SECURE PASSWORD MANAGER AND PASSWORD TOOLS

KINGSLEY NAING SCOTT

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

SECURE PASSWORD MANAGER AND PASSWORD TOOLS

KINGSLEY NAING SCOTT

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

SECURE PASSWORD MANAGER AND PASSWORD TOOLS

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ________________ ________________ Date : 6/9/2024

(KINGSLEY NAING SCOTT)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Computer Security) with Honours.

SUPERVISOR : ________________________________ Date : 6/9/2024

(TS. DR. MOHD. FAIRUZ ISKANDAR OTHMAN)

DEDICATION

This report is being presented to the Faculty of Information and Communication Technology at Universiti Teknikal Malaysia Melaka (UTeM) to meet the criteria for my final year project. This report signifies the conclusion of my diligent efforts, practical encounters, and expertise gained during my tenure at this university. I would want to express my gratitude to all the lecturers who have provided guidance and support during my educational journey. I would want to express my gratitude to my parents and family, who have provided unwavering support throughout my academic journey. Their support has played a crucial role in moulding my character and establishing my identity. With the knowledge I have acquired, I have effectively finished my degree project and hereby give this report as proof of my achievements.

ACKNOWLEDGEMENT

.

I would like to start by expressing my gratitude to my family especially my parents who have given me their unwavering support throughout my life and in this case my time here in this fine institution UTeM. I am indeed humbled and grateful for the opportunities bestowed onto me and I consciously try to go out of my way to make the best of it.

I would like to extend my appreciation to the knowledgeable Dr Fairuz my supervisor, I have had a few encounters with him before and he has always been consistent and very informative. I am grateful for having him as my supervisor as I would consider him a subject expert in the field of Computer Science. I have greatly benefited from his constructive suggestions.

I would like to sincerely thank Assoc. Prof. Gs. Dr. Othman Mohd, my evaluator, for his thoughtful assessment and valuable critique of this project. His feedback has played a key role in shaping my ideas and improving the overall quality of the work.

I want to express my deepest gratitude to my friends, especially Garcia and Zarif, for their incredible support and contributions to my final year project. Their advice, feedback, and encouragement were truly invaluable in helping me bring this project to life. To all my friends who have been there for me throughout this journey, thank you for your constant support and motivation. I am truly blessed to have you all by my side.

ABSTRACT

The Secure Password Manager and Password Tools is a web-based project that aims to provide a secure and convenient solution for managing passwords. With the increasing number of online accounts and the need for strong, unique passwords, password management has become crucial in maintaining digital security. This project utilizes web-based programming languages such as PHP, JavaScript and HTML just to name a few. This is to develop a user-friendly and simplistic interface that allows users to generate, check and store passwords. The password manager will be a web application where users can register, login, delete and update an account and securely store their site passwords. The password tools will be is developed and integrated into a Google Chrome extension, this includes a password generator and checker. This provides users to have easier access to generate strong passwords and check new or existing ones at any time. To fortify the security of user data, the master password used for accessing the password manager is hashed using the bcrypt algorithm. Additionally, two-factor authentication using Google Authenticator is implemented to enhance login security. To protect the confidentiality of stored site passwords, the project incorporates the AES (Advanced Encryption Standard) algorithm for encryption. This industry-standard encryption technique ensures that the site passwords are securely stored and remain inaccessible to unauthorized individuals.

TABLE OF CONTENT

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

2FA	-	Two Factor Authentication
AES	-	Advanced Encryption Standard

CHAPTER 1: INTRODUCTION

1.1 Introduction

Passwords are the most essential part of security may it be physical or digital security. In a fast-paced evolving world, a strong password can go a long way in securing your digital accounts. Nowadays passwords have certain requirements to ensure their strength and security, while this is great at catering to cyber-attacks it may pose a challenge to users in memorizing such complex passwords. To further expand on this is remembering and storing numerous usernames and passwords for various accounts. This is the objective of this project, to build a Secure Password Manager and Password Tools. This would be a web-based program that can store and managed passwords using security methods such as two factor authentications, hashing and encryption. Alongside this program will be a set of password tools that will be able to generate strong passwords and a password strength checker. The password manager and tools can also be accessed through a google chrome extension. This can be done by leveraging programming languages such as HTML, PHP, SQL and more. The password manager and tools will prioritize user experience and security.

1.2 Problem Statement

With the fast advancement of technology, the increase in password attacks poses a serious threat to digital security. Internet users in general are at risk no matter old or young, everyone is susceptible to password attacks, susceptibility to password attacks depends more on an individual's awareness, knowledge, and habits regarding cybersecurity. Habits such as prevalence of weak password choices and risky behaviours like password reuse and sharing. Part of the reason to this behaviour is the ability to memorize such a password and the complexity of using full-fledged password managers that has paid features that is essential for security. This leaves users vulnerable to various attacks, including dictionary and brute force attacks. To address these issues, this project aims to develop a solution. A user-friendly and simplistic yet robust password manager that implements security measures such as two-factor authentication, password hashing and encryption. Part of the solution also includes chrome extension implementation of the password manager tools offering password generation and a password strength checker.

Table 1.1 Summary of Problem Statement

PS	Problem Statement
PS1	Most internet users use weak password combinations, making them easy targets for password attacks such as dictionary attacks or brute force attacks.
PS2	There is a need for robust security measures in password managers to prevent data leakage or unauthorized access.
PS3	The complexity and paid features of existing password managers make them less accessible, leaving users vulnerable to security risks.

1.3 Project Question

Based on the problem statement above, the following questions arise to address the identified issues and objectives. These questions will guide the development of The Secure Password Manager and Password Tools ensuring both user-friendliness and enhanced security measures.

Table 1.2 Project Question

PS	PQ	Project Question
PS1	PQ1	How to create an effective password manager and password tools?
PS2	PQ2	What security features should be integrated into the password manager to ensure sufficient protection?
PS3	PQ3	How can we create effective password tools, such as a generator and strength checker, that are user-friendly and secure?

1.4 Project Objective

The project aims to address the problem statements by achieving the following objectives. To study the current methods and technologies used in password management and security, including password generation, strength checking, and encryption techniques. To design a user-friendly and simplistic interface for the password manager that allows for easy creation, storage, and management of passwords, along with a secure system architecture incorporating advanced security features such as two-factor authentication (2FA), password hashing, and encryption and to develop a robust password manager that securely stores and manages user passwords for multiple accounts and platforms. This includes a Google Chrome extension that will ease users in accessing passwords for accounts in a quick manner. Along with this would be the password tools which includes a random password generator that will greatly increase security and can address attacks such as brute force or dictionary attacks. Besides that, another password tool would be a robust password checker to ensure new and current passwords are strong and secure.

Table 1.3 Project Objective

PS	PQ	PO	Project Objective
PS1	PQ1	PO1	To research effective methods and technologies used in password management and security, including password generation, strength checking, and encryption techniques.
PS2	PQ2	PO2	To develop a user-friendly and secure password manager and password tools incorporating advanced security features such as two-factor authentication, password hashing, and encryption.
PS3	PQ3	PO3	To evaluate the effectiveness and user experience of the developed password manager and tools, ensuring they meet the objectives of enhancing security and usability.

1.5 Project Scope

1.5.1 Random Password Generation.

The Google extension will feature a robust password generator that utilizes parameters set by the user such as uppercase, lowercase, digits and symbols to create strong passwords. Developed with HTML and JavaScript,

1.5.2 Password Strength Checker.

To assist users in evaluating the robustness of their passwords, the Google extension will include a password strength checker. This feature will utilize a visual bar indicator to display the strength of passwords, providing real-time feedback on their security level. By assessing factors such as length, complexity, and use of diverse character sets, the strength checker will help users create passwords that meet recommended security standards, reducing the risk of password-related vulnerabilities.

1.5.3 Password Management.

Users will have access to a user-friendly interface designed for efficient password management. This interface will allow users to create, view, update, and organize their stored passwords effortlessly. The interface will be intuitive, ensuring that even users with limited technical knowledge can manage their passwords effectively and securely.

1.5.4 Two-Factor Authentication (2FA).

To enhance login security, the system will integrate two-factor authentication (2FA). This feature will require users to provide a secondary form of verification, such as a unique code generated by a mobile application like Google Authenticator. By adding this additional layer of security, the system will significantly reduce the risk of unauthorized access, even if a user's password is compromised. 2FA ensures that only authenticated users can access their stored passwords and sensitive information.

1.5.5 Master Password Hashing.

To protect user master passwords, the password manager will employ strong password hashing techniques, such as bcrypt. Hashing will convert passwords into irreversible hash values, making it exceedingly difficult for attackers to decipher or compromise user credentials. By using secure hashing algorithms, the system ensures that master passwords are stored safely, preventing unauthorized access even in the event of a data breach.

1.5.6 Site Password Encryption.

To further safeguard stored passwords, the system will utilize advanced encryption algorithms, such as AES (Advanced Encryption Standard). This encryption process will secure passwords within the database, ensuring they can only be accessed with the appropriate decryption key. By encrypting passwords, the system enhances overall data confidentiality and integrity, protecting user information from unauthorized access and potential data breaches.

1.6 Project Contribution

Implementing a strong password strategy is crucial in protecting user accounts from being breached by attackers. The password generator and management system will streamline the process of creating and managing secure passwords, making it less likely for attackers to succeed with methods such as dictionary attacks and brute force attacks.

Table 1.4 Project Contribution

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC1	Research on current methods and technologies for password management and security will provide a foundation for developing a secure and effective password manager.
PS2	PQ2	PO2	PC2	Development of a password manager and tools with advanced security features such as two-factor authentication, password hashing, and encryption will enhance user password security.
PS3	PQ3	PO3	PC3	Evaluation of the developed password manager and tools will ensure they are user-friendly and effective in enhancing password security, leading to improved digital security for users.

1.7 Report Organization

Chapter 1: Introduction

Introduction chapter discusses about the overall picture of the project. This chapter also explains about the scope and explains the gist of the whole project.

Chapter 2: Literature Review

In this chapter, it will discuss about the problem statement in more detail. Next, this chapter will also discuss the findings found in research papers that are related to this topic. Things that are required to be inserted here are such as citation of the research papers.

Chapter 3: Project Methodology

Project Methodology will discuss the method completing this project. It will follow the project milestone given to make sure that each chapter are completed in time.

Chapter 4: Analysis and Design

Design chapter show the prototype of the application of the project and how the password can be managed and used in the user's account. For example, it will show the process of how the password generated.

Chapter 5: Implementation

Implementation chapter will show about the development of the secure password manager and tools. It includes the requirement and tools used in the development of the application.

Chapter 6: Testing

Testing and Analysis chapter will test the secure password manager and tools and analysis the output result from the testing.

Chapter 7: Project Conclusion

Conclusion chapter captures the result of the whole project. Starting from the design until testing and analysis chapter. This chapter will also conclude whether the project is a success or vice versa.

1.8 Conclusion

The introduction chapter presents a comprehensive outline of the project's objective to improve digital security by creating a robust password manager and related technologies. This solution tackles the present difficulties that users encounter while generating and handling robust passwords, as well as the dangers associated with feeble passwords and sophisticated password attacks. The project aims to provide a user-friendly solution that incorporates sophisticated security mechanisms, like two-factor authentication, password hashing, and encryption, to safeguard user credentials.

The problem statements highlight the importance of efficient password management and strong security features, while the project questions direct the development process towards resolving these concerns. The project aims to provide a secure password generator, an easy to understand password manager, and a Google Chrome extension to make it more accessible.

The initiative seeks to greatly improve the digital security of users by applying these technologies, hence facilitating password management and safeguarding online accounts. In the upcoming chapter, we will thoroughly explore the literature study, doing a comprehensive analysis of current research and technology pertaining to the secure storage of passwords.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter discusses the literature review for the project. A literature review is a summary of previous research on a particular topic. It looks through scientific journals, books, and other sources that are relevant to the study. The literature review lists, defines, summarizes, critically assesses, and clarifies this previous work. It serves as a theoretical framework for the research and helps to determine the scope of the study. By acknowledging the findings of previous researchers, the literature review ensures that the reader understands that the current work is well-informed.

It is assumed that the author has read, analyzed, and absorbed previous work on the subject by mentioning it in the current work. A literature review provides the reader with an overview of the field's progress, allowing them to better understand it.

2.2 Related Work/Previous Work

2.2.1 Password authentication

Password authentication remains a fundamental aspect of computer security despite its inherent vulnerabilities. According to Taneski et al. (2019), alphanumeric passwords, although widely used, have long been recognized as a weak point in information system security due to their susceptibility to being short, simple, and easily compromised through dictionary attacks. Their systematic literature review indicates that while some studies propose solutions to these issues, few provide statistically significant results, underscoring the persistent problem of weak password security behaviour among users. Issues such as password reuse, writing down passwords, and poor understanding of secure password practices continue to undermine password security.

Proactive password checking, as explored by Proctor et al. (2002), offers a method to mitigate these weaknesses by enforcing restrictions on password creation. Their research demonstrates that while additional restrictions increase the time required to generate passwords, they significantly reduce the crackability of passwords. In their experiments, increasing the minimum password length and adding complexity requirements substantially improved security, indicating that more stringent password policies can enhance the robustness of password authentication systems.

2.2.2 Character Combination

Character combination refers to the complexity of passwords, which includes the use of uppercase and lowercase letters, numbers, and special characters. This approach aims to create passwords that are difficult for attackers to guess or crack using brute force methods.

Studies by Shay et al. (2010) indicate that users often struggle with creating complex passwords that meet security policies. They found that while users understand the need for secure passwords, the complexity requirements can lead to frustration and reduced compliance. This issue is exacerbated when users are required to change their passwords frequently, leading to patterns or reuse that undermine the security benefits of complexity requirements.

Bonneau and Preibusch (2010) explored the impact of password policies on user behavior and password strength. Their findings suggest that while policies requiring a mix of character types do increase password strength, they also significantly increase the cognitive load on users. This often results in users employing coping mechanisms, such as writing down passwords or using similar passwords across multiple accounts.

Yan et al. (2004) proposed alternative methods to enhance password security without overly burdening users. One approach is to use passphrases, which are longer and composed of multiple words, making them easier to remember yet difficult to crack. Their research shows that passphrases can provide a good balance between security and usability, although they are not without challenges, such as the potential for predictable structures.

2.2.3 Password Manager

Password managers are widely recommended by security experts as tools that enhance both the security and usability of password management. Despite these recommendations, their adoption and effective use remain limited.

Pearman et al. (2019) conducted a semi-structured interview study involving 30 participants to explore the mindsets behind the adoption and effective use of password managers. Their study included participants who used no password-specific tools, those who used built-in browser or operating system password managers, and those who used separately installed password manager applications. The findings highlighted a significant difference in motivations: users of built-in managers prioritized convenience, whereas users of standalone tools were more security focused. The study suggested tailored designs to cater to these different user mentalities and provided actionable suggestions to improve the adoption and effective usage of password managers. Furthermore, it was observed that users of built-in managers tend to use weaker and reused passwords compared to those using standalone managers with integrated password generation features.

Li et al. (2014) conducted a security analysis of five popular web-based password managers and identified critical security vulnerabilities in four out of the five managers studied. These vulnerabilities allowed attackers to potentially gain access to users' credentials for arbitrary websites. The study identified issues ranging from logic and authorization mistakes to misunderstandings about web security models. The diversity of these vulnerabilities underscored the need for a defence-in-depth approach to ensure the security of password managers. Their findings emphasized that while password managers are generally considered beneficial for security, they also introduce unique risks that must be carefully managed.

In summary, the literature indicates that while password managers offer significant benefits in terms of security and convenience, their adoption is hindered by usability concerns, security vulnerabilities, and varying user perceptions of their necessity and trustworthiness. Future efforts should focus on addressing these barriers by improving the usability and security of password managers and by educating users on their benefits and proper use.

2.2.4 Password Generator

Password generators create complex and unique passwords to enhance security. Studies have shown that these tools can significantly reduce the risk of password-related breaches. For example, research by Golla et al. (2019) demonstrated that password generators help users create strong passwords that are less susceptible to brute force and dictionary attacks. Their study also found that generated passwords tend to be more secure compared to those created manually by users, who often fall into predictable patterns.

However, the usability of password generators remains a challenge. As noted by Kelley et al. (2012), users often find it difficult to remember complex, randomly generated passwords, leading to reliance on password managers for storage and retrieval. This dependency introduces a trade-off between security and convenience, as users must trust the security of the password manager itself.

