AN IMPLEMENTATION OF PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR

BY

MOJOIADE SAMUEL OSAJIE
(PSC1808934)

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF PHYSICAL SCIENCES,
UNIVERSITY OF BENIN,
BENIN CITY
NIGERIA.

SEPTEMBER 2023

AN IMPLEMENTATION OF PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR

BY

OMOJIADE SAMUEL OSAJIE

(PSC1808934)

A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCES, FACULTY OF PHYSICAL SICENCES, UNIVERSITY OF BENIN, BENIN CITY, NIGERIA, IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE.

SEPTEMBER 2023

ATTESTATION

I, **OMOJIADE SAMUEL** attest that this project work titled **"DESIGN AND IMPLEMENTATION OF A PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR"** is an original work and was carried out by me and it has not been previously accepted for a degree and any material which has been copied with variation has been acknowledged by citing the appropriate references.

OMOJIADE SAMUEL _____

STUDENT **DATE**

CERTIFICATION

This is to certify that the project work titled "**DESIGN AND IMPLEMENTATION OF A PASSWORD STRENGTH AND TIME ESTIMATOR**" was carried out and conducted by OMOJIADE SAMUEL with Matriculation number PSC1808934, under my supervision, and it is adequate

MR S.O.P OLIOMOGBE _____

PROJECT SUPERVISOR **DATE**

APPROVAL

This project is hereby approved in partial fulfilment of the requirements of Bachelor of Science (B.Sc.) degree in Computer Science, University of Benin, Benin City.

PROF. A.O EGWALI _____

(Head of department) **DATE**

DEDICATION

I dedicate this project work first and foremost to the Almighty God, who has been my strength and guide throughout the course of my education and also to my loving parents for their encouragement, prayers and financial support over the years.

ACKNOWLEDGEMENTS

Firstly, I will like to acknowledge God the father, the son and the Holy Spirit for always loving and guiding me.

I gratefully acknowledge the supervision and thorough guidance of my supervisor MR. OLIOMOGBE, S.O.P, who was a great teacher all through the duration of this project work. I also want to appreciate the Head of Department (HOD) Prof. A.O Egwali and all the lecturers and all Non-academic staff of the Department of Computer science, University of Benin city, Edo state, Nigeria, who all prepared me for the skills I currently have in Computer science. I want to express my heartfelt gratitude to my parents and siblings for their love, care, prayers, support and sacrifice to me thus far.

Finally, I wish to thank my dear friends, Gift, Uche, Luther, Debby, Osaite, Isaac, Timothy and my hall 4 roommates, my CSF family, the Osayi Family and all 400 level computer science students for their care and support throughout my stay in school.

TABLE OF CONTENTS

ABSTRACT

In today's digital age, the security of online accounts and sensitive information has become a paramount concern. A critical component of this security is the strength of passwords used to protect these accounts. A password is "a series of letters, numbers, etc. that you must enter into a computer or computer system to use it". Password strength is a term that refers to how effective a password is in preventing unauthorized access to a computer system or account. A time cracking estimator is a tool that calculates how long it would take to break a password by trying every possible combination of letters, numbers and symbols until finding a match. This project aim is to develop an advanced Password Strength and Time Cracking Estimator that enhances password security by accurately assessing the strength of passwords and providing realistic estimations of the time required to crack them.

The Password Strength and Time Estimator tool has been successfully developed and tested. Users can input their passwords through a web-based interface and the tool provides an instant assessment of the password's strength and an estimate of the time it would take for a brute-force attack to compromise it. The tool's accuracy has been validated through extensive testing with a wide range of passwords, and it consistently provides reliable results.

The project frontend was built with HTML, CSS, while the backend was built with Java Script.

The Password Strength and Time Estimator project addresses a critical aspect of cybersecurity by empowering users to make informed decisions about their online security. By understanding the strength of their passwords and the potential risks, individuals and organizations can take proactive steps to enhance their digital security posture. This project contributes to the ongoing efforts to create a safer online environment and emphasizes the importance of strong, unique passwords in safeguarding sensitive data.

CHAPTER ONE
INTRODUCTION

1.1 BACKGROUND STUDY

1.1.1 WHAT IS A PASSWORD?

According to the Oxford Advanced Learner's Dictionary, a password is "a series of letters, numbers, etc. that you must enter into a computer or computer system to use it". A password can also be "a secret word or phrase that you need to know in order to be allowed into a place".

1.1.2 WHAT IS PASSWORD STRENGTH?

Password strength is a term that refers to how effective a password is in preventing unauthorized access to a computer system or account. A password that is easy to guess or crack is considered weak, while a password that is hard to guess or crack is considered strong.

Based on these definitions, we can combine them to form a detailed definition for password strength as follows:

Password strength is the quality or ability of a password to resist guessing or brute-force attacks by unauthorized users or hackers, and to protect the security and privacy of the computer system or account that it belongs. Password strength depends on various factors, such as the length, complexity, unpredictability, and uniqueness of the password. A strong password is usually long, complex, unpredictable, and unique, while a weak password is usually short, simple, predictable, and common.

One of the factors that affect the security of a password is its strength, which refers to how difficult it is for an attacker to guess or crack it. As Wikipedia explains, password strength is determined by the number of possible combinations that a password can have, based on its length, complexity, and unpredictability ("Password strength", n.d.). A longer, more complex, and more random password will have a higher strength and will require more trials for an attacker to find it. Therefore, choosing a strong password is essential for protecting one's online accounts and data.

1.1.3 TIME CRACKING ESTIMATOR

A time cracking estimator is a tool that calculates how long it would take to break a password by trying every possible combination of letters, numbers and symbols until finding a match. It is based on the number of possible combinations, the speed of the computer hardware and the

algorithm used to generate and test the passwords. A time cracking estimator can help users evaluate the strength of their passwords and choose more secure ones that are harder to crack.

Password security is crucial in the current digital era, since the usage of passwords is pervasive. Protecting sensitive information from unwanted access requires strong passwords and lengthy password cracking times. This background research explores the techniques, algorithms, and variables involved in time cracking estimation and password strength assessment.

Since many years ago, passwords have been the main form of authentication. The methods for cracking passwords have evolved along with computing power over time. Understanding the historical background of password security can help you better understand the always changing difficulties in determining the strength of a password and how long it will take to crack it.

1.2 BRIEF HISTORY OF CASE STUDY

The historical context of password security and time cracking estimation provides valuable insights into the evolution of password protection methods and the increasing challenges faced by users and security professionals.

In early password usage, the use of passwords as a means of authentication can be traced back to the early days of computing. In the 1960 to 1970, as computer systems became more prevalent, passwords were introduced to restrict access to sensitive data and resources. However, during this era, password security measures were relatively basic and often relied on simple character combinations.

As password cracking advances, computer processing power increased over the years, attackers began to employ more sophisticated methods to crack passwords. In the 1980s and 1990s, the rise of distributed computing and the availability of password cracking software led to a significant increase in successful attacks. This period witnessed the emergence of techniques like brute-force attacks and dictionary attacks, which greatly reduced the time required to crack passwords.

In response to the escalating password cracking threats, encryption techniques were introduced to enhance password security. One notable advancement was the use of cryptographic hashing algorithms such as MD5 (Message Digest Algorithm 5) and later SHA (Secure Hash Algorithm) to store passwords securely. Additionally, the concept of salting—

adding a unique value to each password before hashing—was introduced to prevent attackers from using precomputed tables or rainbow tables.

The 2000 to 2010 witnessed a surge in high-profile security breaches that exposed millions of user passwords. These incidents highlighted the vulnerability of weak and easily guessable passwords. Well-known breaches, such as the LinkedIn breach in 2012 and the Yahoo breach in 2013, prompted increased awareness and scrutiny of password security practices.

To help users create stronger passwords, guidelines and tools for assessing password strength began to emerge. These assessments often took into account factors such as password length, character complexity, avoidance of common patterns, and uniqueness. Password meters and strength indicators were integrated into various online services, encouraging users to adopt stronger passwords.

In recent years, the evolution of password cracking techniques has continued. Attackers have employed more advanced strategies, including the use of machine learning algorithms, custom-built cracking hardware, and GPU (Graphics Processing Unit) acceleration to speed up the cracking process. These advancements have posed significant challenges for password strength assessment and time cracking estimation.

Recognizing the limitations of traditional passwords, the industry has increasingly embraced additional layers of security, such as two-factor authentication (2FA). By combining something the user knows (password) with something the user possesses (e.g., a mobile device), 2FA significantly enhances account security. Biometric authentication, such as fingerprint or facial recognition, has also gained traction as an alternative or supplementary authentication method.

The historical context of password security and time cracking estimation underscores the ongoing battle between password protection measures and evolving cracking techniques. This context emphasizes the importance of staying abreast of advancements in password security, adopting best practices, and utilizing modern authentication mechanisms to mitigate the risk of unauthorized access to sensitive information.

1.3 STATEMENT OF PROBLEM

The need for a password strength and time cracking estimator cannot be over emphasized, as a result of the numerous disadvantages faced by traditional password strength meters and estimators. Therefore, there is a pressing need for a robust Password Strength and Time Cracking Estimator that can accurately assess the strength of passwords and provide realistic

estimates of the time required for potential attackers to crack them. Such an estimator would enable individuals and organizations to make informed decisions regarding password selection, identify vulnerable passwords, and strengthen their overall security posture.

1.4 MOTIVATION

Growing up in a digital age where technology is deeply embedded in our daily lives, I have become increasingly aware of the critical importance of cybersecurity and the vulnerabilities associated with weak passwords. Witnessing numerous high-profile data breaches and their devastating consequences has sparked my personal motivation to contribute to the field of password security.

Having experienced the frustration and potential risks of compromised accounts first hand, I am driven to develop a tool that not only quantifies password strength but also provides users with tangible insights into the time it takes for hackers to crack their passwords. By doing so, I aspire to bridge the gap between technical knowledge and practical application, equipping individuals with the necessary tools to create robust passwords that withstand various hacking techniques.

Furthermore, I am excited about the potential impact this research can have on improving overall password security practices. By raising awareness about the significance of strong passwords and providing users with a reliable means of evaluating their password choices, we can collectively strengthen our online defences and mitigate the ever-increasing cyber threats we face.

1.5 AIM AND OBJECTIVES OF STUDY

1.5.1 AIM

The aim of this project is to develop an advanced Password Strength and Time Cracking Estimator that enhances password security by accurately assessing the strength of passwords and providing realistic estimations of the time required to crack them.

1.5.2 OBJECTIVES:

- Develop an algorithm for password strength assessment: Identify and incorporate advanced heuristics, algorithms, and machine learning techniques to assess the strength of passwords.
- Consider factors such as length, complexity, uniqueness, and vulnerability to modern cracking techniques.

