PASSWORD MANAGER

BY

COMPUTER ENGINEERING

COLLEGE OF ENGINEERING

BELLS UNIVERSITY OF TECHNOLOGY – NEW HORIZON

January 2025

ROBOTICS 1

(ICT 215)

SUBMITTED TO

AYUBA MUHAMMAD

DECLARATION

We hereby declare that this is our original work of the project design reflecting the knowledge acquired from research on the project about "**Password Manager"**. We therefore declare that the information in this report is original and has never been submitted to any other institution, university or college for any award other than Bells University of Technology, College of Engineering, and Department of Computer Engineering.

Name: ……

Signature: ……

Name: ……

Signature: ……

Name: ……

Signature: ……

Name: ……

Signature: ……

Name: ……

Signature: ……

ACKNOWLEDGEMENT

We would like to thank our lecturer, **Mr. Ayuba Muhammad** for his enormous guidance. Also, we would like to thank our friends and classmate of our college who were able to show us different perspectives which were eventually incorporated into the project.

TABLE OF CONTENT

METHODOLOGY

3.0 System Architecture

3.1 Tools and Libraries Used

3.2 Working of the system

3.3 GUI Design

3.4 Algorithm and Flowchart of Core Features

3.5 Database and File Structure

3.6 Development Workflow

CHAPTER FOUR

IMPLEMENTATION AND RESULTS

4.1 Screenshots of All Major Windows/Pages

4.2 Sample Input to Output Demonstration

4.3 Key Implementation Details and Code Explanation

4.4 Performance Evaluation and System Efficiency

4.5 User Testing and Feedback Analysis

CHAPTER FIVE

CONCLUSION

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

The incorporation of digital devices in daily activities has been at the forefront. People are heavily reliant on computers and the internet to perform activities such as online bank transactions, communication via social media, e-commerce, education, and work. The need to be heavily reliant on digital devices necessitates the creation of various accounts that are protected by unique passwords.

Passwords have been identified as the most common form of security for digital accounts based on the simplicity that comes with it. However, as the number of services that require an online presence continues to grow, many individuals have found it challenging to come up with a wide array of complex and varied passwords. The difficulty often forces individuals to compromise their security by sharing passwords, making the passwords guessable, or writing the passwords in a notebook.

Poor management and control of passwords have been cited as one of the main causes of breaches in cyber security. If an attacker succeeds in accessing a single weak and often shared password, he/she may have access to all the accounts associated with the user's identity. These risks and breaches include loss, identity theft, financial fraud, and breaches in privacy.

This has led to the creation of applications referred to as password manager applications, which have been developed to assist in overcoming this problem by providing users with an opportunity to store their login details securely. The applications utilize encryption techniques for storing passwords, and a master

password is required for gaining access, thus ensuring that unauthorized access to stored data does not compromise its security.

Python is an extremely popular programming language that is famous for its simplicity and flexibility. It can handle files, encryption, and create GUI applications. It is therefore often used for creating secure applications. It is therefore often used for creating applications in both academia and industry. The present project focuses on designing a Python-based Password Manager with Encryption. In this regard, it can be asserted that the proposed application shall be designed as a Desktop Application with a Graphical User Interface, which will assist users in managing their password details offline while ensuring data privacy.

1.2 Problem Statement

Users encounter difficulties with secure password management because cybersecurity problems have become more known. Users who must remember various passwords tend to develop security problems because they reuse passwords and create weak passwords. Users sometimes save their passwords in plain text files or notebooks or unsecured applications which creates a security risk because unauthorized users can access the stored information.

Most password manager applications require web access because they operate as online services and need continuous internet connectivity. The paid subscription model used by some services creates financial difficulties for users who need more affordable options. Users who need to store sensitive data face challenges because they must use third-party servers which create privacy and data control problems.

The existing problems demonstrate that users require a password management system which provides both security and usability through offline storage of their credentials which will remain encrypted. Users continue to face security threats

which security experts consider avoidable because the system lacks this essential feature.

The project aims to develop a secure desktop password manager application through Python which enables users to manage their passwords through an interface which includes encrypted local storage capabilities.