Additionally, some research suggests that while password generators can improve security, their effectiveness is contingent upon user adoption and proper use. Pearman et al. (2019) highlighted that users who do not fully understand the benefits of password generators or who find them cumbersome are less likely to use them consistently. This underscores the importance of user education and interface design in promoting the adoption of secure password practices.

Manoj Ramashish Gupta and his colleagues from the University of Mumbai explored this issue in their mini project, "Password Generator as Chrome Extension." They developed a password generator that allows users to create complex passwords by selecting letters and numbers, making these passwords easier to remember. Their approach underscores the importance of user-friendly design in the adoption of secure password generation practices.

Further supporting this perspective, Zibaei et al. (2022) conducted a study on the efficacy of nudges used by web browsers' built-in password managers to encourage the adoption of randomly generated passwords. They found that Safari's implementation was significantly more effective than those of Chrome and Firefox. This suggests that thoughtful design and user experience enhancements can significantly impact the adoption rates of secure practices like using password generators.

In conclusion, password generators play a critical role in enhancing password security by creating complex and unique passwords. However, their effectiveness is influenced by usability factors and user behaviour, necessitating further research and development to improve their adoption and usability.

2.2.5 Password Hashing

The bcrypt hashing function, created by Niels Provos and David Mazières, is extensively employed for enhancing password security. The name of this system is derived from the combination of "Blowfish" and "crypt," which is the hashing function utilised by the UNIX password system. By employing bcrypt, a password security framework may be established that dynamically adapts to computational capabilities and automatically incorporates salt for each password. This guarantees heightened security against password decoding and theft. Bcrypt's strong resistance to hacking, especially against rainbow table password cracking, making it a secure option for protecting user passwords, unlike readily compromised encryption approaches (Sriramya & Karthika, 2015). Bcrypt provides several security features: • Salt Integration: Each password is hashed with a unique salt, preventing attackers from using precomputed tables to crack the passwords. • Adaptability: Bcrypt allows for the configuration of the cost factor (number of iterations), making it scalable with increasing computational power. • Memory Intensive: The algorithm includes a computationally expensive key setup phase, making it resistant to attacks using specialized hardware like GPUs and FPGAs (Provos & Mazieres, 1999). Bcrypt's method involves generating a salt, combining it with the password, and applying the Blowfish cipher in multiple iterations. The resulting output includes the version identifier, cost factor, salt, and hash value, ensuring secure and verifiable password storage.

2.2.6 AES Encryption

The Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm, renowned for its robustness and efficiency. Developed by NIST in 2001, AES has become the standard for encrypting sensitive data due to its strong security features and performance benefits.

AES operates on fixed block sizes of 128 bits, with key sizes of 128, 192, or 256 bits. The algorithm employs a series of transformations, including substitution, permutation, mixing, and key addition, performed in multiple rounds depending on the key size (10, 12, or 14 rounds). These operations provide a high level of security against various cryptographic attacks.

Key Features and Security

AES's strength lies in its complex key schedule and resistance to common attacks such as brute force, differential, and linear cryptanalysis. The algorithm's design ensures that even small changes to the plaintext or the key result in vastly different ciphertexts, a property known as the avalanche effect.

According to Lu and Mohamed (2021), AES's performance can be further enhanced by optimizing the key schedule and reducing the number of rounds, while still maintaining security. Their study suggests integrating AES with the RSA algorithm to leverage the strengths of both symmetric and asymmetric encryption. This hybrid approach enhances key management and security functions, particularly for secure file sharing over insecure channels.

Implementation and Performance

Implementing AES in Java, as discussed by Lu and Mohamed, takes advantage of Java's extensive library support, enabling efficient encryption and decryption operations. The researchers compared AES's performance with other algorithms by evaluating parameters such as encryption/decryption speed, entropy, and memory consumption. The results indicated that the optimized AES algorithm performs well across these metrics, demonstrating its suitability for high-security applications.

Practical Applications

- AES is extensively used in various security protocols and applications, including:

- Secure Communications: AES secures data transmitted over networks, ensuring confidentiality and integrity.
- Storage Security: Encrypting sensitive data stored on devices and in databases protects against unauthorized access.
- Cryptographic Services: AES serves as a core component in many cryptographic protocols, including SSL/TLS for secure web browsing and IPsec for secure network communications.

2.2.7 Two Factor Authentication

Two-factor authentication (2FA) is a security process in which users provide two different authentication factors to verify themselves. This method enhances security by requiring not only a password and username but also something that only the user has on them, such as a physical token or a mobile device. 2FA is considered a subset of multi-factor authentication (MFA), which involves using multiple distinct categories of credentials to confirm user identity.

The Evolution and Implementation of 2FA

Traditional single-factor authentication, primarily using passwords, has been found insufficient to safeguard sensitive information due to the rise in sophisticated cyber-attacks like phishing and password breaches. 2FA introduces an additional layer of security, making unauthorized access more difficult.

One innovative approach to enhancing 2FA security is combining honeytokens and Google Authenticator, as proposed by Papaspirou et al. (2023). Honeytokens are decoy passwords designed to detect unauthorized access attempts. When integrated with Google Authenticator, which provides a time-based one-time password (TOTP), this method significantly enhances security. The proposed system operates like a multi-factor authentication system by mixing honeytokens with real passwords, making it challenging for attackers to identify the correct password even if they gain access to the password file. The combination of these techniques offers a higher security level while maintaining user-friendliness.

Benefits of 2FA

1. Enhanced Security: By requiring two forms of verification, 2FA makes it significantly harder for attackers to gain unauthorized access.
2. Phishing Resistance: Even if a password is compromised, the second factor, such as a TOTP or a hardware token, prevents unauthorized entry.
3. Usability and Adoption: Modern implementations, like those integrating mobile apps for TOTP, balance security with user convenience.

Challenges and Considerations

While 2FA provides robust security, it is not without challenges. Usability concerns, such as the inconvenience of carrying an additional device or the complexity of setup, can hinder widespread adoption. Additionally, some advanced attacks, like SIM swapping, can still compromise 2FA systems.

Advanced 2FA Systems

Papaspirou et al. (2023) highlight the novel use of honeytokens in 2FA to provide additional security layers. Their system not only incorporates Google Authenticator for generating TOTPs but also embeds honeytokens within the authentication process to alert administrators of any unauthorized access attempts. This combination offers a highly secure yet user-friendly authentication mechanism, resilient against various threats like SIM swapping and side-channel attacks.)

2.3 Critical review of current problem and justification

Passwords are a cornerstone of modern digital security, safeguarding access to personal and sensitive information across various online platforms. Despite their critical role, passwords pose significant challenges due to their inherent vulnerabilities and user management difficulties. One major problem is the complexity required for strong passwords, which makes them difficult to remember. Users often cope with this by reusing passwords across multiple accounts or writing them down, thus compromising security (Bonneau & Preibusch, 2010).

Password generators and managers have emerged as essential tools to address these issues. A password generator creates random, complex passwords that are difficult to guess, while a password manager securely stores and organizes these passwords, reducing the cognitive load on users (Pearman et al., 2019). However, many current solutions fall short in several key areas:

1. Password Generation: Some password generators produce passwords that are not sufficiently complex or random, leaving users vulnerable to attacks. Moreover, users may lack understanding of how to create secure passwords, leading to weak security practices (Taneski et al., 2019).
2. Password Management: Existing password managers may lack essential features like password strength checking, easy updates, or a user-friendly interface, which can deter effective use. The usability of these tools is crucial, as complicated interfaces can discourage users from adopting them fully (Shay et al., 2010).
3. Integration and Usability: The integration of password generators and managers in a seamless, user-friendly interface is often inadequate. Users need a cohesive solution that combines strong password generation with easy management to encourage proper security practices (Pearman et al., 2019).

By addressing these gaps, my project aims to develop a robust and user-friendly Secure Password Manager and Password Tools. This tool will enhance password security by combining advanced password generation capabilities with intuitive management features, ultimately simplifying the user's interaction with password security and encouraging better practices.

2.4 Proposed Solution/Further project

The proposed solution involves developing a secure password manager and password tools using a combination of web-based programming languages, such as HTML, PHP, JavaScript, and jQuery. The project will utilize the Agile Software Development Life Cycle

(SDLC) methodology to ensure flexibility and continuous improvement throughout the development process.

Key Features of the Proposed Solution:

1. Password Generation:
    - Options for different levels of complexity and length.
    - Inclusion of special characters, upper and lowercase letters, and numbers.
    - User-friendly interface for generating strong and secure passwords.
2. Password Management:
    - Secure storage of passwords with easy retrieval through a master password.
    - Features to create, view, update, and delete passwords effortlessly.
    - Ability to generate and store passwords for multiple accounts and platforms.
3. User Interface:
    - Intuitive and easy-to-navigate design to promote user adoption.
    - Real-time password strength indicators and suggestions for improving security.

Agile Methodology Justification:

The Agile methodology is well-suited for this project due to its iterative nature and flexibility, allowing for continuous feedback and improvement. Agile facilitates:

- Adaptive Planning: Adjusting requirements as the project evolves based on user feedback and emerging needs.
- Incremental Development: Delivering functional components in stages, ensuring each part is thoroughly tested and validated.
- Collaboration and Communication: Promoting close cooperation between developers and stakeholders, ensuring the final product meets user expectations and security standards (Beck et al., 2001).

2.5 Comparative Studies

2.5.1 Feature Comparison Table

Feature	LastPass	Bitwarden	Dashlane	1Password
Random Password Generation	Yes, customizable settings for length and character types	Yes, customizable settings for length and character types	Yes, customizable settings, including pronounceable passwords	Yes, customizable settings, including memorable passwords
Password Strength Checker	Yes, visual feedback with a strength meter	Yes, real-time feedback with a strength meter	Yes, visual feedback and detailed security improvement suggestions	Yes, real-time feedback with a strength indicator
Password Management	Yes, comprehensive and user-friendly interface	Yes, clean and intuitive interface	Yes, elegant interface with automatic password change for supported sites	Yes, highly intuitive and organized interface
Two-Factor Authentication (2FA)	Yes, integrates with Google Authenticator,	Yes, supports TOTP apps and	Yes, integrates with Google	Yes, supports TOTP and

	Authy, and Duo	hardware tokens	Authenticator and Authy	hardware tokens
Master Password Hashing	Yes, PBKDF2-SHA256 with high iteration count	Yes, PBKDF2-SHA256 with high iteration count	Yes, PBKDF2-SHA256	Yes, PBKDF2-SHA256
Site Password Encryption	Yes, advanced encryption algorithms (AES)	Yes, advanced encryption algorithms (AES)	Yes, advanced encryption algorithms (AES)	Yes, advanced encryption algorithms (AES)

LastPass, Bitwarden, Dashlane, and 1Password are prominent password managers renowned for their strong features designed to improve password security and organisation. Each platform provides a range of functionalities, beginning with their capacity to generate passwords at random. LastPass and Bitwarden offer configurable options for password length and character composition, allowing users to generate robust and distinctive passwords that are personalised to their requirements. Dashlane provides additional features including pronounceable passwords to accommodate various user preferences, whereas 1Password prioritises the creation of memorable passwords that strike a balance between security and usability (Li et al., 2014; Carreira et al., 2020; Oesch & Ruoti, 2020; Pearman et al., 2019).

Regarding the evaluation of password strength, all four password managers excel in delivering immediate feedback to users. LastPass and Bitwarden utilise visual strength indicators, which enable users to assess the strength of their passwords as they are being generated or modified. Dashlane expands this functionality by providing customers with comprehensive security enhancement recommendations, instructing them on how to further fortify their passwords. 1Password provides a strength indicator that updates in real-time, allowing users to accurately evaluate and enhance the security of their login information (Li et al., 2014; Carreira et al., 2020; Oesch & Ruoti, 2020; Pearman et al., 2019).

When it comes to interfaces for managing passwords, LastPass is notable for its extensive and user-friendly design. It makes it easy to store and retrieve passwords across many platforms and devices. Bitwarden features a streamlined and user-friendly design that improves the user's experience by combining simplicity and functionality. Dashlane provides a sophisticated interface that not only handles passwords but also automates password changes for compatible websites, offering increased convenience and security. 1Password is renowned for its very user-friendly interface, which efficiently organises passwords and other confidential data in a well-structured manner. This enhances both accessibility and security, as supported by studies conducted by Li et al. (2014), Carreira et al. (2020), Oesch and Ruoti (2020), and Pearman et al. (2019).

All four platforms offer two-factor authentication (2FA) as an extra security mechanism to enhance account security. LastPass effortlessly interacts with popular authentication applications such as Google Authenticator, Authy, and Duo, providing versatile and strong two-factor authentication (2FA) choices. Bitwarden offers compatibility for a range of time-based one-time password (TOTP) applications and hardware tokens, which improves security by enabling multi-factor authentication. Dashlane seamlessly incorporates Google Authenticator and Authy, augmenting the security measures for user accounts. 1Password has support for TOTP (Time-Based One-Time Password) and hardware tokens, providing users with flexible and dependable choices for two-factor authentication (2FA) to protect their login credentials (Li et al., 2014; Carreira et al., 2020; Oesch & Ruoti, 2020; Pearman et al., 2019).

In addition, all of these password managers utilise robust approaches for hashing the master password in order to safeguard user credentials. LastPass and Bitwarden employ PBKDF2-SHA256 with a significant number of iterations, which strengthens the security of master passwords by making them more resistant to brute-force attacks. Dashlane utilises the PBKDF2-SHA256 algorithm to securely hash passwords, guaranteeing strong security of master passwords. 1Password also implements PBKDF2-SHA256, demonstrating its dedication to strong security measures (Li et al., 2014; Carreira et al., 2020; Oesch & Ruoti, 2020; Pearman et al., 2019).

LastPass, Bitwarden, Dashlane, and 1Password are password management tools that offer a wide range of capabilities to cater to different user requirements. These tools prioritise both security and usability, aiming to enhance online safety. (Li et al., 2014; Carreira et al., 2020; Oesch & Ruoti, 2020; Pearman et al., 2019). These platforms are constantly improving, integrating sophisticated security features and user-friendly interfaces to offer complete password management solutions in the current digital environment.

2.6 Conclusion

In conclusion, this project aims to develop a secure, user-friendly password generator and manager. By leveraging web-based technologies and following the Agile SDLC methodology, the tool will address current shortcomings in password security and management. This solution will enable users to generate strong passwords and manage them effectively, thereby enhancing their overall security posture. The next chapter will delve into the methodology that will be employed to bring this project to fruition.

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

This chapter provides an overview of the project methodology employed in the development of the secure password manager and password tools. The selected methodology guarantees a methodical and organised approach, facilitating comprehensive planning, development, testing, and deployment. The Agile Software Development Life Cycle (SDLC) technique has been chosen for this project because of its iterative nature and flexibility, which make it well-suited for managing evolving requirements and promoting continuous improvement through feedback.

3.2 Methodology

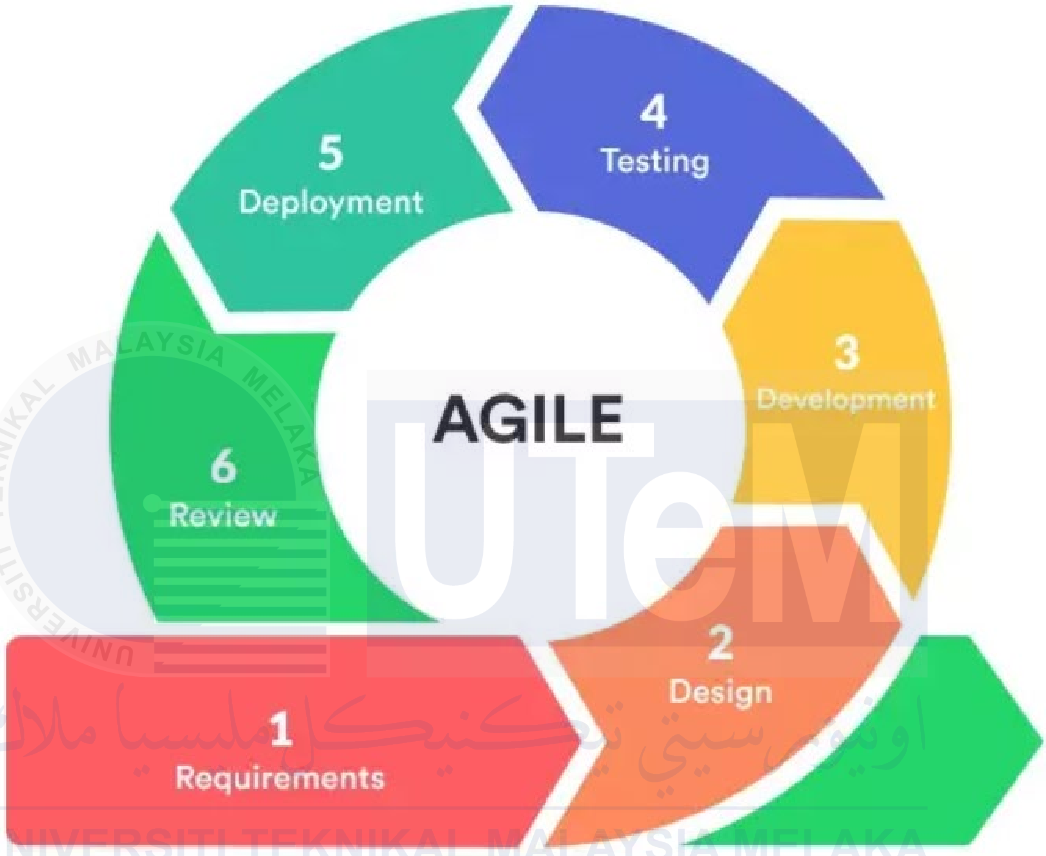


Figure 3.1 Methodology

Explanation of Activities in Relation to the Project:

- Requirement Gathering and Analysis: This phase involves understanding the requirements of users who need secure and manageable passwords. It ensures that all essential features are identified and documented.

