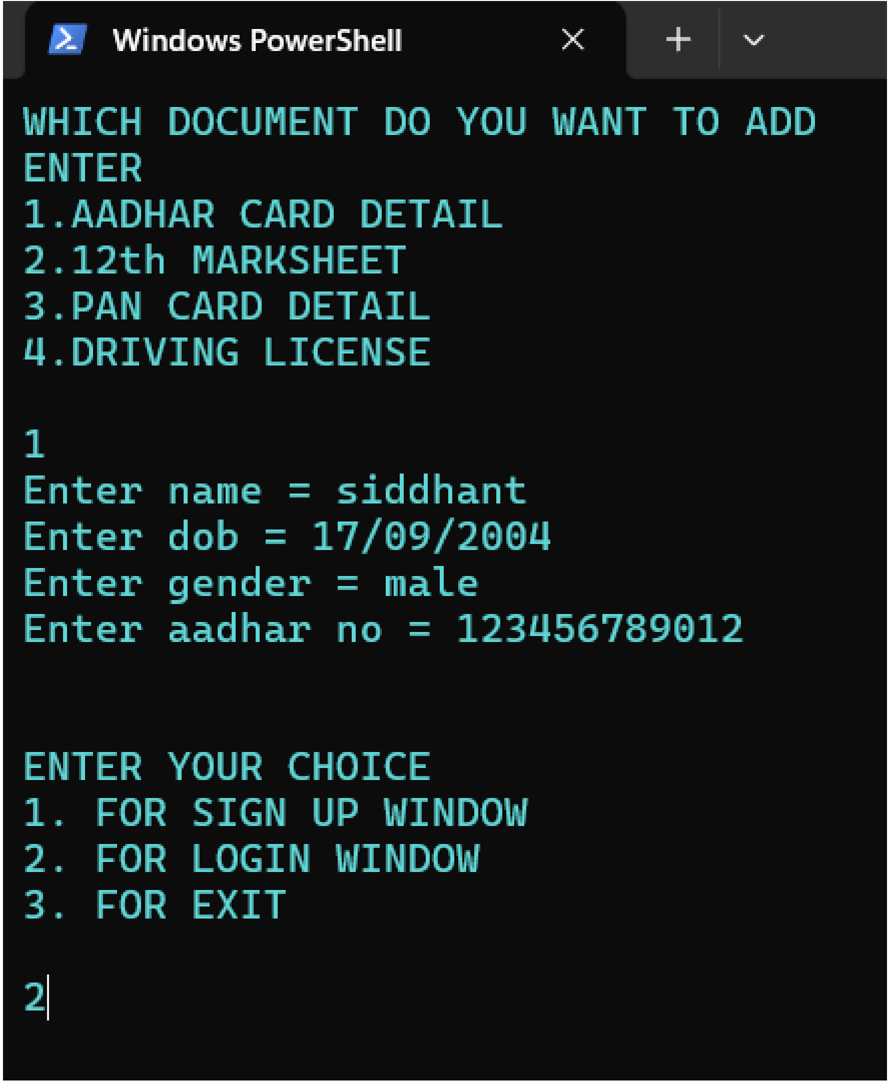
```
LOGGEDIN  SUCCESSFULLY...


DO YOU WANT TO PROCEED OR NOT.
PRESS 'Y' FOR YES ,'N' FOR NOT

Y
```



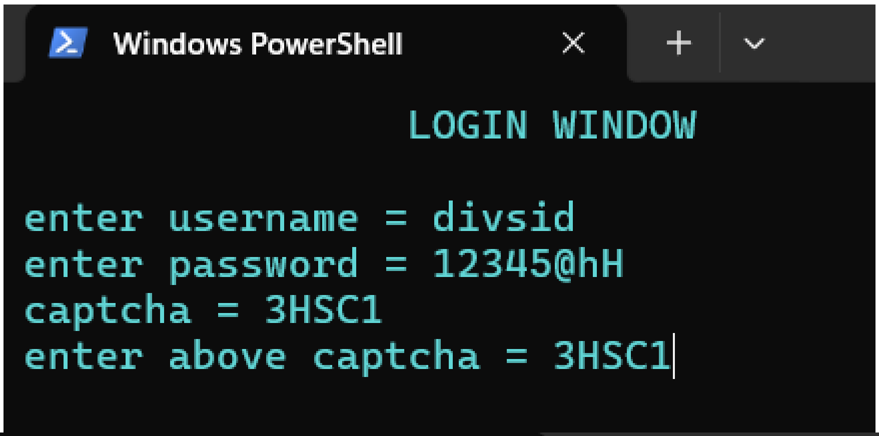
```
ENTER YOUR CHOICE FOR DOCUMENT REQUIRED
PRESS
1.ADD DOCUMENT
2.DELETE DOCUMENT
3.VIEW DOCUMENT
4.EXIT

1
```