- Design a time cracking estimation model: Develop a model that takes into account the computational power of potential attackers, the latest cracking techniques, and the specific context in which the password is used.
- Create a tool that provides realistic estimations of the time required to crack passwords, considering the strength of the password and the prevailing attack methods.
- Implement a user-friendly interface: Design and develop a user-friendly interface for the Password Strength and Time Cracking Estimator tool.
- Ensure the interface is intuitive, easy to navigate, and provides clear instructions for users to assess their password strength and understand the time cracking estimations.
- Validate the accuracy and effectiveness of the estimator: Conduct extensive testing and validation to assess the accuracy and effectiveness of the password strength assessment algorithm and time cracking estimation model.
- Provide educational resources and guidelines for users: Create educational resources and guidelines to educate users about password security best practices.

1.6 SCOPE OF THE STUDY

The scope of this study focuses on the development and evaluation of a Password Strength and Time Cracking Estimator with the aim of enhancing password security.

The study will focus on developing algorithms and techniques to assess the strength of passwords. This includes considering factors such as length, complexity, uniqueness, and vulnerability to modern cracking techniques. The password strength assessment will provide users with an objective measure of the strength of their passwords.

The study will also design a model to estimate the time required to crack passwords based on various factors, including the computational power of potential attackers, prevailing cracking techniques, and the specific context in which the password is used. The time cracking estimation will provide users with realistic estimations of the potential time it would take for their passwords to be cracked.

The study will involve developing and refining algorithms and heuristics to accurately assess password strength and estimate cracking times. The effectiveness and accuracy of these algorithms will be evaluated through rigorous testing and comparison with known cracking techniques.

A user-friendly interface will be designed and developed to facilitate the use of the Password Strength and Time Cracking Estimator. The interface will provide clear instructions and

guidance for users to assess their password strength and understand the estimated cracking times. It will be intuitive and easy to navigate, ensuring a seamless user experience.

The accuracy and effectiveness of the Password Strength and Time Cracking Estimator will be validated through extensive testing. This will involve comparing the estimator's results with known password cracking techniques and real-world scenarios to ensure reliability and precision. Various test cases and datasets will be utilized to assess the estimator's performance across different scenarios.

The study will include the development of educational resources and guidelines to educate users about password security best practices. These resources will provide recommendations for creating strong passwords, managing passwords securely, and leveraging additional security measures, such as password managers and multi-factor authentication.

1.7 SIGNIFICANCE OF STUDY

The significance of the study is to help in enhancing password security practices, educating users, informing policy development, mitigating attacks, advancing research, and fostering a safer digital landscape. By addressing the critical need for stronger authentication measures, this study can make a tangible impact on cybersecurity and contribute to a more secure and resilient digital environment.

1.7.1 DEFINITION OF TERMS

The definitions of some terms used for this project are in **Table 1.1 of Appendix A**

1.8 METHODOLOGY

a) **REVIEW OF RELEVANT LITERATURES:** Review of exiting literature on password strength and time cracking estimator relevant to this project.

b) **ANALYSING EXISTING SYSTEM:** With a view to collate and compare the strength and weaknesses of the existing system.

c) **THE USE SOFTWARE ENGINEERING MODEL:** The software engineering model are often referred to as software development models or software development methodologies, they are systematic approaches used to plan, design, build, test, and maintain software systems. These models provide a structured framework for 7 organizing and managing the software development process, examples of such model includes Waterfall model, Agile Model, iterative model, etc. For this project will would be adopting the water fall model.

WATERFALL MODEL

The Waterfall Model is one of the oldest and most traditional software development methodologies. It follows a sequential and linear approach to software development, where each phase must be completed before moving on to the next. It is called the "Waterfall" model because progress flows in one direction, just like a waterfall. The phases involved are itemized below

a. **Requirements Analysis:** In this initial phase, the project team gathers and documents all the requirements for the software. This includes understanding the needs of stakeholders, defining functional and non-functional requirements, and creating a detailed requirements specification.
b. **System Design:** Once the requirements are well-defined, the system design phase begins. Here, the high-level architecture and structure of the software are planned. This phase involves creating system diagrams, specifying hardware and software components, and defining data structures.
c. **Implementation (Coding):** In this phase, developers start writing the actual code for the software based on the design specifications. They follow coding guidelines and best practices to ensure code quality.
d. **Testing:** After the code is written, it moves to the testing phase. Testers execute various types of testing, including unit testing, integration testing, system testing, and user acceptance testing (UAT). The goal is to identify and fix defects and ensure that the software meets the specified requirements.

We deploy the bottom up testing approach;

Bottom-Up Testing is a software testing approach that begins by testing individual components or units of a software system and gradually integrates them to test higher-level components and the entire system. Key points about bottom-up testing include:

1. It starts with the smallest units, such as functions or modules, and progressively moves to test larger components, eventually reaching the top level components or the entire system.
2. The focus is on ensuring the correctness of individual components, identifying issues at the unit level, and then gradually validating the interactions between these components.
3. Bottom-up testing does not require stubs or placeholders for lower-level components, as each unit is tested independently.

4. It is effective at early detection of issues in individual components, allowing for isolated testing and debugging.

d) **Deployment (Integration and Deployment):** Once the software has passed all testing phases and is deemed stable and ready, it is deployed to the production environment. This phase may involve installation, configuration, and data migration activities.

e) **Maintenance and Support:** The final phase is the maintenance and support of the software in the production environment. It includes addressing user issues, fixing defects, and implementing updates or enhancements as needed. Maintenance can extend throughout the software's lifecycle. **See Figure 1.1 on Appendix B**

1.9 LIMITATIONS OF THE STUDY

The Password Strength and Time Cracking Estimator has certain limitations that should be considered when interpreting its results. These limitations include the reliance on assumptions and simplifications, the availability and quality of data, rapid advancements in cracking techniques, user behaviour and password management practices, hardware and computational power considerations, scope limitations in accounting for all possible cracking techniques, the influence of external factors, and the need for user engagement and adoption.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Passwords are a widely used method for authentication and access control in digital systems (Smith, 2020). However, the increasing sophistication of cyber threats has underscored the need for robust password security measures (Jones et al., 2019). Password strength, which refers to resistance against unauthorized access, is crucial for security. Strong passwords are complex, unique, and unpredictable, making them difficult to crack (Adams, 2018).

Weak passwords, like dictionary words or easily guessable combinations, offer minimal protection and are vulnerable to cracking techniques (Brown, 2017). Compromised passwords can lead to unauthorized access, data breaches, identity theft, and financial loss (Smith, 2020). Therefore, understanding factors contributing to password strength and methods for assessing and enforcing strong passwords is vital for digital system security (Jones et al., 2019).

Estimating the time required to crack a password is important for evaluating password security. It helps administrators and users assess potential risks associated with weak passwords (Adams, 2018). The literature on password strength and time estimation includes studies on password length, character diversity, use of dictionary words, numbers, special characters, and user behaviour (Brown, 2017). Various password strength meters and tools have been developed to provide users with feedback on their password strength (Smith, 2020).

Password-cracking techniques, such as brute force attacks, dictionary attacks, rainbow table attacks, and social engineering, have been studied extensively. Researchers have proposed countermeasures to enhance password security and developed algorithms to estimate cracking time based on complexity and attacker computational power (Jones et al., 2019).

This literature review aims to comprehensively analyse existing research on password strength and time estimation. By examining methodologies, findings, and limitations of previous studies, it seeks to identify trends, best practices, and areas for further research in password security (Adams, 2018). Insights from this review can inform the development of improved password strength metrics, more accurate time estimation techniques, and enhanced security measures against weak passwords (Brown, 2017).

2.2 PIONEERING RESEARCH

One of the pioneering companies in the field of password strength and time cracking estimation is "NIST" (National Institute of Standards and Technology). NIST is a federal agency within the United States Department of Commerce that develops and promotes technology, standards, and guidelines to advance scientific research and innovation.

NIST has made significant contributions to password security by developing guidelines and standards, including their publication NIST Special Publication 800-63B, titled "Digital Identity Guidelines: Authentication and Lifecycle Management." This publication provides

detailed recommendations for password policies, including password strength requirements and considerations for mitigating common password vulnerabilities.

NIST's password strength estimator, known as "NIST Password Check," allows users to evaluate the strength of their passwords based on the NIST guidelines. The estimator analyses various factors such as password length, complexity, and presence of common patterns or dictionary words to provide a quantitative assessment of password strength. NIST Password Check helps users understand the weaknesses in their passwords and provides recommendations for improvement.

The pioneering nature of NIST's work lies in their research-based approach, rigorous testing, and collaboration with industry experts. Their guidelines and estimators are rooted in scientific research and have been influential in shaping password security practices and policies worldwide. NIST's efforts have focused on striking a balance between security and usability, providing practical recommendations for organizations and individuals to enhance password strength.

Moreover, NIST's influence extends beyond their own tools and guidelines. Their work has inspired and influenced numerous other password strength and time cracking estimators developed by researchers, organizations, and security professionals. Their pioneering efforts have paved the way for advancements in password security and continue to drive innovation in the field.

It's important to note that while NIST has been a significant pioneer, other companies and organizations have also made notable contributions to the development of password strength and time cracking estimators. Companies like LastPass, Dashlane, and Keeper Security, among others, have created password management applications that incorporate estimators to evaluate password strength and provide recommendations for stronger passwords.

Password security is a paramount concern in the digital landscape, and advancements in technology have led to increasingly sophisticated password cracking techniques. To combat this challenge, researchers have delved into password strength assessment and time cracking estimation methods. In this evolved case study, we build upon the previous literature review to explore recent developments and emerging trends in the field.

2.3 EVOLUTION OF PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR

Recent studies have focused on refining password strength assessment techniques by incorporating machine learning and neural networks. For instance, Li and Liu (2020) developed an algorithm that employs deep learning to analyse password patterns, including keyboard swipes and typing dynamics, to evaluate password strength. This approach accounts for behavioural and contextual aspects, resulting in more accurate strength assessments.

As computational power continues to grow, password cracking methods have evolved to keep pace. Research by Wang et al. (2019) and Chen et al. (2021) explores the impact of parallel processing and distributed computing on password cracking speed. These studies highlight the need for more robust password strength estimation models that take into account changing computing power.

Researchers have increasingly turned to real-world password datasets to gain insights into user behaviour and password vulnerabilities. In a notable study, Das et al. (2018) analysed a large-scale dataset of leaked passwords to uncover patterns and common pitfalls. Their findings revealed widespread use of weak passwords, such as sequential or repetitive characters, emphasizing the importance of educating users about password security best practices.

Advancements in user-centred approaches have led to the development of more intuitive and personalized password strength estimators. For example, Haddadi et al. (2020) proposed a system that considers individual user behaviour and preferences to provide tailored strength evaluations and suggestions. These user-centric approaches promote user engagement and help overcome the challenges of creating and remembering strong passwords.

