Prevention of Shoulder Surfing Attacks

Project Review Report
Submitted by:

Aman Anand – 19CBE0521
Lokesh Mishra – 19BCE2672

Prepared for the subject

Information Security and Analysis and Audit (CSE3501) – J Component

Submitted to:

Prof. Aju D (Associate Professor Grade 1)
Dept. of Information Security, School of Computer Science Engineering

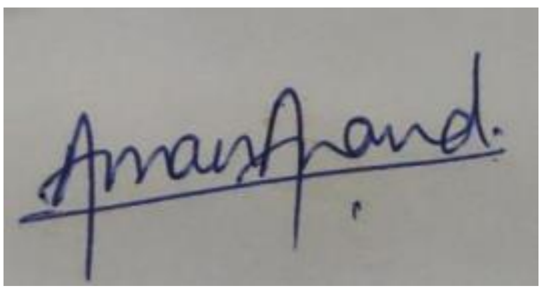
School of Computer Science Engineering

December 2021

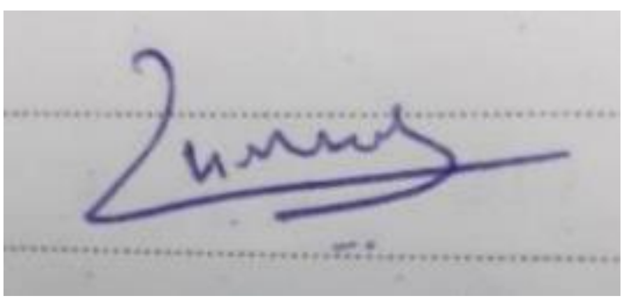
Declaration

We as a team of students from the esteemed institution of Vellore Institute of Technology, hereby declare that the project work entitled “**Prevention of Shoulder Surfing Attacks**” is a record of original work completed by us under the unparalleled and invaluable guidance of our professor, Mr. Aju D, Associate Professor Grade 1, School of Computer Science Engineering. Our project draws inspiration from various current architectures being implemented and in no way is intended to be a duplication of others' works. We further declare that this project will not intentionally be misused and replicated for any other ongoing courses that we have or may have shortly.

(Student Signatures)



Aman Anand (RN – 19BCE0521)



Lokesh Mishra (RN – 19BCE2672)

Acknowledgment

We as a team have taken many efforts in this project. However, this journey would not have been possible without the immense support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

Primarily we would like to extend our thanks to God Almighty for providing us with everything that we required to complete this project.

We take immense pleasure in thanking **Dr. G Vishwanathan**, our beloved chancellor of VIT University, our respected Dean **Dr. Ramesh Babu K**, and our HoD **Dr. Vairamuthu S**, for having permitted our team in carrying out this project.

We are highly indebted to our teacher in charge **Prof. Aju D** for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing this project.

Words are inadequate to express our gratitude towards our parents and fellow peers for their kind co-operation and encouragement while developing this project which helped us in the completion of this project.

Abstract

1. Motivation, aim, and objective of the project.

While going throughout this course we learned that a lot of the websites have password encryptions where the entered password is encrypted using Algorithms like DES or AES. In computer security, shoulder surfing is a type of social engineering technique used to obtain information such as **Personal Identification Number** (PIN), login passwords, and other confidential data by looking over the victim’s shoulder, either from keystrokes from a device or sensitive information being spoken and heard, also known as eavesdropping. This attack can be performed either at close range (by directly looking over the victim's shoulder) or from a longer range, for example by using a pair of binoculars or similar hardware. To implement this technique attackers do not require any technical skills; keen observation of victims' surroundings and the typing pattern is sufficient. This attack can also be performed by the use of keylogging where the attackers track the key presses the user does on the keyboard instead of looking over one’s shoulder.

A user enters a password based on the type of input system. If the input system is pattern-based, then the user enters a pattern to log in. If the input system is code-word based then the user enters a code word to log in. In this project, we intend to completely remove the chances of any type of Shoulder surfing attack. We introduce a new algorithm, which is effortless to implement but very effective. The algorithm along with the way the information is stored and fetched from the database (without any encryption and hashing method) makes it so much secure, that even if a user tells the person his/her login password, he will never be able to log in to the system.

2. About Methodology.

The methodology used in this project is that instead of using the keyboard as the primary input device. Instead of using the keyboard we will be using the mouse and giving the password through a dynamic graphical input system. This type of system accepts a pattern-based input that is entered in a code word form.

3. **Expected Outcome.**

The outcome that we expect through this project is to stop all the attackers who are trying to get the different types of confidential data of the users by either eavesdropping or keylogging.

Keywords

Shoulder surfing, keylogging, MD5, pattern, authentication

Introduction

1. **Overall idea about the project.**

The idea behind the project is to use this algorithm so that the users are safe from key-loggers and eavesdropping by the attackers. Since every time the user enters the password through the **Dynamic Graphical Input**, the pattern changes every time, so the attacker records and enters the wrong password every time.

2. **Background of the project.**

Shoulder Surfing is using direct observation techniques, such as looking over someone's shoulder, to get information. Shoulder surfing is an effective way to get information in crowded places because it's relatively easy to stand next to someone and watch as they fill out a form, enter a PIN at an ATM, or use a calling card at a public payphone. Shoulder surfing can also be done long distances with the aid of binoculars or other vision-enhancing devices. To prevent shoulder surfing, experts recommend that you shield paperwork or your keypad from view by using your body or cupping your hand. Shoulder surfing can also be categorized as **eavesdropping**.

Another attack that is discussed in this project is **Key-logging.**

Keyloggers are a type of monitoring software designed to record keystrokes made by a user. One of the oldest forms of cyber threat, these keystroke loggers record the information you type into a website or application and send it back to a third party. Keyloggers collect information and send it back to a third party – whether that is a criminal, law enforcement, or IT department. "Keyloggers are software programs that leverage algorithms that monitor keyboard strokes through pattern recognition and other techniques.

3. Advantages and disadvantages of the various methods as well as the projects.

There are mainly two types of **Secure Hash Algorithms** (SHA) and **Message Digest**(MDx).

Secure Hash Algorithms, also known as **SHA**, are a family of cryptographic functions designed to keep data secured. It works by transforming the data using a hash function: an algorithm that consists of bitwise operations, modular additions, and compression functions. The hash function then produces a fixed-size string that looks nothing like the original. These algorithms are designed to be one-way functions, meaning that once they're transformed into their respective hash values, it's virtually impossible to transform them back into the original data. A few algorithms of interest are **SHA-1**, **SHA-2**, and **SHA-3**, each of which was successively designed with increasingly stronger encryption in response to hacker attacks. **SHA-0**, for instance, is now obsolete due to the widely exposed vulnerabilities.

A common application of SHA is to encrypt passwords, as the server-side only needs to keep track of a specific user's hash value, rather than the actual password. This is helpful in case an attacker hacks the database, as they will only find the hashed functions and not the actual passwords, so if they were to input the hashed value as a password, the hash function will convert it into another string and subsequently deny access.

Message digest algorithms rely on cryptographic hash functions to generate a unique value that is computed from data and a unique symmetric key. A cryptographic hash function inputs data of arbitrary length and produces a unique value of a fixed length. Because message digest algorithms generate a value that is always used in encrypted form (never decrypted), they are sometimes known as encryption-only algorithms.

Adding a unique symmetric key that is shared between a sender and receiver to compute a message digest value provides confidentiality to ensure that the message digest cannot be easily changed if the data is changed in an unauthorized or unexpected manner.

Both the sender and receiver of the data (including the senders' message digest) must share the same key for the receiver to generate an identical message digest.

In this project, we have used **MD5 Message-Digest Algorithm**

MD5 message-digest algorithm is the 5th version of the **Message-Digest Algorithm** developed by Ron Rivest to produce a 128-bit message digest. MD5 is quite fast than other versions of message digest which takes the plain text of 512-bit blocks which is further divided into 16 blocks, each of 32 bit, and produces the 128-bit message digest which is a set of four blocks, each of 32 bits. MD5 produces the message digest through five steps i.e. padding, append length, dividing the input into 512-bit blocks, initializing chaining variables a process blocks, and 4 rounds, using different constants in each iteration.