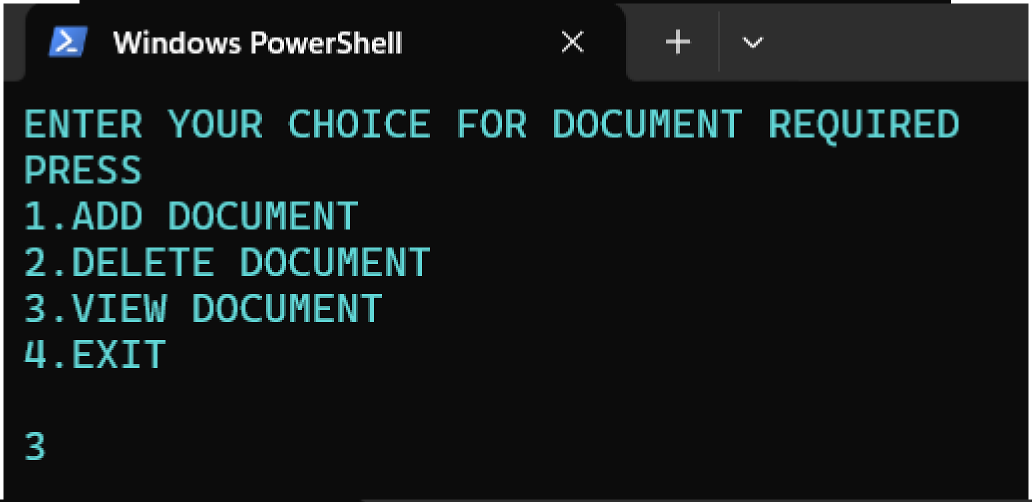
```
DO YOU WANT TO ADD PERSONAL DOCUMENT OR OFFICIAL DOCUMENT
PRESS
1.PERSONAL DOCUMENT
2.PROFESSIONAL DOCUMENT

1
```



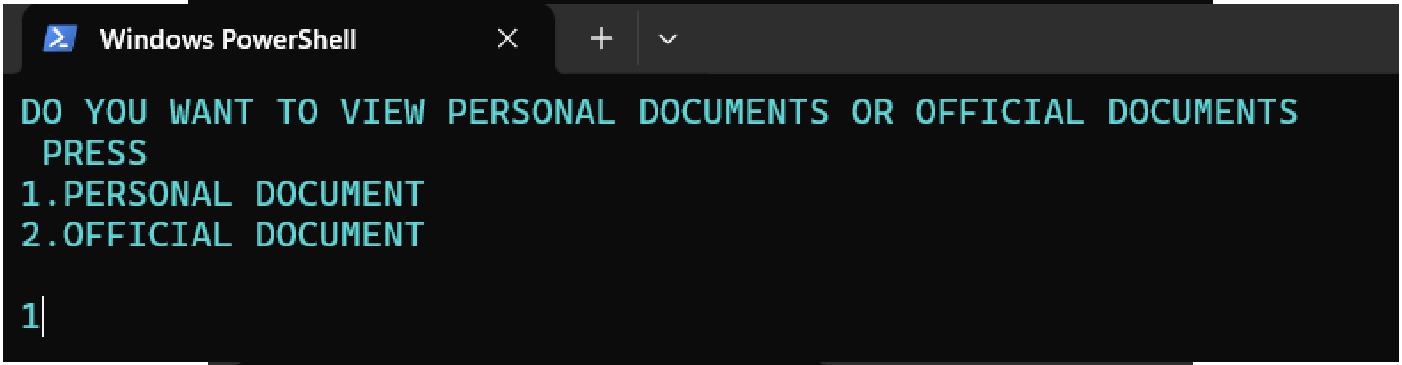
```
WHICH DOCUMENT DO YOU WANT TO ADD
ENTER
1.AADHAR CARD DETAIL
2.12th MARKSHEET
3.PAN CARD DETAIL
4.DRIVING LICENSE

1
Enter name = siddhant
Enter dob = 17/09/2004
Enter gender = male
Enter aadhar no = 123456789012


ENTER YOUR CHOICE
1. FOR SIGN UP WINDOW
2. FOR LOGIN WINDOW
3. FOR EXIT

2
```



