APPENDIX-I

Password Authentication and Strength Checking System

A PROJECT REPORT

Submitted By

RAJENDRA KISHOR	10000220033
SOUMIL KONAR	10000220045
SOUMYA SINGH	10000220046
SRISTY KUMARI	10000220049
TAMANNA SULTANA	10000220053

Under the Guidance of

Dr. Nabanita Ganguly

In the partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

NH-12, Haringhata, Post Office- Simhat, Police Station- Haringhata, W.B - 741249

APPENDIX-II

BONAFIDE CERTFICATE

Certified that this Project Report "PASSWORD AUTHENTICATION AND STRENGTH CHECKING SYSTEM" is the Bonafide work of RAJENDRA KISHOR, SOUMIL KONAR, SOUMYA SINGH, SRISTY KUMARI and TAMANNA SULTANA who carried out the project work under my supervision

SIGNATURE

DR. SOMDATTA CHAKROVARTY
HEAD OF THE DEPARTMENT
Department of Information Technology
Maulana Abul Kalam Azad University of Technology, W.B.

SIGNATURE

DR. NABANITA GANGULY
SUPERVISOR
ASSISTANT PROFESSOR
Department of Information Technology
Maulana Abul Kalam Azad University of Technology, W.B.

ACKNOWLEDGEMENT

First and foremost, we would like to express my earnest gratitude and heartfelt indebtedness to my advisor, Dr. Nabanita Ganguly, Department of Information Technology, for the privilege and the pleasure of allowing me to work under her towards my Degree of Bachelor of Information Technology. This work would not have been materialized, but for her whole-hearted help and support. Working under her has been a great experience. We sincerely thank my supervisor, particularly for all the faith he had in me. We are thankful to Dr. Somdatta Chakrovarty who has acted as Head of the Department of Information Technology during the tenure of my studentship. We would also like to show my gratitude to the respected professors of the Department of Information Technology for their constant guidance and valuable advice.

Rajendra Kishor	Roll No.:10000220033
Soumil Konar	Roll No.: 10000220045
Soumya Singh	Roll No.:10000220046
Sristy Kumari	Roll No.:10000220049
Tamanna Sultana	Roll No.:10000220053

APPENDIX-III

CONTENTS

ABSTRACT - Passwords are a way to authenticate the user and are meant to gain access to specific data or a protected account. If the password is not strong enough then it increases the likelihood of unauthorized individuals or hackers guessing or cracking the password. Once the password is compromised, attackers can gain unauthorized access to user accounts. The risk of data breaches becomes more significant. In this project report we focused on exploring the design for password strength testers and implemented verification and user-friendly password management, allowing users to create strong passwords. We implemented verification mechanisms and user-friendly password management solutions. To enhance security beyond passwords, we also examined the implementation of two-factor authentication (2FA). By requiring an additional authentication factor, such as a mobile-generated OTP, we provide an extra layer of protection. This dual-layer approach ensures that even if passwords are compromised, unauthorized access can be effectively prevented.

1. INTRODUCTION

In an increasingly digitized world, where our personal and confidential information is often stored and accessed online it is important to protect our data from unauthorized access. The most commonly used methodology for achieving this is to use a password as a part of online access process.

Hackers are well aware that a significant number of passwords are inadequately designed, making them susceptible to attack. As long as passwords continue to be utilized, password attacks will persist as a prevalent method employed by hackers to compromise accounts and systems.

With the rising frequency of data breaches and cyber threats, it has become imperative for individuals and organizations to prioritize the security of their online accounts. To enhance security, it is generally recommended to create strong, complex passwords that are not easily guessable. A strong password generally involves a fusion of uppercase and lowercase letters, numbers, and special characters. Length is also important, as longer passwords are generally harder to crack. Additionally, using a unique password for each online account and periodically updating passwords can further enhance security.

Weak passwords, such as those that are easily guessable or commonly used, pose a significant risk to the integrity and confidentiality of sensitive information. By designing a password strength tester, we aim to provide users with a means to assess the effectiveness of their passwords and make informed decisions to fortify their online security. The password strength tester will employ advanced algorithms and techniques to analyse various elements that contribute to password strength, including complexity, length, and uniqueness. It will then generate a comprehensive evaluation, offering users valuable feedback on the strengths and weaknesses of their chosen passwords. By highlighting vulnerabilities and suggesting improvements, the tester will empower users to create stronger and more resilient passwords.

Traditional single-factor authentication methods, primarily relying on passwords, are no longer sufficient to protect against unauthorized access. Two-Factor Authentication (2FA) provides an additional layer of security by requiring two forms of verification, significantly enhancing the protection of sensitive information.

Our project aims to address these vulnerabilities by developing a comprehensive solution that combines a password strength tester with an advanced authentication system and improved encryption techniques. This report outlines the objectives, methodology, implementation, and potential impact of our project in enhancing cybersecurity.

2. LITERATURE SURVEY

Password attacks are a prevalent method used by cybercriminals to gain unauthorized access to corporate and personal accounts by attempting to obtain or crack the passwords. Passwords are a way to authenticate the user and are meant to gain access to specific data, an account, or a protected space. [2] It is a secret character string only the user knows and its hashed code is stored on the server that provides access to the data. When the user requests for data access, he enters the password along with other identification information, such as user name or email.[1] The key to a powerful password is length and complexity. While setting a password for protected data one should keep in mind that Using common words, sequential numbers, or easily obtainable personal information as part of a password makes it easier for automated tools and algorithms to guess or crack the password. Additionally, if an attacker knows or can easily find personal information about the user, they can exploit that information to guess the password more easily.[2]

The objective of a password attack is to exploit weak or compromised passwords to gain entry into an individual's or an organization's system. According to a survey. In 2020, 81% of data breaches were due to compromised credentials [5]. This is because passwords can only contain letters and numbers, passwords are becoming less safe.

The reason password attacks remain effective is due to various factors:

Weak Password Practices: Many individuals still choose passwords that are easy to guess or use common patterns. Passwords like "123456," "password," or personal information like birthdates or names are frequently employed, making it relatively easy for attackers to crack them.

Password Reuse: Password reuse across multiple accounts is a common practice, making it easier for attackers to gain access to multiple systems once one password is compromised. Cybercriminals often obtain passwords from breaches or leaks and test them across various platforms to exploit this behaviour.

Lack of Password Complexity Policies: Some systems or organizations may not enforce strong password requirements, allowing users to create weak passwords without any guidance. This oversight creates an opportunity for attackers to exploit vulnerabilities easily.

Limited User Awareness: Despite increased awareness of cybersecurity risks, many individuals still lack sufficient knowledge about password best practices. This lack of awareness leaves them more susceptible to falling victim to password attacks.