1.3 Aim and Objectives

The aim and objectives of this project define what the system is expected to achieve and guide the development process.

The main aim of this project is to **design and develop a secure Python-based desktop Password Manager application** that allows users to safely store, retrieve, and manage their passwords using encryption techniques within a graphical user interface. The system is intended to function as an offline application, giving users full control over their data and reducing exposure to external security threats.

To achieve this aim, the following objectives were set: to design an easy-to-use graphical interface for managing passwords; to implement encryption methods that protect stored credentials from unauthorized access; to ensure secure file handling and local data storage for password persistence; to enable basic password management functions such as adding, editing, deleting, viewing, and searching stored records; to include proper authentication and error-handling mechanisms to improve system reliability; and to document and deploy the completed application on GitHub with clear usage instructions.

1.4 Significance of the Project

The project holds important value because it contains multiple important reasons. Organizations can manage their password requirements through the security system which establishes a protected space for keeping their confidential

information. The system protects user data through password encryption and master password access control which prevents unauthorized access to the system and stops data breaches.

The project enables students to gain practical experience with programming and software development skills which they learned during their academic studies. The project requires users to apply Python programming skills together with encryption methods and graphical user interface development and file management skills. Users of the project will develop better problem-solving abilities through their experience with secure software development techniques which they will learn about.

The application provides value to users who need to manage their passwords because it offers offline password management capabilities which do not require cloud services. The project functions as a base for upcoming developments which will include features like cloud synchronization and advanced authentication and mobile application capabilities.

1.5 Scope and Limitations

The project requires developers to build a Python-based password manager application that functions as a separate desktop application. The system enables users to securely store their login information which includes website names and usernames and passwords. The system protects all sensitive information through encryption which secures it before permanent storage in local storage. The application provides users with a graphical user interface and a master password system for authentication purposes, and it delivers essential password management capabilities. The project includes testing work and documentation production and GitHub deployment activities.

The project has some restrictions which limit its functionality. Users who want to access their data from different devices through cloud synchronization cannot use the application. The project lacks advanced security features which include biometric authentication and automatic password strength assessment and breach detection because of time limitations and academic restrictions. The system operates with strong encryption technology, but its development occurred to serve educational and personal use instead of business deployment purposes.

1.6 Organization of the Study

The project report contains five separate chapters. Chapter One presents the project introduction together with the study background information which includes the problem statement and project objectives and significance and project boundaries and project limits. Chapter Two focuses on the review of related literature and existing password management systems. Chapter Three explains the methodology used in the design and development of the system. Chapter Four presents the implementation details and results obtained from the system. Chapter Five concludes the project and provides recommendations for future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Review of Existing Password Management Systems

Users utilize password management systems to store their login information in a secure and organized manner. People need to create multiple accounts because their online activities now require them to access various platforms. The need to remember numerous passwords creates difficulties because security experts recommend using strong and distinct passwords. Password managers exist in multiple ways to solve this security problem.

Web-based password manager systems serve as the most common type of password management software. LastPass, Dashlane, and Bitwarden serve as examples of these web-based password management systems. The systems maintain user passwords in online databases which users can reach from any device they own. Users receive assistance through features that let them create passwords and automatically complete online forms. Users can use web-based password managers to access their system features, but they need an active internet connection and functioning third-party services to operate. Users who have privacy concerns or need to protect their data from breaches will find this situation to be a major issue.

Users keep their password information on their personal computers through desktop password managers which function as another password management solution. Users can choose from different options through these systems which include KeePass and Password Safe. The systems function without requiring users to connect to the internet while they provide users with complete control over their personal information. Many users consider desktop password managers to be safer because the information is not stored online. The applications present challenges to

user comprehension because they create complex interfaces which are difficult for non-technical users to handle.

Mobile password management applications enable users to manage their passwords through their smartphone devices. The applications enable users to access their passwords whenever they need them. Users who do not implement suitable security methods face risks because their mobile devices can be easily lost or stolen, which exposes their stored passwords to potential threats.