Literature Survey / Related Works

S.No.	Paper Title	Citation (APA Format)	Information/Knowledge Gained
1.	**Password Encryption Key**	Utin, D. (2013). *U.S. Patent No. 8,447,990*. Washington, DC: U.S. Patent and Trademark Office.	A password-encrypted key (PEK) is generated from a user-supplied password or other identifying data and then used to encrypt the user's password. This password is stored and at log in, the user enters the password which is also encrypted and which is then used to decrypt and compare with the stored password.
2.	**Two-factor graphical password for text password and encryption key generation**	Gyorffy, J. C., & Miller, J. (2010). U.S. Patent Application No. 12/659,264.	This invention details systems, methods, and devices for providing a two-factor graphical password system to a user so that the user may obtain access to a restricted resource. The first previously selected image which is selected by the user is presented to the user to enter his password by sequentially selecting predetermined areas on the first image. The user's input is used to create an encryption/decryption key which is used for communicating between a user application and a device. If the user has entered the correct password, then the device can communicate with the user application.
3.	**Password-Based Encryption Analyzed**	Abadi, M., & Warinschi, B. (2005,	We offer two models for reasoning about the concurrent use of symmetric,

		July). Password-based encryption analyzed. In International Colloquium on Automata, Languages, and Programming (pp.664-676). Springer, Berlin, Heidelberg.	asymmetric, and password-based encryption in protocol messages. In each of the models, we define a notion of equivalence between messages and also characterize when passwords are used securely in a message or a set of messages. Our new definition for the computational security of password-based encryption may be of independent interest. The main results of this paper are two soundness theorems. We show that under certain (standard) assumptions about the computational implementation of the cryptographic primitives, symbolic equivalence implies computational equivalence.
4.	**Encryption method and system for portable data**	Hardy, D. A., Fossey, C. R., Balogh, C. R., & Tugenberg, S. R. (1997). U.S. Patent No. 5,623,546. Washington, DC: U.S. Patent and Trademark Office.	The system and method allow portable, encrypted data to be accessible through multiple hosts, including new hosts without requiring a secure link to the new hosts. A split key encryption stores the encrypted data on a portable device. A split of the data is stored on the portable device and another split of data is stored in a home host. Then a combined password is made and stored.
5.	**Password Encryption for hybrid cloud services**	Singleton IV, L. C., & Cooper, A. (2019). U.S. Patent No. 10,432,592. Washington, DC: U.S. Patent and Trademark Office.	Methods, systems, computer-readable media, and apparatuses may provide password encryption for hybrid cloud services. A workspace cloud connector internally residing with an entity may intercept user credentials associated with an internal application being transmitted to an external cloud service. The workspace cloud connector may generate an encryption key and encrypt the user credentials via a reversible encryption methodology. The workspace cloud connector may encrypt the encryption key using an irreversible encryption methodology (e.g., use a hashing function to produce a first hash).
6.	**Securely generating a computer system password by utilizing an external encryption algorithm**	Angelo, M. F. (2002). U.S. Patent No. 6,400,823. Washington, DC: U.S. Patent and Trademark Office.	A method for generating system passwords derived from an external encryption algorithm and plain text user passwords entered during a secure power-on procedure. At some point during the secure power-up procedure, the computer system checks for the presence of an external token or smart card that is coupled to the computer through specialized hardware. The token or smart card is used to store an encryption algorithm furnished with a unique encryption key or of limited

			production. Following detection of the external token, the computer user is required to enter a user password. The user password is encrypted using the encryption algorithm contained in the external token, thereby creating a system password. This system password is compared to the stored password
7.	**A security system that uses indirect password-based encryption**	Gutnik, Y.(2010). U.S. Patent No. 7,836,310. Washington, DC: U.S. Patent and Trademark Office.	An improved system and approaches for protecting passwords are disclosed. A file security system for an organization operates to protect the files of the organization and thus prevents or limits users from accessing some or all of the files (e.g., documents) associated with the organization. According to one aspect, a password entered by a user is used, provided it is authenticated, to obtain a respective authentication string (a relatively long string of numbers or characters). The retrieved authentication string is then used to enable the user to enter the file security system and/or to access secured files therein.
8.	**A PIN-entry method resilient against shoulder surfing**	Roth, V., Richter, K., & Freidinger, R. (2004, October). A PIN-entry method is resilient against shoulder surfing. In Proceedings of the 11th ACM conference on Computer and communications security (pp. 236-245).	Personal identification numbers (PINs) are obtained by shoulder surfing, through the use of mirrors or concealed miniature cameras. In this paper, we present alternative PIN entry methods to which we refer as cognitive trapdoor games. These methods make it significantly harder for a criminal to obtain PINs even if he fully observes the entire input and output of a PIN entry procedure. We also introduce the idea of probabilistic cognitive trapdoor games, which offer resilience to shoulder surfing even if the criminal records a PIN entry procedure with a camera.
9.	**A Shoulder-Surfing Resistant Graphical Password Scheme**	Man, S., Hong, D., & Matthews, M. M. (2003, June). A Shoulder-Surfing Resistant Graphical Password Scheme-Wiw. In Security and Management (pp. 105-111).	We propose a new graphical password scheme. It is defined as a challenge-response identification. Hence, a password in our scheme is time-variant. The user who knows the password can meet the challenge and respond correctly. As a consequence, our graphical password scheme is shoulder-surfing resistant. An attacker still cannot tell what the password is, even if he/she has filmed a user's login process. Primary experiments on our graphical password scheme showed the scheme is promising.
10.	**An Association-Based Graphical Password Design Resistant to**	Li, Z., Sun, Q., Lian,Y., & Giusto, D. D. (2005, July). An association-based	In line with the recent call for technology on Image-Based Authentication (IBA) in the JPEG

	Shoulder-Surfing Attack	graphical password design resistant to shoulder-surfing attack. In 2005 IEEE international conference on multimedia and expo (pp. 245-248). IEEE.	committee, they present a novel graphical password design in the paper. It rests on the human cognitive ability of association-based memorization to make the authentication more user-friendly, compared with a traditional textual password. Based on the principle of zero-knowledge proof protocol, they further improve our primary design to overcome the shoulder-surfing attack issue without adding any extra complexity to the authentication procedure. System performance analysis and comparisons are presented to support our proposals.
11.	**A Shoulder Surfing Resistant Graphical Authentication System**	Sun, H. M., Chen, S.T., Yeh, J. H., & Cheng, C. Y. (2016). A shoulder-surfing resistant graphical authentication system. IEEE Transactions on Dependable and Secure Computing, 15(2), 180-193.	Inputting the passwords is the weakest link in the authentication chain. Users usually enter passwords that are short and/or easy to memorize. To overcome this problem, we proposed a novel authentication system PassMatrix, based on graphical passwords to resist shoulder surfing attacks. With a one-time valid login indicator and circulative horizontal and vertical bars covering the entire scope of pass-images, PassMatrix offers no hint for attackers to figure out or narrow down the password even they conduct multiple camera-based attacks.
12.	**Novel Shoulder-Surfing Resistant Haptic-based Graphical Password**	Malek, B., Orozco, M., & El Saddik, A. (2006, July). Novel shoulder-surfing resistant haptic-based graphical password. In Proc. EuroHaptics (Vol. 6, pp. 1-6).	Graphical password schemes are believed to be more secure and more resilient to dictionary attacks than textual passwords, but more vulnerable to shoulder surfing attacks. In this work, we design a new graphical password that is larger in the possible passwords' space than in similar schemes and it is more resilient to shoulder surfing attacks. Personal entropies are integrated into the system in the user-aware behavior that reduces the False Acceptance and False Rejection Rates. The user-aware personal entropy we employ is the binary pressure when drawing a secret on the screen; unlike conventional authentication schemes that use personal entropies, the binary pressure in our scheme is varied arbitrarily by the users and is not intuitive. This method yields the authentication scheme that acquires all the advantages of graphical passwords and behavioral authentication schemes altogether; our scheme is resilient to both dictionary attacks and shoulder surfing attacks
13.	**A simple text-based shoulder surfing**	Chen, Y. L., Ku, W. C., Yeh, Y. C., &	Since conventional password schemes are vulnerable to shoulder surfing, many

	resistant graphical password scheme	Liao, D. M. (2013, February). A simple text-based shoulder surfing resistant graphical password scheme. In 2013 International Symposium on Next-Generation Electronics (pp. 161-164). IEEE.	shoulder-surfing-resistant graphical password schemes have been proposed. Unfortunately, none of the existing text-based shoulder surfing resistant graphical password schemes is both secure and efficient enough. In this paper, we propose an improved text-based shoulder surfing resistant graphical password scheme by using colors. In the proposed scheme, the user can easily and efficiently log in to the system.
14.	**Integrated network security system**	Weiss, K. P. (1993). U.S. Patent No. 5,237,614. Washington, DC: U.S. Patent and Trademark Office.	An integrated network security system is provided which permits log-on to a normally locked client on the network in response to at least one coded non-public input to the client by a user. At least a selected portion of the coded input is encrypted and sent to a network server where the user is authenticated. After authentication, the server preferably returns a decryption key, an encryption key for future use, and any critical files previously stored at the server to the client. The decryption key is utilized to decrypt any material at the client which was encrypted when the client was locked, including any material sent from the server, thereby unlocking the client. The decryption key may be combined with untransmitted portions of the original coded input in a variety of ways to generate an encryption key for the next time the terminal is to be locked.
15.	**Network Security and Surveillance System**	Trcka, M. V., Fallon, K. T., Jones, M. R., & Walker, R. W. (2002). U.S. Patent No. 6,453,345. Washington, DC: U.S. Patent and Trademark Office.	A network security and surveillance system passively monitors and records the traffic present on a local area network, wide area network, or another type of computer network, without interrupting or otherwise interfering with the flow of the traffic. Raw data packets present on the network are continuously routed (with optional packet encryption) to a high-capacity data recorder to generate low-level recordings for archival purposes. The raw data packets are also optionally routed to one or more cyclic data recorders to generate temporary records that are used to automatically monitor the traffic in near-real-time. A set of analysis applications and other software routines allows authorized users to interactively analyze the low-level traffic recordings to evaluate network attacks