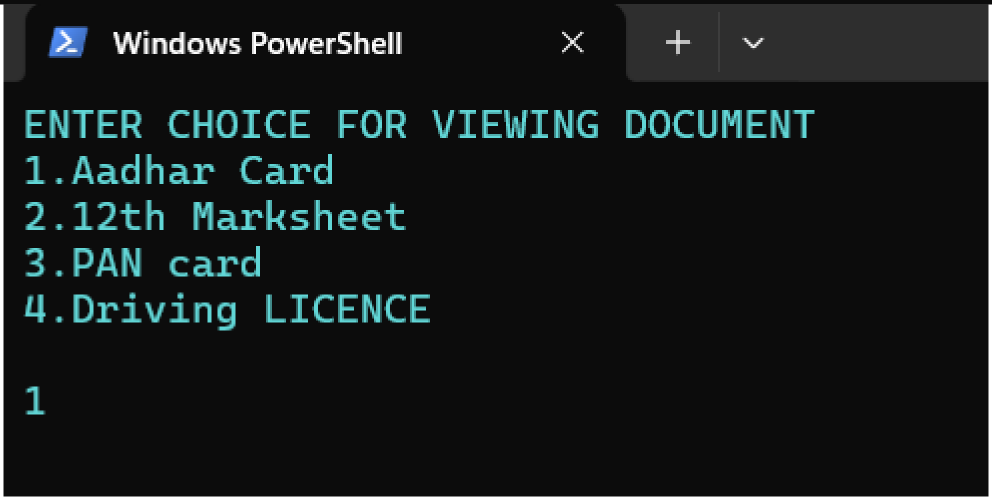
```
               LOGIN WINDOW

enter username = divsid
enter password = 12345@hH
captcha = 3HSC1
enter above captcha = 3HSC1
```