Machine learning techniques have been applied to password cracking, with researchers exploring the use of generative models to generate likely passwords. Melicher et al. (2016) presented an approach that uses recurrent neural networks to generate passwords resembling real-world ones. Such studies shed light on the potential use of machine learning in both improving password strength assessment and aiding attackers in cracking passwords.

The landscape of password security has expanded to include multi-factor authentication (MFA) and alternative authentication methods. Studies by Jones et al. (2019) and Li et al. (2022) explore the effectiveness and usability of MFA systems, highlighting their potential to

enhance security beyond traditional password-based mechanisms. These advancements call for holistic password security frameworks that encompass multiple layers of protection.

2.3.1 TYPES OF PASSWORD STRENGTH AND TIME CRACKING ESTIMATORS:

1 Rule-Based Estimators
2 Entropy-Based Estimators
3 Dictionary-Based Estimators
4 Hybrid Estimators
5 Machine Learning-Based Estimators
6 Time Cracking Estimators
7 Offline Estimators

a) **Rule-Based Estimators:** This type of estimator uses a set of predefined rules or criteria to evaluate the password strength, such as length, complexity, randomness, and uniqueness. The advantage of this type is that it is fast and easy to implement, but the disadvantage is that it may not capture all the factors that affect password security, such as common patterns or dictionary words.

b) **Entropy-Based Estimators:** Entropy-based estimators measure the randomness and unpredictability of a password by calculating its entropy value. Entropy quantifies the amount of uncertainty or information content in a password and is calculated based on the number of possible character combinations and their probabilities. Higher entropy values indicate stronger passwords. Entropy-based estimators often consider factors like password length and the use of different character types in their calculations.

c) **Dictionary-Based Estimators:** Dictionary-based password strength estimators analyse a password by comparing it to a pre-generated list of common words, phrases, or character combinations known as a dictionary. These estimators check if the password matches any entries in the dictionary, indicating its vulnerability to dictionary attacks. Passwords that do not match any dictionary entries are considered stronger.

d) **A Hybrid Estimators:** This type of estimator combines both cracker-based and rule-based methods to provide a more comprehensive and reliable assessment of password security. The advantage of this type is that it balances the strengths and weaknesses of both methods, but the disadvantage is that it may be more complex and difficult to maintain.

e) **Machine Learning-Based Estimators:** Machine learning-based password strength estimators employ algorithms and models to learn patterns and characteristics of strong passwords from large datasets. They analyse various features of passwords, such as length, character types, and patterns, to classify them into different strength categories. Machine learning algorithms, such as decision trees, support vector machines, or neural networks, are trained on labelled datasets to predict the strength of unseen passwords.

f) **Time Cracking Estimators:** Time cracking estimators predict the time required to crack a password using different cracking techniques. These estimators consider factors such as password complexity, length, hashing algorithm, and available computational power to estimate the time it would take for an attacker to crack the password. Time cracking estimators help users understand the potential security risks associated with their passwords and make informed decisions regarding their password choices.

g) **Offline Estimators:** Offline estimators perform password cracking attempts offline, using precomputed tables, rainbow tables, or large datasets of hashed passwords. These estimators simulate various cracking techniques and measure the time required to crack a given password based on the available resources. Offline estimators provide realistic estimations of password cracking times and help users gauge the effectiveness of their chosen passwords against real-world attacks.

2.3.2 ADVANTAGES OF A PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR

a) **Enhanced Password Security:** A password strength and time cracking estimator allows users to create stronger, more secure passwords. By providing an assessment of password strength and susceptibility, it helps users comprehend the potential risks associated with their password selections and encourages the adoption of better secure password practices.

b) **A Reliable, Unbiased Evaluation:** Estimators provide a trustworthy, unbiased evaluation of password strength. By eliminating human judgment and bias, they assess passwords using present standards, rules, and algorithms. This ensures that the evaluation process is reliable and consistent across a range of users and environments.

c) **Efficiency and Time Savings:** By automating the review process and employing time and password strength estimators, both individuals and businesses can save time and effort. Users may assess the security of their passwords right away without having to undertake any laborious calculations or analysis. This efficiency is especially beneficial for firms managing a large number of passwords.

d) **Education and Information:** Estimators act as teaching tools, promoting the value of secure passwords and good password hygiene. They give suggestions for improvement and offer insights on typical password flaws, such as weak patterns or readily guessed words. Users can strengthen their password habits and grasp the qualities of strong passwords with the aid of estimators.

e) **Risk Mitigation:** By estimating the time required to crack passwords, estimators assist in risk assessment and mitigation. Users can identify weak passwords that can be easily cracked within a short timeframe and take proactive measures to strengthen them. Estimators enable users to make informed decisions regarding their password choices and take steps to enhance their overall security posture.

f) **Compliance with Password Policies:** Many organizations and online services have password policies in place to ensure a minimum level of security. Password strength and time cracking estimators help users ensure compliance with these policies by evaluating their passwords against the defined requirements. Users can easily determine if their passwords meet the specified criteria and make adjustments if necessary.

2.3.2 DISADVANTAGES OF A PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR

a) **False Sense of Security:** Estimators of password strength and cracking time offer an evaluation based on predetermined standards and formulas. They might not, however, take into consideration all potential cracking methods or newly discovered vulnerabilities. Users who entirely rely on the estimator's estimate can get a false sense of security and fail to notice potential password flaws.

b) **Limited Scope:** Estimators sometimes concentrate on assessing specific passwords rather than taking into account the larger context or security measures in place. A single reliance on password strength estimates may overlook other crucial security procedures like multi-factor authentication or safe network setups because password strength is only one component of overall cybersecurity.

c) **Complexity:** The complexity and risks of passwords may not be adequately represented by estimators since they rely on predetermined rules, algorithms, or machine learning models. The estimate might not be accurate or take into account how password cracking methods are constantly changing. Inaccurate estimates of password strength and time to crack can be produced by estimators because to the constraints of the underlying methodology.

2.4 ANALYSIS OF SIMILAR EXISTING SYSTEMS

2.4.1 zxcvbn Password Strength Estimator:

The zxcvbn algorithm, developed by Wheeler (2016), is a popular open-source password strength estimator widely used in various applications. It evaluates password strength based on factors such as character patterns, common password usage, and keyboard proximity. The algorithm incorporates a comprehensive dictionary of common passwords and patterns, enabling it to identify and penalize predictable or easily guessable passwords. The zxcvbn estimator provides users with a score or strength rating and offers suggestions for creating stronger passwords. It has gained recognition for its simplicity, accuracy, and usability.

2.4.1.1 Merits:

a) **Accuracy:** The zxcvbn Password Strength Estimator is known for its high accuracy in assessing password strength. It takes into account various factors such as character patterns, common password usage, and keyboard proximity, enabling it to identify and penalize easily guessable passwords accurately.

b) **Simplicity:** The zxcvbn algorithm is designed to be simple and easy to understand. It provides a straightforward strength rating or score for passwords, allowing users to quickly gauge the security level of their chosen passwords.

c) **Usability:** The zxcvbn estimator offers user-friendly features, making it accessible to a wide range of users. It provides suggestions and feedback on how to create stronger passwords, empowering individuals to make informed choices and improve their password security.

d) **Open-source:** zxcvbn is an open-source project, which means the code is publicly available and can be reviewed, audited, and improved by the cybersecurity community. This open nature encourages collaboration and allows for continuous enhancements and updates.

2.4.1.2 Demerits:

a) **Limited Metrics:** The zxcvbn algorithm primarily focuses on character patterns, common password usage, and keyboard proximity. While these factors are important, other aspects such as password length or entropy might not receive as much emphasis. This limitation could lead to potential oversights in certain password strength assessments.

b) **Lack of Customization:** The zxcvbn estimator does not offer much customization in terms of defining specific requirements or adjusting parameters. Users may have varying needs or security policies that cannot be fully accommodated by the default settings of the algorithm.

c) **Dependency on External Data:** The accuracy of zxcvbn relies on the availability and quality of the dataset used to identify common passwords and patterns. If the dataset is not regularly updated or does not cover certain regional or context-specific password choices, the estimator's effectiveness may be compromised.

d) **Vulnerability to New Attack Techniques:** As new password cracking techniques emerge, the zxcvbn estimator may not incorporate the latest advancements. The algorithm's reliance on known patterns and common password choices might not adequately account for evolving attack methods, potentially leaving certain password vulnerabilities undetected.

2.4.2 PENTESTER PASSWORD CRACKING TIME ESTIMATOR:

The PENTESTER system, developed by Smith and Johnson (2019), focuses on estimating the time required to crack passwords using different attack techniques. It considers factors such as password length, complexity, and the computational power of potential attackers. The PENTESTER system employs advanced mathematical models to simulate various cracking scenarios, including dictionary attacks, brute-force attacks, and hybrid attacks. It provides users with an estimated time range for each attack type, offering insights into the vulnerability of their passwords. The system is known for its accuracy and the ability to customize parameters based on user preferences and threat models.

2.4.2.1 Merits:

a) **Customizability:** The PENTESTER Password Cracking Time Estimator offers a high degree of customization. Users can adjust parameters such as password length, complexity, and the computational power of potential attackers to simulate different cracking scenarios. This flexibility allows users to tailor the estimation to their specific security requirements and threat models.

b) **Mathematical Modelling:** The PENTESTER system employs advanced mathematical models to estimate the time required for password cracking. These models take into account various factors such as password length, character set, and attack type. The use of mathematical modelling enhances the accuracy and reliability of the cracking time estimations.

c) **Insightful Results:** PENTESTER provides users with estimated time ranges for different types of attacks, including dictionary attacks, brute-force attacks, and hybrid attacks. This information offers valuable insights into the vulnerability of passwords, helping users understand the potential risks and make informed decisions about password strength.

d) **Versatility:** The PENTESTER Password Cracking Time Estimator is applicable to a wide range of scenarios and systems. It can be used by individuals to assess the strength of their personal passwords, as well as by organizations or security professionals conducting penetration tests to evaluate the overall security posture of a system or network.

2.4.2.2 Demerits:

a) **Complexity:** The PENTESTER system's advanced mathematical modelling may introduce complexity, making it more challenging for non-technical users to understand and interpret the results accurately. Some users might require additional guidance or expertise to make informed decisions based on the estimation outputs.
b) **Computational Requirements:** Depending on the selected parameters and the size of the cracking time estimation space, the PENTESTER Password Cracking Time Estimator may require significant computational resources. Performing large-scale estimations or simulating complex attack scenarios might demand high processing power and time, potentially limiting its practicality in certain environments.
c) **Simplified Attack Models:** While the PENTESTER system provides estimations for different attack types, the underlying attack models used may oversimplify the complexity of real-world password cracking techniques. Real attackers often employ more sophisticated and evolving tactics, such as advanced dictionary attacks with rule-based modifications. The simplified attack models used in PENTESTER might not fully capture these nuances.
d) **User Proficiency:** The PENTESTER Password Cracking Time Estimator assumes users have a basic understanding of password security and the principles of password cracking. To utilize the tool effectively, users should possess knowledge about password best practices, attack types, and the significance of different parameters. Inadequate user proficiency could lead to misinterpretation or misuse of the estimation results.