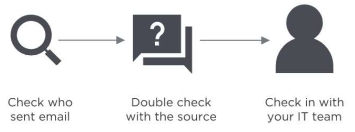
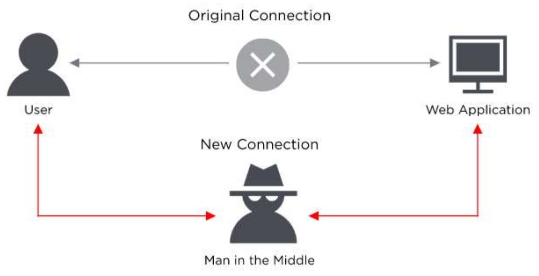
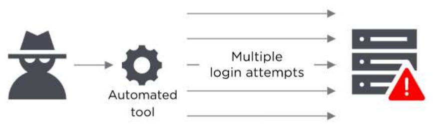
existing research tends to focus on specific attack methods, such as brute force attacks or dictionary attacks, which can uncover passwords with particular vulnerabilities. However, these studies often fail to quantify the strength of passwords that have not been compromised, especially against more sophisticated or general attack techniques. [4]

To address the persistence of password attacks, it is crucial for individuals and organizations to Educate User raise awareness about the importance of creating strong passwords, avoiding common pitfalls, and the risks associated with password reuse. This can be achieved by adding prompt on our password tester which would help the user to get an idea about how to frame a strong and unique password.

On the other hand, we can Enforce Strong Password Policies, implement strict password requirements that mandate the use of complex, unique, and regularly updated passwords. This can be achieved by enforcing minimum length, complexity, and expiration rules.

2.1 PASSWORD ATTACKS

Most common password attacks are as follows:[5]

ATTACK	METHOD	PREVENTION
PHISHING [5]	Phishing is a type of online scam or fraudulent activity where attackers try to trick people into sharing their sensitive information, such as usernames, passwords, credit card details, or personal identification numbers (PINs).	• **Check who sent the email** • **Double check with the source** • **Check in with your IT team**  Check who sent email Double check with the source Check in with your IT team
MAN-IN-THE-MIDDLE ATTACK [5]	In this attack, the attacker positions themselves between the sender and the receiver, secretly eavesdropping on their communication and potentially manipulating the data exchanged.	• **Enable encryption on your router.** • **Use strong credentials and two-factor authentication.** • **Use a VPN.**  Original Connection User Web Application New Connection Man in the Middle
BRUTE FORCE ATTACK [5]	In a brute force attack, the attacker employs automated software or scripts to generate and test an extensive range of passwords. The software starts with the simplest and most common passwords, such as "123456" or "password," and then progresses through all possible combinations, including different character types (lowercase letters, uppercase letters, numbers, and special characters) and various password lengths.	• Use a complex password. • Enable and configure remote access. • Require multi-factor authentication. Hackers likely won't have access to your mobile device or thumbprint, which means they'll be locked out of your account. 
DICTIONARY ATTACK [5]	In dictionary attack, the hackers gain access to unauthorised accounts by systematically trying a list of common words,	• Dictionary word should never be part of your password. • Lock accounts after too many password failures.

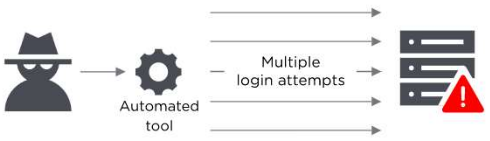
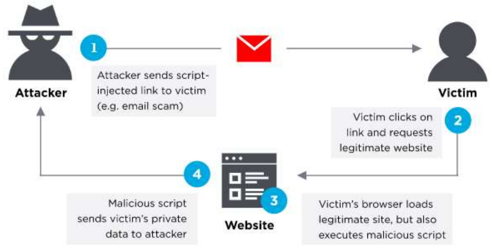
	phrases, or commonly used passwords as potential login credentials.	
CREDENTIAL STUFFING [5]	In a credential stuffing attack, hackers use automated scripts or tools to systematically input the stolen credentials into various websites, applications, or services. The attackers aim to identify accounts where users have reused their credentials.	• Monitor your accounts which would check whether your email address is connected to any recent leaks. • Regularly change your passwords. Monitor your accounts Regularly change your passwords Use a password manager
KEYLOGGERS [5]	Keyloggers are types of malicious software or hardware devices designed to monitor and record keystrokes made on a computer or mobile device without the user's knowledge or consent. They can be installed through various means, including phishing emails, malicious downloads, infected websites, or physical access to the device.	• Regularly inspect your computer and physical hardware.  Attacker 1 Attacker sends script-injected link to victim (e.g. email scam) Victim 2 Victim clicks on link and requests legitimate website 3 Website Victim's browser loads legitimate site, but also executes malicious script 4 Malicious script sends victim's private data to attacker

TABLE-01: Different types of attacks

From the data shown in the above table it is evident that there are already many techniques available in the market to intrude into someone's system and break security rules. So, it is not only the responsibility of the user to keep strong passwords in order to secure their information but also there is a need of adopting strong Encryption and Decryption mechanism working at the back to make the users data safe and secure. Encryption is a must for maintaining secure authentication and protecting sensitive data during the authentication process.

2.2 ENCRYPTION

Encryption is the process of converting data or information into a code to avoid unauthorized access. It is a fundamental technology that highlights many aspects of cybersecurity, that ensures the confidentiality, integrity, and authenticity of data.

Secrecy is the heart of cryptography. Encryption is a practical means to achieve information secrecy [a]. By converting data into a coded format, encryption ensures that only authorized parties with the correct decryption key can access the original information.

2.2.1 TYPES OF ENCRYPTIONS

(a) Symmetric Encryption

Symmetric encryption uses singular key for both encryption and decryption. It is efficient and suitable for encrypting large amounts of data [19].

Common symmetric encryption algorithms include:

Algorithm	Key Length	Block Size	Security Level	Performance	Use Cases
AES (Advanced Encryption Standard) [20]	128, 192, 256 bits	128 bits	High (considered very secure)	Fast and efficient	General-purpose encryption, SSL/TLS, VPNs, Wi-Fi security
DES (Data Encryption Standard) [20]	56 bits	64 bits	Low (considered insecure due to short key length)	Relatively slow	Legacy systems, historical use
3DES (Triple DES) [20]	112 or 168 bits (112 bits effective)	64 bits	Medium (more secure than DES but less than AES)	Slower than DES and AES	Legacy systems, financial services
Blowfish [20]	32 to 448 bits	64 bits	High (considered secure with key lengths > 128 bits)	Fast in software	General-purpose encryption, software encryption
Twofish [20]	Up to 256 bits	128 bits	High (secure)	Slower than AES but flexible	File and disk encryption, open-source software
RC4 (Rivest Cipher 4) [20]	40 to 2048 bits	Stream cipher (variable size)	Low (vulnerabilities found)	Very fast	Deprecated, was used in SSL/TLS, WEP/WPA
IDEA (International Data Encryption Algorithm) [20]	128 bits	64 bits	Medium to high (patent expired, secure)	Efficient	Email encryption (used in PGP), some VPNs

TABLE-02: Different types of attacks

(b) Asymmetric Encryption

Asymmetric encryption involves the use of two keys: a public key for encrypting data and a private key for decrypting it. It is more secure for key distribution but computationally more intensive [19].