16.	**Graphical network security policy management**	Wiegel, S. L. (2002). U.S. Patent No. 6,484,261. Washington, DC: U.S. Patent and Trademark Office.	A method of establishing a representation of an abstract network security policy is disclosed. The representation is established in the form of a decision tree that is constructed by assembling graphical symbols representing policy actions and policy conditions. A user modifies properties of the graphical symbols to create a logical representation of the policy. Concurrently, the logical representation is transformed into a textual script that represents the policy, and the script is displayed as the user works with the logical representation. When the policy representation is saved, the script is translated into machine instructions that govern the operation of a network gateway or firewall.
17.	**Graphical Password Authentication Using Cued Click Points**	Chiasson, S., Van Oorschot, P. C., & Biddle, R. (2007, September). Graphical password authentication using cued click points. In European Symposium on Research in Computer Security (pp. 359-374). Springer, Berlin, Heidelberg.	We propose and examine the usability and security of Cued Click Points (CCP), a cued-recall graphical password technique. Users click on one point per image for a sequence of images. The next image is based on the previous click-point. We present the results of an initial user study which revealed positive results. Performance was very good in terms of speed, accuracy, and several errors. Users preferred CCP to PassPoints (Wiedenbeck et al., 2005), saying that selecting and remembering only one point per image was easier and that seeing each image triggered their memory of where the corresponding point was located. We also suggest that CCP provides greater security than PassPoints because the number of images increases the workload for attackers.
18.	**Detection of keylogging software**	Xu, Y. (2010). U.S. Patent No. 7,823,201. Washington, DC: U.S. Patent and Trademark Office.	The detection hook function determines whether the request indicates that the hook procedure is keylogger software. If so, an action is taken such as denying the request or alerting the user. A detection hook function also intercepts a request to remove a hook procedure. A dynamic detection function intercepts a call to a hook chain function attempting to pass an event to a hook procedure.
19.	**A Robust Technique of Anti- Key-logging using Key-Logging Mechanism**	Baig, M. M., & Mahmood, W. (2007, February). A robust technique of anti-keylogging using keylogging	Key-loggers have gained so much supremacy in their execution that they have become serious intimidation to the privacy and security of a computer. The fact which makes the key-loggers more perilous is their undetectable nature

		mechanism. In 2007 Inaugural IEEE- IES Digital Ecosystems and Technologies Conference (pp. 314 318). IEEE.	against anti-virus and spy-where applications. This paper discusses some existing techniques of fortification against key loggers and also exemplifies a new technique along with its proven advantages
20.	**A Novel Friendly Jamming Scheme in Industrial Crowdsensing Networks against Eavesdropping Attack**	Li, X., Wang, Q., Dai, H. N., & Wang, H. (2018). A novel friendly jamming scheme in industrial crowdsensing networks against eavesdropping attack. Sensors, 18(6), 1938.	An eavesdropping attack is one of the most serious threats in industrial crowdsensing networks. In this paper, we propose a novel anti-eavesdropping scheme by introducing friendly jammers to an industrial crowdsensing network. In particular, we establish a theoretical framework considering both the probability of eavesdropping attacks and the probability of successful transmission to evaluate the effectiveness of our scheme. Our framework takes into account various channel conditions such as path loss, Rayleigh fading, and the antenna type of friendly jammers. Our results show that using jammers in industrial crowdsensing networks can effectively reduce the eavesdropping risk while having no significant influence on legitimate communications
21.	**Keylogging-Resistant Visual Authentication Protocols**	Nyang, D., Mohaisen, A., & Kang, J. (2014). Keylogging-resistant visual authentication protocols. IEEE Transactions on Mobile Computing, 13(11), 2566-2579.	In this paper, we demonstrate how careful visualization design can enhance not only the security but also the usability of authentication. To that end, we propose two visual authentication protocols: one is a one-time password protocol, and the other is a password-based authentication protocol. Through rigorous analysis, we verify that our protocols are immune to many of the challenging authentication attacks applicable in the literature. Furthermore, using an extensive case study on a prototype of our protocols, we highlight the potential of our approach for real-world deployment: we were able to achieve a high level of usability while satisfying stringent security requirements.
22.	**Improved keylogging and shoulder-surfing resistant visual two-factor authentication protocol**	Khedr, W. I. (2018). Improved keylogging and shoulder-surfing resistant visual two-factor authentication protocol. Journal of Information Security and Applications, 39, 41-57.	Due to their high user convenience, the password is the most widely used means of authentication. However, passwords are vulnerable to compromise by disclosure using various forms of information tapping like Keylogging, phishing attacks, human shoulder-surfing, and camera-based recording. In this paper, the deficiencies of the original scheme are demonstrated, then a two-factor authentication scheme that eliminates

			these deficiencies is presented. A prototype of the proposed scheme is implemented and a secured virtual on-screen keyboard (SVOSK) comprising a dynamic emoticon keyboard layout is also proposed. Formal security proof and usability analyses show that the proposed scheme is secure, efficient, and has a high level of usability.
23.	**Visual Authentication Using QR Code to Prevent Keylogging**	Divya, R., & Muthukumarasamy, S. (2015). Visual authentication using QR code to prevent keylogging. International Journal of Engineering Trends and Technology, 20(3), 149-154.	To prevent keylogging, strict authentication is required. The QR code can be used to design visual authentication protocols to achieve high usability and security. The two authentication protocols are Time based One-Time-Password protocol and the Password-based authentication protocol. Through accurate analysis, the protocols are proved to be robust to several authentication attacks. And also, by deploying these two protocols in real-world applications especially in online transactions, the strict security requirements can be satisfied.

Overall Architecture

Register:

- Input username, email, and password. We have included the following constraints on the user details.
    - The username should be unique
    - Password must be an 8-digit number

  Eg. Username: amanxanand

  Password: 12345678

  Email Id: a.anand2k19@gmail.com

- Check if entered details are valid.

- Convert password into a string. Following is the algorithm to do that
    - Divide the 8-digit number into four, two-digit numbers.
    - E.g. 12345678 = [12, 34, 56, 78]

- ○ Convert each chunk into corresponding character to generate a string of four characters. Below is the list of conversions:
- ○ [0-5: ‘a’, 6-11: ‘b’, 12-17: ‘c’, 18- 23: ‘d’, 24-29: ‘e’, 30-35: ‘f’, 36-41: ‘g’, 42-47: ‘h’, 48-53: ‘i’, 4-59: ‘j’, 60-65: ‘k’, 66-71: ‘l’, 72-77:‘m’, 78-83: ‘n’, 84-91: ‘#’, 91-99: ‘*’]
    - ○ Therefore: 12345678 = [12, 34, 56, 78] = ‘cfjn’
- Compute every permutation of the string
    - ○ [‘cfjn’, ‘fcjn’, ‘jcfn’, ‘cjfn’, ‘fjcn’, ‘jfcn’, ‘nfcj’, ‘fncj’, ‘cnfj’, ‘ncfj’, ‘fcnj’, ‘cfnj’, ‘cjnf’, ‘jcnf’, ‘ncjf’, ‘cnjf’, ‘jncf’, ‘njcf’, ‘njfc’, ‘jnfc’, ‘fnjc’, ‘nfjc’, ‘jfnc’, ‘fjnc’]
    - ○ Total 4! = 24 permutations.

- Generate a hash of each permutation using any hashing algorithm. We have used the MD5 hash algorithm in this project.

  [‘501a93259d8f00aa7e4c2e9eda8e561e’, ‘7edea98f68d1d1e69ba39b6c6788dc45’, ‘2b5444b24e15b955a7455fcd49d7b897’, ‘c3dff954f4b96447ae2dfa96b95d0c17’, ‘99dde74f2613c8a304e0d02a8aeb5907’, ‘f46b40c76c99accb81cee28e01c7af4f’, ‘1ebe262273b1d93cb165726b19051b1b’, ‘439acea80e1d0124b1caf7fdbebc690f’, ‘ff9f526a2d7aed1da71d767d41445409’, ‘5d8999eb19e557dd865d4e5114fb06b9’, ‘cf83685381e1384c721ad9fd1a4879a4’, ‘b5b5e2088cd03f62b3a1d34e12568353’, ‘ac670e2a0e7de505f6f6b02d0949b966’, ‘f5e56809a788ec7baac614b21e6ac6ea’, ‘5398b787cb9e80c2c29c171d7c5f531f’, ‘a7a290d301d09f8aab46d440c1c0bc45’, ‘0f8698e809e4205ae5441f94dfa4b00d’, ‘b6b66056736b722b5e512f0492e1e36e’, ‘3dd8d0eb7c4d67179dd1c22c3d777a8e’, ‘12008658a74e6974ba34ddda2f2eda0a’, ‘3353f6fc45c701c655ad648ef7857815’, ‘545d3e0ae7566fb264df9d02f6cf49d6’, ‘097ee3281992b6f025cbc9a3f5e3b973’, ‘3475bb0f27f3a9614a79eae11cec1cf8’]

- Save the username, email, and each hash in the database.
- Send a mail to the user on the registered email-id informing the user about the process on how to login to the system.