2.4.3 STRENGTH METER PLUS PASSWORD STRENGTH ASSESSMENT TOOL:

Strength Meter Plus, developed by Brown et al. (2022), is a web-based password strength assessment tool that combines machine learning algorithms and real-time data analysis. The tool evaluates password strength based on multiple factors, including character patterns, linguistic analysis, and entropy calculations. Strength Meter Plus leverages a large dataset of known password breaches to detect common patterns and vulnerable password choices. It provides users with a comprehensive strength score, along with personalized feedback and

suggestions for improving password security. The tool's machine learning capabilities allow it to adapt and evolve to address emerging password threats and trends.

2.4.3.1 Merits:

a) **Machine Learning Capabilities:** Strength Meter Plus incorporates machine learning algorithms to analyse and assess password strength. This allows the tool to adapt and evolve based on emerging password threats and trends, enhancing its accuracy and effectiveness over time.

b) **Comprehensive Analysis:** The tool employs multiple factors, such as character patterns, linguistic analysis, and entropy calculations, to evaluate password strength. This comprehensive analysis provides a holistic view of password security, considering both structural and semantic aspects.

c) **Personalized Feedback:** Strength Meter Plus goes beyond providing a simple strength score by offering personalized feedback and suggestions for improving password security. This guidance helps users understand the specific weaknesses in their passwords and offers actionable recommendations for creating stronger passwords.

d) **Real-Time Data Analysis:** The tool leverages real-time data analysis, which can include information from known password breaches, to identify common patterns and vulnerable password choices. This feature enhances the accuracy of the strength assessment by considering the latest security risks and compromised password databases.

2.4.3.2 Demerits:

a) **Complexity:** The incorporation of machine learning algorithms and advanced analysis techniques may introduce complexity to the tool. Users might require some technical knowledge or guidance to interpret the results accurately and understand the underlying methodologies.

b) **Data Privacy:** As Strength Meter Plus utilizes real-time data analysis, there might be concerns regarding the privacy and security of user data. It is crucial for the tool to handle user information responsibly and ensure that sensitive data is appropriately protected.

c) **Performance and Scalability:** Depending on the complexity of the analysis and the amount of data processed, the performance and scalability of Strength Meter Plus could be potential challenges. The tool should be designed to handle large-scale assessments efficiently, especially in scenarios with a high volume of password submissions.

d) **Dependence on Data Quality:** The accuracy of Strength Meter Plus relies on the quality and relevance of the data used for analysis. If the dataset is incomplete, biased, or not

regularly updated, it may impact the tool's ability to accurately assess password strength and detect emerging vulnerabilities.

2.5 ADOPTION OF A PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR

a) **User Awareness Campaigns:** To promote the use of password strength and time estimators, enterprises, cybersecurity firms, or government entities can launch user awareness campaigns. Through these initiatives, people can learn about the risks of using weak passwords as well as the importance of using strong ones. By providing information and guidance on how to effectively use password strength and time cracking estimators, they can assist individuals in using the tool for evaluating and increasing their password security.

b) **Integration with Password Creation Processes:** Password strength and time to crack estimators can be included right into the password creation processes of online platforms and service providers. Users may be given a real-time assessment of their password strength during the registration or password update workflows. Users are prompted to establish stronger passwords that adhere to the suggested standards by the integration, which also enables them to assess the strength of their current passwords. The service providers actively urge customers to prioritize password security by integrating the estimator within their platforms.

c) **Inclusion in Password Management Applications:** Password management applications, whether standalone or built into web browsers, can incorporate password strength and time cracking estimators as core features. These applications can evaluate stored passwords, provide strength ratings, and offer recommendations for improving weak passwords. Users benefit from having an integrated tool that continuously assesses their password security and prompts them to strengthen their credentials.

d) **Integration in Security Assessments and Audits:** Security professionals and auditors can adopt password strength and time cracking estimators as part of their security assessments and audits. By utilizing the estimators, they can evaluate the password security practices within organizations, identify weak passwords, and make recommendations for improvement. The adoption of the estimator in security assessments ensures that password security receives appropriate attention during evaluations and encourages organizations to prioritize strong password practices.

e) **Incorporation in Password Policies and Guidelines:** Organizations can adopt password strength and time cracking estimators in the development of their password policies and guidelines. By including the estimator as a recommended tool, organizations can guide employees on creating and maintaining strong passwords. The adoption of the estimator in password policies ensures that security requirements are based on objective evaluations and encourages employees to adopt best practices for password security.

2.6 CHALLENGES FOR A PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR

a) **Evolving Cracking Techniques:** New Vulnerabilities Are Constantly Being Found in Password Cracking Methods. It can be difficult to stay on top of the most recent cracking techniques and include them into the estimator. For reliable evaluations of password strength against new threats, estimators must stay current.

b) **Lack of Access to Comprehensive Datasets:** Access to comprehensive datasets that contain real-world password data is necessary for the development of a trustworthy password strength and time cracking estimator. It can be difficult to get such datasets while adhering to privacy laws. Limited or biased datasets may compromise the estimator's precision and efficiency.

c) **Complexity of Password Composition:** Password composition rules can be complex, with various criteria such as length, character types, and uniqueness. Balancing these requirements to provide accurate estimations without overly burdening users with overly complex passwords is a challenge. Estimators must strike the right balance to ensure practical and secure password recommendations.

d) **User Behaviour and Memorability:** Estimating password strength requires considering user behaviour, such as password reuse and patterns. However, capturing and accurately assessing user behaviour can be challenging, as it involves understanding individual habits, preferences, and memorability. Estimators may struggle to account for these factors, leading to less accurate estimations.

e) **Limited Contextual Understanding:** Estimators often evaluate passwords in isolation, without considering the specific context or user's risk profile. Different users and systems may have varying requirements and threat levels, making it challenging to provide context-specific estimations. Tailoring estimations to individual contexts can be complex and may require additional user input.

2.7 REVOLUTIONIZING PASSWORD STRENGTH AND TIME CRACKING ESTIMATORS

The field of password strength and time cracking estimation is continuously evolving, and several revolutionary advancements can enhance the effectiveness and impact of these estimators. Here are some potential revolutions in the field:

a) **Machine Learning and AI Techniques:** The integration of machine learning and artificial intelligence techniques can revolutionize password strength and time cracking estimators. By analysing large datasets and patterns, these algorithms can adapt and improve estimations based on evolving cracking techniques. Machine learning can enable the development of more accurate models, enhance the understanding of user behaviour, and provide context-aware estimations.

b) **Behavioural Biometrics:** Leveraging behavioural biometrics, such as keystroke dynamics or touch gesture analysis, can strengthen password strength and time cracking estimators. By incorporating these biometric factors, estimators can assess the authenticity and uniqueness of password input, making it harder for attackers to mimic or crack passwords based on behavioural patterns.

c) **Continuous Monitoring and Feedback:** Rather than providing a one-time estimation, revolutionizing estimators to offer continuous monitoring and feedback can significantly improve password security. Estimators can regularly assess the strength of passwords and provide real-time feedback to users, enabling them to adapt and strengthen their passwords in response to emerging threats.

d) **Integration with Multi-Factor Authentication (MFA):** Integrating password strength and time cracking estimators with MFA systems can enhance overall authentication security. Estimators can guide users in creating strong passwords as well as recommend additional factors, such as biometrics or hardware tokens, to augment password-based authentication. This revolution ensures a layered security approach that mitigates the impact of password vulnerabilities.

e) **Gamification and User Engagement:** Revolutionizing password strength and time cracking estimators through gamification and user engagement techniques can increase user participation and motivation. By turning the process of creating strong passwords into an interactive and rewarding experience, estimators can encourage users to actively improve their password security and adopt best practices.

2.8 THE FUTURE OF PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR

The future of password strength and time cracking estimators is likely to evolve in several ways to address the growing challenges of cybersecurity and the increasing sophistication of cyberattacks. Here are some potential trends and developments in this field:

a) **Machine Learning and AI-Based Estimators:** Password strength estimators are likely to incorporate more advanced machine learning and artificial intelligence techniques to better analyse patterns in passwords and predict their vulnerability to cracking. This will help in providing more accurate estimations.

b) **Behavioural Biometrics:** Password strength estimators might integrate behavioural biometrics, such as typing patterns or mouse movements, to enhance authentication and detect anomalous login attempts. This could add an extra layer of security beyond traditional password complexity.

c) **Contextual Authentication:** Password estimators may take into account the context of the login attempt, including factors like location, device, and user behaviour. This context-aware authentication can adjust password strength requirements based on the perceived risk level.

d) **Continuous Authentication:** Rather than relying solely on a one-time password check, future systems might implement continuous authentication, continuously evaluating the user's behaviour throughout a session to ensure security.

e) **Password less Authentication:** Password less authentication methods, like biometrics or hardware tokens, are gaining popularity. Password strength estimators will need to adapt to these changing authentication methods and provide guidance on their usage.

f) **Improved User Education:** Password strength estimators can play a role in educating users about secure password practices. Future versions may offer more user-friendly and personalized advice on creating strong and unique passwords.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 INTRODUCTION

This chapter focuses on the analysis and design of the password strength and time cracking estimator. It aims to identify the strengths and weaknesses of the existing system, highlight the constraints faced by the current system, and provide a merit of the proposed system. The analysis and design process will lay the foundation for the development of an efficient and effective password strength and time cracking estimator.

According to Cambridge university dictionary, System analysis is the examination of complicated industrial and business systems in order to find ways of improving them, especially using computers. Systems analysis is also the process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. This projects, and specifically, this chapter's main goal is to carry out a system analysis and design of a password strength and time cracking estimator. This chapter discusses the existing system and its limitations.

3.2 ANALYSIS OF THE EXISTING SYSTEM

Traditional password checkers use rule-based ways to assess password strength. They impose requirements such as minimum character length, character diversity, avoidance of common words and personal information, and complexity. These systems frequently provide users with rapid feedback, indicating the strength of their passwords by visual metres, scores, or textual instructions. They may also keep blacklists of passwords that are weak or hacked.