Key algorithms include:

Algorithm	Key Length	Security Level	Performance	Use Cases
RSA (Rivest-Shamir-Adleman) [21]	1024, 2048, 3072, 4096 bits	High (secure with longer keys, e.g., 2048 bits or more)	Slower compared to ECC; computationally intensive	Secure key exchange, digital signatures, SSL/TLS
ECC (Elliptic Curve Cryptography) [21]	160-521 bits (e.g., 256 bits commonly used)	High (provides comparable security to RSA with shorter keys)	Faster than RSA for the same security level	Mobile devices, secure key exchange, digital signatures, SSL/TLS
DSA (Digital Signature Algorithm) [22]	1024, 2048, 3072 bits	High (depends on key length)	Efficient for signature generation, slower for verification	Digital signatures, used in digital certificates
DH (Diffie-Hellman) [22]	1024, 2048, 3072, 4096 bits	High (secure with longer keys)	Moderately fast	Secure key exchange, establishing shared secrets
ElGamal [21]	Variable (2048 bits or higher recommended)	High (based on the difficulty of computing discrete logarithms)	Slower than RSA and ECC	Encrypted communication, digital signatures

TABLE-03: Different types of attacks

2.2.2 <u>SYMMETRIC V/S ASYMMETRIC CRYPTOGRAPHY WHICH IS BETTER?</u>

Symmetric Encryption [23] is better when:

- There is a need to encrypt large amount of data efficiently and quickly.
- Key distribution is manageable, or a secure method is already in place [23].

Asymmetric Encryption [23] is better when:

- You need a secure path to exchange keys over an edgy channel.
- Digital signatures and authentication are required.
- You have large number of users and need a scalable key management system [23].

Hybrid Encryption often provides the best of both worlds, using asymmetric encryption for secure key exchange and symmetric encryption for data transfer, combining security, efficiency, and scalability [24].

2.3 HASHING

Hashing is a fundamental technique in computer science and cryptography used to ensure data integrity, enable efficient data retrieval, and provide secure authentication mechanisms [25].

Hashing involves transforming input data of arbitrary size into a fixed-size string of characters, typically a hexadecimal number. The output, known as a hash value or digest, is unique to each unique input, and small changes in the input produce significantly different hash values [25].

2.3.1 VARIOUS HASHING ALGORITHMS

Common Hashing algorithms include: MD5(Message Digest Algorithm), SHA-1(Secure Hash Algorithm-1), SHA-256, SHA-3, Blake-2, Bcrypt [26].

Here's a comparison of the hashing algorithms mentioned earlier (MD5, SHA-1, SHA-256, SHA-3, Blake2) with bcrypt, focusing on their suitability for different applications, particularly password hashing:

Feature	**MD5 [26]**	**SHA-1 [26]**	**SHA-256 [26]**	**SHA-3 [26]**	**Blake2**
Output Size	128 bits (16 bytes)	160 bits (20 bytes)	256 bits (32 bytes)	Variable (up to 512 bits)	Variable (up to 512 bits)
Speed	Fast	Fast	Moderate	Moderate	Fast
Security	Broken	Broken	Secure	Secure	Secure
Collision Resistance	Poor	Poor	Good	Good	Good
Preimage Resistance	Poor	Poor	Good	Good	Good
Second Preimage Resistance	Poor	Poor	Good	Good	Good
Cryptographic Uses	No	No	Yes	Yes	Yes
Optimal for Checksums	Yes	Yes	No	No	Yes
Adoption	Widely used (legacy)	Widely used (legacy)	Widely used	Increasing adoption	Increasing adoption
Password Hashing	No	No	No	No	No
Adjustable Cost Factor	No	No	No	No	No
Salt Usage	Not typical	Not typical	Not typical	Not typical	Not typical

TABLE-04: Different types of attacks

MD5 [26] is known for its speed in comparison to the SHA-512 [26] algorithm. However, while SHA-512 offers stronger security, it is still susceptible to attacks due to advancements in hardware technology. Text-only passwords are particularly vulnerable to being cracked, so they are not recommended. Though MD5 has some security features, its effectiveness is significantly enhanced when combined with salting. On the other hand, SHA-512 is generally more secure, but its hashes can still be compromised with modern hardware capabilities. To counteract this and slow down potential attacks, techniques such as PBKDF2, Bcrypt, and Scrypt are utilized [26].

2.4 NEED FOR 2-FACTOR AUTHENTICATION

With advancement in technology there are numerous emerging attacks performed on regular basis to break through any highly Encrypted system.

While encryption is essential for data security, it is important to be aware of and mitigate the associated risks. The security of encrypted data heavily depends on how encryption keys are managed. Poor key management practices can lead to unauthorized access, loss of data, or data breaches. Therefore, it is very important to add an extra security layer that can be achieved with a hardware token that generates one-time passwords (OTPs) [14]. These OTPs are used for single sessions or transactions, adding an extra layer of security beyond just a password.

However, hardware tokens have several drawbacks. They are expensive to purchase, issue, and manage. From a user's perspective, using multiple two-factor authentication (2FA) [17] systems can be inconvenient as it requires carrying several tokens or cards, which can easily be lost or stolen.

To address these issues, OTPs can also be generated on mobile devices. While this eliminates the need for physical tokens, it introduces new challenges. Users must install OTP generation software on their mobile devices, and the time on both the mobile device and the server must always be synchronized. Additionally, if a user gets a new mobile device, it must be registered and the OTP software must be reinstalled, adding to the complexity [14].

3. COMPARATIVE SUMMARY TABLE OF LITERATURE SURVEY

Sr No.	Work considered	Author(s)	Contribution	Focus of the paper	Technique(s) Used	Analysis
1	Password Strength: A Survey and Analysis [7]	Gongzhu Hu	Performed clustering analysis on a set of passwords with different levels of quality(weak, medium, strong) and determine the difficulty of guessing and quality of the passwords.	• Quality of password	• Entropy analysis to check uncertainty • Levenshtein Distance • Password complexity metric	Quality of password depend on the length of the password and size of the charset.
2	Empirical study of Password Strength Meter Design [8]	1. Yi Yang 2. Kheng Cher Yeo 3. Sami Azam 4. Asif Karim 5. Ronju Ahammad 6. Rakib Mahmud	Developed a software Plug-in named "Password Strength Meter" that visually accesses the strength of user chosen password and provide information about the durability of the password and estimate the time it may take to break it using standard cracking mechanism.	• To implement contemporary password cracking algorithms. • Identify the weaker components in the given password and guide user accordingly.	• The plug-in program combines Brute Force algorithm and Dictionary attack .	To identify the weak links and overcome it by guiding the user
3	Real Time Password Strength Analysis on Web Applications using Machine Learning Approaches [9]	1. Umar Farooq	Implemented various Machine Learning algorithms on 250 accounts to determine an efficient way to defend against online and offline attacks. Created a dataset that contains numerous password with different patterns	• Force the user to choose strong password with complex pattern • To determine the efficiency of each machine learning algorithm used	• Decision Tree • Naive Bayes • Linear Regression • Random Forest • Neural Network	On implementing all the Machine Learning algorithms on the dataset, the evaluation of decision tree had the highest accuracy rate of 99%.
4	Password strength and Memorability [10]	1. Hanna Julkunen 2. Josefin Ceder Molander	Survey on password strength and find the strongest and memorable one.	• To determine a strong password and its memorability	• It's a comparitive survey	We should choose a password that is both strong and Memorable
5	Password Security: An Analysis of Password Strength and Vulnerability [11]	1. Katha Chanda	This paper contributes towards minimizing the risk posed by those seeking to	• Provides solutions for making password breaking more difficult.	• Performed tests on password: -Numeric test -Alpha numeric test	Avoid passwords containing similar char set to make it less vulnerable to attack.