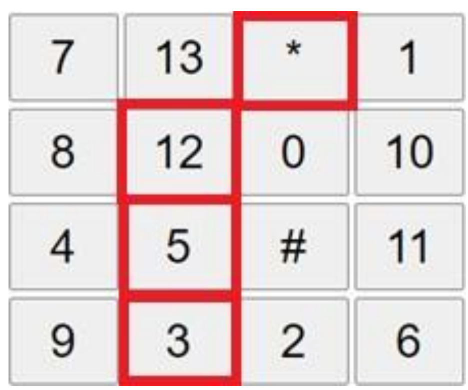
The way it works is

The Password that was registered is **NOT** the one used to log in to the system. Instead, the password is a string generated by the four-button clicks as shown below.

First, the registered mobile number is entered. Then a question is being sent to the registered mobile device, and there the pattern for that login session is shown

On the main login window, the Username is the one you registered with. For passwords - Two patterns are incorporated in our login system:

For vertical pattern, any permutation of these four buttons is the password for a particular login session



For a Horizontal pattern, any permutation of these four buttons is the password for a particular login session.

Click Submit and welcome to the DASHBOARD.

Login Module:

The architecture of Security:

There are three levels of security in this project. All of these levels of work in a cascading manner as described below:

First layer: The user will enter the Phone Number of the mobile device, which is registered, into the system. If the phone number is not registered, the process will not proceed.

Second Layer: If the Phone number entered is registered, then verification will be processed on the registered device. This way provides two additional sub-level of security and an additional feature.

1. As only the genuine user will have the registered mobile device, we will be sure that it is the real user and not the attacker who is accessing the system.

2. In case, the registered mobile device gets lost. What a normal user does is deactivate the SIM card and device remotely. If that happens, the Verification question will not be sent on the device and the attacker will not be able to reach the final level of Login.

3. Additional feature: The loophole in the Static Login system is that what if the attacker recognizes and remembers the pattern. The obvious solution is changing the pattern in each Login Session. That gives rise to another problem. We need to inform the user about that current session pattern. We cannot flash it directly onto the screen. When the user gets the verification question on the registered mobile device, there will also be an additional piece of information telling the user about the pattern distribution. The user remembers this distribution and enters the password following that.

Third Layer: The user will be asked to enter the details – Username and password through a unique input system. The user can enter anyone's password from the set as a code word but actually, it is a pattern. For demo purposes, we included only two patterns: Horizontal distribution and Vertical distribution, as shown below:

R1C1 (a) **0**	R1C2 (b) **1**	R1C3 (c) **2**	RC14 (d) **3**
R2C1 (e) **4**	R2C2 (f) **5**	R2C3 (g) **6**	R2C4 (h) **7**
R3C1 (i) **8**	R3C2 (j) **9**	R3C3 (k) **10**	R3C4 (l) **11**
R4C1 (m) **12**	R4C2 (n) **13**	R4C3 (#) **#**	R4C4(*) *****

Horizontal Distribution

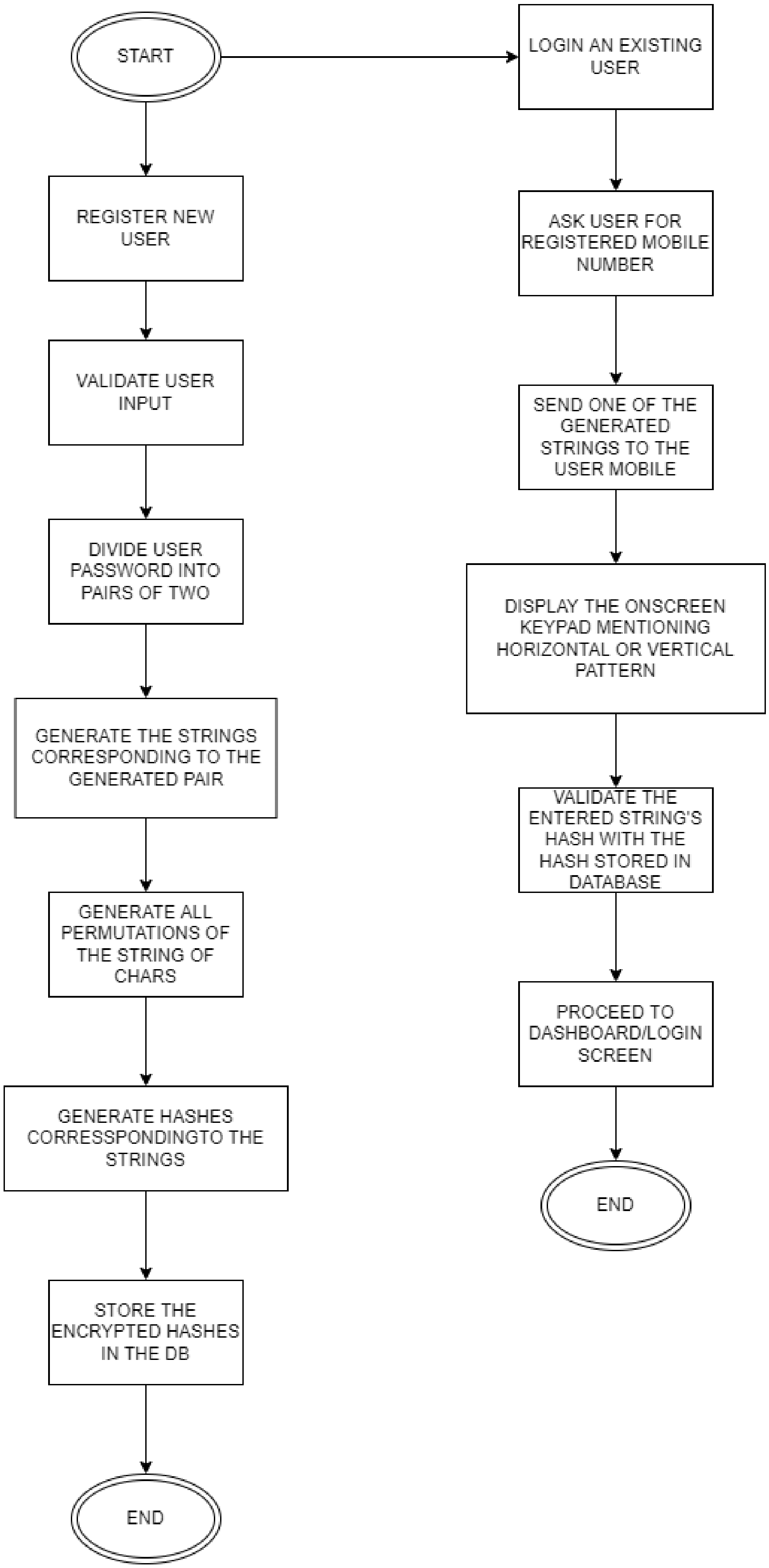
R1C1 (a) **0**	R1C2 (b) **4**	R1C3 (c) **8**	RC14 (d) **12**
R2C1 (e) **1**	R2C2 (f) **5**	R2C3 (g) **9**	R2C4 (h) **13**
R3C1 (i) **2**	R3C2 (j) **6**	R3C3 (k) **10**	R3C4 (l) **#**
R4C1 (m) **3**	R4C2 (n) **7**	R4C3 (#) **11**	R4C4(*) *****

Vertical Distribution

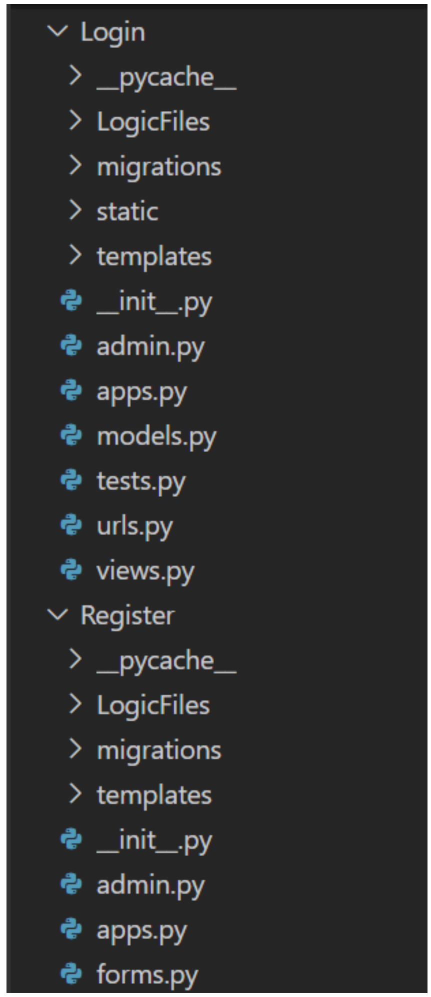
Now,

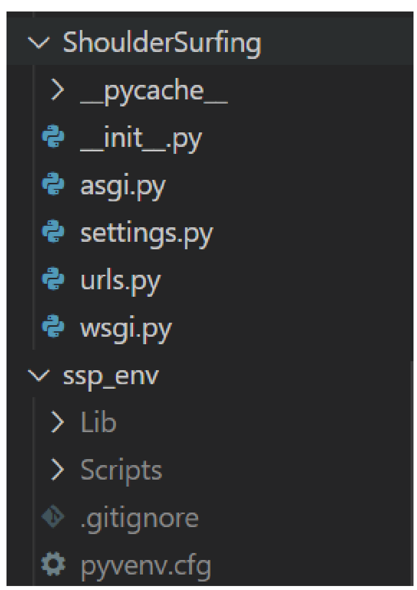
- Take an array of characters to display on the input buttons.
- Generate the pattern from this array as required.
- Check the database for the existing pattern.
- If the pattern matches: Login Successful
- Next Login Session, Change the display values of the grid.