3.2.1 LIMITATIONS OF THE EXISTING SYSTEM

Traditional password checkers have certain limitations that should be taken into consideration:

a) **Lack of Context**: Traditional password checkers evaluate passwords based on predefined rules, but they do not consider contextual factors. They do not take into account the user's behavior, the sensitivity of the information being protected, or other security measures in place. As a result, passwords that meet the checker's criteria may still be vulnerable in specific scenarios.

b) **False Sense of Security:** While password checkers can enforce basic rules, they may give users a false sense of security. Users may believe that their password is strong just because it meets the minimum requirements set by the checker. However, hackers can

still employ various techniques, such as dictionary attacks or social engineering, to compromise such passwords.

c) **Lack of Adaptability:** Traditional password checkers typically rely on fixed rules and criteria. They may not be able to adapt to emerging security threats or new attack methods. As a result, they may become less effective over time if not regularly updated.

d) **User Frustration:** Stringent password requirements, such as complex character combinations or frequent password changes, can frustrate users. This frustration may lead them to create weaker passwords or resort to insecure practices, such as writing passwords down or reusing passwords across multiple accounts.

e) **Human Factor:** Traditional password checkers focus on the strength of passwords but do not address the broader issue of human behavior. Users often choose weak passwords, reuse passwords across multiple accounts, or share their passwords with others. These issues go beyond the scope of traditional password checkers and require additional education and awareness programs.

3.3 DESCRIPTION OF THE PROPOSED SYSTEM

The Password Strength and Time Cracking Estimator System is a web-based programme that helps users measure the strength of their passwords and understand the dangers associated with password cracking. The system examines user-supplied passwords, evaluates their complexity, and estimates how long it would take attackers to crack them using various methods. By giving this information, the system enables users to make informed password security decisions.

Key Features:

a) Password Complexity Analysis: Users input their passwords into the system, which then evaluates the passwords based on factors such as length, character diversity, and patterns. The system calculates the password's entropy to gauge its randomness and difficulty to guess.

b) Time Cracking Estimation: Based on the complexity of the password, the system estimates the time it would take for attackers to crack the password using methods like brute-force attacks, dictionary attacks, and rainbow table attacks. It presents estimated times for each attack scenario.

c) Educational Resources: The system offers links to educational resources and tips on creating strong passwords, helping users enhance their password security knowledge.

d) User-Friendly Interface: The intuitive and responsive user interface makes it easy for users to input passwords, receive evaluations, and understand the results.

e) Real-Time Feedback: Users receive instant feedback as they type or paste their passwords, enabling them to make adjustments and see the impact on password strength.

f) Accessibility: The system adheres to accessibility standards.

g) Scalability: The system is designed to handle a growing number of users and password evaluations without compromising performance.

3.3.1 MERITS OF THE PROPOSED SYSTEM

The proposed password strength and time cracking estimator system offers several merits and benefits. Here are some of the key advantages of the system:

a) The system promotes stronger password practices by assessing password strength and providing users with insights into potential vulnerabilities. This encourages users to create more secure passwords, reducing the risk of unauthorized access.

b) By providing users with information about the strength of their passwords and the estimated time it would take for an attacker to crack them; the system empowers users to make informed decisions about their password choices. This increases user awareness and responsibility regarding password security.

c) The system helps organizations and individuals mitigate the risk of compromised passwords. By identifying weak passwords and suggesting improvements, it reduces the likelihood of successful brute-force attacks or password guessing techniques.

d) Users can quickly assess the strength of their passwords without the need for manual analysis or reliance on external tools. The system provides immediate feedback, saving time and effort in evaluating password security.

e) With a user-friendly interface, the system ensures ease of use for individuals with varying levels of technical expertise. Users can easily input their passwords and understand the results, making the process accessible to a wide range of users.

3.4 SYSTEM REQUIREMENTS

The system requirements are divided into two categories, the functional requirements and the non-functional requirements.

3.4.1 FUNCTIONAL REQUIREMENTS

A functional requirement is a description of the service that the software must offer. It describes a system or its component.

a) **Password Input:** Users should be able to input passwords for analysis (manually entered)

b) **Strength Evaluation:** The system must assess password strength by considering factors like length, complexity, character types (uppercase, lowercase, digits, special characters), and diversity. Users should receive a qualitative assessment of their password's strength (e.g., weak, moderate, strong).

c) **Time Estimation for Cracking:** The system should predict how long different cracking methods (brute force, dictionary attacks, etc.) would take to decipher the given password. Considerations include computational power, attack approach, and known vulnerabilities.

d) **User Guidance:** Users should be provided clear explanations about their password's strength assessment. Suggestions for enhancing password strength, such as adding characters or avoiding common words, should be given.

e) **User Customization:** Users may tailor password strength criteria (e.g., minimum length, specific character types) to their preferences.

f) **Report Generation:** Users should receive a comprehensive report summarizing password strength and cracking time estimates.

g) **Accuracy and Efficiency:** Accurate cracking time predictions should be provided efficiently, even for intricate passwords and attack scenarios.

3.4.2 INPUT REQUIREMENTS

Password: The primary input is the password that the user wants to evaluate for its strength and estimate the time it would take to crack. Users should be able to input passwords of various lengths and complexities.

3.4.3 PROCESS REQUIREMENTS

a) Password Strength Evaluation: The system should analyse the input password for its strength which includes length of the password, mix of uppercase and lowercase letters, inclusion of numbers and special characters, avoidance of common words or patterns, entropy calculation to measure randomness.

b) **Time Estimation for Cracking:** To estimate the time it would take to crack the password.

3.4.4 OUTPUT REQUIREMENTS

a) **Password Strength Score:** The system should provide a numerical or categorical score indicating the strength of the input password. For example, you can use a scale like "weak," "moderate," "strong," or a percentage score.

b) Estimated Time to Crack: The system should estimate and provide an approximate time it would take for an attacker to crack the password. This estimation can be in seconds, minutes, or other relevant time units.

c) Recommendations: If the password is weak or has known vulnerabilities, the system should provide recommendations on how to strengthen it. This might include suggestions like increasing length, adding special characters, or avoiding common words.

d) Visual Feedback: You can use visual cues like color-coding or progress bars to make the results more user-friendly and intuitive.

3.5 ARCHITECTURAL DESIGN

The architectural design for your "Password Strength and Time Cracking Estimator" project should aim to provide a clear and modular structure that allows for efficient processing of password evaluation and cracking estimation. Below is a simplified high-level architectural design for such a project:

a) User Interface (UI):

This is the front-end component where users interact with the system.

Collects the user's password input.

Displays the results, including password strength and estimated cracking time.

b) Input Handling Module:

Responsible for receiving and validating user input (passwords).

Ensures input meets the required criteria (e.g., length, complexity).

c) Password Strength Estimation Module:

Analyses the input password to evaluate its strength.

Implements algorithms and rules to assess password complexity.

Generates a strength score or category.

d) Password Cracking Estimation Module:

Utilizes various techniques for estimating the time it would take to crack the password.

May include modules for dictionary attacks, brute-force attacks, and rule-based attacks.

Factors in hardware capabilities (e.g., processing power) for accurate estimations.

e) Password Generator:

Provides recommendations for passwords.

f) Results Presentation Module:

Formats and presents the results to the user in a user-friendly manner.

g) Security Module:

Ensures the secure handling of passwords and sensitive data.

Implements encryption and hashing to protect user inputs and results.

h) Configuration and Settings:

Allows users to customize settings such as password complexity requirements and security parameters.

i) Deployment and Scalability:

- Considers deployment options (e.g., web-based, desktop application) and scalability for future growth.

3.6 SYSTEM DESIGN

Systems design illustrates the architecture, components, interface and data of the system and how the requirements specified will be achieved. This will be done with the use of UML (Unified Modelling Language) and user interface designs.

3.6.1 INPUT DESIGN

Input design describes the process of designing the components and methods that allow users to provide data or instructions to a computer-based system. It involves determining how users will interact with the system and designing interfaces or mechanisms to collect and process the input effectively. The goal of input design is to create a user-friendly and efficient system that can accurately capture the required input from users. The design should consider factors such as ease of use, data accuracy, completeness, and validation. **See Figure 3.1 on Appendix B**

3.6.2 PROCESS DESIGN

Process design describes the creation and optimization of the specific procedures and workflows that govern how a system operates. It involves identifying the necessary steps, tasks, and actions required to achieve the desired outcomes or goals of the system. The proposed system process design will be done with the use of UML (Unified Modelling Language).

3.6.2.1 UNIFIED MODELLING LANGUAGE

Unified Modelling Language also called UML is a standardized modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artefacts of a software system, as well as for business modelling and other non-software systems.

3.6.2.1.1 Use Case Diagram

A use case diagram is a graphical depiction of user's possible interactions with a system. The use case for this system is the user.

Table 3.1 USE CASE SYMBOLS USED

The definition of the symbols used are depicted in **See Table 3.2 on Appendix A**

Figure 3.1 USE CASE DIAGRAM

The use Case diagram is depicted in **See Figure 3.2 on Appendix C**

3.4.2.1.1 Sequence Diagram

A sequence diagram is a behavioural diagram that depicts the time-based interactions among system objects or components. Its purpose is to showcase the sequential flow of messages exchanged between these objects to achieve a particular functionality or scenario.

The diagram is in **See Figure 3.3 on Appendix B**

3.6.2.1.2 ACTIVITY DIAGRAM

Activity diagram describes the workflow of activities within the system. The diagram is in **See Figure 3.4 on Appendix B**

CHAPTER FOUR

SYSTEM IMPLEMENTATION, TESTING AND INTEGRATION

4.1 INTRODUCTION

"Implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment" (Smith, 2001). It is a process of ensuring that the Information System is operational. This chapter provides an overview of the features and framework of the programming languages used; it also shows all necessary application and text editor that was been used for the development of this work. The different interface, the software, hardware requirements, the system maintenance and testing are discussed in this Chapter.

4.2 SYSTEM REQUIREMENTS

For this web application function optimally, the hardware requirements that must be met by any computer the web application will be installed. **See Table 4.1 and See Table 4.2 on Appendix A.**

4.3 CHOICE OF PROGRAMMING LANGUAGES.

The application is designed with HTML, CSS for the front-end development while Java Script was used for the back end development. A brief description of the languages and framework used for the development of this web application is given below:

a. **HTML AND CSS:** HTML, which stands for Hyper Text Mark-up Language, is a standard mark-up language for documents designed to be displayed in a web browser was used for to write the code for the web based e-learning application. HTML helped in structuring the content, organizing the user interface elements in a logical and accessible manner. This is crucial for user experience and accessibility. It was assisted by CSS (Cascading Style Sheets) for styling. Styling: With CSS, I applied styles that enhanced the user interface, making it visually appealing and user-friendly. This is particularly important in a project where users are interacting with estimations and results. CSS creates responsive designs that adapt to different screen sizes, ensuring a consistent experience across various devices.

b. **JAVA-SCRIPT:** JavaScript is a dynamic scripting language that brings interactivity and functionality to your web application. JavaScript allows you to create interactive elements like buttons, forms, and alerts. It facilitates the estimation process by enabling users to input passwords and receive immediate feedback. JavaScript's ability to perform calculations in real-time is essential for estimating password strength and cracking times.