			expose sensitive digital data.		-multiple case test • Entropy evaluation to determine uncertainity.	
6	Augmenting Password Strength Meter Design using the Elaboration Likelihood Model: Evidence from Randomized experiments [12]	1. Warut Khern-am-nuai 2. Matthew Hashim 3. Alian Pinsonneault 4. Weining Yang 5. Ninghui Li	Proposed design of augmented password strength meters	• Enhance the effectiveness of password strength meter	• It's a survey	Performance of password strength meters can be enhanced by incorporating meaningful persuasive messages such as the rank of the entered password in comparison to other common passwords
7	Guess again(again and again) [13]	1. Patrick Gage Kelley 2. Saranga Komanduri 3. Michelle L. Mazurek 4. Richard Shay 5. Timothy Vidas 6. Lujo Bauer 7. Nicolas Christin 8. Lorrie Faith Cranor 9. Julio L´opez	This paper analyzes different password composition policies and develops a method to measure how easily passwords can be guessed. The findings help improve our understanding of password security and composition policies.	• Develop an efficient distributed method for calculating how effectively several heuristic password guessing algorithms guess password.	• Guess-number calculators • Entropy calculation	the effectiveness of a dictionary check depends heavily on the choice of dictionary
8	Two Factor Authentication [14]	1. Asif Amin 2. Israr Ul Haq 3. Monisa Nazir	Proposed a computer-based software token which is supposed to replace existing hardware token device.	• implementation of two-factor authentication methods by using both users friendly traditional Alphanumeric Password and graphical Password as gateway for authentication	• involves generation of secured OTP using cryptographic algorithm	Integrated two factor authentication gives the best convenience to better security
9	A Comparative Usability Study of Two-Factor Authentication [15]	1. Emiliano De Cristofaro 2. Honglu Du 3. Julien Freudiger 4. Greg Norcie	This paper presents an exploratory comparative study of two-factor authentication technologies	• Conduct a pre-study interview to identify popular technologies as well as contexts and motivations in which they are used • Study their impacts on perceived usability	• Data sanitization • Demographics	Usability of different two factor authentication technologies
10	OTP Based Secure Authentication [16]	1. Mukesh Mehta 2. Aditya A. Belhe	Proposed an authentication service that makes use of one-time	• Authenticating and verifying an authorised person	• Python Scripting language	To improve the standard of security provided, OTP

		3. Prince Raj Singh 4. Maqsood Choudhary 5. Smita Chopade	passwords to prove security at the organizational level		• Time based OTP using Python	can be combined with other stand-alone security systems like RFID and Bio-metric authentication.
11	Development of Two-Factor Authentication login system using Dynamic Password with SMS Verification [17]	1. Abimbola Rhoda Iyanda 2. Mayokun Ebenezer Fassasi	Used the Obafemi Awolowo University (OAU) e-portal login system as a case study to find the solutions to overcome the weakness to provide a more secure environment.	• Adding another step of authentication to individual identity making it more difficult for an attacker to gain access to personal data.	• Dynamic password generation (OTP)	Developing a strong Authentication system which is two-way factor using SMS verification
12	Analysis and Performances of Different Hashing Algorithms [26]	3. S Govinda Rao	It emphasizes the vulnerabilities of cryptographic ciphers to various attacks and proposes solutions like salting passwords and using CPU-intensive algorithms to slow down attacks.	• analyzing and comparing different hashing algorithms to maintain the authenticity and integrin • ulnerabilities of cryptographic ciphers to various types of attacks	• SHA-256, SHA-384, and SHA-512 (Hashing methods) • salting passwords with random text to enhance security	the comparative analysis of different hashing techniques provides a basis for understanding the strengths and weaknesses of various algorithms, aiding in the selection of appropriate methods to secure data effectively.

TABLE-05: Comparative summary and analysis

4. METHODOLOGY AND IMPLEMENTATION

The methodology in the Password Strength Checker involves defining a function (password Strength) to evaluate the strength of a password and creating a validation schema for user registration data, including password strength requirements.

While designing a methodology for Password strength testing the password attacks (Brute force attack, Dictionary attack, etc) should be kept in mind as the main motive is to generate a password which cannot be cracked easily with the existing password attacks.

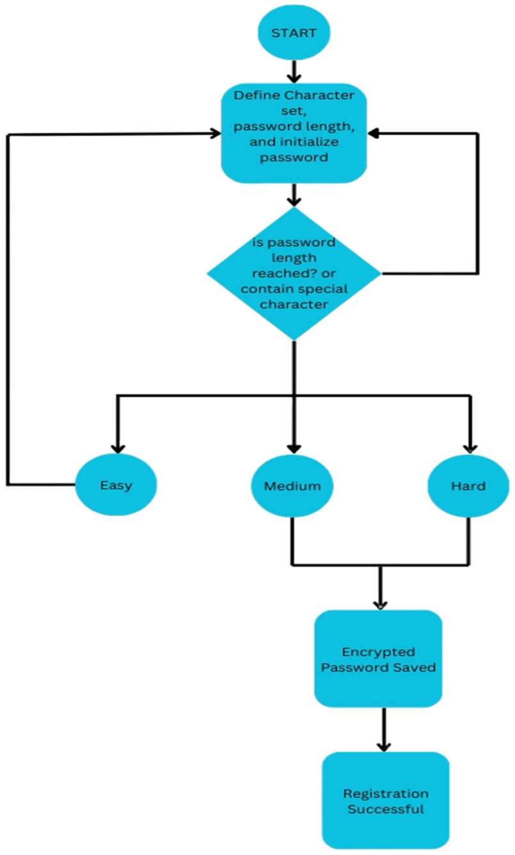

START
Define Character set, password length, and initialize password
is password length reached? or contain special character
Easy
Medium
Hard
Encrypted Password Saved
Registration Successful