Proposed Methodology



File structure:

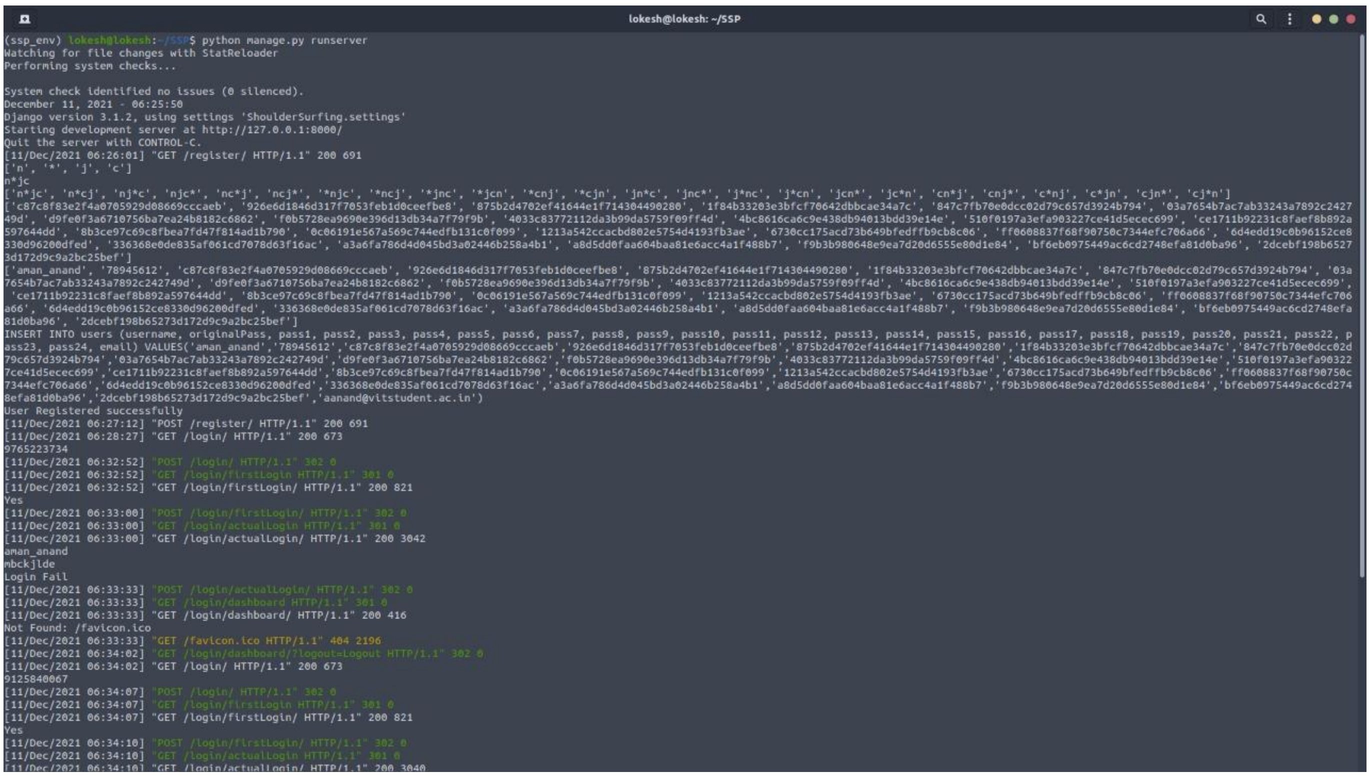




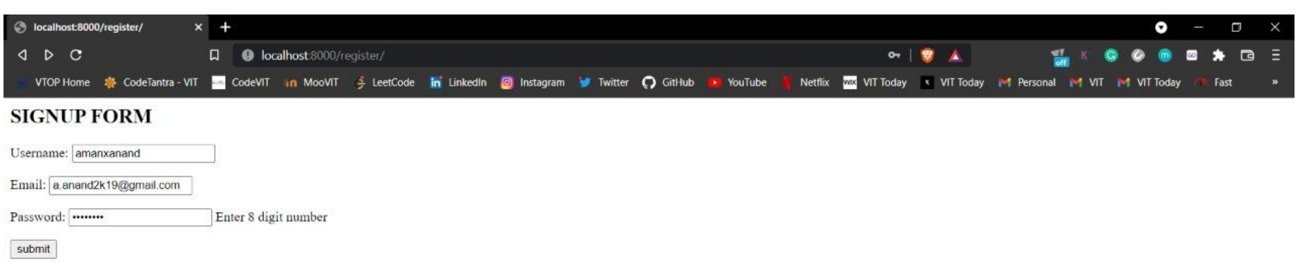
Working of the system:

This project is based on python. We have used a Django-based web app to implement the frontend mechanism. The project works in the following manner:

First, we run the python-based server on Django. Make sure Apache server and MySQL are running on Xampp.