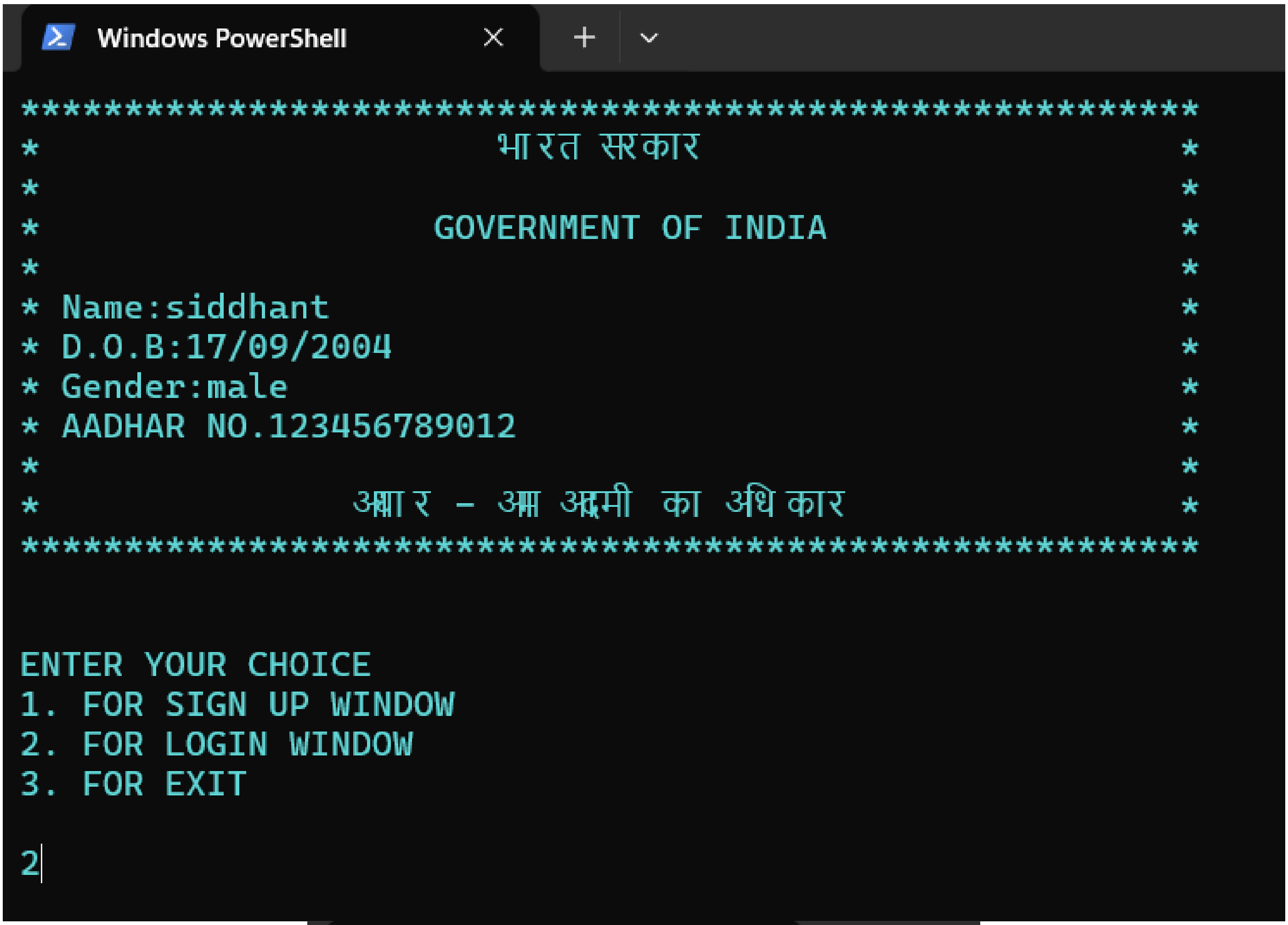
```
ENTER YOUR CHOICE FOR DOCUMENT REQUIRED
PRESS
1.ADD DOCUMENT
2.DELETE DOCUMENT
3.VIEW DOCUMENT
4.EXIT

3
```



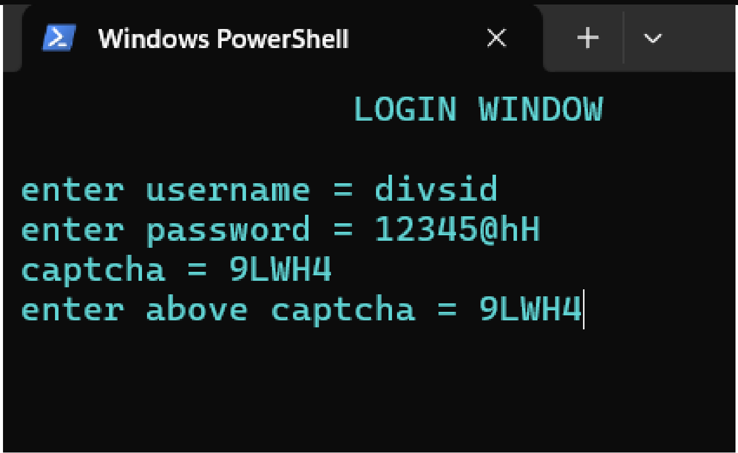
```
DO YOU WANT TO VIEW PERSONAL DOCUMENTS OR OFFICIAL DOCUMENTS
 PRESS
1.PERSONAL DOCUMENT
2.OFFICIAL DOCUMENT

1
```