2.2 Strengths and Weaknesses of Existing Tools

The design of existing password management tools enables them to deliver multiple advantages which their users can access. The system uses encryption to protect stored passwords which stands as its primary advantage. Users who obtain access to stored data cannot read passwords because the encryption system prevents password access through data verification. This feature stands as a critical element which enhances security protection for users. The system provides two main advantages which make it convenient to use. Password managers enable users to keep all their passwords in a single location which decreases their need to memorize multiple login credentials. The tools assist users in developing secure passwords which strengthens their overall security. The autofill function provides users with a speedier and simpler way to reach online sites. The tools come with several shortcomings that affect their performance. The security of cloud-based password managers becomes compromised because they depend on external servers which enable attackers to exploit their vulnerabilities during service disruption events. Some tools need users to pay subscription fees which become unaffordable for certain users. Desktop password managers protect user privacy better than other solutions. The advanced design of these systems creates difficulties for beginners who try to use them. The systems require users to possess

technical expertise while some systems function with outdated user interface designs. The system lacks proper user guidance along with essential error corrector systems which handle user mistakes.

2.3 Python Libraries Used in Similar Projects

Python functions as a programming language which people utilize for their educational studies and their real-world software development work across different fields of study. Developers choose Python as their preferred programming language because its code structure enables easy comprehension. Password manager projects typically use Python together with Cryptography and Fernet encryption libraries for security purposes. Libraries enable developers to protect sensitive information through encryption and decryption processes without their need to create complete security systems from the ground up.

Python offers multiple libraries for users who want to create graphical user interfaces between Tkinter PyQt and PySide. Students commonly select Tkinter for their projects because it provides basic functionality and comes pre-installed with Python. The software enables developers to create essential desktop application elements such as windows and buttons and input fields.

Python provides multiple options for users who want to store their data. JSON files serve as a storage solution for projects because they offer simple management capabilities that fit small applications. The system uses SQLite databases to create an organized method that enables users to store and access their data.

Most Python password manager projects focus on creating functional applications but they lack essential user experience features and documentation content. The project studies existing examples to develop better results through its focus on

clear presentation and user-friendly design together with organized content delivery.

2.4 Identified Research Gap

The existence of multiple password management systems creates security weaknesses which need to be fixed. Most password managers depend on cloud systems which fail to fulfill the needs of users who want to keep their data safe and maintain their privacy. Some tools become inaccessible to beginners because their complete features require users to make payment.

Most student password management projects at academic institutions only aim to achieve essential project requirements. The project fails to meet basic usability standards because it lacks both proper encryption methods and sufficient clear user documentation. The applications function properly but users face difficulties in learning how to operate them.

This project addresses these gaps by developing a simple, offline password manager using Python. The system delivers user-friendly functionality because it combines a secure encryption system with comprehensive user documentation. The project offers a practical solution to personal password management needs while it serves as an educational tool for software development students.

CHAPTER THREE

METHODOLOGY

3.1 System Architecture

The password manager system architecting process required developers to create three design elements which provide security to the application while delivering operational simplicity and maintenance efficiency. The system architecture required security measures to protect confidential data because password management needs to protect sensitive information through its entire development process.

The password manager system uses a modular system design which operates through multiple layers that handle distinct purposes for each component. The design system enables users to better understand system operations while system development work can proceed without interrupting current system functions.

The system architecting process establishes four primary layers which include the User Interface Layer, Application Logic Layer, Security and Encryption Layer, and Data Storage Layer. The User Interface Layer exists as the system's front end which enables users to access all visible system elements. The user interface includes the login page and dashboard together with the input forms and buttons and message prompts. Users can create new accounts through this layer which also allows them to log in using their master password and control their password records by adding new passwords and accessing existing passwords and modifying stored credentials and removing records. The system interface presents a straightforward design which enables users with basic technical skills to navigate the system without difficulties.

The Application Logic Layer exists as the middle component which links the user interface to the backend system. The application framework manages user activities through automatic process control which handles all execution steps.