- Design: The design phase creates the entire architecture of the system, ensuring that both the user interface and the backend are properly organized and capable of efficiently managing secure data.

- Coding (Implementation): This phase involves the actual development of the system, where all components are implemented, and the source code is created.

- Testing: Comprehensive testing is performed to identify and fix any faulty parts of the code. Security testing is crucial for this project to protect sensitive user data.

- Deployment: After testing, the software is deployed to end users in a secure and stable environment. Proper deployment protocols and thorough documentation ensure a smooth transition to live implementation.

- Maintenance: Continuous maintenance is necessary to address any emerging security threats or user issues promptly, ensuring the long-term effectiveness and security of the application.

By adopting the Agile SDLC methodology, this project aims to deliver a high-quality, secure password manager and password tools that meets user needs and adapts to evolving security challenges.

3.3 Project Milestones

A project milestone is used to identify the various stages of the project timetable through a scheduling method based off Agile SDLC. This will help in establishing project goals and ensuring that the execution goes well.

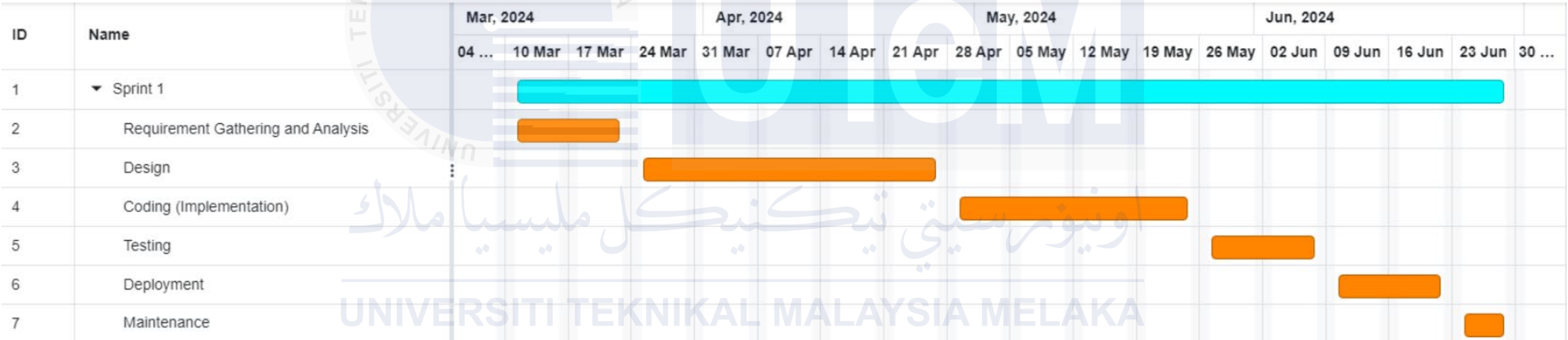
Table 3.1 Milestones Table

Process/Phases	Activities	Completion Date
Requirement Gathering and Analysis	Week 1 (11/03 → 15/03) • Choosing project topic • Proposal discussion, assessment, and verification Week 2 (18/03 → 22/03) • Proposal correction, approval, and submission	Start Date: 11/03 End Date: 22/03
Design	Week 3 (25/03 → 29/03) • Chapter 1 writing • Meeting 2	Start Date: 25/03 End Date: 26/04

	Week 4 (01/04 → 05/04) • Chapter 1 writing • Report Writing Progress 1 • Log progress – ePSM Week 5 (08/04 → 12/04) • Chapter 2 writing Week 6 (15/04 → 19/04) • MID-SEMESTER BREAK Week 7 (22/04 → 26/04) • Chapter 2 writing • Report Writing Progress • Project Progress 1 • Log progress – ePSM • Deliverable – Chapter 2 – ePSM • Progress Presentation 1 (KP1)	
Coding (Implementation)	Week 8 (29/04 → 03/05) • Chapter 3 writing Week 9 (06/05 → 10/05) • Chapter 3 writing • Report Writing Progress • Log progress – ePSM • Deliverable – Chapter 3 – ePSM Week 10 (13/05 → 17/05) • Chapter 4 writing • Project Progress 2	Start Date: 29/04 End Date: 24/05

	• Meeting with supervisor • Log progress – ePSM • Progress Presentation 2 (KP2) Week 11 (20/05 → 24/05) • Chapter 4 writing • Report Writing Progress 2 • Log progress – ePSM • Deliverable – Chapter 4 – ePSM	
Testing	Weeks 12-13 (27/05 → 07/06) • PSM1 Draft Report preparation	Start Date: 27/05 End Date: 07/06
Deployment	Week 14 (10/06 → 14/06) • PSM1 Draft Report submission to SV & Evaluator • Report Evaluation • Log Progress – ePSM • Deliverable – Complete PSM1 Draft Report – ePSM Week 15 (17/06 → 21/06) • FINAL PRESENTATION	Start Date: 10/06 End Date: 21/06
Maintenance	Week 16 (24/06 → 28/06) • PSM 1 Demo and Report Presentation to Supervisor & Evaluator • Presentation Skill • Submission of PSM 1 documents to PSM supervisor, evaluator, and committee in ePSM	Start Date: 24/06 End Date: 28/06

	• Log Record – ePSM • Submission of logbook in ePSM • Submission of Project Report PSM 1 to ePSM	



ID	Name
1	Sprint 1
2	Requirement Gathering and Analysis
3	Design
4	Coding (Implementation)
5	Testing
6	Deployment
7	Maintenance

Figure 3.2 Gantt Chart

3.4 Conclusion

This chapter has presented the Agile Software Development Life Cycle (SDLC) approach as it is used to the development of a secure password manager and password tools. The project follows a systematic approach by going through iterative phases such as Requirement Gathering and Analysis, Design, Coding (Implementation), Testing, Deployment, and Maintenance. This approach allows for careful planning, development, and deployment of the project, while also taking into account changing requirements and user feedback.

The project milestones, outlined in Table 3.1 and depicted in Figure 3.2 Gantt Chart, establish a well-organized schedule for accomplishing important deliverables and milestones, guaranteeing effective project management and assessment.

The project intends to produce a secure and user-friendly application that fulfils current security standards and adapts to future problems through continuous improvement and proactive maintenance by adopting Agile Software Development Life Cycle (SDLC). This methodology fosters flexibility, responsiveness to change, and rigorous security measures throughout the software development process.

CHAPTER 4: ANALYSIS AND DESIGN

4.1 Introduction

This chapter covers the analysis and design of the Secure Password Manager and Tools project. It details the methodologies used, the design of the system architecture, and the planned user interfaces. The goal is to create a comprehensive blueprint that will guide the development and implementation phases, ensuring that all components of the system function cohesively and meet user needs for security and usability.

4.2 Problem Analysis

One significant issue with password authentication is the tendency for individuals to use weak passwords that are easily guessed or hacked. This vulnerability can lead to unauthorized access to systems. Additionally, the reuse of passwords across multiple platforms exacerbates the problem, as a breach in one system can compromise others. Recording passwords or sharing them with others further increases the risk of unauthorized access.

A password generator and manager is a tool designed to securely store and oversee user credentials. The system can generate robust passwords for users and securely store and manage them for various websites or applications. This feature is beneficial as it enables the use of strong and unique passwords without the need to memorize them all.

Password attacks, such as dictionary attacks, brute force attacks, and rainbow table attacks, exploit weak passwords. To safeguard against these attacks, it is crucial to use strong, unique passwords and ensure they are stored securely and not shared.

4.3 Requirement analysis

4.3.1 Data Requirement

First-time users must create an account by providing their full name, phone number, email, username and password. Users can set up two-factor authentication (2FA) using a secret key or QR code in the Google Authenticator application and verify the code, although this step is optional. Before being stored in the database, passwords will be hashed. Users need to enter their email and password to log in. If 2FA is enabled, users must verify the time-based code from Google Authenticator. When adding a new password in the manager, the password will be encrypted.

Table 4.1 Data dictionary of tbl_user

USER				
Attribute	Description	Data Type and Size	Primary Key	Constraints
tbl_user_id	Unique identifier	INTEGER(11)	PK	Not Null
name	Full name	VARCHAR(255)		Not Null
phone_number	Contact number	VARCHAR(255)		Not Null
email_address	Email address	VARCHAR(255)		Not Null
username	Username	VARCHAR(255)		Not Null
password	Hashed password	VARCHAR(255)		Not Null

Table 4.2 Data dictionary of tbl_accounts

USER				
Attribute	Description	Data Type and Size	Primary Key	Constraints
tbl_account_id	Unique identifier	INTEGER(11)	PK	Not Null
tbl_user_id	User identifier	INTEGER(11)		Not Null
account_name	Name of the account	VARCHAR(255)		Not Null

username	Username for the account	VARCHAR(255)		Not Null
password	Encrypted password	VARCHAR(255)		Not Null
link	URL or application name	text		Not Null
description	Description of the account	text		Not Null

4.3.2 Functional Requirement

Functional requirements describe the system’s functionality, including inputs, behaviors, and outputs. The use case diagram below shows that the Secure Password Manager and Tools offers system authentication, password generation, and password management.

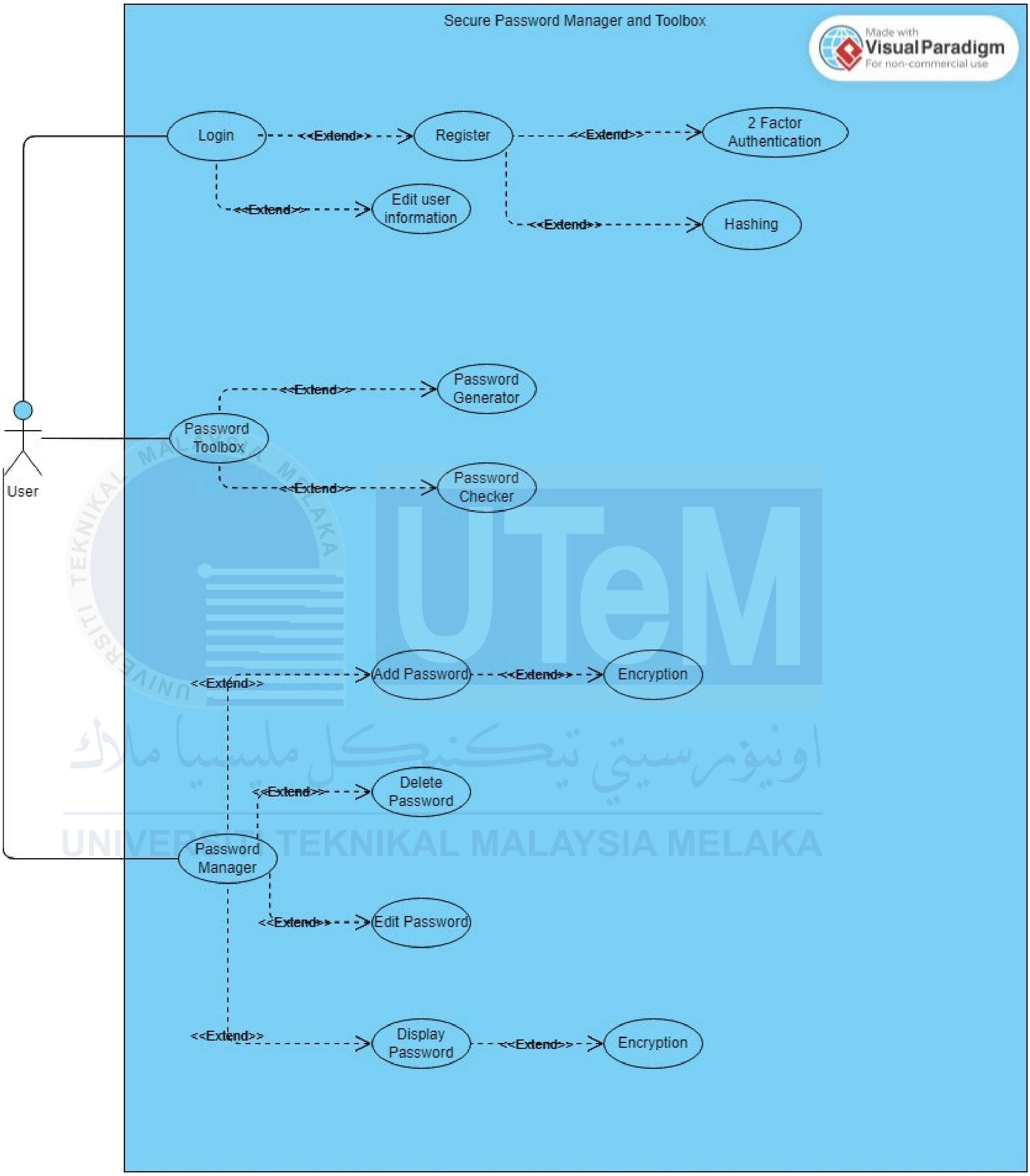


Figure 4.2 Use Case diagram of the proposed application

4.3.3 Non-functional Requirement

Non-functional requirements are essential for the effective operation and management of a system, despite not being directly tied to its functionality. These requirements encompass aspects like performance, security, scalability, maintainability, and more, which are necessary for the system's proper functioning and oversight.

Table 4.3 Non-Functional Requirement Table

Non-Functional Requirement	Description
Performance	The system should minimize response time and enhance speed and efficiency, potentially by optimizing code or ensuring capability to handle substantial requests per second.
Security	The system must prevent unauthorized access, ensuring only authorized individuals can access it.
Usability	The system should be user-friendly, requiring minimal effort to learn and operate, with a clear and intuitive user interface and informative error messages.
Reliability	The system should ensure that a failure in any individual component does not lead to a complete system outage, sustaining performance under stress and recovering from errors.

4.3.3.1 Justification choice of security algorithm

- Password Hashing

The choice of the bcrypt algorithm as the password hashing technique is based on multiple factors. Firstly, this algorithm is widely utilised and renowned. Empirical evidence has demonstrated its robustness and resilience against various cryptographic assaults. Unlike other algorithms, bcrypt was not specifically developed to encrypt vast volumes of data, making it more ideal for password hashing. Bcrypt is intentionally designed to have a long and computationally demanding process, which enhances its resistance against brute-force assaults. These attacks involve attempting to guess passwords by hashing several possible combinations. In addition, bcrypt automatically generates and adds a salt to the password, which increases the difficulty for attackers attempting to decrypt passwords using rainbow tables or dictionary assaults.

- Password Encryption

The AES-128 encryption algorithm was chosen for the selected site password encryption for multiple reasons. AES is the most secure encryption algorithm currently available, surpassing previous encryption algorithms. It is widely utilised in government and military applications, as well as by enterprises in heavily regulated sectors. The encryption keys are produced using the PBKDF2 technique. PBKDF2 was specifically designed to derive cryptographic keys from passphrases or passwords. This process employs both a salt and a sequence of hashes to transform a password into a cryptographic key. By employing this method, it ensures that the resulting key possesses a high level of strength and is impervious to brute-force attacks. The AES algorithm selected a key size of 128 bits instead of 192 or 256 bits due to the widespread use and excellent level of security provided by AES-128, even at its lowest level. No serious vulnerabilities have been found after doing thorough research and testing. Thus, AES-128 is considered to offer a sufficiently robust level of security for the majority of applications, including password management systems. AES-128 is computationally less intensive than AES-256. The increased key length of AES-256 necessitates greater computational resources for encryption and decryption processes, potentially affecting performance.