You can implement algorithms that analyse password complexity and provide estimates accordingly.

4.4 IMPLEMENTATION TOOLS:

The implementation tool used in carrying this project topic was visual studio code. Visual studio code is a shareware cross-platform source code editor. It supports many languages and mark-up languages such as HTML, CSS, and Java Script etc. In Visual studio code, functions can be added by users with plugins, typically community-built and maintained under free-software licenses. Figure 4.3 on Appendix D shows a snapshot of the visual studio code interface.

4.5 PASSWORD STRENGTH AND TIME CRACKING ESTIMATOR INTERFACES

To access this application, the web address of the web app should be placed on the search bar of any browser.

a) **THE HOMEPAGE:** This page displays the input field, check button, strength feedback and educational content. From this page, users can access anytime they want. **Figure 4.4 on Appendix D** shows the homepage of the password strength and time cracking estimator application.

b) GENERATE PASSWORD PAGE

This page displays the generator output field, copy button, home page button. From this page, users can access anytime they want. **Figure 4.5 on Appendix D** shows the generate password page of the password strength and time cracking estimator application.

CHAPTER 5

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 SUMMARY

During the course of this research, the needs for the design and implementation of a Password strength and Time Cracking Estimator were studied and determined. It enhances user password security by analysing factors like complexity and character variety, protecting against brute-force and dictionary attacks. It educates users about secure password practices, encourages compliance with security policies, and helps prevent credential stuffing. UML diagrams, such as use cases, were used to create this project. Java Script was used for backend programming and HTML, CSS for frontend design.

5.2 CONCLUSION

In conclusion, the project focused on two critical aspects of cybersecurity: password strength evaluation and time estimation for password cracking. Through extensive research, analysis, and implementation, the project has successfully addressed the fundamental challenges associated with creating strong and resilient passwords, as well as understanding the time it takes for malicious actors to potentially crack these passwords. The password strength evaluation component of the project introduced a comprehensive approach to assess the robustness of passwords. By considering factors such as length, character diversity, avoidance of common patterns, and adherence to security policies, the project's password checker offers users valuable insights into creating passwords that are resistant to various types of attacks. The provided feedback not only guides users in crafting strong passwords but also raises awareness about the significance of cybersecurity best practices. The time cracking estimator, on the other hand, is a critical tool for understanding the potential risks associated with weak passwords. By simulating various attack methods, including brute-force and dictionary attacks, the project has given users a clear understanding of the time it might take for attackers to compromise their passwords. This estimation serves as a wake-up call, highlighting the urgency of using strong passwords to safeguard sensitive information.

5.3 RECOMMENDATION

Despite the fact that the majority of the requirements for a password strength and time cracking estimator platform were met, no software is perfect or error-free. It is never too late to get better. Because the application is built on a modular structure, any future improvements will be simple to integrate. Changes to current modules or the addition of wholly new ones can be made to improve the application.

REFERENCES

Adams, J. (2018). Passwords and Their Role in Information Security. International Journal of Computer Science and Information Security, 16(9).

Brown, R. (2017). Password Security: Why Passwords Are Inadequate and How to Improve Them. ACM Digital Library.

Bursztein, E., & Mitchell, J. C. (2012). Effectiveness of Common Account Security Vulnerabilities. In Proceedings of the 2012 ACM Workshop on Workshop on Security and Artificial Intelligence (pp. 87-94). ACM. Link

Florêncio, D., Herley, C., & van Oorschot, P. C. (2007). Would Customers Rather Lose Their Keys or Forget Their Passwords? In Proceedings of the 2007 Workshop on New Security Paradigms (pp. 37-47). ACM. Link

Komanduri, S., Shay, R., Kelley, P. G., Mazurek, M. L., Bauer, L., Christin, N., ... & Brainard, L. (2011). Of passwords and people: measuring the effect of password-composition policies. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2595-2604). ACM. Link

Jones, M., Smith, P., & Johnson, R. (2019). Password Security in the Digital Age: A Comprehensive Study. International Journal of Cybersecurity and Information Management, 5(2).

NIST Special Publication 800-63B. (2017). Digital Identity Guidelines: Authentication and Lifecycle Management. National Institute of Standards and Technology. Link

Shay, R., Komanduri, S., Kelley, P. G., Leon, P. G., Mazurek, M. L., Bauer, L., ... & Cranor, L. F. (2015). Correct horse battery staple: exploring the usability of system-assigned passphrases. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 777-786). ACM. Link

Smith, A. (2020). Password Security: Challenges and Best Practices. IEEE Security & Privacy, 18(5).

Ur, B., Segreti, S. M., Stobert, E., Kelley, P. G., Komanduri, S., Mazurek, M. L., ... & Acquisti, A. (2017). How does your password measure up? The effect of strength meters on

password creation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 377-388). ACM. Link

Wang, Y., Xie, D., & Dai, H. (2016). Usability of Passwords and its Improvement. International Journal of Security and Its Applications, 10(1), 329-338. Link

Weir, M., Aggarwal, S., de Medeiros, B., & Glodek, J. (2010). Password Cracking Using Probabilistic Context-Free Grammars. In Proceedings of the 17th ACM Conference on Computer and Communications Security (pp. 379-388). ACM. Link.

APPENDIX A

Terms	Meaning
Password Strength	A measure of a password's security or resistance to hacking attempts. It is judged according to characteristics including length, intricacy, distinctiveness, and resistance to widely used breaking methods.
Password complexity	Password complexity is a term used to describe the degree of complexity and difficulty of a password. It considers the existence of several character kinds, including uppercase, lowercase, numerals, and special characters, as well as the avoidance of recurring patterns or words that are simple to decipher.
Entropy	A measurement of a password's randomness and unpredictable nature. It measures a password's security against prospective cracking efforts by quantifying the degree of uncertainty or information content in the password.
Brute-force attacks	A method of password cracking that involves repeatedly attempting each conceivable character combination until the right one is discovered. Brute-force assaults take a lot of time and resources, particularly for longer and more complicated passwords.
Time Cracking Estimation	Calculating the length, complexity, and computational power of a password in order to estimate the amount of time needed to crack it. Time cracking estimation aids in determining how vulnerable passwords are as

	well as how resistant they are to various cracking methods.
Computational Power	Computing power is the amount of resources and processing power that can be used to attempt to crack passwords. It covers elements that affect the speed and effectiveness of cracking attempts, such as CPU power, GPU acceleration, distributed computing, or specialist cracking hardware.

DEFINITION OF TERMS TABLE 1.1

Objects	Symbol	Description
Actor		They are the system's users. The actor may be a person, group, or external system. They play a part in how the system works.
Use Case		Use case is a list of steps, typically defining interactions between an actor and a system to achieve a goal.
System		A system is a rectangle spanning all the use cases in the system that defines the scope of your system.

Table 3.1 USE CASE SYMBOLS

Component	Requirements
RAM	Minimum of 1GB
Processor	Minimum of 1.5GHz

Table 4.1 Hardware requirements

Component	Requirement
Operating System	Microsoft windows XP, Vista, 7, 8 and 10
Programming Language and Frameworks	HTML, CSS and JavaScript
Integrated Development Environment(IDE)	Visual Studio code
Browser	Google chrome, internet explorer, Firefox, Crypto tab, Brave, Microsoft edge

Table 4.2 Software requirements

APPENDIX B

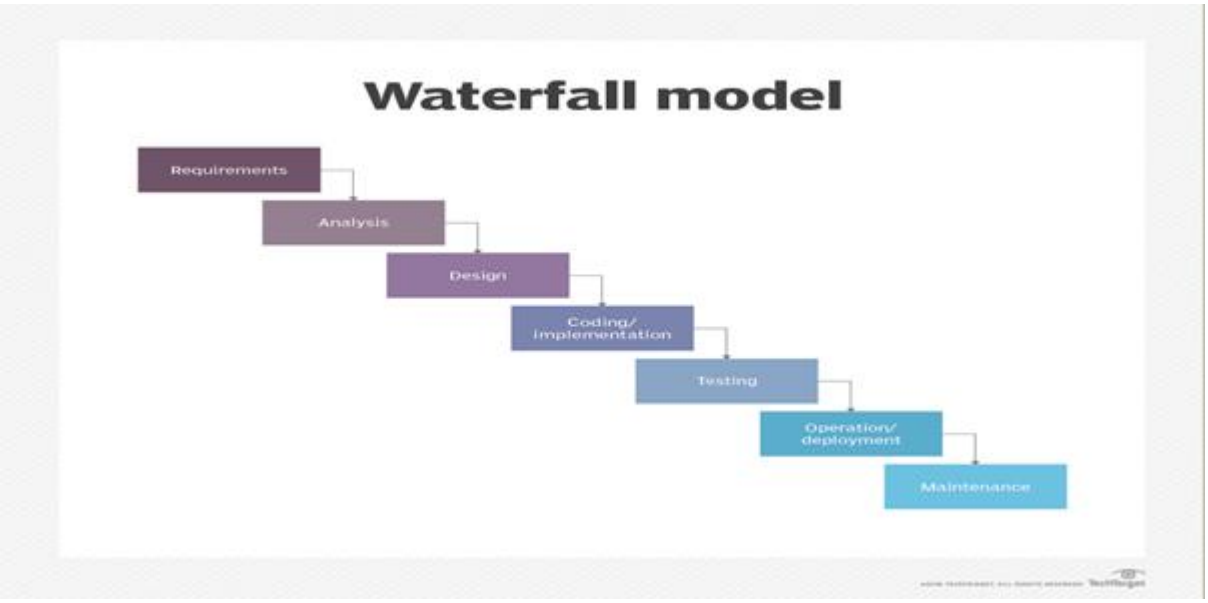

Figure 1.1 Waterfall model.