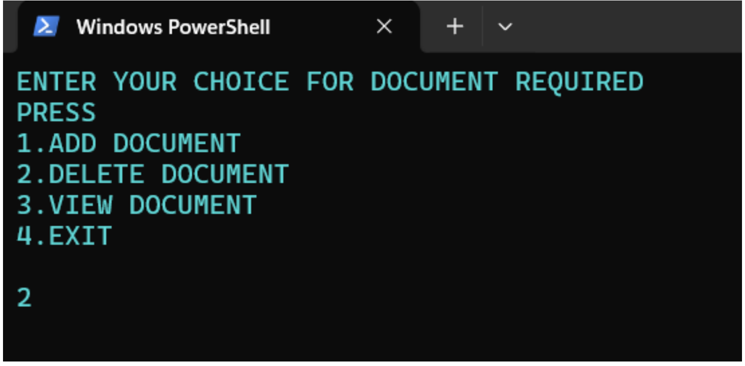
```
ENTER CHOICE FOR VIEWING DOCUMENT
1.Aadhar Card
2.12th Marksheet
3.PAN card
4.Driving LICENCE

1
```



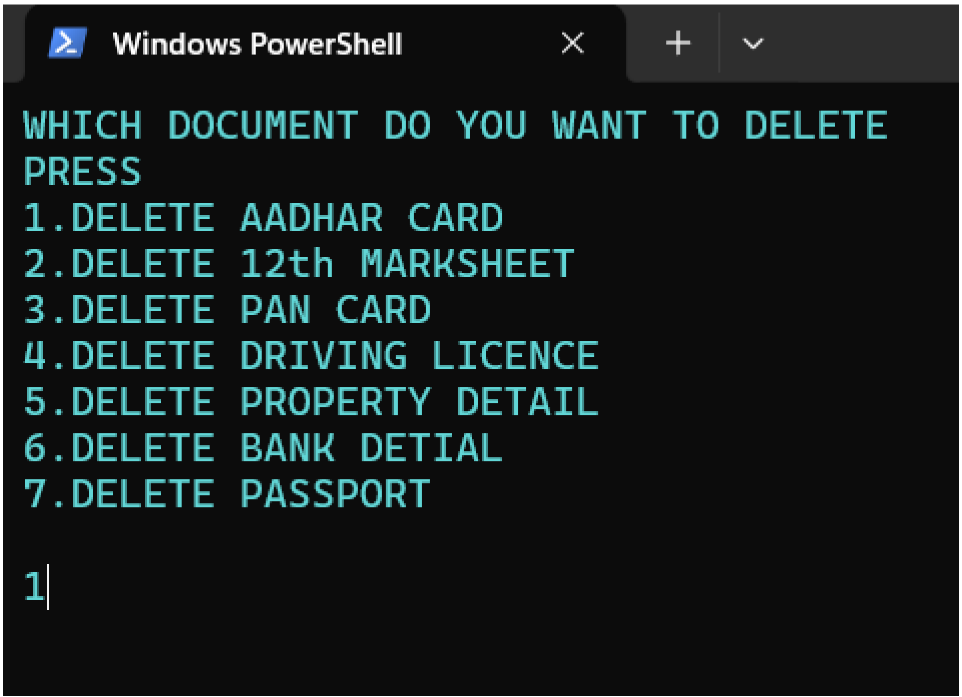
```
*********************************************************
*                       भारत सरकार                      *
*                                                       *
*                    GOVERNMENT OF INDIA                *
*                                                       *
* Name:siddhant                                         *
* D.O.B:17/09/2004                                      *
* Gender:male                                           *
* AADHAR NO.123456789012                                *
*                                                       *
*               आधार - आम आदमी का अधिकार                *
*********************************************************


ENTER YOUR CHOICE
1. FOR SIGN UP WINDOW
2. FOR LOGIN WINDOW
3. FOR EXIT

2
```



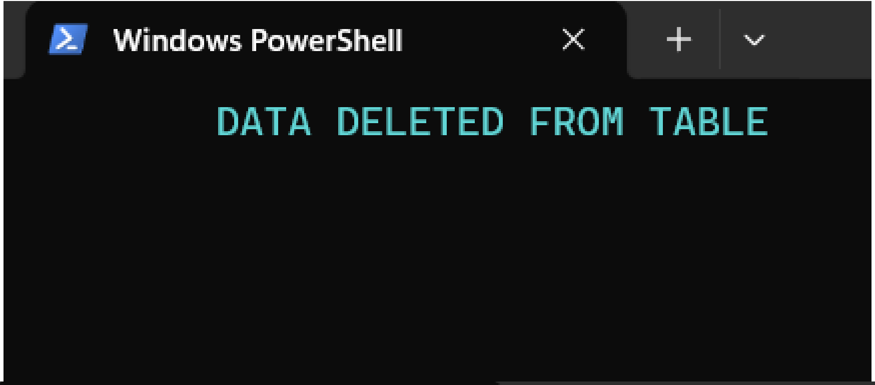
```
              LOGIN WINDOW

enter username = divsid
enter password = 12345@hH
captcha = 9LWH4
enter above captcha = 9LWH4
```