4.3.4 Other Requirement

Table 4.4 Software Requirement Table

Software
Windows 10
Visual Studio Code
XAMPP
Diagram.io

Table 4.5 Hardware Requirement Table

Hardware
Personal Computer

4.4 High-Level Design

The main objective of the system is to generate random passwords based on user-specified conditions while prioritizing robust password manager security. This will be accomplished through the incorporation of two-factor authentication and password hashing techniques. The password manager feature will allow users to securely store passwords for different websites, encrypting them for enhanced protection. Additionally, users will have the ability to edit their account data and access the password data table, enabling them to view, edit, and delete stored passwords.

4.4.1 System Architecture

The system architecture is established by identifying the necessary software components, providing developers with a conceptual understanding of the proposed system domain. In this architecture, data is stored in the MySQL database, which is made accessible through the XAMPP Web Server. Microsoft Visual Code is utilized for the design and coding phases of the system development. The system architecture serves as the most abstract representation of the system, outlining its structure and components.

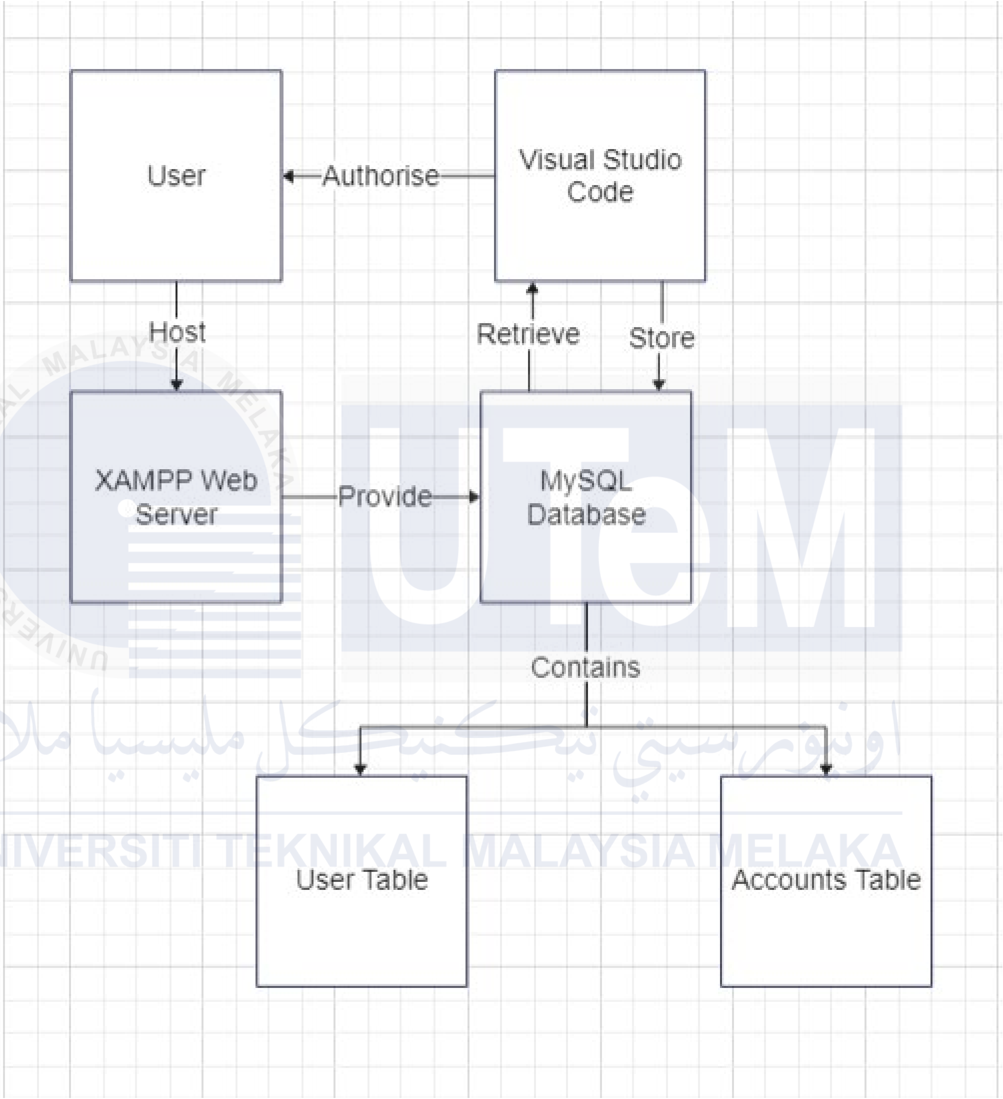


Figure 4.3 System Architecture of the proposed system

4.4.2 User Interface Design

User interface design is the process of designing the graphical user interface for a software application. The objective of this design process is to ensure that the user's interaction with the application is as straightforward and effective as can be, striving to optimize usability and overall user experience.

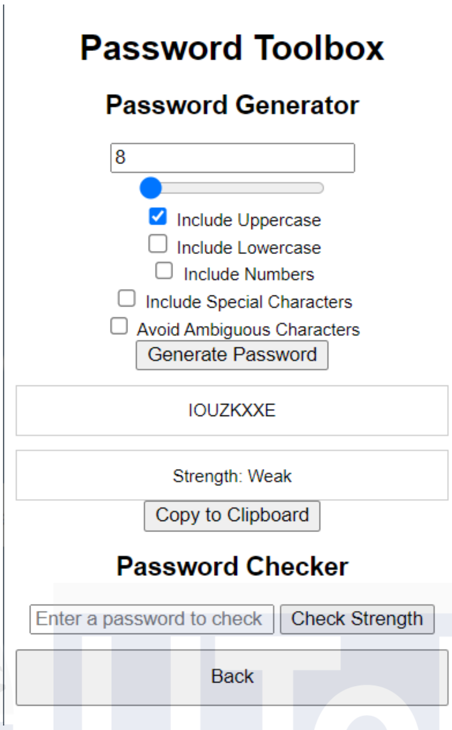
Secure Password Manager and Password Tools Prototype Version

a) Character combination choices

Table 4.6 Password Combination Table Examples

Password combination	Example Password generated
• Include Uppercase	IOUZKXXE
• Include Lowercase	hzzkwvyw
• Include Numbers	05649699
• Include Special Characters	|:+`{>).
• Include Uppercase • Include Lowercase	tTxgFomG
• Include Uppercase • Include Numbers	GE16IEDF
• Include Uppercase • Include Special Characters	Y]/T;D*D
• Include Uppercase • Avoid Ambiguous Characters	IPSZFSWV

After passwords are generated, user can copy to clipboard to fill in the password manager.



Password Toolbox

Password Generator

8

☑ Include Uppercase
□ Include Lowercase
□ Include Numbers
□ Include Special Characters
□ Avoid Ambiguous Characters

Generate Password

IOUZKXXE

Strength: Weak

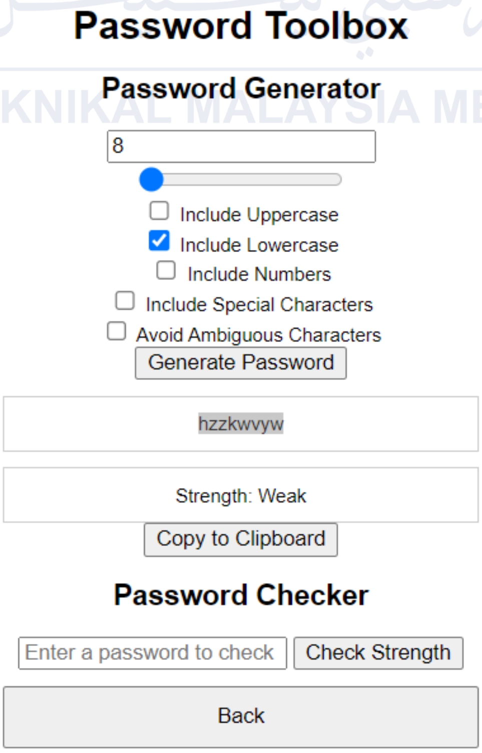
Copy to Clipboard

Password Checker

Enter a password to check | Check Strength

Back

Figure 4.4 Include Uppercase



Password Toolbox

Password Generator

8

□ Include Uppercase
☑ Include Lowercase
□ Include Numbers
□ Include Special Characters
□ Avoid Ambiguous Characters

Generate Password

hzzkwvyw

Strength: Weak

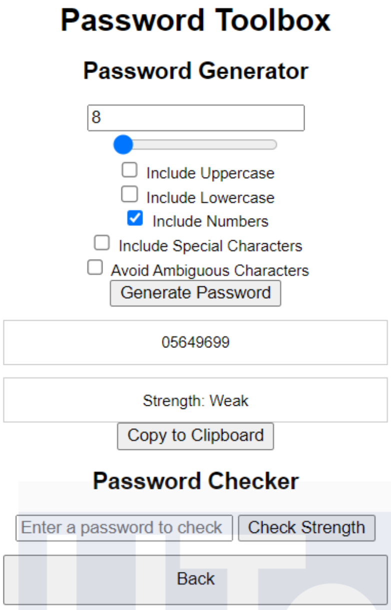
Copy to Clipboard

Password Checker

Enter a password to check | Check Strength

Back

Figure 4.5 Include Lowercase



Password Toolbox

Password Generator

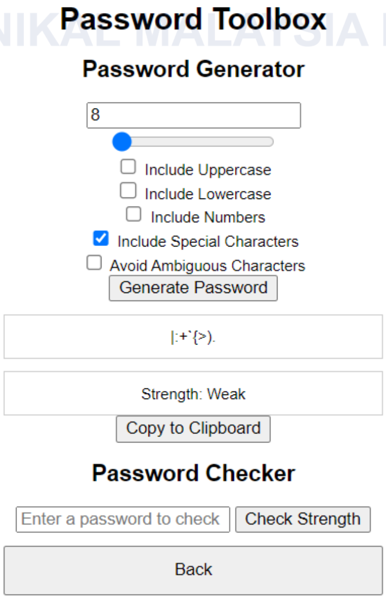
8

□ Include Uppercase
□ Include Lowercase
☑ Include Numbers
□ Include Special Characters
□ Avoid Ambiguous Characters

05649699

Strength: Weak

Figure 4.6 Include Numbers



Password Toolbox

Password Generator

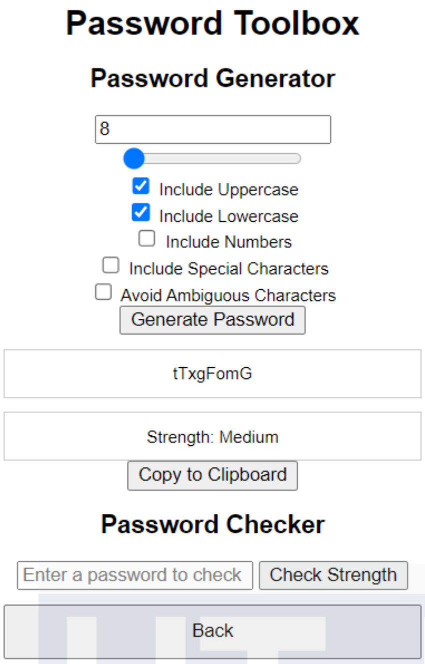
8

□ Include Uppercase
□ Include Lowercase
□ Include Numbers
☑ Include Special Characters
□ Avoid Ambiguous Characters

|:+`{>).

Strength: Weak

Figure 4.7 Include Special Characters



Password Toolbox

Password Generator

8

☑ Include Uppercase
☑ Include Lowercase
□ Include Numbers
□ Include Special Characters
□ Avoid Ambiguous Characters

Generate Password

tTxgFomG

Strength: Medium

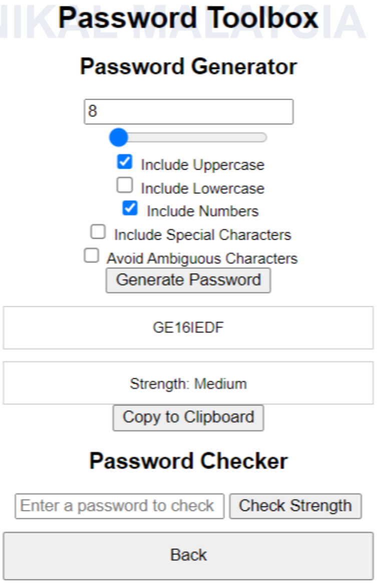
Copy to Clipboard

Password Checker

Enter a password to check | Check Strength

Back

Figure 4.8 Include Uppercase and Lowercase



Password Toolbox

Password Generator

8

☑ Include Uppercase
□ Include Lowercase
☑ Include Numbers
□ Include Special Characters
□ Avoid Ambiguous Characters

Generate Password

GE16IEDF

Strength: Medium

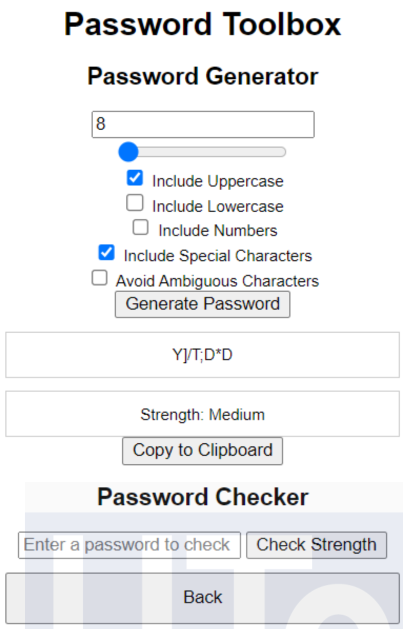
Copy to Clipboard

Password Checker

Enter a password to check | Check Strength

Back

Figure 4.9 Include Uppercase and Numbers



Password Toolbox

Password Generator

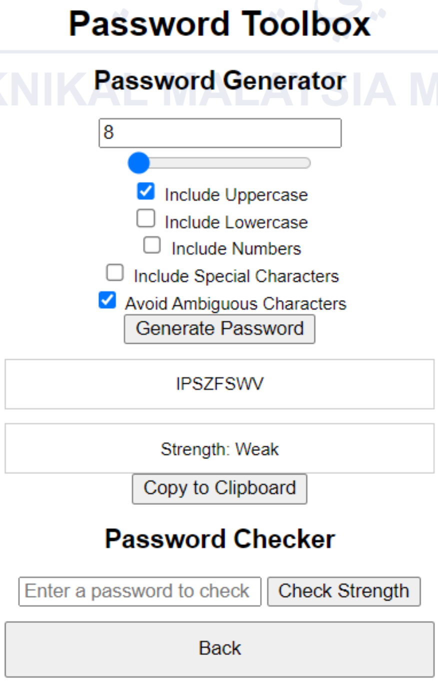
8

☑ Include Uppercase
□ Include Lowercase
□ Include Numbers
☑ Include Special Characters
□ Avoid Ambiguous Characters

Y]/T;D*D

Strength: Medium

Figure 4.10 Include Uppercase and Special Characters



Password Toolbox

Password Generator

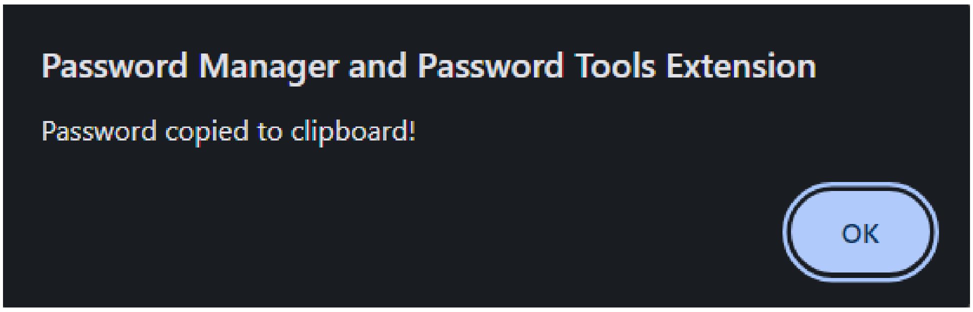
8

☑ Include Uppercase
□ Include Lowercase
□ Include Numbers
□ Include Special Characters
☑ Avoid Ambiguous Characters

IPSZFSWV

Strength: Weak

Figure 4.11 Include Uppercase and Avoid Ambiguous Characters



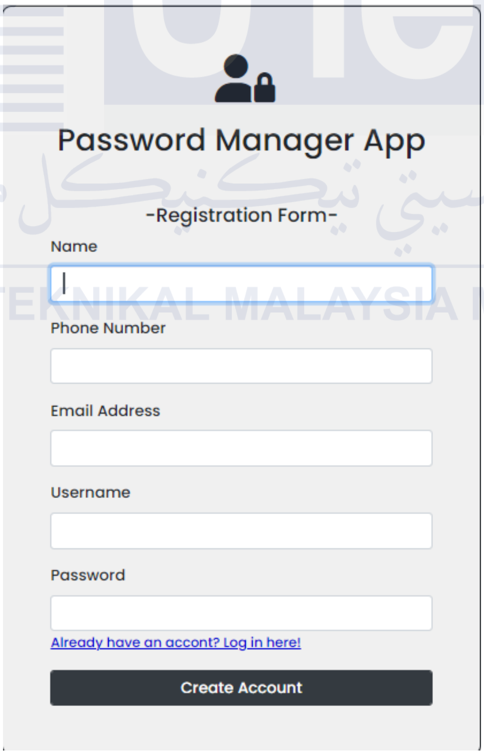
Password Manager and Password Tools Extension

Password copied to clipboard!