FIG-01: Flow Diagram for Strength determination and registration

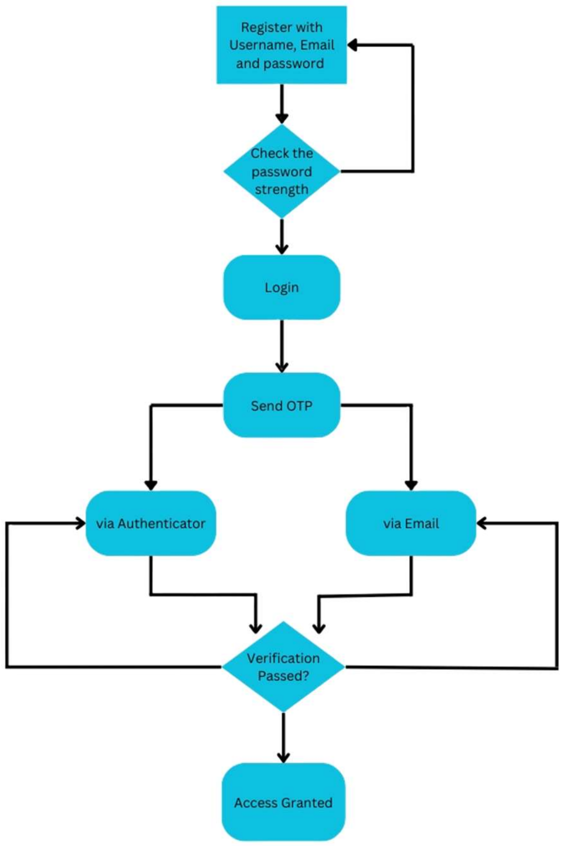


FIG-02: Backend for the login page (Postman).

Password strength criteria: According to OWASP standards

Determine the factors that contribute to password strength, such as length, character types, uniqueness, and avoidance of common patterns or dictionary words. Clearly define the requirements for a strong password.

Developing a Scoring System:

To create a scoring system that assigns points or a rating to passwords based on their adherence to the defined criteria. Assign higher scores to passwords that meet or exceed the desired strength requirements.

We can set the criteria as:

Weak Passwords

The first set of passwords generated by using numerals (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) only.
The second set of passwords generated by using lowercase characters (az) only
The third set of passwords generated by using upper case characters (A-Z).
The fourth set of passwords generated by using symbols (! @, #, $, %, ^, &, *, (,), , /, and ?).[6]

Medium Passwords

For a password to be of medium strength, it can be generated by using the following six sets:
The first set contains passwords that are generated with numerals and lowercase characters only
The second set with numerals and uppercase characters only
The third set with numerals and symbols only
The fourth set with uppercase and lowercase characters only
The fifth with uppercase and symbols only,
The sixth set with lowercase and symbols only.[6]

Strong Passwords

The scoring system will give high score to those passwords that are generated by using a mixture of numerals, uppercases, lowercases and symbols with length greater than 8 characters.[6]

4.1 PASSWORD STRENGTH FUNCTION:

Objective: Assess the strength of a password based on predefined criteria.

Implementation:

1. Define Regular Expressions:
   Two regular expressions are established to represent strong and medium password criteria.
2. Testing Password Strength:
   The function takes a password as input and tests it against the defined regular expressions.
   If the password fulfils the criteria of any regular expression (discussed above) is sorted as Medium or strong.
   If the password doesn't match either expression, it is considered "weak."
3. Return Strength Level:
   The checker is a string indicating the strength level of the password ('strong', 'medium', or 'weak').

4.2 REGISTRATION PAGE SCHEMA:

Objective : Validate user registration data, including password strength requirements.

Implementation:

1. Validation Schema:
A validation schema is defined using a validation library (possibly Yup or similar). Fields include 'name,' 'email,' 'password,' and 'password Confirmation.'

2. Constraints and Rules:
Constraints ensure data meets specific requirements, such as length limits and valid email format.

3. Password Confirmation Check:
A refine method is used to check if the 'password' and 'password Confirmation' fields match. An error message is set if they don't.

4. Password Strength Check:
Another refine method is employed to check if the password strength is at least medium.
It calls the password Strength function and ensures the returned strength is either 'strong' or 'medium.' If not, an error message is set for the 'password' field.

4.3 <u>LOGIN PAGE SCHEMA:</u>

Objective : Validate user login data.

<u>Implementation:</u>

1. Validation Schema:
A validation schema is defined using a validation library.
Fields include 'email,' 'password'.

2. Constraints and Rules:
Constraints ensure data meets specific requirements, such as valid email format.

3. Password Confirmation Check:
A method is used to check if the 'password' and 'password saved during registration' of a respective user fields match. An error message is set if they don't.

4.4 HASH FUNCTION: WORKING AND IMPLEMENTATION

Objective: Encrypting and verifying Password

Implementation

Password Hashing with our encryption:

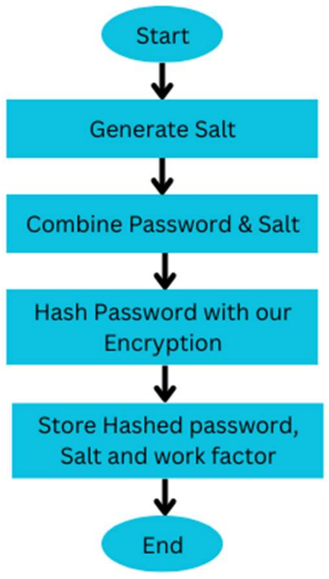

Start
Generate Salt
Combine Password & Salt
Hash Password with our Encryption
Store Hashed password, Salt and work factor
End

FIG-03: Flow Diagram for Encrypting Password

Workflow with example:

1. Generate Random Salt
   Example Salt: c8a2b8c0f7b5c3e8d4e2f1a6b1d0c9f8

2. Combine Password and Salt
   Password: myPassword
   - Salt: c8a2b8c0f7b5c3e8d4e2f1a6b1d0c9f8

3. Hash Password with our encryption (Include Work Factor)
   -fn.hash('myPassword', 'c8a2b8c0f7b5c3e8d4e2f1a6b1d0c9f8', 12)
   - Resulting Hash:
   $2b$12$C8a2b8c0F7b5c3E8D4e2F1A6B1D0C9F8u/UudISBuZQYxkBy9VVu

4. Store Hashed Password, Salt, and Work Factor

- Stored in Database:
- Hashed Password:
$2b$12$C8a2b8c0F7b5c3E8D4e2F1A6B1D0C9F8u/UudISBuZQYxkBy9VVu
- Salt: c8a2b8c0f7b5c3e8d4e2f1a6b1d0c9f8 - Work Factor: 12

5. End

Password Verification with our encryption:

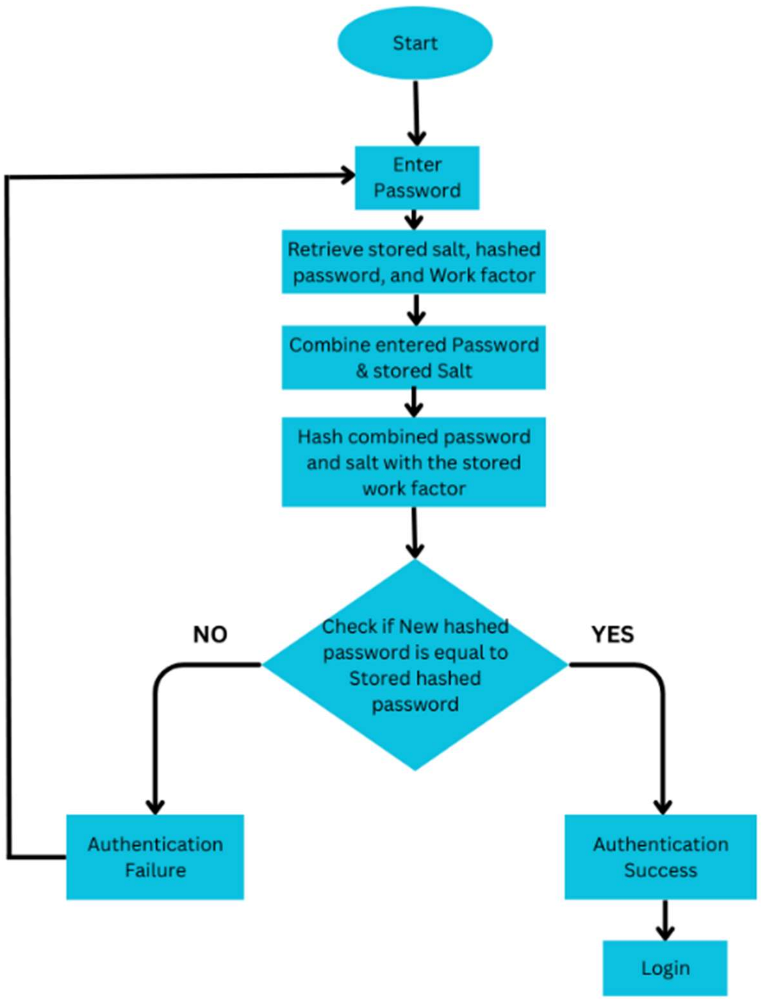

Start
Enter Password
Retrieve stored salt, hashed password, and Work factor
Combine entered Password & stored Salt
Hash combined password and salt with the stored work factor
Check if New hashed password is equal to Stored hashed password
NO
YES
Authentication Failure
Authentication Success
Login

FIG-04: Flow Diagram for Password Verification

Workflow with example:

1. Retrieve Stored Salt, Hashed Password, and Work Factor
   - Retrieved from Database:
   - Hashed Password:
   $2b$12$C8a2b8c0F7b5c3E8D4e2F1A6B1D0C9F8u/UudISBuZQYxkBy9VVu
   - Salt: c8a2b8c0f7b5c3e8d4e2f1a6b1d0c9f8 - Work Factor: 12
2. Combine Entered Password and Stored Salt
   - Entered Password: myPassword
   - Stored Salt: c8a2b8c0f7b5c3e8d4e2f1a6b1d0c9f8
3. Hash Combined Password and Salt with Stored Work Factor
   - fn. hash ('myPassword', 'c8a2b8c0f7b5c3e8d4e2f1a6b1d0c9f8', 12)
   - Resulting Hash:
   $2b$12$C8a2b8c0F7b5c3E8D4e2F1A6B1D0C9F8u/UudISBuZQYxkBy9VVu
4. Compare Newly Hashed Password with Stored Hashed Password
   - Newly Hashed Password:
   $2b$12$C8a2b8c0F7b5c3E8D4e2F1A6B1D0C9F8u/UudISBuZQYxkBy9VVu
   - Stored Hashed Password:
   $2b$12$C8a2b8c0F7b5c3E8D4e2F1A6B1D0C9F8u/UudISBuZQYxkBy9VVu
   - Match: Yes
5. If Match - Authentication Success
6. End

4.5 TWO-FACTOR AUTHETICATION: IMPLEMENTATION

Objective: To add 2-FA using TOTP.

Implementation:

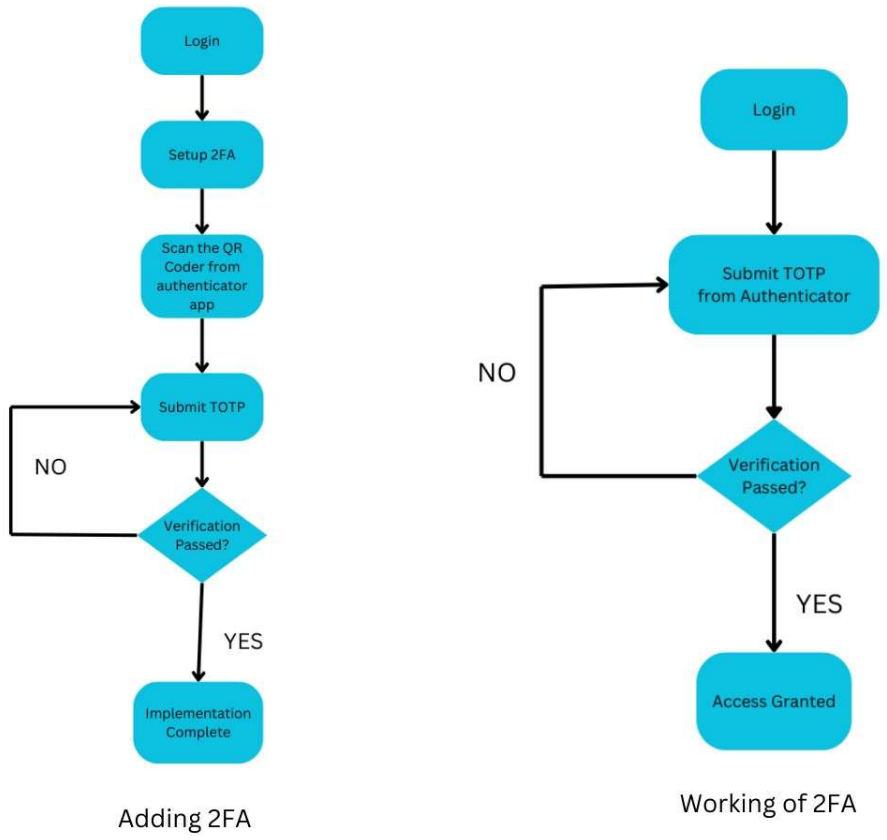


FIG-05: Flow Chart for working of 2FA

BACKEND

1. Create an endpoint to generate a 2FA secret and QR code for the user.
2. generate a secret and create a URI.
3. Save the secret in the user's data store entry.
4. generate a QR code image URL with the help of secret.
5. Provide the QR code to the user to scan with an authenticator app like Google Authenticator or Authy to create a Token (in our case it is TOTP).
6. Create an endpoint to verify the 2FA token provided by the user.

7. Retrieve the user's secret from the data store.
8. check if the provided token is valid.

FRONTEND

1. Registration
2. After successful registration, prompt the user to set up 2FA.
3. Show the QR code generated by the backend to the user for scanning with their authenticator app.
4. Create a form for the user to enter the 2FA token.
5. Send the token to the backend for verification.
6. Display appropriate messages based on whether the token is valid or not.
7. Login

Workflow of 2FA

1. User logs in with email and password.
2. Backend verifies credentials and prompts for 2FA token.
3. User enters the 2FA token.
4. Backend verifies the token.
5. Access is granted if the token is valid.