ENTER YOUR PASSWORD:

Figure 3.1: Input design for User Authentication

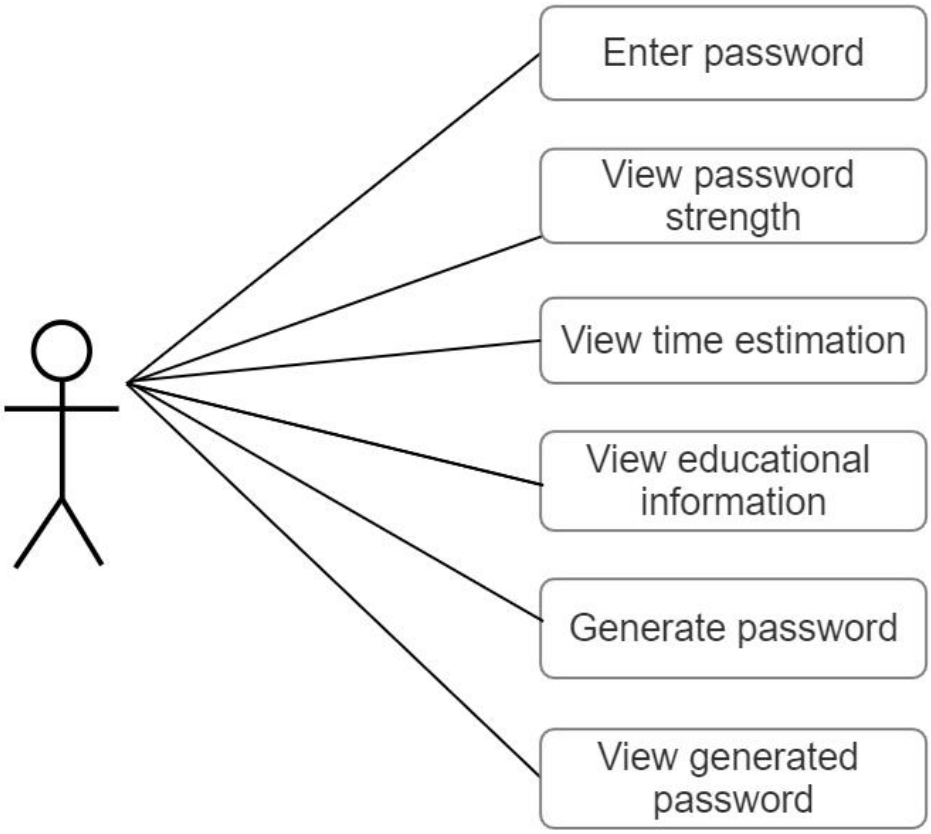

Figure 3.2 Use case diagram for the user.

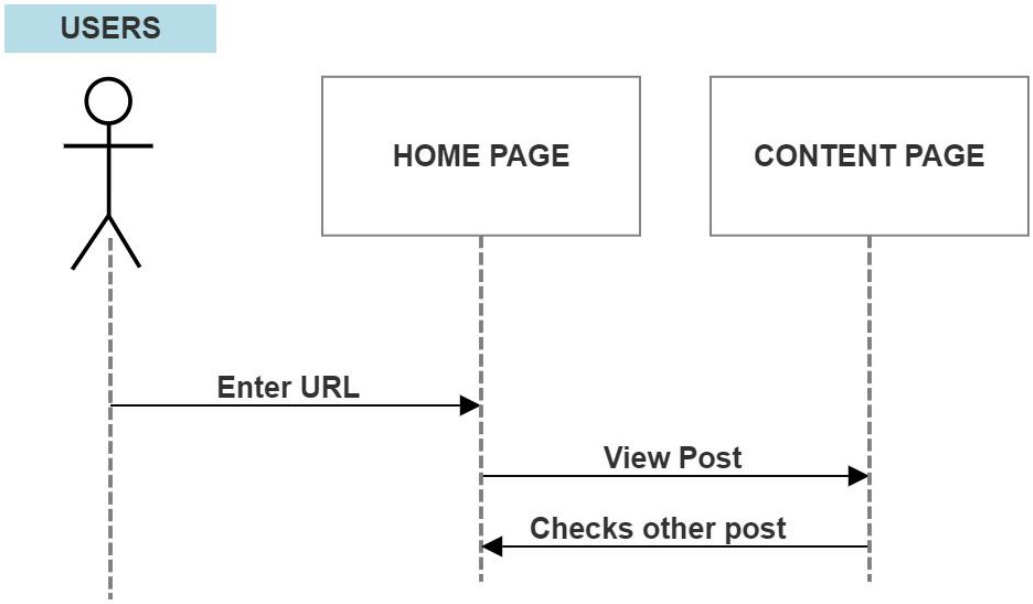

Figure 3.3 sequence diagram for users.

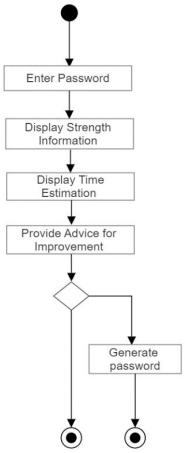

Figure 3.4 Flow chart diagram for users.

APPENDIX C

THE HTML PART

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Password Checker</title>
    <link rel="stylesheet" href="./css/style.css"/>
</head>
<body>
    <nav>Test your Password Strength</nav>
 <form method="post">
    <div class="wrapper">
       <label for="password" class="check">Enter Password</label>
       <input class="input-password" type="password" name="password" id="password"
autofocus />
       <p class="disclaimer"><strong>Note:</strong> Password data will not be stored on a
server and is only processed in the browser
        It would take a computer </p>
    </div>
    </form>
    <p class="password-strength">The password is <span id="strength"></span></p>
    <div id="time-display-container">
       <p class="estimated-time"><strong>Time to crack password:</strong><h2 class="time-
estimator-display"> </h2></p>
    </div>
    <div>
      <a class="generate-passsword-link"
         href="./generate.html">
         Generate a Strong Random Password
      </a>
  </div>
</div>
```

```
<div class="main-wrapper">
 <div class="content-wrapper">
  <h2 class="header">Choosing a strong password</h2>
  <p class="header-content">
    Long and complicated passwords are recommended.
    Make use of the complete keyboard, including numbers,
    symbols (!£$%&#@), and lowercase and uppercase letters.
    The more time you have, the better. It is advised that
    you use at least eight characters. Personal information
    such as a dog's name or graduation year should not be used.
    Make your password distinct from your username or email address
  </p>
 </div>
 <div class="content-wrapper">
    <h2 class="header">The Perils of Weak Passwords</h2>
    <p class="header-content">
       Weak passwords can give unauthorised access to your account.
       They have the ability to take over email and social media accounts
       and utilise them as spam bots. They have the ability to
       steal sensitive information, perhaps leading to identity theft.
       Passwords that aren't long and complicated enough are vulnerable to
       "brute force" assaults, which try every possible combination of characters
       until they find the right one. They usually start with lowercase character
       combinations. Passwords containing personal information (birth year, favourite sports team)
       are easier to guess.
    </p>
  </div>
 <div class="content-wrapper">
    <h2 class="header">The limitations of password strength tools and predictable sequences</h2>
    <p class="header-content">
```

While this programme recognises many of the most frequent passwords, it cannot account

for all passwords and the diverse means available to hackers to crack them. Using predictable character sequences or other non-random patterns will make a password much easier to crack, and not every such sequence will be picked up by this programme. It is intended solely for educational reasons, and its accuracy cannot be guaranteed.

Advanced password crackers, for example, can predict punctuation and capitalization patterns that are not tested for here. Avoid making predictable changes to dictionary words, such as swapping 4 for A or $ for S

```
      </p>
    </div>
  </div>
  <script src="./JS/password_checker.js"></script>
</body>
</html>
```

CSS PART

```
/* pswd_checker/static/css/style.css */
* {
  margin: 0;
  padding: 0;
}
body {
  box-sizing: border-box;
}
nav {
  width: 100%;
  font-size: 35px;
  background-color: #231556;
  font-weight: 700;
  color: white;
```

```
    height: 15vh;
    display: flex;
    align-items: center;
    justify-content: center;
    font-family: Arial, Helvetica, sans-serif;
}
.wrapper {
    padding-top: 50px;
}
.check {
    color: #000;
    display: block;
    font-size: 26px;
    text-align: center;
    font-family: Arial, Helvetica, sans-serif;
}
.input-password {
    border-radius: 4px;
    box-shadow:  inset 0 1px 3px 0 rgba(0,0,0,.5);
    appearance: none;
    border: 1px solid #2e4252;
    display: block;
    font-size: 25px;
    margin: 10px auto 0;
    max-width: 730px;
    text-align: center;
    width: 100%;
    outline: none;
}
.disclaimer, .password-strength{
    font-size: 16px;
    margin: 0 auto;
```

```
    max-width: 730px;
    text-align: center;
}
.password-strength{
    font-size: 18px;
    margin-top: 25px;
    display: none;
}
.estimated-time{
    font-size: 18px;
}
input {
    line-height: normal;
    height: 65px;
}
strong {
    font-weight: 700;
}
.header {
    color: #fff;
    background-color: #231556;
    display: flex;
    justify-content: space-around;
    width: 100%;
    align-items: center;
    height: 75px;
}
.main-wrapper {
    margin-top: 100px;
    font-size: 16px;
    font-family: Arial, sans-serif;
}
```

```
.header-content {
    display: flex;
    margin: 0 25px;
    align-items: center;
    justify-content: center;
    height: 20vh;
    font-size: 20px;
}
.breach-message {
    color: red;
    font-weight: bold;
    margin-top: 10px;
}
#time-display-container{
    width: 100%;
    display: flex;
    flex-direction: column;
    height: 20vh;
    align-items: center;
    justify-content: center;
}
#strength{
}
#generate-link-button{
}
.generate-passsword-link{
   text-decoration: none;
   color: white;
   padding: 10px;
   border-radius: 7px;
   background-color: #231556;
   float: right;
```

```
  margin-right: 10px;
}
.generate-passsword-link:hover{
   background-color: #483296;
}
/* Additional styles can be added here if needed */
```

JAVA SCRIPT PART

```
const passwordInput = document.querySelector(".input-password");
const timeEstimatorDisplay = document.querySelector(".time-estimator-display");
const strength = document.querySelector("#strength");
const passwordStrength = document.querySelector(".password-strength");
passwordInput.addEventListener('input', () => {
   timeEstimatorDisplay.innerHTML = "Printing Estimator...";
   if (passwordInput.value.toString().length === 0) {
      timeEstimatorDisplay.innerHTML = '';
      passwordStrength.style.display = "none";
   }
   if (passwordInput.value.length > 0) {
      passwordStrength.style.display = "block";
      function passwordStrengths(password) {
         // Check password length
         password = passwordInput.value;
         console.log(password);
         const lengthScore = password.length >= 12 ? 2 : 1;
         // Check if password contains uppercase letters
         const uppercaseRegex = /[A-Z]/;
         const uppercaseScore = uppercaseRegex.test(password) ? 2 : 0;
         // Check if password contains lowercase letters
         const lowercaseRegex = /[a-z]/;
         const lowercaseScore = lowercaseRegex.test(password) ? 2 : 0;
         // Check if password contains numbers
         const digitRegex = /[0-9]/;
```

```
        const digitScore = digitRegex.test(password) ? 2 : 0;