Figure 4.12 Notification when copy to clipboard

b) Storing generated password

The password generated that are copied to clipboard can be pasted on the password manager form that receive input email, password, username and the url or app name used. The figure below shows the prototype of the password manager.



Password Manager App

-Registration Form-

Name

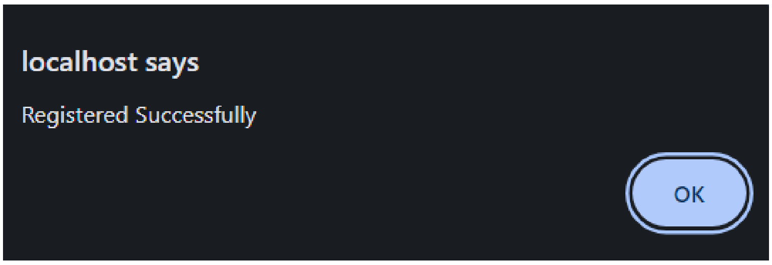
Phone Number

Email Address

Username

Password

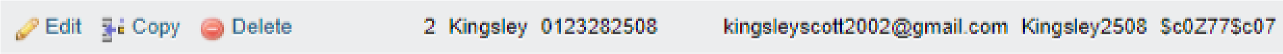
Figure 4.13 Password manager form



localhost says

Registered Successfully

Figure 4.14 Notification when form are saved



```
2  Kingsley  0123282508        kingsleyscott2002@gmail.com  Kingsley2508  $c0Z77$c07
```

Figure 4.14 Data saved in database

<u>Secure Password Manager and Password Tools Final Version</u>

4.4.2.1 Navigation Design

- Password Toolbox

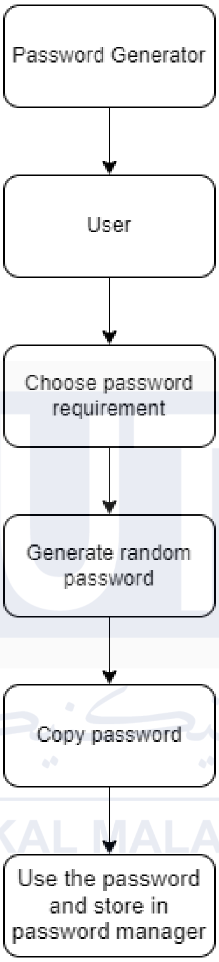

Password Generator
User
Choose password requirement
Generate random password
Copy password
Use the password and store in password manager

Figure 4.15 Password Toolbox Flow

The figure shows the flow of using the password generator. Starting from the user will open the password manager at in chrome extension, and then they will choose the password requirement and satisfied the minimum requirement from the website they are registering for. After that, they will click generate random password and user can copy the password to paste into the site’s form and stored in password manager.

-
- Password Manager

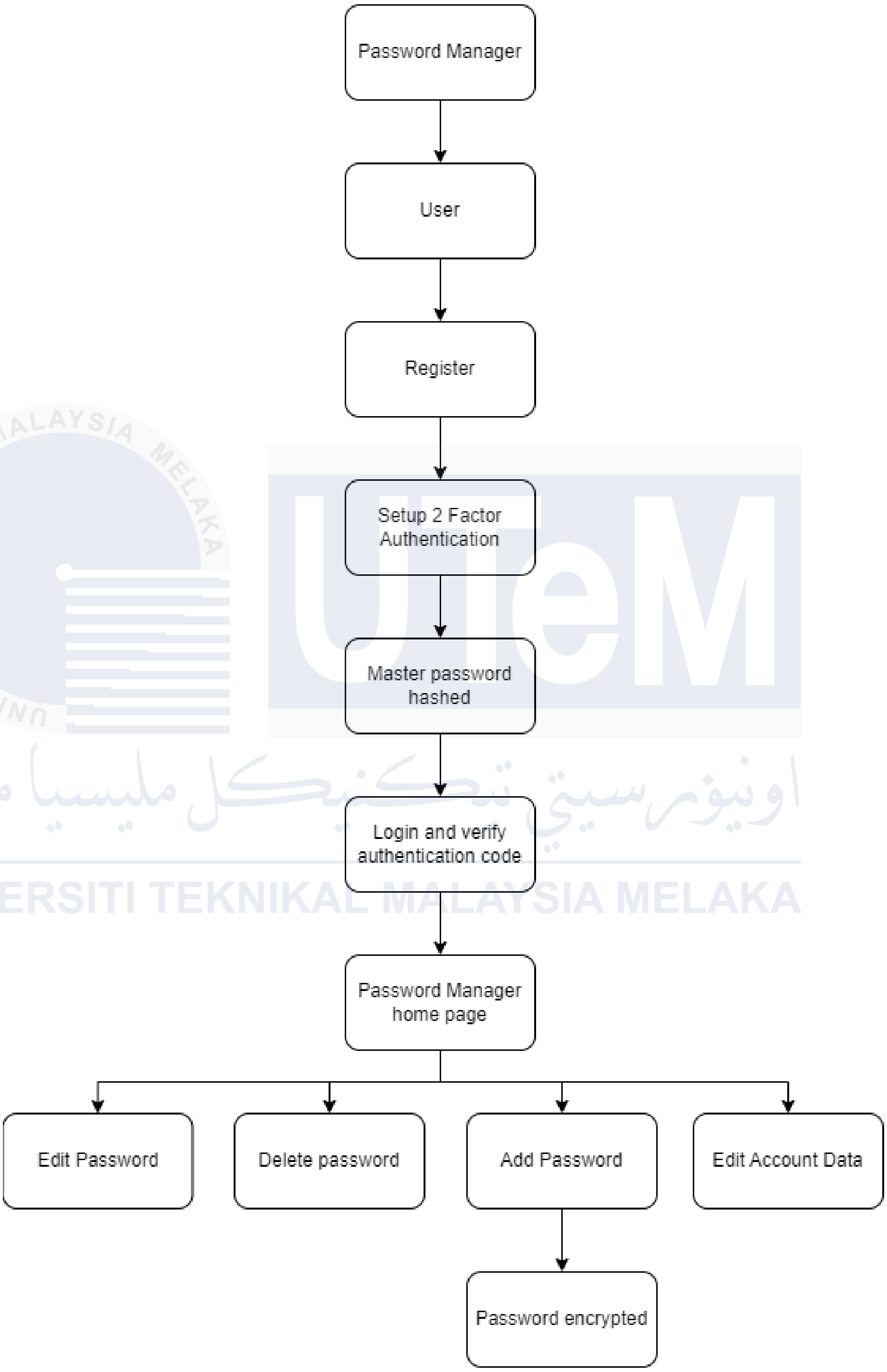


Figure 4.16 Password Manager Flow

The figure shows the flow of using the password manager. Starting from the user will need to register their password manager account. After the form is filled, the user will need to setup 2 factor authentication with Google Authenticator application. User can proceed the setup or skip it.

-

The master password will be hashed when stored in the database. When the user need to login, they need to verify the authentication code if they use the 2-factor authentication. In the password manager, user can add the site's password, edit and delete it. When the site password is stored in the database, it will be encrypted. User also can edit their account data if needed.

4.4.3 Database Design

Database design is a plan for how a database will be structured. It includes information on what data will be stored in the database, how it will be organized, and how the data will be accessed. A good database design can make a big difference in the performance and usability of a database.

4.5 Detailed Design

The Secure Password Manager and Tools application incorporates three security systems. The first is two-factor authentication, which utilizes the Google Authenticator application. The second is the hashing of the master password using the bcrypt algorithm. Lastly, the site passwords stored in the password manager are encrypted using the AES algorithm. To generate an encryption key, it employs the PBKDF2 algorithm, enhancing the strength of the key derivation process.

4.5.1 Software Design

Table 4.7 Function table of the registration form

Function	Details
Name input textbox	User enter their full name for registration
Email input textbox	User enter their email for registration
Master password input password	User enter their master password for registration. Master password will be
	hashed with bcrypt algorithm before stored in the database.
Repeat password input password	User enter repeat password to make sure password entered correctly
Password must have:	Password requirement for password manager
Register button	Register user and open two-factor authentication setup modal
Expected outcome: When the form are filled, and user clicked the Register button, it will open a modal to setup two-factor authentication.	

Table 4.8 Function table of the login form

Function	Details
Email input textbox	User enter their email for login
Master password input password	User enter their master password for login
Login button	Authenticate user and open 2FA verification
Register here anchor	Go to register page
Expected outcome: After the user clicks the login button, the credentials provided will be checked in the database. If authorized, the user will be presented with a 2FA verification modal.	

Table 4.9 The function table of the 2FA setup

Function	Details
Google authenticator key code	Secret key generated used for Google Authenticator
Scan QR Code	QR code generated used for Google Authenticator
Verification Code	Verification code obtained from Google Authenticator
Skip button	User can skip the 2FA setup to not use it
Verify button	Verify the verification code after scan the QR code or entered the secret key in Google Authenticator.
Expected outcome: After user scan the QR code or entered the secret key in Google Authenticator, there will be verification code can be obtained, user enter the verification code in the 2FA setup and click verify to complete the register process. User can also choose to skip the 2FA setup to not use it.	

Table 4.10 Function table of the 2FA verification

Function	Details
Verification code input textbox	User enter the verification code obtained from Google Authenticator
Scan QR Code	QR code used for Google Authenticator

Verification Code	Verification code obtained from Google Authenticator
Skip button	User can skip the 2FA setup to not use it
Verify button	Verify the verification code after scan the QR code or entered the secret key in Google Authenticator.
Expected outcome: Verification code obtained from Google Authenticator entered in verification code input textbox and then user click the verify button to login into password manager.	

Table 4.11 Function table of the Password Generator

Function	Details
Password generated textbox	To display the generated password to the user
Copy button	Copy password generated to clipboard
Password length input	Set the length of the password
Lowercase checkbox	Select the lowercase checkbox for the password generation requirement.
Uppercase checkbox	Select the uppercase checkbox for the password generation requirement.
Numbers checkbox	Select the numbers checkbox for the password generation requirement.
Randomized symbol checkbox	Select the randomized symbol checkbox for the password generation requirement.
Generate password button	Generate random password after set the password requirement
Expected outcome: Verification code obtained from Google Authenticator entered in verification code input textbox and then user click the verify button to login into password manager.	

Table 4.12 The function table of the Password Datable

Function	Details
Add password button	Open add password modal

Logout button	Clear the login session and logout
Dropdown entries	Set how many row displayed in the password data table.
Search bar	User can enter URL, site name, username and site password in the password data table
Edit button in column Action	Open edit password modal
Delete button in column Action	Open delete confirmation password modal
Every column in data table	Used to sort in ascending or descending order, either alphabetically or numerically.
Page number	Choose which page in data table
Expected outcome: Site password data are shown in this data table, there is also logout button and add password modal button.	

Table 4.13 The function table of the add new password form

Function	Details
URL input textbox	User enter URL
Site name input textbox	User enter site name
Username or email input textbox	User enter username or email
Site password input password	User enter site password
Show password button	Change input type from password to text
Close modal button	Close add new password modal
Add password button	The input field data entered will be stored in database. Site password will be encrypted with AES algorithm in the database
Expected outcome: You can insert site password data here.	

Table 4.14 Function table of the edit password form

Function	Details
URL input textbox	User enter URL
Site name input textbox	User enter site name
Username or email input textbox	User enter username or email
Site password input password	User enter site password

Show password button	Change input type from password to text
Close modal button	Close edit password modal
Edit password button	The input field data entered will be updated in database. Site password will be encrypted with AES algorithm in the database
Expected outcome: You can edit site password data here.	

Table 4.15 The function table of the edit user account form

Function	Details
Full Name input textbox	User enter URL
Email input textbox	User enter site name
Old master password input password	User enter old master password to verify if the old password entered are correct
New master password input password	User enter new master password
Show password button	Change input type from password to text
Close modal button	Close edit password modal
Edit password button	The input field data entered will be updated in database. Master password will be hashed with bcrypt algorithm before stored in the database
Expected outcome: You can edit user account data here.	

4.5.2 Flowchart

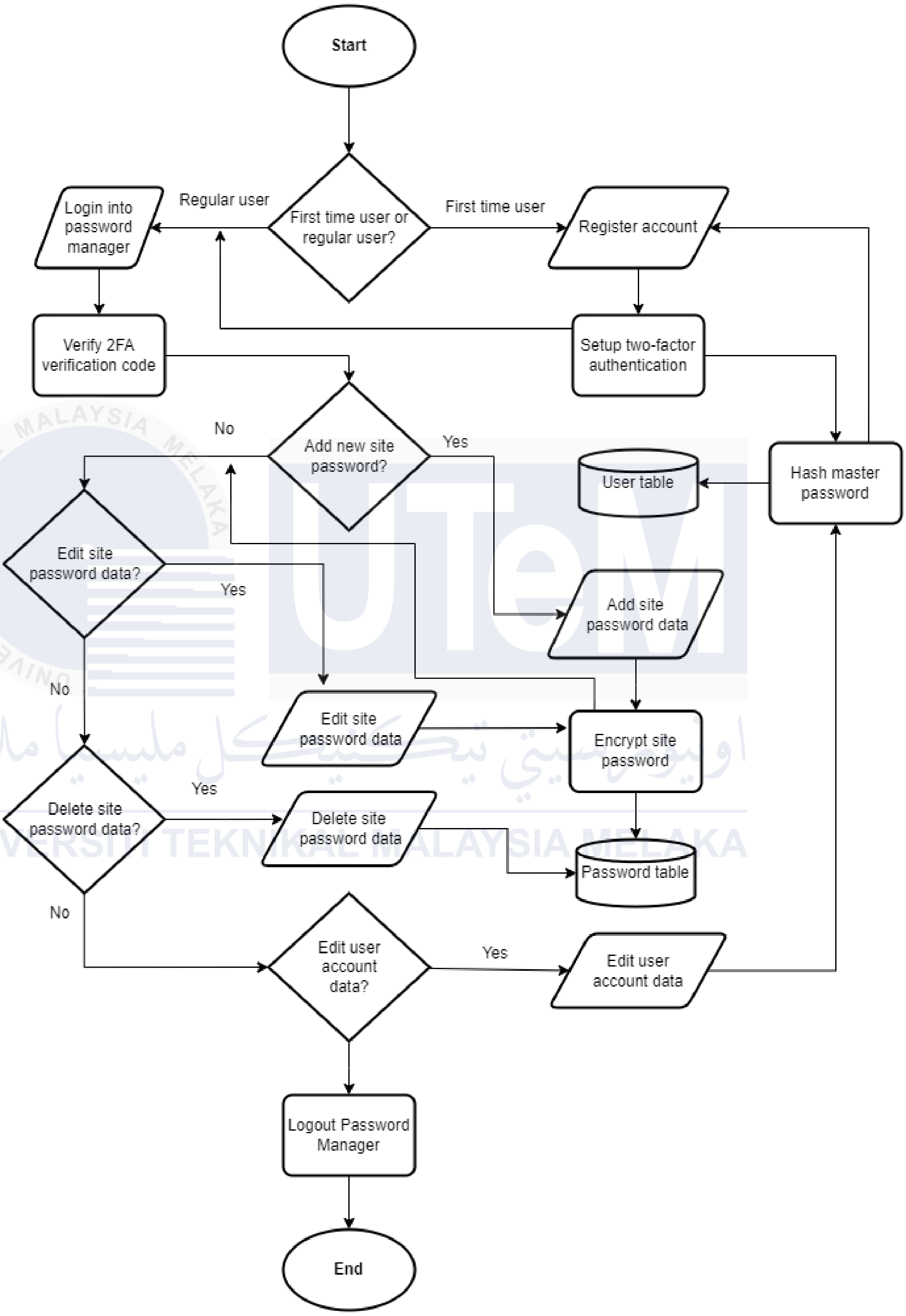
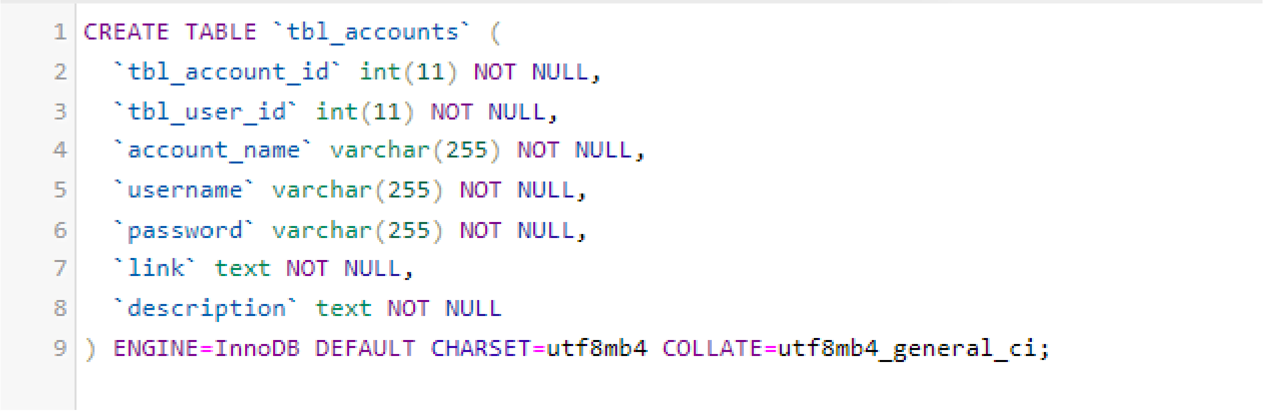