3.2 Tools and Libraries Used

The password manager application was constructed using various software development tools and libraries. The project needed development tools which matched its requirements and matched the software development skills of the students. Python Programming Language serves as the main programming language because its simple syntax and widespread use enable developers to build secure software through its comprehensive library resources. Python enables software development at a faster rate than developers who use lower-level programming languages. The application used Tkinter to build its graphical user interface because Tkinter provides a Python library that comes included with Python and requires no extra setup. The application created its login page and dashboard and additional application windows using Tkinter widgets which include buttons and labels and text fields and message boxes. Cryptography and Hashing Libraries The project used hashlib and other encryption libraries to create password hashing and encryption systems. The master password is hashed so that it is never stored in plain text, while individual account passwords are encrypted before storage. Libraries provide tools which help maintain user data security by stopping unauthorized persons from accessing protected information. JSON File Handling The system used JSON as its primary data storage format. The system uses JSON to create a data storage format which maintains both structured data storage and easy human reading. The system used JSON files to organize stored usernames and encrypted passwords along with other related information.

Operating System (OS) Library, the OS library provided functions for handling files and managing directories and checking whether files exist.

3.3 GUI Design

The password manager interface was created to achieve three design goals which include providing users with a simple and clean interface that is easy to use. The interface was built to support users who will access the application multiple times because it needs to show them exact paths for discovering its hidden features. The application begins with a login and registration interface which allows users to create new accounts or access their existing accounts. The system provided users with clear labels and input fields which showed them the required information they needed to enter. The main dashboard serves as the central control panel which users' access after they complete their login process. The dashboard enables users to create new password records while they can also view their stored credentials and make changes to their existing records and remove unnecessary entries. The system created two distinct windows which users can use to both create new passwords and view saved passwords. The system divided tasks into separate windows which allowed users to concentrate on their current work. The system used clear button labels which showed users the available functions while the system displayed confirmation messages after every completed task. The system provides error messages and warning dialogs to notify users about invalid input which includes empty fields and incorrect login credentials. The graphical user interface design achieves an overall simple design which enables users to access all necessary system functions.

3.4 Algorithm and Flowchart of Core Features

The password manager uses structured algorithms to guarantee that its operations maintain both secure and accurate execution.

User Registration Process: The system requires users to provide a username together with a master password when they create an account. The system converts the master password into a hashed secure format through a secure hashing algorithm. The data file contains only the hashed version of the password. This method prevents the system from obtaining any knowledge about the actual password. The system creates a hash from the entered password during login which it uses to check against the stored hash. The user gains access to the dashboard when both hashes match. The system denies access to the user while showing an error message. The user provides website name together with username and password information when saving a password. The system encrypts the password before it gets stored in the database. The system maintains protection for sensitive information by storing it in an encrypted format. The user needs to authenticate himself before retrieving encrypted data which contains a saved password from storage. The user sees the password after the system decrypts it. The system protects its operations through these algorithms which maintain confidentiality together with data integrity and access control protection.

3.5 Database and File Structure

The project achieves lightweight deployment through its file-based storage system which replaces the need for complete database management systems. This approach is suitable for a desktop application developed at the student level. User data gets stored in JSON files which they organize into designated folders. Each user has associated encrypted data stored in a structured format. The system provides a straightforward method for programmers to handle data operations which include reading and updating and managing data.

The project file structure contains specific components which include

- A data folder for storing encrypted user credentials

• An assets folder for images and icons used in the interface

• A src folder for application logic and supporting modules

• A main.py file that serves as the entry point of the application.

The project contains an organized structure which helps evaluators to read and understand the project more easily.

3.6 Development Workflow

The development process of the password manager followed a systematic workflow which enabled effective project planning and implementation. Requirement analysis constituted the first stage because the investigators needed to investigate the security issues associated with password storage. System design work began when the team developed all system components which included Architectural design and graphical user interface design and encryption method selection. The developers used Python programming language to create individual modules which they assembled into a complete system during the implementation stage. The system underwent testing after implementation to find bugs and logic errors and usability problems. Testing required the application to run multiple times while checking different scenarios which included incorrect login attempts and empty inputs and file access errors. The team made all necessary changes based on test results. The project team used GitHub to manage their documentation and version control needs. The team fulfilled all documentation requirements which explained system operation in accordance with established procedures.