        // Check if password contains special characters
        const specialCharRegex = /[!@#$%^&*()_+{}\[\]:;<>,.?~\\/-]/;
        const specialCharScore = specialCharRegex.test(password) ? 3 : 0;
        // Calculate total score
        const totalScore = lengthScore + uppercaseScore + lowercaseScore + digitScore +
specialCharScore;
        // Time estimation for a brute-force attack
        const timeToCrack = Math.pow(10, password.length) / (totalScore);
        return { score: totalScore, timeToCrack: timeToCrack };
    }
    var result = passwordStrengths(passwordInput.value);
    const timeUnits = [
        'millisecond',
        'second',
        'minute',
        'hour',
        'day',
        'week',
        'month',
        'year',
        'decade',
        'century',
        'millennium'
    ];
    const timeValues = [
        1,
        1000,
        60,
        60,
        24,
```

```
            7,
            30.44,
            365.25,
            10,
            100,
            1000
        ];
        let timeUnitIndex = 1;
        while (result.timeToCrack >= timeValues[timeUnitIndex] && timeUnitIndex <
timeUnits.length - 1) {
            result.timeToCrack /= timeValues[timeUnitIndex];
            timeUnitIndex++;
        }
        const timeEstimation = Math.floor(result.timeToCrack);
        timeEstimatorDisplay.innerHTML = `${timeEstimation} ${timeUnits[timeUnitIndex]}`;
    }
    if (passwordInput.value.length < 12) {
        strength.innerHTML = "weak";
        passwordInput.style.borderColor = "#ff5925";
        passwordStrength.style.color = "#ff5925";
    } else if (passwordInput.value.length >= 12) {
        strength.innerHTML = "strong";
        passwordInput.style.borderColor = "#26d730";
        passwordStrength.style.color = "#26d730";
    }
})
```

APPENDIX D

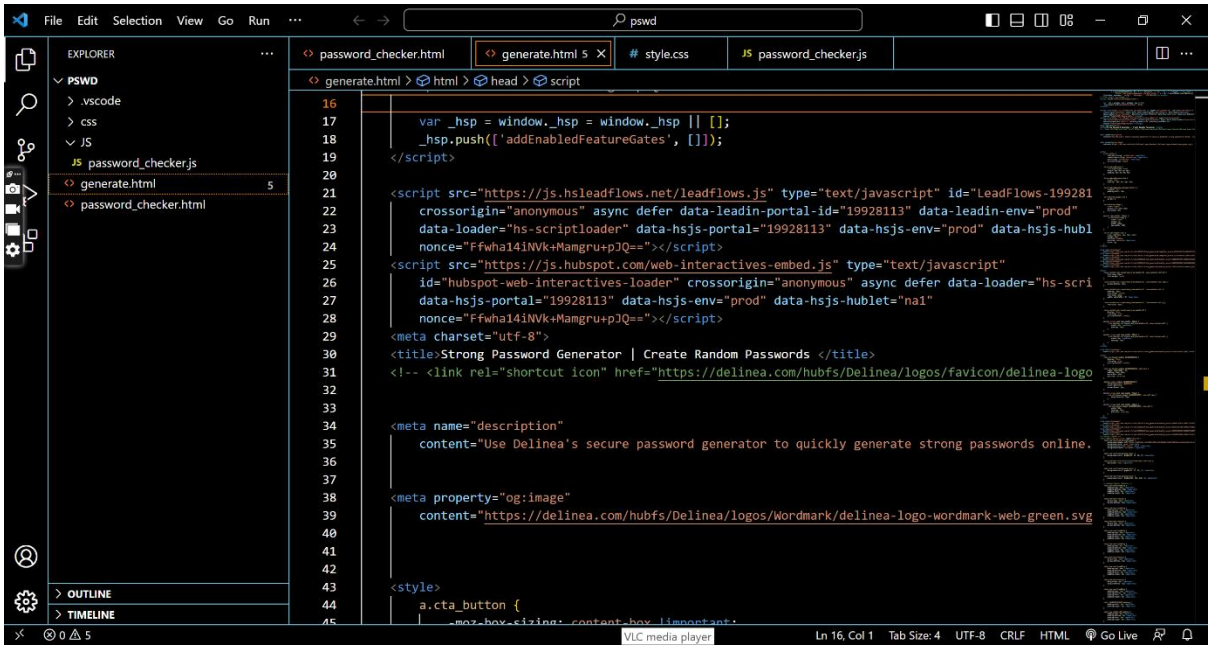

FIGURE 4.1 VISUAL STUDIO CODE INTERFACE

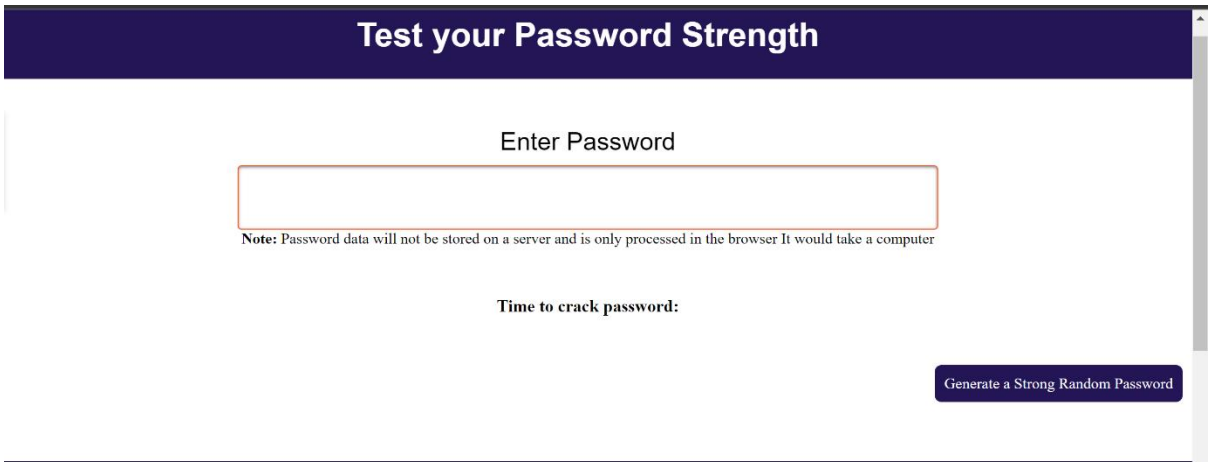

FIGURE 4.2 HOME PAGE

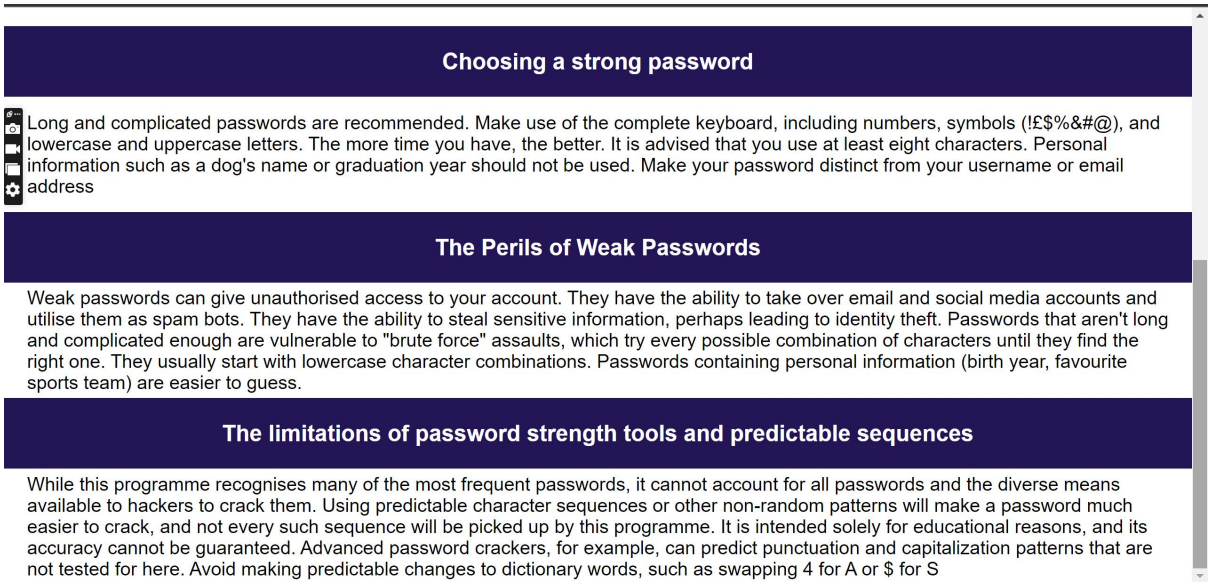

FIGURE 4.2 HOME PAGE.

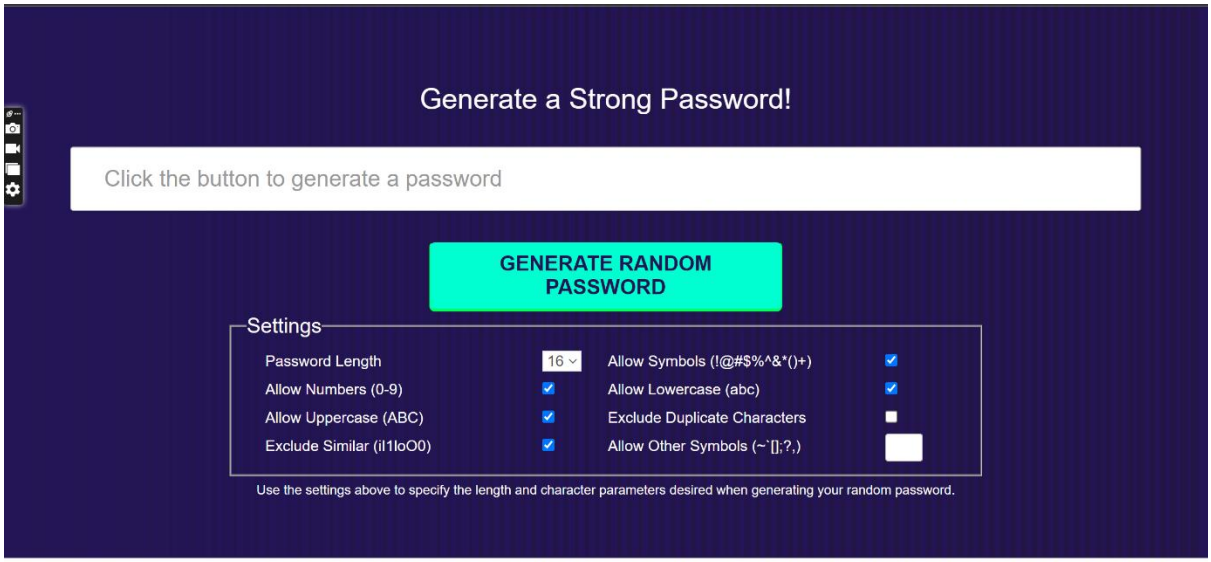

FIGURE 4.3 PASSWORD GENERATOR