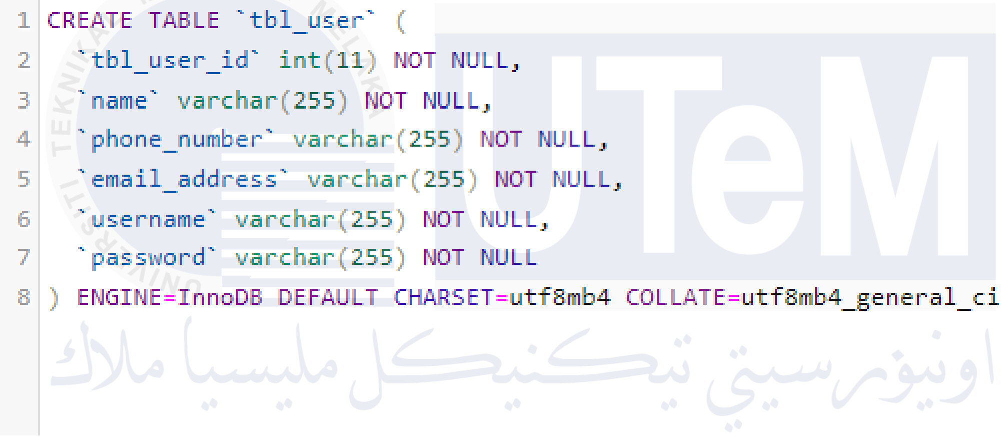
Figure 4.40 The flowchart of Password Manager

4.5.3 Physical Database Design



```
CREATE TABLE `tbl_accounts` (
  `tbl_account_id` int(11) NOT NULL,
  `tbl_user_id` int(11) NOT NULL,
  `account_name` varchar(255) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `link` text NOT NULL,
  `description` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 4.41 Create table accounts in password manager database



```
CREATE TABLE `tbl_user` (
  `tbl_user_id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `phone_number` varchar(255) NOT NULL,
  `email_address` varchar(255) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Figure 4.42 Create table user in password manager database

4.6 Conclusion

The analysis and design phase are critical to the success of the Secure Password Manager and Tools project. By thoroughly understanding the problems with existing password systems and defining clear requirements, we can develop a robust, user-friendly solution. The detailed design ensures that each component of the system is well-planned and aligned with the overall objectives, providing a solid foundation for the implementation phase.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter offers a thorough overview of the implementation phase of the project, where the system's functionalities are fully developed and detailed. It includes an in-depth explanation of the implementation status for each module, as well as discussions on software configuration management and the development environment setup. The main objective of this phase is to ensure the successful deployment and operation of the system.

5.2 Software Development Environment Setup

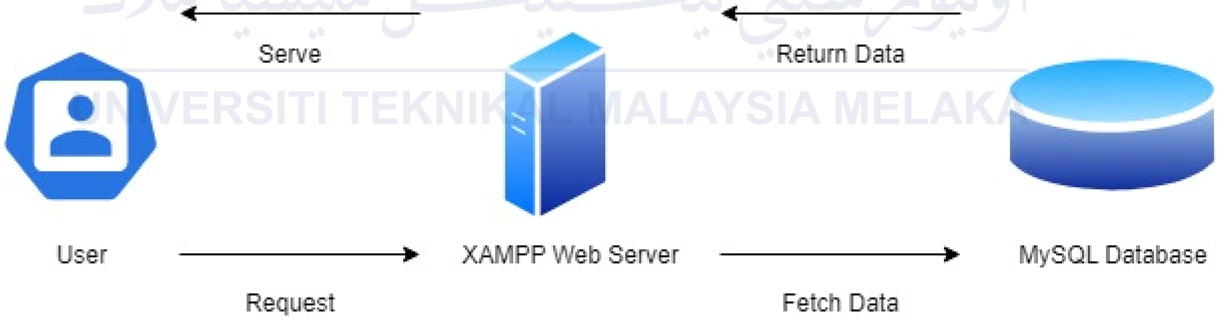


Figure 5.1 Deployment Diagram

Figure 5.1 shows the deployment diagram for the Secure Password Manager and Password Tools system, illustrating its structure and workflow. Users interact with the system via their personal devices, like a computer, by visiting the website through a specific URL. When a user makes a request, the server processes it and sends the appropriate response back to the user.

For local installation, we've used the Xampp web server, which provides the necessary infrastructure to host and run the website locally. This setup allows us to develop and test the system within our chosen development environment. We manage the system's MySQL databases using phpMyAdmin, a popular tool for this purpose. The Secure Password Manager and Password Tools system relies on MySQL, a widely used relational database management system, to store and manage data.

The deployment diagram gives a detailed overview of how the different components connect and highlights the underlying architecture that supports the system's operation. It serves as a foundation for the implementation phase, ensuring that the development environment is properly configured for the project's success.

Table 5.1 Software Development Environment Configuration

No	Items	Details
Xampp (Local Web Server)		
1	URL	
MySQL (Database)		
2	Database	password-manager
3	Username	Root
4	Password	
5	Port	3306

5.3 Software Configuration Management

This section provides an overview of the software configuration control implemented to support the development of this project.

5.3.1 Configuration Environment Setup

5.3.1.1 Xampp Configuration

To set up the configuration environment using Xampp, the following steps were taken:

Step 1: **Download and Install Xampp**: Obtain the latest version of Xampp from the official Apache Friends website and follow the installation instructions specific to your operating system.

Step 2: **Open Xampp Control Panel**: Once installed, access the Xampp Control Panel to manage its components.

Step 3: **Start Apache and MySQL**: In the Control Panel, click the "Start" button next to Apache and MySQL to initiate the Apache web server and MySQL database server.

Step 4: **Access phpMyAdmin**: Launch a web browser and type "localhost/phpmyadmin" in the address bar to open phpMyAdmin, a web-based tool for managing MySQL databases.

Step 5: **Create Database**: In phpMyAdmin, navigate to the "Databases" tab, enter a name for your project's database, and click "Create" to establish the database.

Step 6: **Edit Apache Configuration**: Locate the main Apache configuration file, "httpd.conf," found by default in the "xampp\apache\conf" directory. Open this file with a text editor.

Step 7: **Verify DocumentRoot**: In the "httpd.conf" file, locate the "DocumentRoot" directive and confirm it points to the directory where your project files are stored.

Step 8: **Check Directory Block**: Find the <Directory> block corresponding to your DocumentRoot in the "httpd.conf" file and ensure the directory path matches your project location.

Step 9: **Save and Restart**: Save your changes to the "httpd.conf" file and close it. Return to the Xampp Control Panel and click "Restart" next to Apache to apply the new configuration.

Access the web project:

To access your web project, open a web browser and type "localhost" in the address bar to load the default Xampp welcome page. To view your project, create a subdirectory under the DocumentRoot directory, place your project files in it, and access it by entering "localhost/subdirectory" in the browser's address bar.

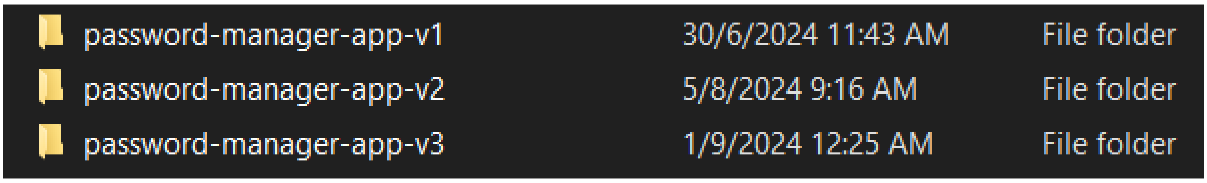
5.3.1.2 Visual Studio Code Configuration

Step 1: **Download and Install Visual Studio Code**: Visit the official Visual Studio Code website, download the installer for your operating system, and follow the installation steps to complete the process.

Step 2: **Open Visual Studio Code**: Once installed, launch Visual Studio Code.

Step 3: **Add Project Folder**: Click "Add Folder" and select the folder containing your project files to add them to the workspace.

5.3.2 Version Control Procedure



password-manager-app-v1	30/6/2024 11:43 AM	File folder
password-manager-app-v2	5/8/2024 9:16 AM	File folder
password-manager-app-v3	1/9/2024 12:25 AM	File folder

Figure 5.2 Version of Secure Password Manager and Password Tools

5.4 Implementation Status

This section gives detail about development status for each of the module in the system.

Table 5.2 Implementation Status Table

Module	Description	Duration to Complete
Password Generator Chrome extension	The Password Generator Chrome extension allows users to create secure passwords directly within their Chrome browser, providing convenience and enhancing security during the password creation process.	1 week
User Register and Login	The User Register and Login module enables users to create accounts and securely authenticate within the system. This module facilitates user registration by collecting essential information such as full name, email, and master password, ensuring a smooth and secure onboarding experience.	5 days
User 2 Factor Authentication	The User Two-Factor Authentication module enhances the security of user accounts by implementing dual verification through the Google Authenticator app. This added layer of security ensures that only authorized users can access their accounts.	1 week
Master password hashing	The Master Password Hashing module uses bcrypt to securely hash the master password. By employing bcrypt, the system ensures that master	4 days

	passwords are stored securely, protecting them from potential breaches.	
Account Site Passwords	The Password Manager Encryption module utilizes the Advanced Encryption Standard (AES) to secure and encrypt the stored passwords in the password manager database.	2 weeks
User Edit Account Data	The User Edit Account Data module allows users to modify and update their account information within the system. This feature provides flexibility for users to keep their account details current and accurate.	4 days
Password Data Table	The Password Data Table module offers a user-friendly interface for efficiently managing password records within the system. It includes features such as paging, sorting, and searching, allowing users to easily navigate and retrieve specific passwords based on criteria like account name or category. Additionally, users can edit existing passwords to make updates or delete passwords from the table, providing comprehensive password management capabilities.	2 weeks

5.5 Conclusion

In conclusion, this chapter outlined the setup of the software development environment, the management of software configuration, and the current implementation status of the project's application. The steps taken to configure the development environment were detailed, and the importance of software configuration management in facilitating the development process was emphasized. The progress of each module was summarized, highlighting the advancements made in the project. The next chapter will focus on the testing phase, discussing the testing strategy and the activities involved.

CHAPTER 6: TESTING

6.1 Introduction

This chapter provides a comprehensive overview of the test plan, detailing key aspects such as the test organization, test environment, and test schedule. It also outlines the testing methodology, focusing on different test classes. Following this introduction, the chapter delves into the test design, which includes a description of the tests and their expected outcomes. The chapter concludes with the presentation and analysis of the test findings, summarizing the results of the testing activities.

6.2 Test Plan

6.2.1 Test Organization

This system is specifically designed for personal computers using the Chrome browser. The password generator is implemented as a Chrome extension, offering users convenient access at any time, while the password manager is accessible via a web-based interface, allowing users to manage their passwords online. To account for potential mistakes and human errors, it is essential to involve a group of targeted users in the testing process. This ensures that all functionalities operate as expected. The table below outlines the plan for organizing and conducting tests on the system

.

Table 6.1 Test organization of the Secure Password Manager and Tools

Tester	Task	Description
Developer	• Programmer • Test analyst	Conduct thorough testing of the application's functionality in line with the provided test plan. This involves designing and executing tests to verify performance and ensure the application complies with requirements. The goal is to confirm the application functions as intended and meets expected standards.

User	• Test the functionality and non-functionality of the application	Carry out tests using the provided test cases to evaluate the system's functionality and non-functionality. Review the system's performance against each test case and provide feedback on the expected outcomes, ensuring the system meets the required specifications.

6.2.2 Test Environment

The testing phase is carried out according to the test schedule, and the hardware and software requirements for testing are shown in Table 24.

Table 6.2 Software requirements

Software	Description
Windows operating system	An operating system developed by Microsoft for use on personal computers.
Microsoft Visual Code	code editor used by developers to write, edit, and debug code.
MySQL Server	A platform used for storing and managing structured data within a database.
XAMPP	A hosting server that provides a local development environment for web applications.
Chrome Browser	A web browser developed by Google. The password tools is installed as an extension in this browser.

Table 6.3 Hardware requirements

Personal Computer
Processor: Intel(R) Core(TM) i5-10500H CPU @ 2.50GHz
Memory: 8.0GB

Operating system: Windows 10 Home

6.2.3 Test Schedule

Table 6.4 Test schedule

Activities and Event Entries			
No.	**Test Module**	**Test Cycle**	**Duration (days)**
1	Registration	6	3
2	Login	6	3
3	Two Factor Authentication	6	3
4	Edit user account data	6	3
5	Add site password	6	3
6	Edit site password	6	3
7	Delete site password	6	3
8	Site password encryption	6	3
9	Master password hashing	6	3
10	Generate random password	6	3
11	Pasword Checking	6	3

6.3 Test Strategy

The test strategy is essential for evaluating how well the project meets its objectives. It includes two main types of testing: black-box and white-box. Black-box testing focuses on identifying behavior-related bugs without needing to understand the internal code. This high-level approach targets the system's functionality and covers System Testing and User Acceptance Testing (UAT). It's done from the perspective of the end-user, so testers don't need to know how the system is built.

On the other hand, white-box testing is carried out by testers or developers who are familiar with the code and system structure. It's designed to find logic-related bugs by analyzing the code itself. White-box testing is considered lower-level and involves Unit Testing and Integration Testing.

6.3.1 Classes of tests

There are two key types of tests: functional and non-functional. Functional testing checks that the software meets its intended functional requirements, while non-functional testing looks at performance and other factors that aren’t directly related to functionality.

Table 6.5 Functional testing of the Secure Password Manager and Tools

No.	Test module	Description
1	Registration	• Test that that users can successfully create new accounts by providing the required information and that the registration process functions correctly. • Verify the connection with the MySQL server by ensuring the data can be stored in database.
2	Login	• Test that the users can log in using their registered credentials and gain access to their accounts securely. • Verify the connection with the MySQL server by ensuring the data can be retrieved from the database.
3	Two Factor Authentication	• Test that when the users are registering, a two-factor authentication setup will be displayed. • Test that users are prompted to provide a second form of authentication, such as a verification code from an authenticator app, to enhance the security of their accounts.

4	Update user account data	• Test that users can modify their account information, such as changing their email address or updating their profile details. • Verify the connection with the MySQL server by ensuring the data can be retrieved, updated and stored in the database.
5	Add site password	• Test that users can successfully add new site passwords to the password manager, ensuring accurate and secure storage of the password information. • Verify the connection with the MySQL server by ensuring the data can be stored in the database.
6	Edit site password	• Test the functionality to edit existing site passwords in the password manager, ensuring that changes are applied correctly and securely. • Verify the connection with the MySQL server by ensuring the data can be retrieved, updated and stored in the database.
7	Delete site password	• Test that the users can delete site passwords in the password manager, ensuring the removal of the selected password entry while maintaining data integrity. • Verify the connection with the MySQL server by ensuring the data can be deleted in the database.
8	Site password encryption	• Test the encryption process applied to site passwords within the Password Manager and Tools system, ensuring that the passwords are securely stored and protected from unauthorized access.