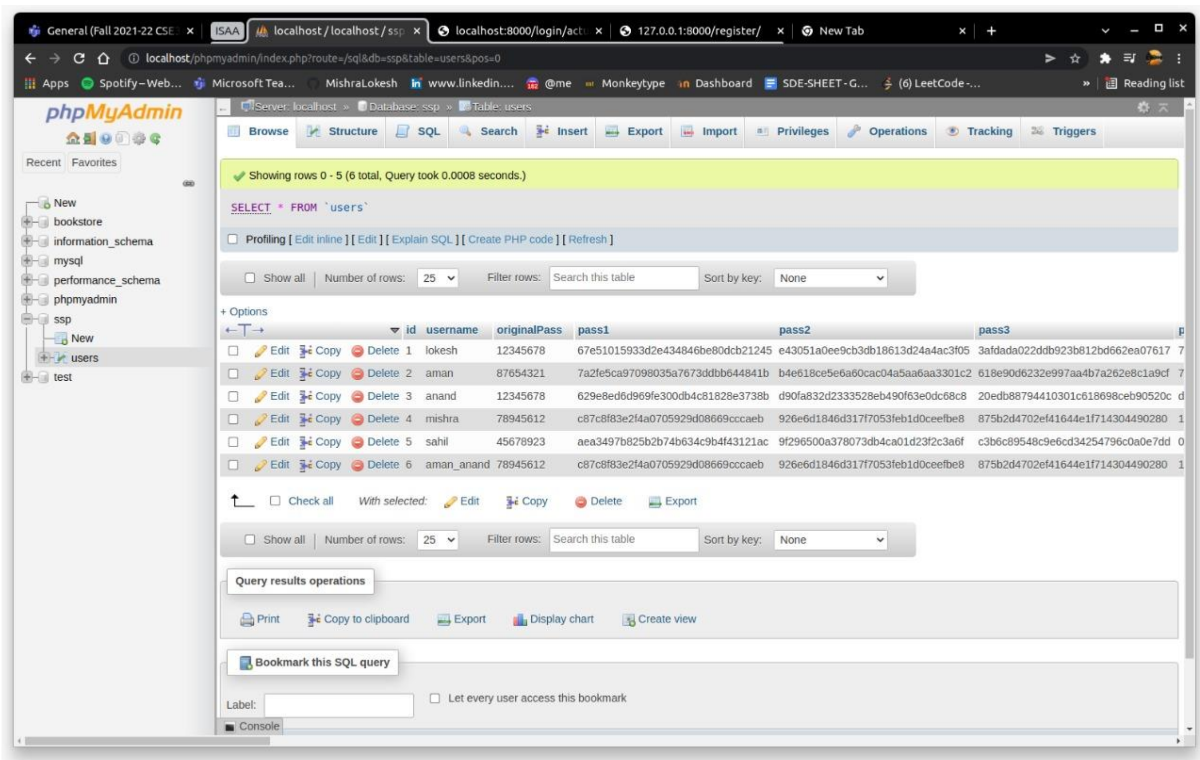


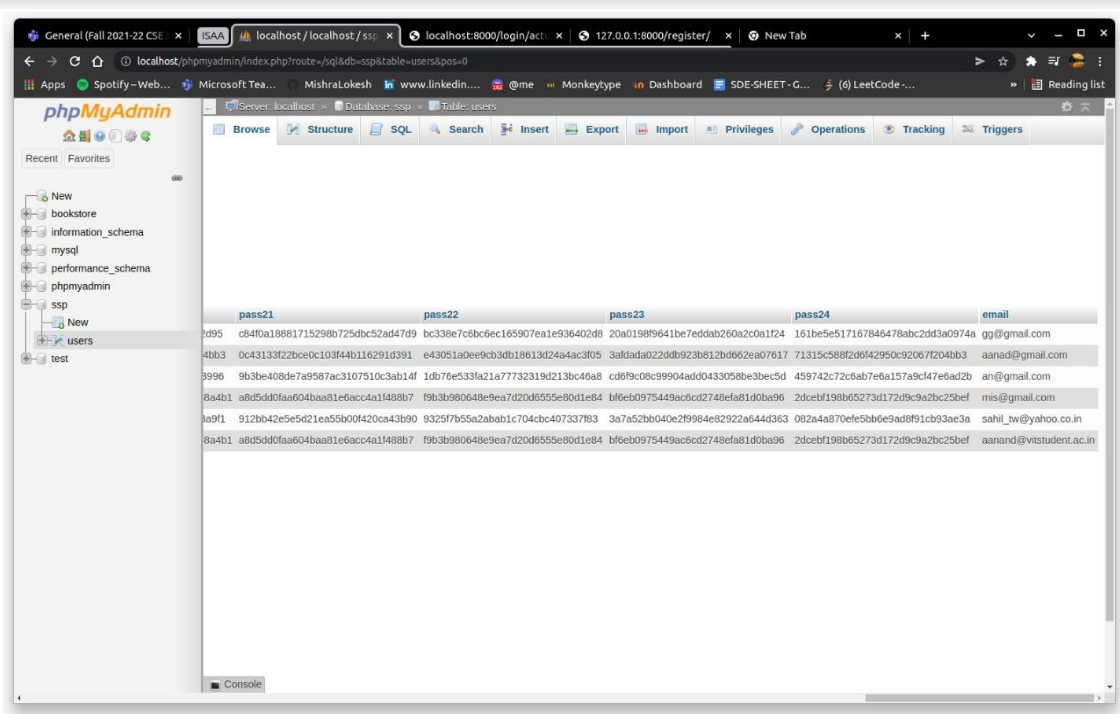
Now, we open the registration page. This is where the user will register for the portal. Here they will set an 8 digit number as their password. For the sake of demonstration, the password set here is *12345678*.



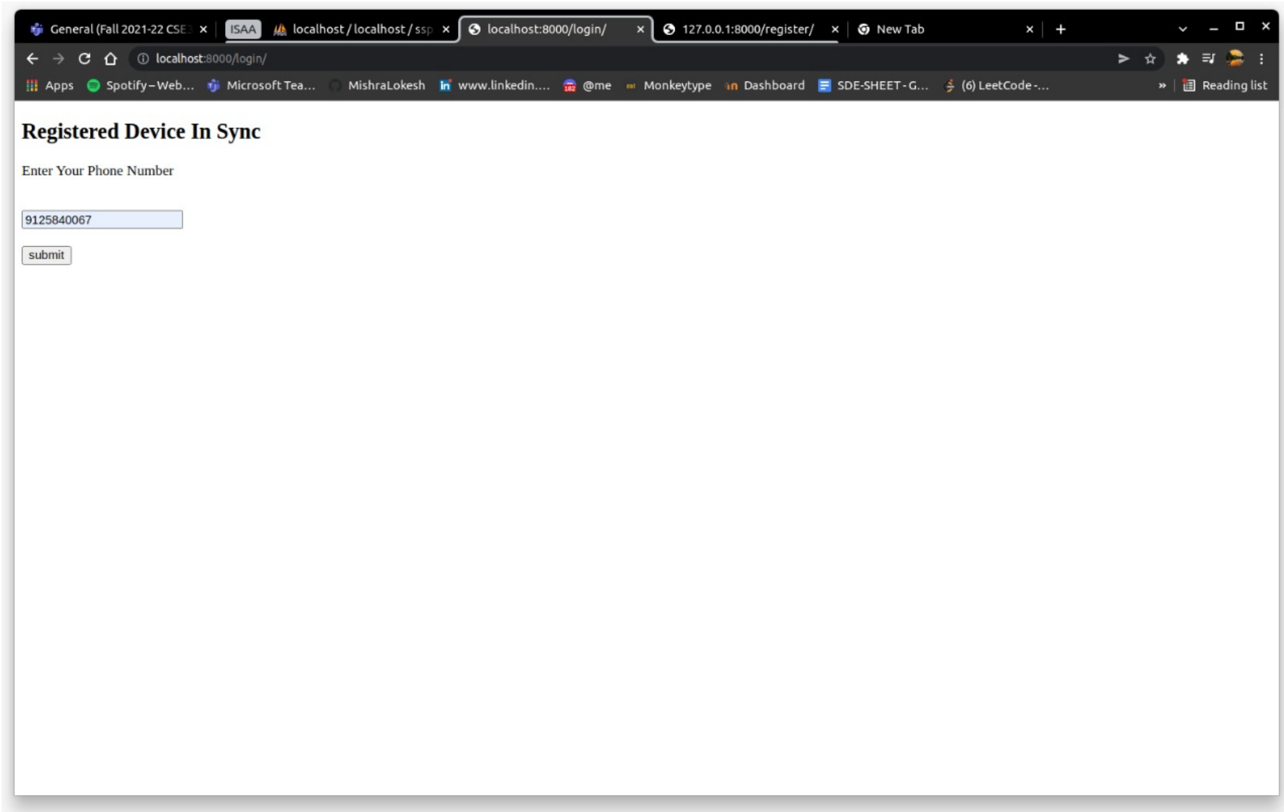
As we can see in our database this account has been registered. The database stored the username, original password, and the 24 hashes of the code that will be their actual password. The user can use any one of those 24 codes.

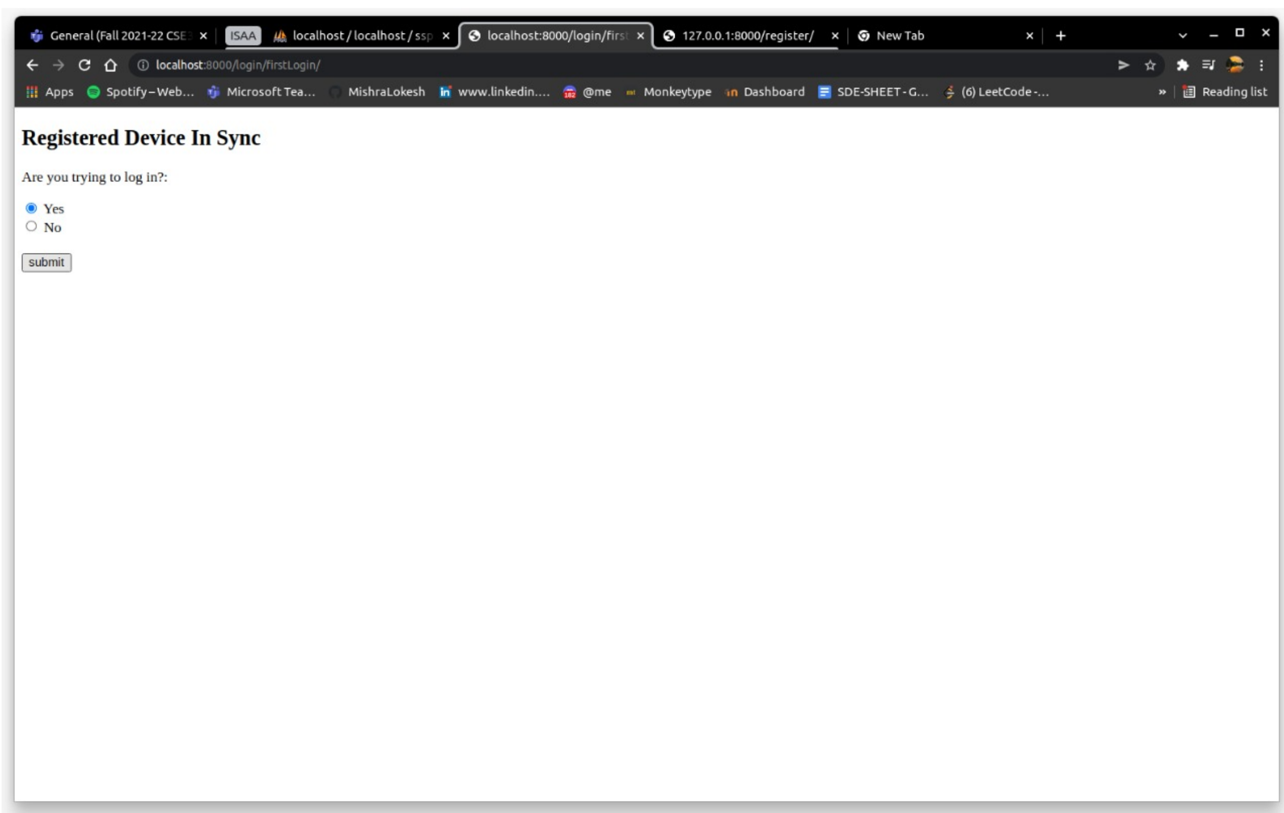
It should be noted that the password will be condensed to a 4 digit code. So even if the database is attacked, the attacker would not be able to log in with the original 8 digit password as the log-in field will only accept a 4 digit code.