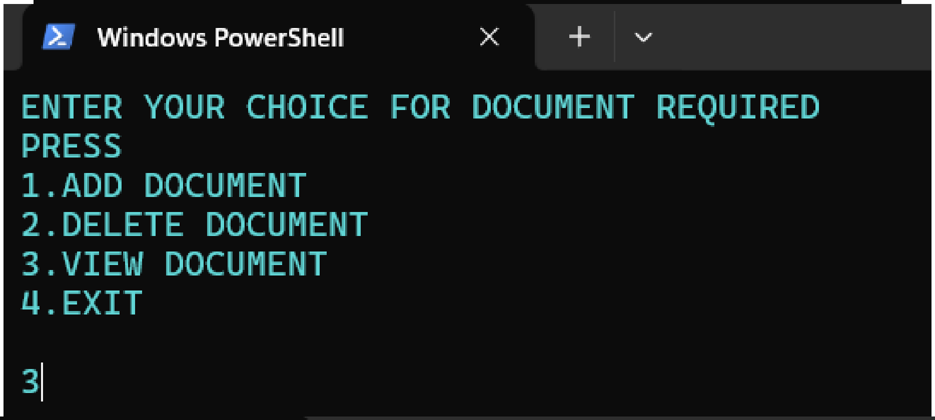
```
ENTER YOUR CHOICE FOR DOCUMENT REQUIRED
PRESS
1.ADD DOCUMENT
2.DELETE DOCUMENT
3.VIEW DOCUMENT
4.EXIT

2
```



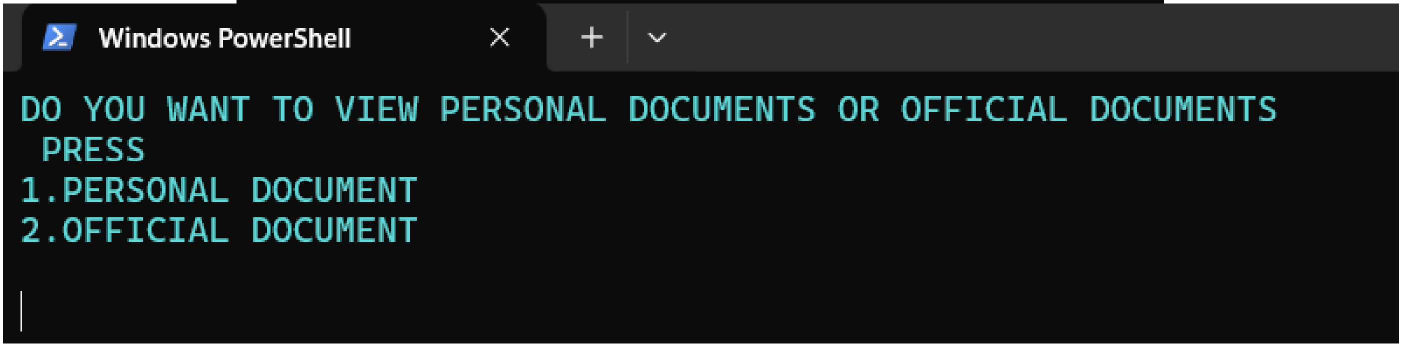
```
WHICH DOCUMENT DO YOU WANT TO DELETE
PRESS
1.DELETE AADHAR CARD
2.DELETE 12th MARKSHEET
3.DELETE PAN CARD
4.DELETE DRIVING LICENCE
5.DELETE PROPERTY DETAIL
6.DELETE BANK DETIAL
7.DELETE PASSPORT

1
```