		• Verify the connection with the MySQL server by ensuring the data can be retrieved from the database.
9	Master password hashing	• Test that the master passwords are properly hashed and stored securely to prevent unauthorized access.
10	Generate Random Password	• Test the functionality of generating random passwords to ensure the passwords are produced according to the defined criteria (e.g., length, complexity). Verify that the generated passwords are sufficiently random and meet the security standards set by the password manager.
11	Password Checking	• Test the password checking feature to ensure it accurately evaluates the strength and validity of user passwords. Verify that the system provides appropriate feedback or suggestions based on the strength of the password and adheres to the defined password policy.
12	Display site password Data Table	• Test the functionality to display the site password data table, ensuring that all relevant information is accurately presented to the user in a clear and organized manner.
		• Verify the connection with the MySQL server by ensuring the data can be retrieved from the database.

Table 6.6 Non functional testing of the Secure Password Manager and Tools

Non-functional testing	Description
Compatibility	Ensuring that the Password Generator and Manager system is compatible with different platforms, browsers, and operating systems. It ensures the system works well and behaves consistently in different environments, providing a smooth user experience.
Performance	Performance testing measures how well the Password Generator and Manager system responds, scales, and operates efficiently with different workloads. It evaluates speed, resource utilization, and stability to ensure it can handle expected user loads and perform well during peak usage. This testing helps identify performance issues and allows for optimizations to improve overall system performance.

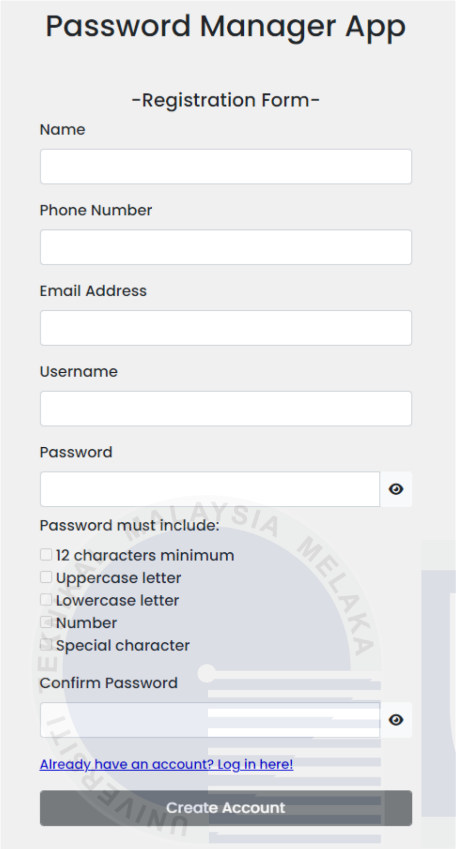
6.4 Test Design

This section explains how the testing process was conducted and the specific test cases that were used. It also includes the data that was utilized for running the tests. By outlining the different stages of the provided test criteria, it provides insights into the methods used for selecting the test cases.

6.4.1 Unit testing

During unit testing, each testing component is examined individually and independently to ensure its proper functioning.

- **Registration**



Password Manager App

-Registration Form-

Name

Phone Number

Email Address

Username

Password

Password must include:

□ 12 characters minimum
□ Uppercase letter
□ Lowercase letter
□ Number
□ Special character

Confirm Password

Already have an account? Log in here!

Table 6.7 Registration module

Test ID	Description	Test Data	Expected Output
PGM001	Fill all the input form, fulfill the password requirement, and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: StrongPass@2024 Confirm Password: StrongPass@2024	The system validates successful registration and displays a modal for setting up two-factor authentication.

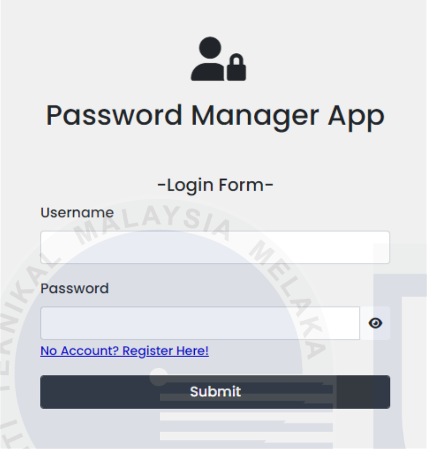
PGM002	Fill some of the input form and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: StrongPass@2024 Confirm Password: StrongPass@2024	The system displays an error regarding the empty input text field.
PGM003	Fill some of the input form and click Register.	Name: Kingsley Smith Phone Number: Email Address: kingsley.smith123@example.com Username: kingsley123 Password: StrongPass@2024 Confirm Password: StrongPass@2024	The system displays an error regarding the empty input text field.
PGM004	Fill some of the input form and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: Username: kingsley123 Password: StrongPass@2024 Confirm Password: StrongPass@2024	The system displays an error regarding the empty input text field.
PGM005	Fill some of the input form and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: StrongPass@2024 Confirm Password:	The system displays an error regarding the empty input text field.

PGM006	Leave all the input form empty and click Register.	Name: Phone Number: Email Address: Username: Password: Confirm Password:	The system displays an error regarding the empty input text field.
PGM007	Fill all the input form but different repeat password and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: StrongPass@2024 Confirm Password: DifferentPass@2024	The system displays an error that the repeat password did not match.
PGM008	Fill all the input form but incorrect email format and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsleysmith123example.com Username: kingsley123 Password: StrongPass@2024 Confirm Password: StrongPass@2024	The system displays an error: invalid email format.
PGM009	Fill all the input form but not fulfill the “At least 12 characters” password requirement and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: Short1! Confirm Password: Short1!	The system displays an error: password does not meet the requirements.

PGM0010	Fill all the input form but not fulfill the “At least 1 lowercase letter” password requirement and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: ALLUPPERCASE1! Confirm Password: ALLUPPERCASE1!	The system displays an error: password does not meet the requirements.
PGM011	Fill all the input form but not fulfill the “At least 1 uppercase letter” password requirement and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: lowercaseonly1! Confirm Password: lowercaseonly1!	The system displays an error: password does not meet the requirements.
PGM012	Fill all the input form but not fulfill the “At least 1 numerical number” password requirement and click Register.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: NoNumber! Confirm Password: NoNumber!	The system displays an error: password does not meet the requirements.
PGM013	Fill all the input form but not fulfill the “At least 1 special	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123	The system displays an error: password does not meet the requirements.

	character" password requirement and click Register.	Password: NoSpecial1 Confirm Password: NoSpecial1	

- Login



Password Manager App

-Login Form-

Username

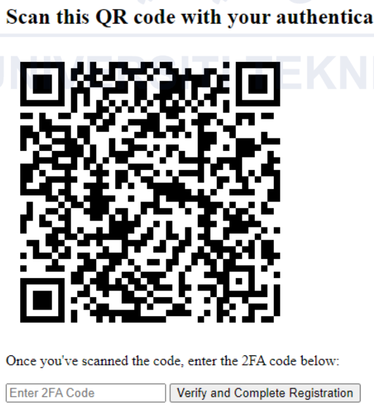
Password

Table 6.8 Login module

Test ID	Description	Test Data	Expected Output
PGM014	Fill all the input form and click Login.	Username: kingsley123 Password: StrongPass@2024 Two-factor authentication: Enabled	The system validates successful login and displays a modal for user to verify the two-factor authentication code.
PGM015		Username: kingsley123 Password: StrongPass@2024 Two-factor authentication: Disabled	The system validates successful login and redirects to the Password Manager homepage.

PGM016	Fill some of the input form and click Login.	Username: Password: StrongPass@2024 Two-factor authentication: Enabled	The system displays an error regarding the empty input text field.
PGM017	Fill some of the input form and click Login.	Username: kingsley123 Password: Two-factor authentication: Enabled	The system displays an error regarding the empty input text field.
PGM018	Leave all the input form empty and click Login.	Username: Password: Two-factor authentication: Enabled	The system displays an error regarding the empty input text field.

- Two Factor Authentication


Scan this QR code with your authenticator app (Google Authenticator, Authy, etc.):

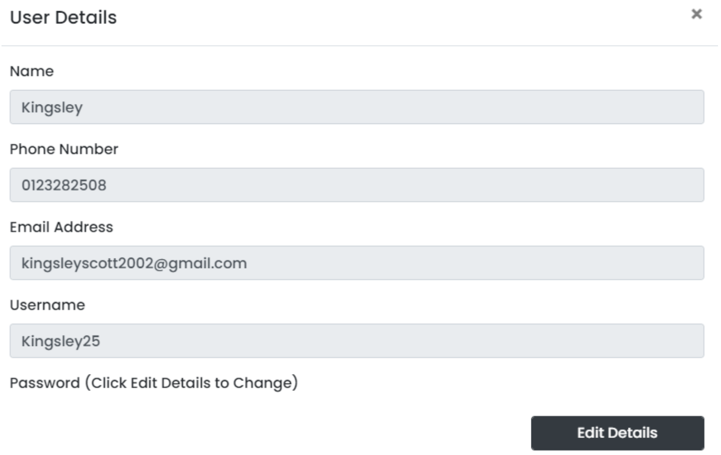
Once you've scanned the code, enter the 2FA code below:

Table 6.9 Two Factor Authentication module

Test ID	Description	Test Data	Expected Output

PGM019	Enter Google Authenticator key or scan QR code and enter the correct code in Verification Code input.	Google Authenticator key: U2FKWYTS4TXDKJT4 QR Code: Verification Code: 906488	The system validates successful two-factor authentication setup and displays that registration is successful.
PGM020	Enter Google Authenticator key or scan QR code and enter an incorrect code in Verification Code input.	Google Authenticator key: U2FKWYTS4TXDKJT4 QR Code: Verification Code: 906400	The system displays an error: invalid verification code.
PGM021	User decides to skip two-factor authentication and clicks the skip button.	Skip button pressed	User immediately registered without two-factor authentication enabled.

- Edit User Account Data



User Details

Name

Kingsley

Phone Number

0123282508

Email Address

kingsleyscott2002@gmail.com

Username

Kingsley25

Password (Click Edit Details to Change)

Table 6.10 Edit User Account module

Test ID	Description	Test Data	Expected Output
PGM022	Fill all the input form, fulfill the password requirement and click Edit account.	Account Name: Instagram Username: kingsley123 Old Password: StrongPass@2024 New Password: NewStrongPass@2024 Confirm New Password: NewStrongPass@2024	The system validates successful edit of user account data.
PGM023	Fill some of the input form and click Edit account.	Account Name: Instagram Username: kingsley123 Old Password: StrongPass@2024 New Password: NewStrongPass@2024 Confirm New Password: NewStrongPass@2024	The system displays an error regarding the empty input text field.
PGM024	Fill some of the input form and click Edit account.	Account Name: Instagram Username: Old Password: StrongPass@2024 New Password: NewStrongPass@2024 Confirm New Password: NewStrongPass@2024	The system displays an error regarding the empty input text field.

PGM025	Fill some of the input form and click Edit account.	Account Name: Instagram Username: kingsley123 Old Password: New Password: NewStrongPass@2024 Confirm New Password: NewStrongPass@2024	The system displays an error regarding the empty input text field.
PGM026	Fill all the input form with mismatched new password and confirmation password, then click Edit account.	Account Name: Instagram Username: kingsley123 Old Password: StrongPass@2024 New Password: NewStrongPass@2024 Confirm New Password: DifferentPass@2024	The system displays an error that the new password and confirmation password do not match.
PGM027	Fill all the input form but old password is incorrect, then click Edit account.	Account Name: Instagram Username: kingsley123 Old Password: WrongPass@2024 New Password: NewStrongPass@2024 Confirm New Password: NewStrongPass@2024	The system displays an error indicating the old password is incorrect.
PGM028	Fill all the input form but new password does not meet requirements, then click Edit account.	Account Name: Instagram Username: kingsley123 Old Password: StrongPass@2024 New Password: short Confirm New Password: short	The system displays an error that the new password does not meet the requirements.

- Add Site Password

Table 6.12 Add Site Password module

Test ID	Description	Test Data	Expected Output

PGM029	Fill all the input form and click Add Site Password.	Account Name: LinkedIn Username: kingsley.linkedIn Password: LinkedInPass@2024 Link: https://www.linkedin.com/ Description: Professional Networking Account	The system validates successful addition of the site password.
PGM030	Fill some of the input form and click Add Site Password.	Account Name: LinkedIn Username: kingsley.linkedIn Password: LinkedInPass@2024 Link: https://www.linkedin.com/ Description: Professional Networking Account	The system displays an error regarding the empty input text field.
PGM031	Fill all the input form but the link is incorrect, then click Add Site Password.	Account Name: LinkedIn Username: kingsley.linkedIn Password: LinkedInPass@2024 Link: www.linkedin.com Description: Professional Networking Account	The system displays an error: invalid link format.
PGM032	Fill all the input form but the description is missing, then click Add Site Password.	Account Name: LinkedIn Username: kingsley.linkedIn Password: LinkedInPass@2024 Link: https://www.linkedin.com/ Description:	The system validates successful addition of the site password with a default description.
PGM033	Fill all the input form but the password does not meet requirements, then click Add Site Password.	Account Name: LinkedIn Username: kingsley.linkedIn Password: short Link: https://www.linkedin.com/ Description: Professional Networking Account	The system displays an error: password does not meet the requirements.

- Edit Site Password

Table 6.13 Edit Site password module

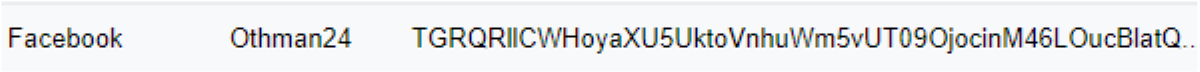
Test ID	Description	Test Data	Expected Output
PGM034	Fill all the input form and click Edit Site Password.	Account Name: LinkedIn Username: kingsley.linkedIn Password: NewLinkedInPass@2024 Link: https://www.linkedin.com/ Description: Updated Professional Networking Account	The system validates successful edit of the site password.
PGM035	Fill some of the input form and click Edit Site Password.	Account Name: LinkedIn Username: kingsley.linkedIn Password: NewLinkedInPass@2024 Link: https://www.linkedin.com/ Description: Updated Professional Networking Account	The system displays an error regarding the empty input text field.
PGM036	Fill all the input form but the password does not meet requirements, then click Edit Site Password.	Account Name: LinkedIn Username: kingsley.linkedIn Password: short Link: https://www.linkedin.com/ Description: Updated Professional Networking Account	The system displays an error: password does not meet the requirements.

- Delete Site Password

Table 6.10 Delete Site Password module

Test ID	Description	Test Data	Expected Output
PGM037	Select site password entry and click Delete.	Account Name: LinkedIn Username: kingsley.linkedIn	The system validates successful deletion of the site password.
PGM038	Select site password entry but click Cancel instead of Delete.	Account Name: LinkedIn Username: kingsley.linkedIn	The system confirms no changes were made and remains on the site password list.

- Site Password Encryption



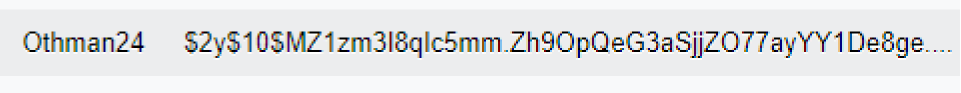
Facebook Othman24 TGRQRIlCWHoyaXU5UktoVnhuWm5vUT09OjocinM46LOucBlatQ..

Table 6.14 Site Password Encryption module

Test ID	Description	Test Data	Expected Output
PGM039	Add a site password and verify that it is encrypted in the database.	Account Name: LinkedIn Username: kingsley.linkedIn Password: LinkedInPass@2024 Link: https://www.linkedin.com/	The system verifies that the password is encrypted in the database.

		Description: Professional Networking Account	
PGM040	Edit a site password and verify that the new password is encrypted in the database.	Account Name: LinkedIn Username: kingsley.linkedIn Password: NewLinkedInPass@2024 Link: https://www.linkedin.com/ Description: Updated Professional Networking Account	The system verifies that the new password is encrypted in the database.