Now when the user goes to log in, they have to provide a phone number, on which device a verification will be done as follows:



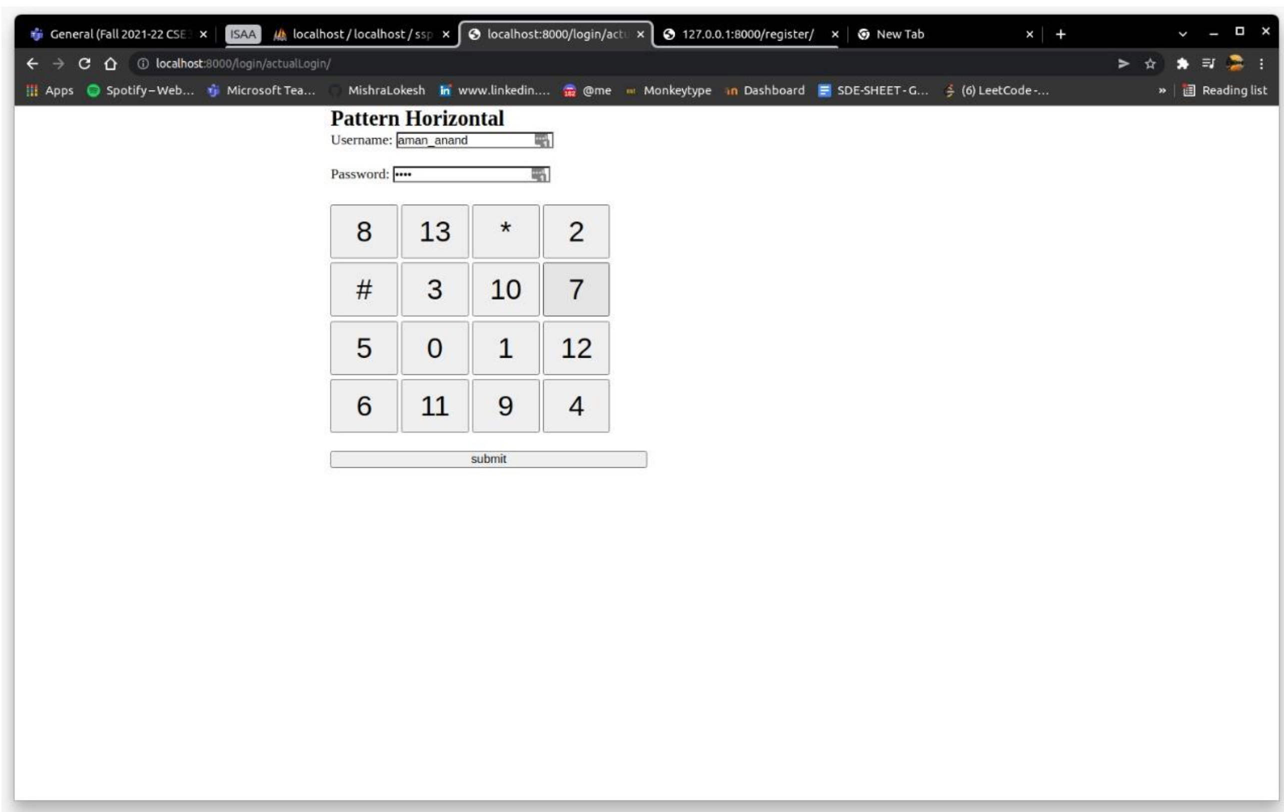


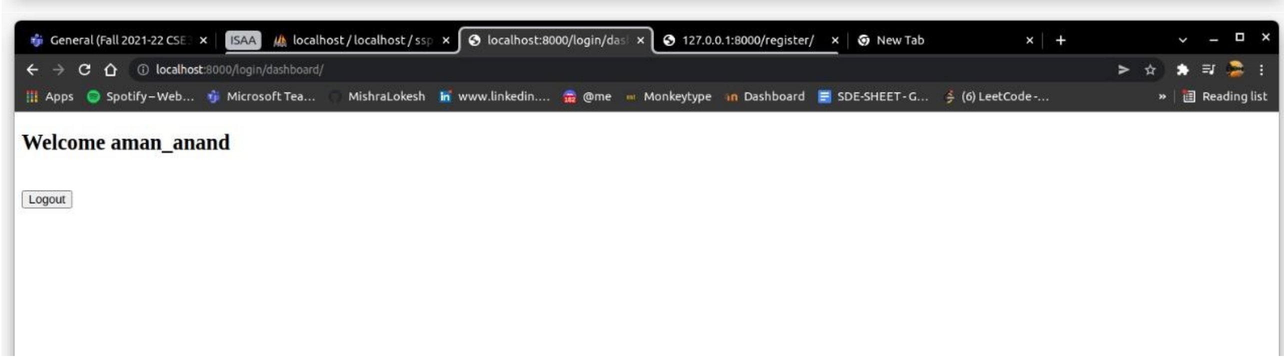
After this step, the user will be asked to give their password. Before this, when they register the user will receive a mail (as per in the proposed system) in which they will be informed about their actual passwords and how to use them.

As you can see below, using registering, the user's codes were generated in the backend. The user can now use any of these 24 codes to log in.

```
['c', 'f', 'j', 'n']
cfjn
['cfjn', 'cfnj', 'cjfn', 'cjnf', 'cnfj', 'cnjf', 'fcjn', 'fcnj', 'fjcn', 'fjnc', 'fncj', 'fnjc', 'jcfn', 'jcnf', 'jfcn',
 'jfnc', 'jncf', 'jnfc', 'ncfj', 'ncjf', 'nfcj', 'nfjc', 'njcf', 'njfc']
```

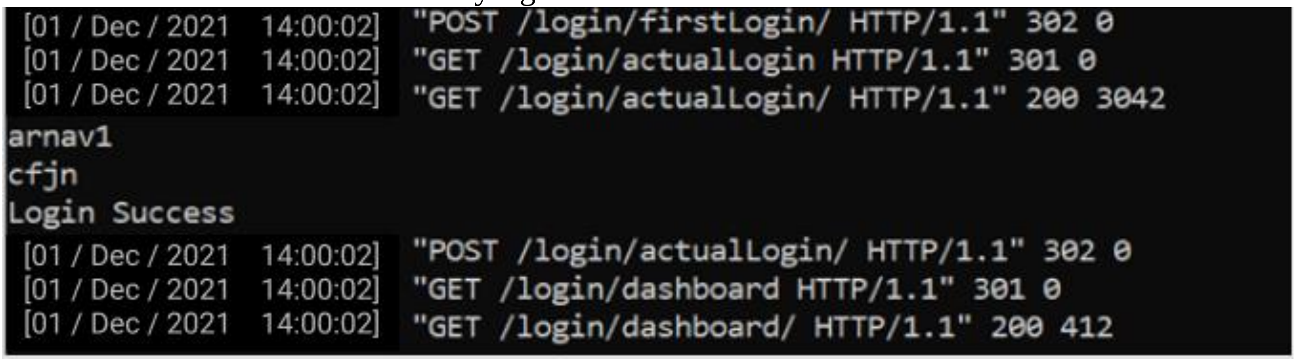
The user will also be notified about the pattern distribution being followed, via mobile. But for demonstration purposes, we have displayed it on the screen (Horizontal in this example).





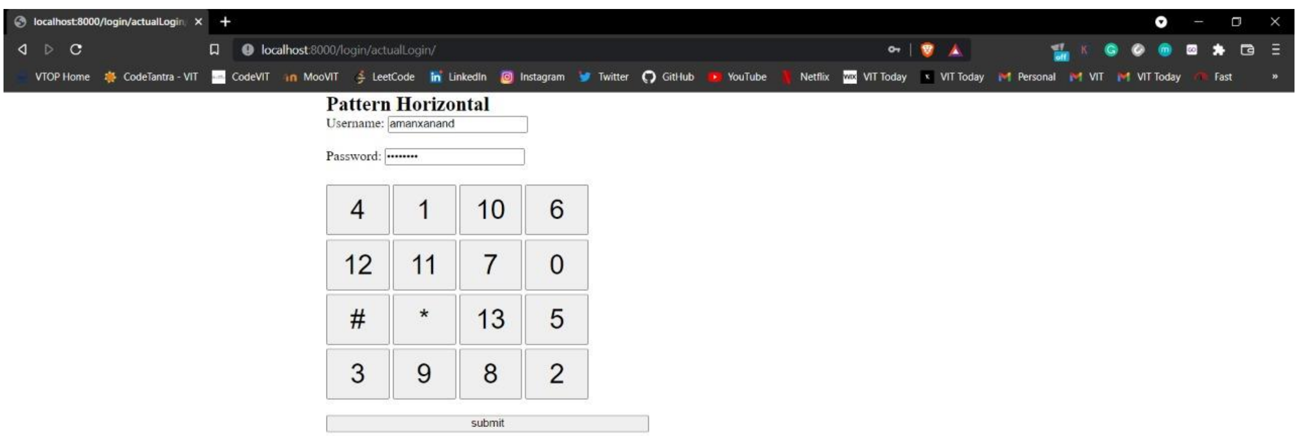
Results

So the user was able to successfully sign in as shown below:



When we put the password as “cfjn”, a shoulder surfing attacker will think it is: 0 5 9 10 (as an example), as can be seen above. As per human psychology, the attacker will remember the password as 0 5 9 10, not knowing that the actual password is the pattern in which the numbers were typed.