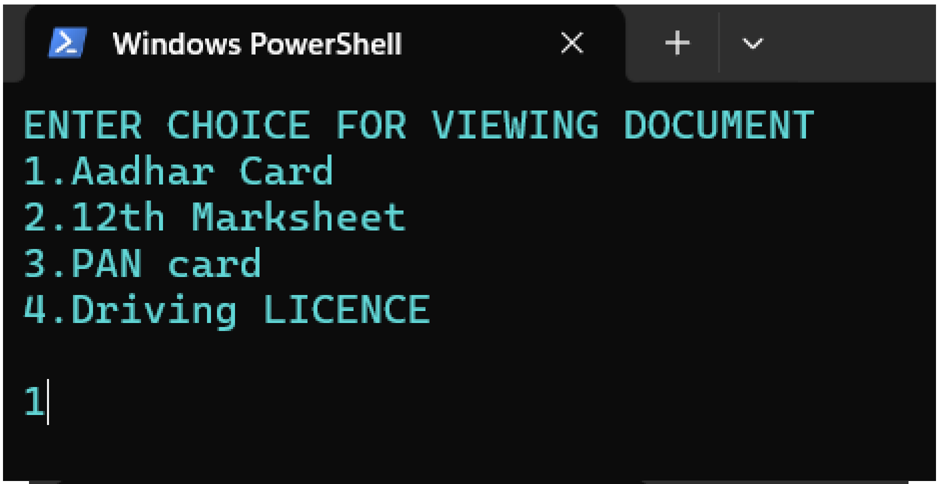
```
          DATA DELETED FROM TABLE
```



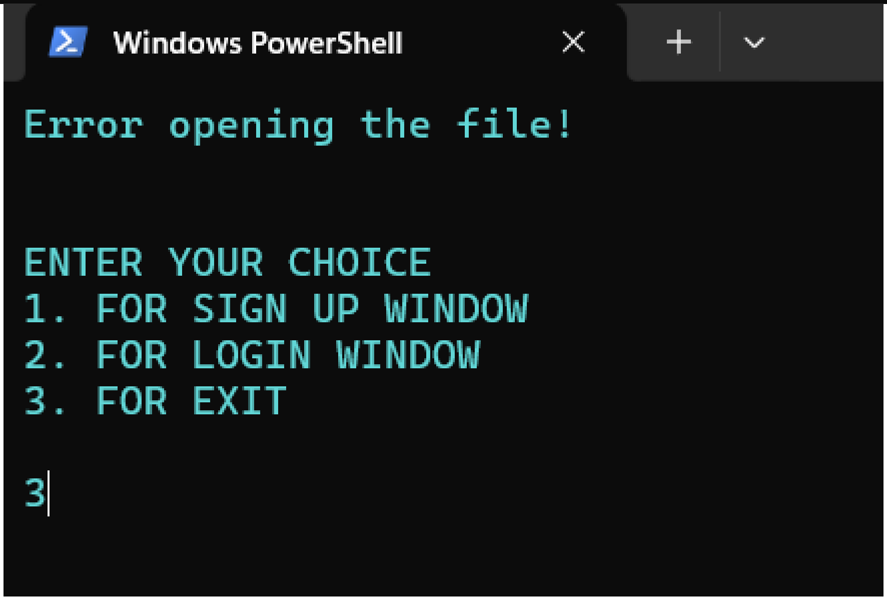
```
ENTER YOUR CHOICE FOR DOCUMENT REQUIRED
PRESS
1.ADD DOCUMENT
2.DELETE DOCUMENT
3.VIEW DOCUMENT
4.EXIT

3
```



```
DO YOU WANT TO VIEW PERSONAL DOCUMENTS OR OFFICIAL DOCUMENTS
 PRESS
1.PERSONAL DOCUMENT
2.OFFICIAL DOCUMENT
```



```
ENTER CHOICE FOR VIEWING DOCUMENT
1.Aadhar Card
2.12th Marksheet
3.PAN card
4.Driving LICENCE

1
```



```
Error opening the file!


ENTER YOUR CHOICE
1. FOR SIGN UP WINDOW
2. FOR LOGIN WINDOW
3. FOR EXIT

3
```

REFERENCES:

1. Robert Lafore, Object Oriented Programming in C++, SAMS, 4th Edition, 2002.
2. Herbert Schildt, C++: The Complete Reference, McGraw-Hill Osborne Media, 4th Edition, 2017.
3. https://youtu.be/CpFx-5J7D7s?feature=shared