5. EXPERIMENTAL RESULT:

Registration Page Interface

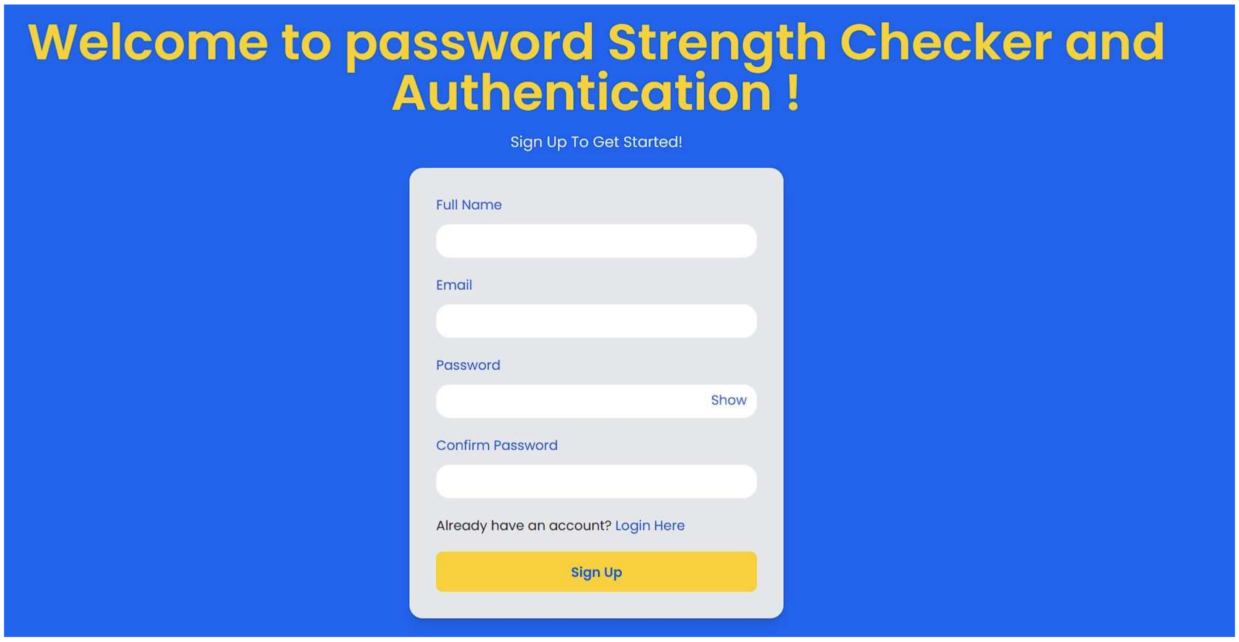

Welcome to password Strength Checker and Authentication !
Sign Up To Get Started!
Full Name
Email
Password
Show
Confirm Password
Already have an account? Login Here
Sign Up

FIG-06: Registration Interface

User Input: User registration data is submitted for validation.
Validation: The register Schema is used to validate the data.
Constraints and rules are applied to ensure the 'name,' 'email,' 'password,' and 'password Confirmation' fields meet specified criteria.
Password Strength Check:
The password Strength function is invoked within the schema to assess the strength of the entered password.
If the password does not meet the required strength, an error message is set for the 'password' field.

Error Handling:

If any validation fails, error messages are provided, indicating which specific fields did not meet the criteria.

Feedback to User:

The user receives feedback on the success or failure of the registration attempt, along with specific error messages if applicable.
This methodology ensures that user registration data is not only validated for basic constraints but also includes checks for password strength, providing comprehensive feedback to users during the registration process.

Login Page Interface

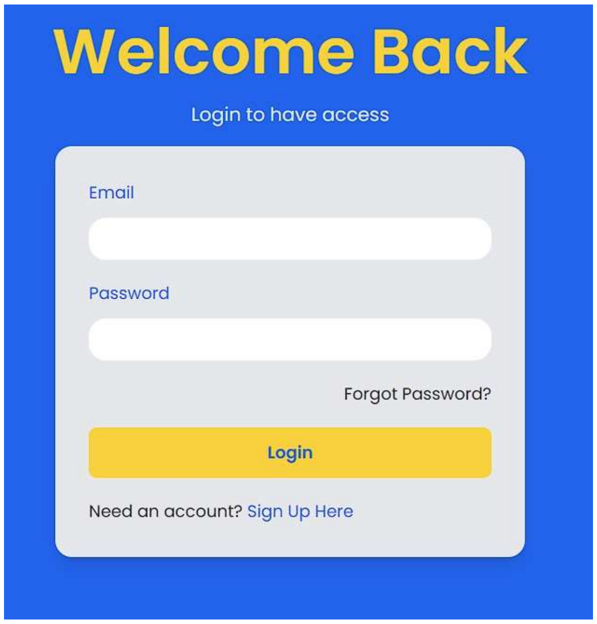


FIG-07: Login Interface

User Input: User credentials are verified
Once the user is authorized, login is successful

Error Handling:

If any validation fails, catch the error and provide user-friendly error messages.
Include specific error messages indicating which fields did not meet the validation criteria.

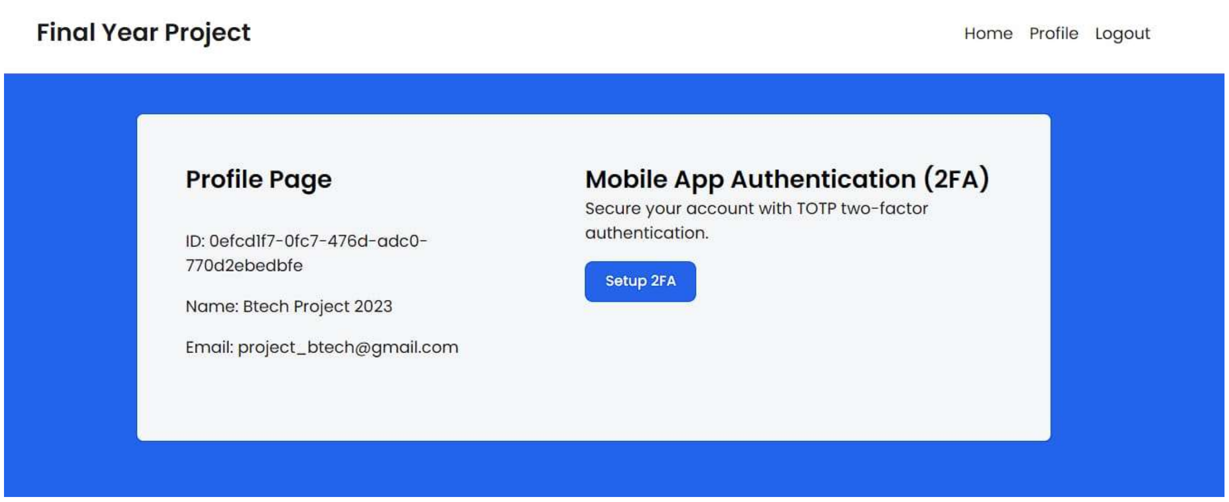
Feedback to User:

Provide feedback to the user on the success or failure of the login attempt.
If login is successful, proceed with the desired user authentication flow.
If there are validation errors, display error messages near the corresponding form fields.