So when they try to log in using the password they know in the following keypad:



Their login would fail as they would be putting the wrong pattern, dissimilar to as shown below:

```
[01 / Dec / 2021   14:00:02]  "POST /login/firstLogin/ HTTP/1.1" 302 0
[01 / Dec / 2021   14:00:02]  "GET /login/actualLogin HTTP/1.1" 301 0
[01 / Dec / 2021   14:00:02]  "GET /login/actualLogin/ HTTP/1.1" 200 3042
arnav1
cfjn
Login Success
[01 / Dec / 2021   14:00:02]  "POST /login/actualLogin/ HTTP/1.1" 302 0
[01 / Dec / 2021   14:00:02]  "GET /login/dashboard HTTP/1.1" 301 0
[01 / Dec / 2021   14:00:02]  "GET /login/dashboard/ HTTP/1.1" 200 412
```

So, the results obtained show that a shoulder surfing attack has been prevented successfully as the attacker will not be able to recognize the fact that the password is not the numbers, but the pattern in which the numbers have been typed.

Analysis

1. Analysis of the results obtained

Since the major aim of this project was to use an algorithm for passwords that can neither be understood by a hacker, a mouse heatmap, etc nor by a person standing beside you trying to see the password we thought it would be best to ask people via screen share if they understood what the password is.
No one could log in to our account even when they saw the password being typed. We believe this could be an important password input mechanism in the future

2. Comparison of the obtained results with the already existing results.

Since this is a novel idea we could not find any such ideas done before

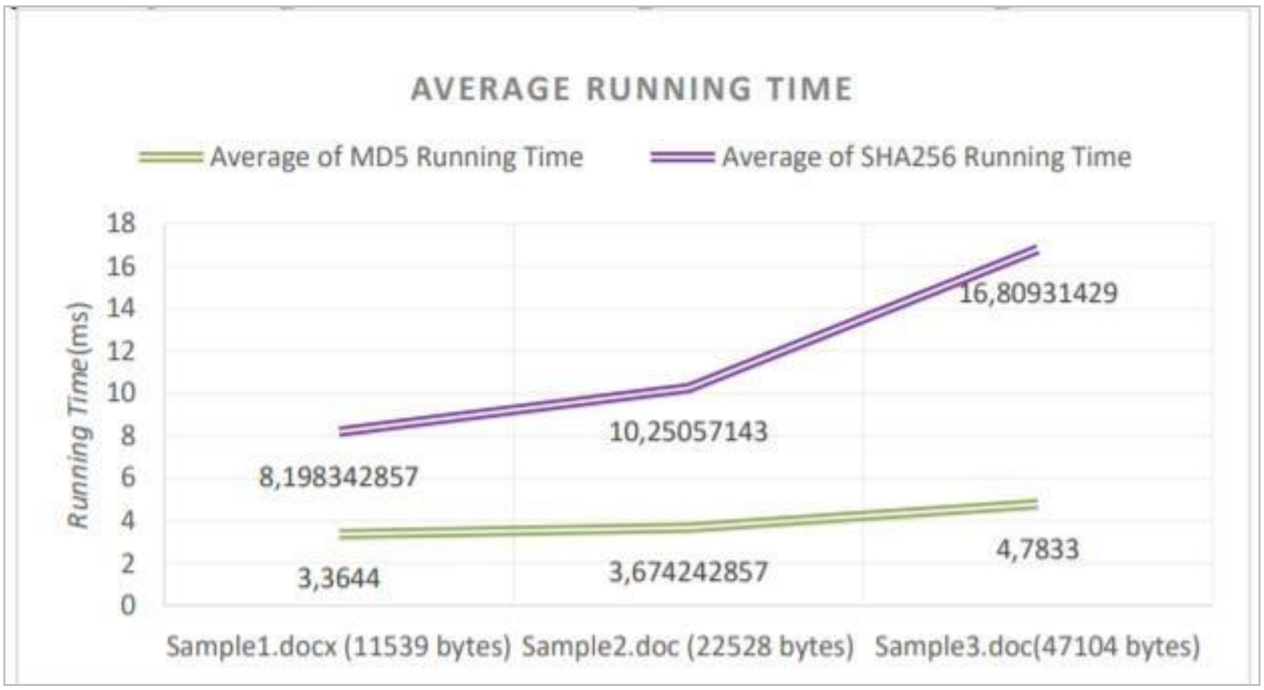
3. The efficiency was obtained along with the metrics used.

This project is not that algorithmically complex and the only encryption essentially being used is the hashing algorithm

- **Case 1** – 36 characters length string, UUID is cached
- **Case 2** – 49 characters length string, UUID is cached and system timestamp is calculated each iteration
- **Case 3** – 49 characters length string, new UUID is generated on each iteration and system timestamp is calculated each iteration
- **Case 4** – 72 characters length string, UUID is cached
- **Case 5** – 85 characters length string, UUID is cached and system timestamp is calculated each iteration
- **Case 6** – 85 characters length string, new UUID is generated on each iteration and system timestamp is calculated each iteration
- SHA-256 is faster with 31% than SHA-512 only when hashing small strings. When the string is longer SHA-512 is faster with 2.9%.
- The time to get the system timestamp is ~121.6 ms per 1M iterations.
- The time to generate UUID is ~670.4 ms per 1M iterations.

- SHA-1 is the fastest hashing function with ~587.9 ms per 1M operations for short strings and 881.7 ms per 1M for longer strings.
- MD5 is 7.6% slower than SHA-1 for short strings and 1.3% for longer strings.
- SHA-256 is 15.5% slower than SHA-1 for short strings and 23.4% for longer strings.
- SHA-512 is 51.7% slower than SHA-1 for short strings and 20% for longer.

We can see that SHA 1 and MD5 are the fastest. Although both are much weaker than SHA 256



Conclusion and Future Work

In the duration of this project, we have formulated a system that helps in preventing shoulder surfing attacks on users while they log in to their accounts in public places. In our project, the user signs up with an 8 digit password which is converted to 4 digit code, whose permutations are stored in the database. Then the user is notified about the way their new log-in works and are told the actual codes via mail. It is interesting to note that this system is also prone to attacks on the database and key loggers as the password is shown in the database is not the actual password, nor is the data that is entered as the password. It has also been demonstrated how a shoulder surfer would fail at logging and an analysis of the results has been prevented.

The project can be extended to add various other features.

- The project can be incorporated with actual apps to see how it works in real-time.
- For demonstration, only two patterns are included in it, but multiple patterns can be added to increase the complexity but the number of patterns shall not be too high, as it may overwhelm and confuse the user more than it can be useful.
- The UI of the system can be improved to make it more interactive and this may reduce the cognitive load on users while using the system.
- We can use another layer of encryption like RSA or ECC before hashing the passwords. We can also use newer and more secure hashing algorithms like SHA256.

References

1. "**Password Encryption Key**" by Utin.
2. "**Two-factor graphical password for text password and encryption key generation**" by Gyorffy, J. C., & Miller.
3. "**Password-Based Encryption Analyzed**" by Abadi, M., & Warinschi.
4. "**Encryption method and system for portable data**" by Hardy, D. A., Fossey, C. R., Balogh, C. R., & Tugenberg.
5. "**Password Encryption for hybrid cloud services**" by Singleton IV, L. C., & Cooper.
6. "**Securely generating a computer system password by utilizing an external encryption algorithm**" by Angelo.
7. "**Security system that uses indirect password-based encryption**" by Gutnik, Y.
8. "**A PIN-entry method resilient against shoulder surfing**" by Roth, V., Richter, K., & Freidinger.
9. "**A Shoulder-Surfing Resistant Graphical Password Scheme**" by Man, S., Hong, D., & Matthews.
10. "**An Association-Based Graphical Password Design Resistant to Shoulder-Surfing Attack**" by Li, Z., Sun, Q., Lian, Y., & Giusto.
11. "**A Shoulder Surfing Resistant Graphical Authentication System**" by Sun, H. M., Chen, S. T., Yeh, J. H., & Cheng.
12. "**Novel Shoulder-Surfing Resistant Haptic-based Graphical Password**" by Malek, B., Orozco, M., & El Saddik.
13. "**A simple text-based shoulder surfing resistant graphical password scheme**" by Chen, Y. L., Ku, W. C., Yeh, Y. C., & Liao.
14. "**Integrated network security system**" by Weiss.
15. "**Network Security and Surveillance System**" by Trcka, M. V., Fallon, K. T., Jones, M. R., & Walker.
16. "**Graphical network security policy management**" by Wiegel.
17. "**Graphical Password Authentication Using Cued Click Points**" by Chiasson, S., Van Oorschot, P. C., & Biddle.
18. "**Detection of keylogging software**" by Xu.
19. "**A Robust Technique of Anti-Key-logging using Key-Logging Mechanism**" by Baig, M. M., & Mahmood.

20. “**A Novel Friendly Jamming Scheme in Industrial Crowdsensing Networks against Eavesdropping Attack**” by Li, X., Wang, Q., Dai, H. N., & Wang.
21. “**Keylogging-Resistant Visual Authentication Protocols**” by Nyang, D., Mohaisen, A., & Kang
22. “**Improved keylogging and shoulder-surfing resistant visual two-factor authentication protocol**” by Khedr
23. “**Visual Authentication Using QR Code to Prevent Keylogging**” by Divya, R., & Muthukumarasamy

Appendix

Work done by every individual student.

Aman Anand (19BCE0521): Code Implementation and documentation

Lokesh Mishra (19BCE2672): Research and documentation