CHAPTER FOUR

IMPLEMENTATION AND RESULTS

4.1 Screenshots of All Major Windows/Pages

The chapter examines two distinct aspects of the password manager system which include both its implementation and performance evaluation after development and testing. The system was developed through Python programming language after the design phase which the previous chapter described. The complete desktop application enables users to store and handle their passwords in a secure manner which results from implementation. The application includes multiple windows which each have their own specific function. The system creates a user interface which combines multiple windows to provide both protection and operational efficiency to users. The application starts with its initial interface which consists of Login and Registration Window. The window enables new users to establish an account through the submission of their chosen username and master password. The existing users can log in to their accounts by entering their established login information. The interface was designed to be simple so that users can easily understand what is required of them. The system shows an error message when users enter incorrect login information while successful registration results in a confirmation message to the user. The Main Dashboard Window becomes accessible to users after they successfully log in. The window functions as the main application hub which users can use to navigate the software. The dashboard enables users to select from multiple functions which include creating a new password and viewing their stored passwords and modifying their existing records and removing passwords which they do not require anymore. The dashboard layout

is simple and well-arranged, making it easy for users to navigate the application without confusion.

The Add Password Window is used when the user wants to save new login details. Users need to fill three input fields which include the website or application name and their login credentials which consist of their username or email and password. The system saves the password after the user clicks the save button by encrypting the password first and then storing it in a secure location. The system presents a success message to the user which shows that their password has been successfully saved.

The View and Manage Passwords Window allow users to see all previously saved password entries. The system presents each saved record in an organized way which enables users to recognize them with ease. The window provides users with three options which include showing the decrypted password and updating existing stored data and removing the complete entry. The system exists to provide users with complete management functions for their saved authentication information.

4.2 Sample Input to Output Demonstration

The system functions correctly because multiple test cases are executed with various input conditions. The tests proved that the application executes its necessary functions with both correct results and secure performance. The user registration test involved a new user who created their account by selecting a username and entering their master password. The system processed the user input after the user clicked the register button by converting the master password into a hashed format which it then stored in the data file. The output confirmed that the account creation process had been completed successfully. The system stores a secure version of the password instead of keeping the actual password in its

database. The user successfully entered their correct username and password during the login test. The system created a password hash from the user input and compared it to its existing password hash. The user was permitted to enter the dashboard area because both values showed an identical match. The system blocked access when the user input an invalid password while showing an error notification. The authentication process operates correctly according to this information.

For the **password storage test**, the user entered details such as a website name, username, and password. After submission, the password was encrypted and stored in the data file. The output was a successful message indicating that the password had been saved. This shows that sensitive information is properly protected before storage.

In the **password retrieval test**, the user selected a saved entry from the list. The system decrypted the stored password and displayed it to the user. This test confirmed that encrypted data can be safely retrieved and used when necessary.

These input-to-output demonstrations show that the system behaves correctly under different scenarios.

4.3 Key Implementation Details and Code Explanation

The developers divided the application code into multiple sections to achieve better control over the code base and to enhance code comprehension. Each section handled a different system component which required separate processing. The authentication logic serves as a critical component which enables the implementation process to succeed. The code section handles all procedures which involve user registration and authentication processes. The system stores the master password after hashing it during the user registration process. The system

hashes the password which users enter during login to compare it against the stored password. The solution improves security through its design which stops attackers from accessing passwords that remain protected in encrypted storage. The implementation process includes both encryption and decryption methods as essential components. All passwords undergo encryption through an encryption algorithm before their storage begins. Users can access their password through a temporary decryption process which displays their encrypted data. The system protects stored information through an encryption method which makes it unreadable to anyone without proper access rights. The file handling logic manages all processes which involve storing and retrieving information from JSON files. The system provides functions that permit users to verify current data files while creating new ones and adding information to existing files. This method guarantees data retention while stopping data deletion from occurring. The development process benefited from modular coding because it created an easier implementation path which decreased development errors.