- Master Password Hashing



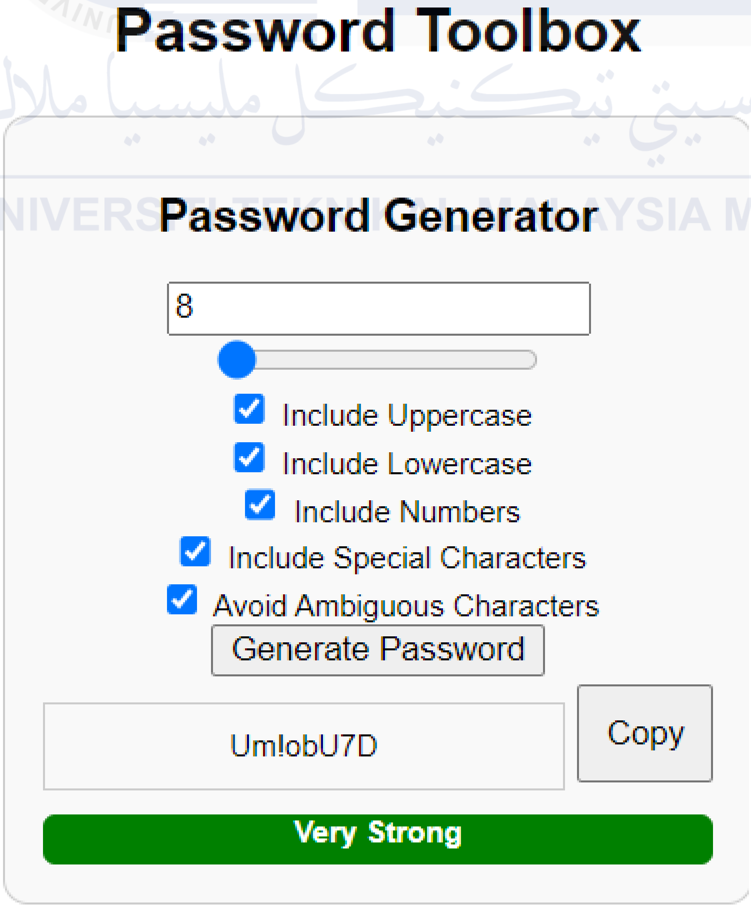
Othman24 $2y$10$MZ1zm3l8qlc5mm.Zh9OpQeG3aSjjZO77ayYY1De8ge....

Table 6.17 Master Password Hashing module

Test ID	Description	Test Data	Expected Output
PGM041	Register a new user and verify that the master password is hashed in the database.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: StrongPass@2024 Confirm Password: StrongPass@2024	The system verifies that the password is encrypted in the database.

PGM042	Edit an existing user's master password and verify that the new password is hashed in the database.	Name: Kingsley Smith Phone Number: 123-456-7890 Email Address: kingsley.smith123@example.com Username: kingsley123 Password: NewStrongPass@2024 Confirm Password: NewStrongPass@2024	The system verifies that the new password is encrypted in the database.

- Generate Random Password



Password Toolbox

Password Generator

8

☑ Include Uppercase
☑ Include Lowercase
☑ Include Numbers
☑ Include Special Characters
☑ Avoid Ambiguous Characters

Um!obU7D

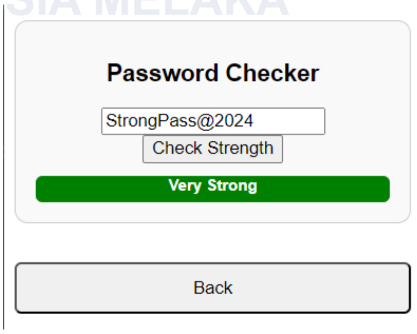
Very Strong

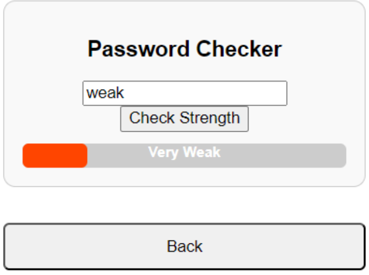
Table 6.18 Generate Random Password module

Test ID	Description	Test Data	Expected Output
PGM043	Click the Generate Random Password button and verify that a new random password is generated and meets the criteria.	Button Clicked	The system generates a password that meets the requirements (at least 12 characters, including uppercase, lowercase, numbers, and special characters).

- Password Checking

Table 6.19 Password Checking module

Test ID	Description	Test Data	Expected Output
PGM044	Enter a password into the Password Strength Checker and verify that the strength is correctly assessed.	Password: StrongPass@2024	The system displays the password as "Very Strong" with appropriate feedback.  Password Checker StrongPass@2024 Check Strength Very Strong Back
PGM045	Enter a weak password into the Password Strength Checker and verify that the strength is	Password: weak Password: 1234 Password: password	The system displays the password as "Very Weak" with appropriate feedback.

	correctly assessed.		 Password Checker weak Check Strength Very Weak Back

6.5 Test Results and Analysis

This chapter looks at how effectively users are kept informed about the current system and explores the project's area of focus and development methods. It also covers the project's domain and construction techniques. The next chapter will dive deeper into the requirement analysis, detailing the project's needs, as well as the software and hardware requirements.

6.5.1 Unit Testing Result

Table 6.20 Unit Testing Result Table

Test Case ID	Results (Pass/Fail)	Remarks if fail
PGM001	Pass	-
PGM002	Pass	-
PGM003	Pass	-
PGM004	Pass	-
PGM005	Pass	-
PGM006	Pass	-
PGM007	Pass	-
PGM008	Pass	-
PGM009	Pass	-
PGM010	Pass	-
PGM011	Pass	-
PGM012	Pass	-
PGM013	Pass	-

PGM014	Pass	-
PGM015	Pass	-
PGM016	Pass	-
PGM017	Pass	-
PGM018	Pass	-
PGM019	Pass	-
PGM020	Pass	-
PGM021	Pass	-
PGM022	Pass	-
PGM023	Pass	-
PGM024	Pass	-
PGM025	Pass	-
PGM026	Pass	-
PGM027	Pass	-
PGM028	Pass	-
PGM029	Pass	-
PGM030	Pass	-
PGM031	Pass	-
PGM032	Pass	-
PGM033	Pass	-
PGM034	Pass	-
PGM035	Pass	-
PGM036	Pass	-
PGM037	Pass	-
PGM038	Pass	-
PGM039	Pass	-
PGM040	Pass	-
PGM041	Pass	-
PGM042	Pass	-
PGM043	Pass	-
PGM044	Pass	-

PGM045	Pass	-

The unit testing for the Secure Password Manager and Tools system was thoroughly conducted, and all test cases were successful, resulting in a perfect 100% success rate. This outcome underscores the effectiveness of our testing approach and confirms that the system meets the expected performance and functionality criteria.

6.6 Conclusion

System testing is a vital component of the Agile SDLC process. It ensures that any potential issues are identified and resolved before the system reaches the customer. This chapter highlighted the various testing methods used to verify and validate the Secure Password Manager and Tools system. By integrating testing throughout the development cycle, we align with Agile principles, allowing for continuous feedback and iterative improvements to enhance the system's quality and reliability.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

The conclusion of this project encapsulates the journey of developing the Secure Password Manager and Password Tools. This tool was conceived to address the growing need for secure password management and has undergone rigorous development to ensure it meets high standards of security and usability. The project involved the integration of multiple security features, such as two-factor authentication, bcrypt hashing, and AES encryption, along with the development of a user-friendly interface for managing passwords effectively. This chapter will present an overview of the strengths and weaknesses identified throughout the development process, propositions for improvement, and the overall contributions of the project.

7.2 Observation on weakness and strength

There will be strengths and weaknesses for every system that is developed, and the Secure Password Manager and Tools system is no exception.

7.2.1 Strengths

The Secure Password Manager and Password Tools project demonstrated several key strengths:

- **Robust Security Features**: The project incorporated strong security measures, including bcrypt hashing for the master password and AES encryption for site passwords. These methods ensure that sensitive data is stored securely and is resistant to common attack vectors such as brute force or dictionary attacks.

- **Comprehensive Password Tools**: The integration of password generation and strength-checking tools directly within the manager offers users the convenience of creating strong passwords without needing external resources. This not only improves user experience but also enhances overall security.

- **User Interface and Experience**: The user interface was designed to be intuitive and user-friendly, making it accessible to a broad audience. The inclusion of features such as password visibility toggles and live password validation enhances the usability of the application.

- **Extension Compatibility**: The development of a Google Chrome extension that interfaces with the password manager broadens the tool’s accessibility, allowing users to manage their passwords directly from their browser. This seamless integration improves the user experience by providing quick access to stored credentials.

.

7.2.2 Weaknesses

Despite its strengths, the project also presented some weaknesses:

- **Limited Encryption Scope**: While the project implemented AES encryption for site passwords, usernames, and email addresses remained unencrypted. This presents a potential vulnerability in case of a data breach, as these critical pieces of information would be exposed in plain text.

- **Single Method of Two-Factor Authentication**: The reliance solely on Google Authenticator for two-factor authentication (2FA) may limit the application's flexibility and accessibility. Users who prefer other methods, such as SMS-based 2FA or hardware tokens, might find the current setup restrictive.

- **Incomplete Account Recovery Features**: The project lacks a comprehensive account recovery mechanism, such as a "Forgot Password" feature. This could pose a significant usability challenge for users who lose access to their master password, potentially locking them out of their accounts without a recovery option.

- **Limited User Data Fields**: The existing database structure only accommodates usernames and passwords for each account. The absence of an email field could limit the system's flexibility, especially in scenarios where users prefer to log in with their email addresses instead of usernames.

7.3 Proposition for Improvement

Several areas for improvement have been identified to enhance the functionality and security of the Secure Password Manager and Password Tools:

- **Encrypt Usernames and Emails**: Encrypting usernames and email addresses would significantly enhance the security of stored data. In the event of a data breach, encrypted usernames and emails would be far less useful to attackers, thereby protecting user identities and reducing the risk of targeted phishing or other forms of attack.

- **Addition of an Email Field in tbl_accounts**: Introducing an email field in the tbl_accounts table would allow the application to support login via email, offering users more flexibility. This addition could also facilitate better account management and recovery options, as many users prefer email-based logins over traditional usernames.

- **Implementation of Autofill Feature**: Incorporating an autofill feature that automatically populates login forms with stored credentials would greatly enhance user convenience. This feature could be particularly useful for users who manage multiple accounts, reducing the need to manually copy and paste credentials each time they log in.

- **Development of a "Forgot Password" Feature**: Implementing a "Forgot Password" feature is crucial for improving user experience and account recovery. This feature could involve sending a password reset link to the user's registered email address, allowing them to securely create a new master password without losing access to their stored data.

- **Exploring Alternative 2FA Methods**: To accommodate a broader range of users, the application could support additional 2FA methods beyond Google Authenticator. Options such as SMS-based codes, hardware tokens, or biometric authentication would offer users more choices and enhance the security of the authentication process.

7.4 Project Contribution

The Secure Password Manager and Password Tools project has made several notable contributions:

- **Enhanced Password Security**: By providing a secure platform for storing and managing passwords, this project contributes to the broader goal of improving online security. The use of advanced encryption techniques and secure password hashing protects users from potential threats.

- **User Empowerment**: The inclusion of tools for generating and evaluating password strength empowers users to create more secure passwords. This not only protects their individual accounts but also contributes to overall cybersecurity awareness and practices.

- **Practical Application of Security Concepts**: The project serves as a practical demonstration of key cybersecurity concepts, such as encryption, hashing, and two-factor authentication. This not only benefits the users but also provides a valuable learning experience in the implementation of these technologies.

- **Flexible and Scalable Design**: The design of the password manager is both flexible and scalable, allowing for future enhancements and integrations. This ensures that the tool can adapt to evolving security requirements and user needs over time.

7.5 Conclusion

In conclusion, the Secure Password Manager and Password Tools project has successfully achieved the primary objectives set out at the beginning of its development. The system effectively addresses the critical need for secure password management by incorporating robust security measures such as strong password generation, bcrypt hashing, and AES encryption, along with the implementation of two-factor authentication. These features work together to offer a reliable solution for users who seek to manage their passwords securely.

However, like any system, there are areas where further refinement and improvement could enhance the overall functionality and user experience. The project's strengths lie in its security framework and the integration of essential tools directly into the password manager, but there are opportunities to enhance user engagement by improving the user interface design. Making the interface more visually appealing and intuitive could significantly enhance user satisfaction and encourage broader adoption of the system.

In addition to the user interface, future work could focus on optimizing the codebase to address potential scalability issues, ensuring that the system performs efficiently even under high user loads. Incorporating additional features, such as a comprehensive account recovery system, alternative methods for two-factor authentication, and encryption of all user-related data, would further solidify the system's robustness and versatility.

Despite the areas identified for enhancement, the Secure Password Manager and Password Tools project represents a significant accomplishment in the realm of cybersecurity. It successfully meets the requirements for a Bachelor of Computer Science (Computer Security) and demonstrates the development of a sophisticated password management solution that prioritizes security and usability.

Looking ahead, ongoing efforts to refine the system's interface, scalability, and overall performance will contribute to an even more effective user experience. This project not only serves as a valuable tool for users but also represents a meaningful contribution to the field of computer security, laying the groundwork for continued advancements in password management practices.

REFERENCE

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for agile software development.

Bonneau, J., & Preibusch, S. (2010). The password thicket: Technical and market failures in human authentication on the web. In WEIS.

Carreira, C., Ferreira, J. F., & Mendes, A. (n.d.). Towards Improving the Usability of Password Managers. INESC-ID and IST, University of Lisbon, Portugal.

Golla, M., De Luca, A., & Dürmuth, M. (2019). The Wi-Fi reformation: A direct and practical attack against Wi-Fi passphrases using password managers. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS).

Gupta, M. R., Koli, Y. P., Patiyane, V. A., & Wagh, K. P. (2022). Password generator as Chrome extension. University of Mumbai, Bachelor of Engineering in Cyber Security, Semester III Mini Project.

Kelley, P. G., Komanduri, S., Mazurek, M. L., Shay, R., Vidas, T., Bauer, L., ... & Lopez, J. (2012). Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. Proceedings of the IEEE Symposium on Security and Privacy.

Li, Z., He, W., Akhawe, D., & Song, D. (2014). The Emperor’s New Password Manager: Security Analysis of Web-based Password Managers. In Proceedings of the 23rd USENIX Security Symposium

Lu, Z., & Mohamed, H. (2021). A complex encryption system design implemented by AES. Journal of Information Security, 12(2), 79-90.

NIST. (2001). Announcing the advanced encryption standard (AES). Federal Information Processing Standards Publication 197.

Oesch, S., & Ruoti, S. (2020). That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers. In Proceedings of the 29th USENIX Security Symposium. USENIX Association.

Papaspirou, V., Papathanasaki, M., Maglaras, L., Kantzavelou, I., Douligeris, C., Ferrag, M. A., & Janicke, H. (2023). A novel authentication method that combines honeytokens and Google Authenticator. Department of Informatics and Computer Engineering, University of West Attica.

Pearman, S., Zhang, S. A., Bauer, L., Christin, N., & Cranor, L. F. (2019). Why people (don’t) use password managers effectively. USENIX Symposium on Usable Privacy and Security (SOUPS).

Proctor, R. W., Lien, M. C., Vu, K. L., Schultz, E. E., & Salvendy, G. (2002). Improving computer security for authentication of users: Influence of proactive password restrictions. Computers & Security, 21(1), 16-26.

Provos, N., & Mazieres, D. (1999). A future-adaptable password scheme. In Proceedings of the USENIX Annual Technical Conference.

Shay, R., Komanduri, S., Durity, A. L., Huh, P. S., Mazurek, M. L., Segreti, S. M., Ur, B., Bauer, L., Christin, N., & Cranor, L. F. (2010). Encountering strong password policies: Information security perceptions and practices. In Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS).

Sriramya, P., & Karthika, R. A. (2015). Providing password security by salted password hashing using bcrypt algorithm. ARPN Journal of Engineering and Applied Sciences, 10(13), 5551-5555.

Taneski, V., Heričko, M., & Brumen, B. (2019). Systematic overview of password security problems. Faculty of Electrical Engineering and Computer Science, University of Maribor.

Yan, J. J., Blackwell, A. F., Anderson, R. J., & Grant, A. (2004). The memorability and security of passwords: Some empirical results. IEEE Security & Privacy, 2(5), 25-31.

Zibaei, S., Malapaya, D. R., Mercier, B., Salehi-Abari, A., & Thorpe, J. (2022). Do password managers nudge secure (random) passwords? USENIX Symposium on Usable Privacy and Security (SOUPS).