Enabling 2FA:

The user needs to login and setup 2FA.Once the 2FA is enabled, the user needs to scan the QR code from Authenticator app (may use free authenticator app by Google). A TOTP will be displayed in the app which will change after every 30s. After submitting the correct TOTP verification will be passed.

2FA Interface



Profile Page

ID: 0efcd1f7-0fc7-476d-adc0-770d2ebedbfe

Name: Btech Project 2023

Email: project_btech@gmail.com

Mobile App Authentication (2FA)

Secure your account with TOTP two-factor authentication.

FIG-08: 2FA Interface

When the user will login again, the TOTP from the authenticator need to be submitted for verification. This is done an additional layer of security by requiring multiple forms of verification, making it significantly more difficult for malicious actors to compromise accounts.

6. FUTURE SCOPE:

1. On-Screen Keyboard:
Objective: Enhance security by providing an on-screen keyboard for password entry.

Implementation:
Create a virtual keyboard component that users can interact with using a mouse or touchscreen.
This helps protect against keyloggers, especially on public computers or devices.

2. Password Policy Enforcement:
Objective: Enforce a strict password policy to ensure strong passwords.

Implementation:
Define and communicate password policies to users.
Implement checks for common passwords, dictionary words, and patterns.
Educate users about creating strong and unique passwords.

4. Magic Links:
Objective: Allow users to log in via one-time links sent to their email.

5. SSO (Single Sign-On):
Objective: systems for a seamless login experience.

7. CONCLUSION:

In conclusion, the literature survey on designing a password strength tester has provided valuable insights into the current state of research and development in this field. The survey highlighted the importance of password security and the need for robust tools to assess the strength of passwords effectively.

Throughout the survey, various approaches and methodologies were explored. The literature emphasized the significance of incorporating multiple criteria, such as length, complexity, uniqueness, and avoidance of common patterns, to determine password strength accurately.

The findings of the literature survey underscored the significance of creating a password strength tester that is user-friendly, efficient, and capable of accurately evaluating the strength of passwords across various dimensions. It should provide actionable feedback to users, empowering them to create stronger passwords and enhance their overall security posture.

The user-friendly interface of the Password Strength Tester and Authenticator contributes to a seamless and positive user experience. Clear feedback and guidance provided by the tool encourage users to adopt stronger security practices without causing frustration or confusion.

It is important to note that ongoing updates and improvements are essential to adapt to evolving cybersecurity threats. Regularly updating the tool with the latest security standards and technologies ensures its continued effectiveness in safeguarding user accounts and sensitive information.

8. REFERENCES:

[1] Darbutaitė, E., Stefanovič, P. and Ramanauskaitė, S., 2023. Machine-learning-based password-strength-estimation approach for passwords of Lithuanian context. *Applied Sciences*, *13*(13), p.7811.

[2] Drašar, M., 2009. *Password based authentication* (Doctoral dissertation, Master Thesis. Universidad de Masaryk. República Checa. 2009. Disponible en: https://is. muni. cz/th/qkn59/thesis. pdf).

[3] Kuriakose, S., Teja, G.K., Duggi, S., Srivatsava, A.H. and Jonnalagadda, V., 2022. Machine Learning Based Password Strength Analysis. In *International Journal of Innovative Technology and Exploring Engineering* (Vol. 11, No. 8, pp. 5-8). Blue Eyes Intelligence Engineering and Sciences Engineering and Sciences.

[4] Dell'Amico, M., Michiardi, P. and Roudier, Y., 2010, March. Password strength: An empirical analysis. In *2010 Proceedings IEEE INFOCOM* (pp. 1-9). IEEE.

[5] https://www.onelogin.com/learn/6-types-password-attacks

[6] Gorle B "Password strength analyzer" . International journal of research publication and reviews, 2022.

[7] Hu, G., 2018. On password strength: a survey and analysis. *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp.165-186.

[8] Yang, Y., Yeo, K.C., Azam, S., Karim, A., Ahammad, R. and Mahmud, R., 2020, June. Empirical study of password strength meter design. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)* (pp. 436-442). IEEE.

[9] Farooq U , "Real Time Password Strength Analysis on Web Applications using Machine Learning Approaches" 2020.

[10] Julkunen, H. and Ceder Molander, J., 2016. Password strength and memorability.

[11] Chanda K, "Password Security: An Analysis of Password Strength and Vulnerability", 2016.

[12] Khern-am-nuai, W., Hashim, M.J., Pinsonneault, A., Yang, W. and Li, N., 2023. Augmenting password strength meter design using the elaboration likelihood model: Evidence from randomized experiments. *Information Systems Research*, *34*(1), pp.157-177.

[13] Kelley, P.G., Komanduri, S., Mazurek, M.L., Shay, R., Vidas, T., Bauer, L., Christin, N., Cranor, L.F. and Lopez, J., 2012, May. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *2012 IEEE symposium on security and privacy* (pp. 523-537). IEEE.

[14] Amin A, Israr Ul Haq, Nazir M, "Two Factor Authentication", 2017.

[15] De Cristofaro, E., Du, H., Freudiger, J. and Norcie, G., 2013. A comparative usability study of two-factor authentication. *arXiv preprint arXiv:1309.5344*.

[16] Mehta M, Aditya A. Belhe, Singh P.R, Choudhary M, Chopade S. "OTP Based Secure Authentication", 2016.

[17] Iyanda, A.R. and Mayokun, E.F., 2022. Development of two-factor authentication login system using dynamic password with SMS verification. *International Journal of Education and Management Engineering*, *12*(3), p.13.

[18] Goyal, Shimpy. "A Study on Encryption and Decryption System." (2021).

[19] Mandal, S.K. and Deepti, A.R., 2019. A review paper on encryption techniques. International Journal of Research and Analytical Reviews, 6(2), pp.70-75.

[20] Alenezi, M.N., Alabdulrazzaq, H. and Mohammad, N.Q., 2020. Symmetric encryption algorithms: Review and evaluation study. International Journal of Communication Networks and Information Security, 12(2), pp.256-272.

[21] Ahmed, A., Xi, R., Hou, M., Shah, S.A. and Hameed, S., 2023. Harnessing big data analytics for healthcare: A comprehensive review of frameworks, implications, applications, and impacts. IEEE Access.

[22] Al-Shabi, M.A., 2019. A survey on symmetric and asymmetric cryptography algorithms in information security. International Journal of Scientific and Research Publications (IJSRP), 9(3), pp.576-589.

[23] Kumar, Y., Munjal, R. and Sharma, H., 2011. Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures**.** International Journal of Computer Science and Management Studies, 11(03), pp.60-63.

[24] El_Deen, A.E.T., 2013. Design and implementation of hybrid encryption algorithm. International Journal of Scientific & Engineering Research, 4(12), pp.669-673.

[25] Tarawneh, M., 2023. Cryptography: Recent Advances and Research Perspectives.

[26] S Govinda Rao , "Analysis and Performances of Different Hashing Algorithms", 2016