4.4 Performance Evaluation and System Efficiency

System performance assessment occurred after system implementation which tested actual user conditions. The application went through testing on a regular personal computer to create a realistic usage environment. The application startup time showed fast performance because the system operated with essential components only and did not require external libraries or online resources. The application starts almost at once after the user launches it. The system needs only small memory space for its operations according to its actual memory consumption. The system achieves low memory consumption because it keeps password data in file-based storage instead of implementing a full database system. The system provides fast response times to user actions which include logging in

and saving passwords and retrieving stored data. The system allows users to access its functions without any interruptions during their entire time using the system. The password manager application achieves desktop performance standards while it operates the system effectively.

4.5 User Testing and Feedback Analysis

The evaluation of the system needed testing through five separate users who provided user testing results. The users interacted with the application to complete various fundamental tasks of the system.

The users had to complete the following tasks:

• Create a new account

• Log in to the system

• Add at least two password entries

• View saved passwords

• Delete one password entry

Users provided their feedback about the process after they completed their assigned tasks. Most users found the system to be understandable and simple for them to operate. The users moved through the application interface without needing any additional help. Users trusted the system security because of its login authentication process and data encryption methods.

Some users suggested possible improvements which would add a search feature that would help users find saved passwords and display a password strength indicator. The development teams will use these suggestions as their guide through upcoming work which they will conduct.

User testing results demonstrate that the application operates properly while providing an easy-to-use experience that achieves all system goals.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Summary of Achievements

The project developed a password manager through its design process which created an encrypted system that functions with Python programming language. The project developed a desktop application which allows users to securely manage their passwords through its dedicated password storage and management features. The team successfully completed all project objectives which they established in Chapter One of the project documentation.

The team built a password management application which enables users to control their accounts through its registration and login features and password management system. The system enables users to handle their passwords through basic functions which include password creation and password retrieval and password record modification and password record deletion. The system protects all confidential data through encryption and hashing methods which prevent user information from being saved as unencrypted text.

The project established effective file management processes through its use of structured JSON files which store encrypted user data. The system permits users to access their saved passwords because the application keeps all data intact after it closes. The team documented all project activities through GitHub which they used to maintain project control according to established project guidelines.

The project achieved its main goals which resulted in a complete working software system.

5.2 Conclusion

The password manager project demonstrated that developers could create secure desktop applications through Python because users find their applications easy to operate. The system resolves poor password management issues through its simple method which allows users to safeguard multiple passwords in one secure storage area. The system obtained increased security through encryption and hashing methods because these techniques prevent unauthorized users from reaching protected information. The graphical user interface created an application which users could operate without needing technical knowledge. The system's modular design enabled people to comprehend and handle application maintenance tasks more effectively. The project showed how software design and security foundations combine with file management and user interface design to create functional systems in actual use. The system implementation achieved complete success because it met all project requirements while proving that the application performs as expected.

5.3 Challenges Faced and Lessons Learned

The project faced multiple difficulties during its development phase. The team faced encryption implementation difficulties as their main project obstacle. The team needed to perform extra research and testing because they required knowledge about encryption and hashing to create system functions which would not disrupt their system functionality. The team needed to develop methods which would control file storage operations while maintaining data integrity throughout their processes. The team needed to execute file operations with precision because they needed to protect data from loss during system updates. The team spent additional time resolving problems which involved the system accessing files and processing data in incorrect formats. The team faced difficulties when they

designed the user interface because they needed to organize widgets and create fluid transitions between different application screens. The team needed to make modifications after conducting tests and receiving user feedback. The project presented students with beneficial educational opportunities which they could use for their future work. The project developed knowledge about secure software development and problem-solving abilities while demonstrating the need for planning work before starting to execute it. The project demonstrated that software testing in conjunction with user feedback will result in higher software quality.

5.4 Recommendations and Future Improvements

The password manager application successfully achieves its current goals yet needs to enhance multiple areas for future development. The application requires a database system implementation which needs proper database system implementation through either SQLite or MySQL to replace its current file storage system. The system requires database implementation because it needs to handle multiple records through its operations. The system needs to implement a password strength checker which will assist users in creating stronger passwords for their accounts. The system needs to implement automatic password generation which will enable users to create secure passwords through this feature. The application needs to provide users with password access across multiple systems through cloud storage and device synchronization features which it must develop. The application needs to be transformed into a web-based and mobile application to improve its accessibility for users. The project will consider these improvements for its upcoming development phases.