Industrial Internship Report on

Password Manager

Prepared by

SHASHANK R R

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time. My project was Password Manager. This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface....3
2 Introduction....6
2.1 About UniConverge Technologies Pvt Ltd....6
2.2 About upskill Campus (USC)....10
2.3 The IoT Academy....12
2.4 Objectives of this Internship program....12
2.5 Reference....12
2.6 Glossary....13
3 Problem Statement....14
4 Existing and Proposed solution....15
4.1 Code submission (Github link)....15
4.2 Report submission (Github link)....15
5 Proposed Design/ Model....16
5.1 High Level Diagram (if applicable)....17
5.2 Interface / Snapshots (if applicable)....17
6 Performance Test....21
7 My learnings....23
8 Future work scope....25

1 Preface

Summary of the whole 6 weeks' work:

During the six-week internship, I had the opportunity to work on the Password Manager project. The project involved developing a secure and user-friendly application for managing passwords. I implemented key functionalities such as password storage, retrieval, and generation. The project utilized Python programming, cryptography libraries, and database management. I also incorporated a graphical user interface (GUI) using the Tkinter library to enhance the user experience. Throughout the internship, I faced various challenges and gained valuable skills in software development and cybersecurity.

About need of relevant Internship in career development:

This internship provided me with practical experience and the opportunity to apply my knowledge in a real-world scenario. It allowed me to enhance my technical skills in Python programming, cryptography, and GUI development. Additionally, I gained insights into software development methodologies and best practices. Working on the Password Manager project enabled me to deepen my understanding of data security and user-centered design principles. This internship experience has undoubtedly contributed to my career development by providing me with a strong foundation in software development and valuable industry exposure.

Brief about Your project/problem statement:

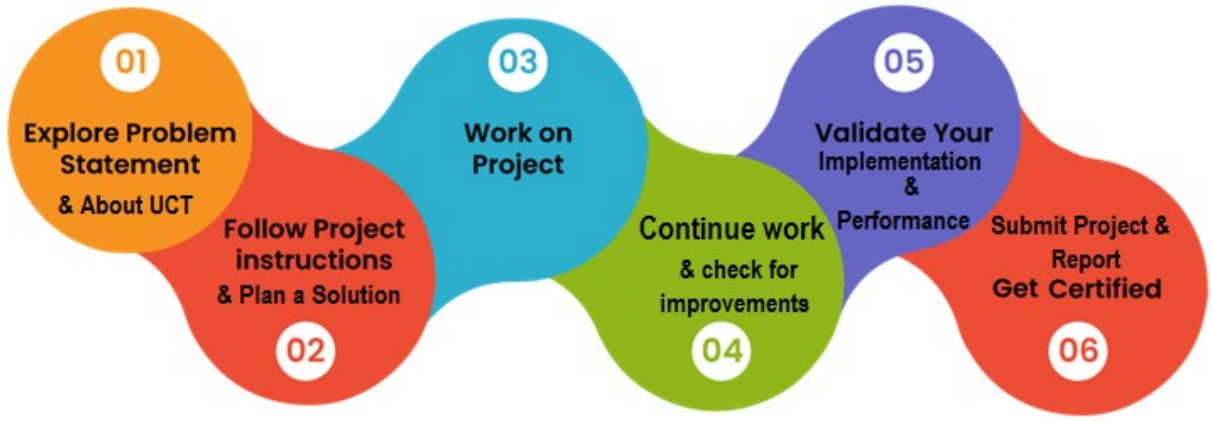
The problem statement for the Password Manager project was to develop a comprehensive and secure application for managing passwords. The project aimed to provide users with functionalities such as storing, retrieving, and generating passwords. Data integrity, encryption, and user authentication were key considerations to ensure the security of stored passwords. The transition from a command-line interface to a GUI was an important aspect of the project, making the application more accessible and visually appealing.

Opportunity given by USC/UCT:

I am grateful to USC/UCT for providing me with the opportunity to undertake this internship. The internship allowed me to gain hands-on experience and apply my knowledge to a practical project. It provided me with exposure to industry-standard

tools and practices, allowing me to understand the real-world challenges faced in software development. The opportunity to work on the Password Manager project specifically enabled me to explore the complexities of data security and develop skills in Python programming, cryptography, and GUI development. This opportunity has been instrumental in my career development and has significantly contributed to my growth as a software developer.

How Program was planned:



My Learnings and overall experience:

Throughout the internship, I learned valuable skills in Python programming, cryptography, and GUI development. I gained proficiency in implementing data encryption techniques to secure passwords. The transition to a GUI enhanced my understanding of user interface design and improved the overall user experience. I also learned the importance of thorough testing, documentation, and collaboration in software development. This internship provided me with a well-rounded learning experience, fueling my passion for software development and preparing me for future challenges in the field.

I extend my gratitude to all my mentors, Upskill Campus, USC/UCT, and all the faculty and staff who have supported me throughout this internship. Their guidance and support have been invaluable in my learning journey.

To my juniors and peers, I would like to share the following message:

Embrace every opportunity to learn and grow in your field of interest. Internships provide valuable hands-on experience and the chance to apply your knowledge in real-world scenarios. Approach every project with enthusiasm, curiosity, and a growth mindset. Be proactive in seeking guidance and collaborating with your teammates. Emphasize the importance of continuous learning, adaptability, and perseverance. Remember to document your progress and learnings, as they serve as a valuable resource for future reference. Lastly, enjoy the journey and make the most of every opportunity to enhance your skills and contribute to meaningful projects.

With dedication and determination, you can make a significant impact and set yourself on a path towards success. Best of luck in your endeavors!

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

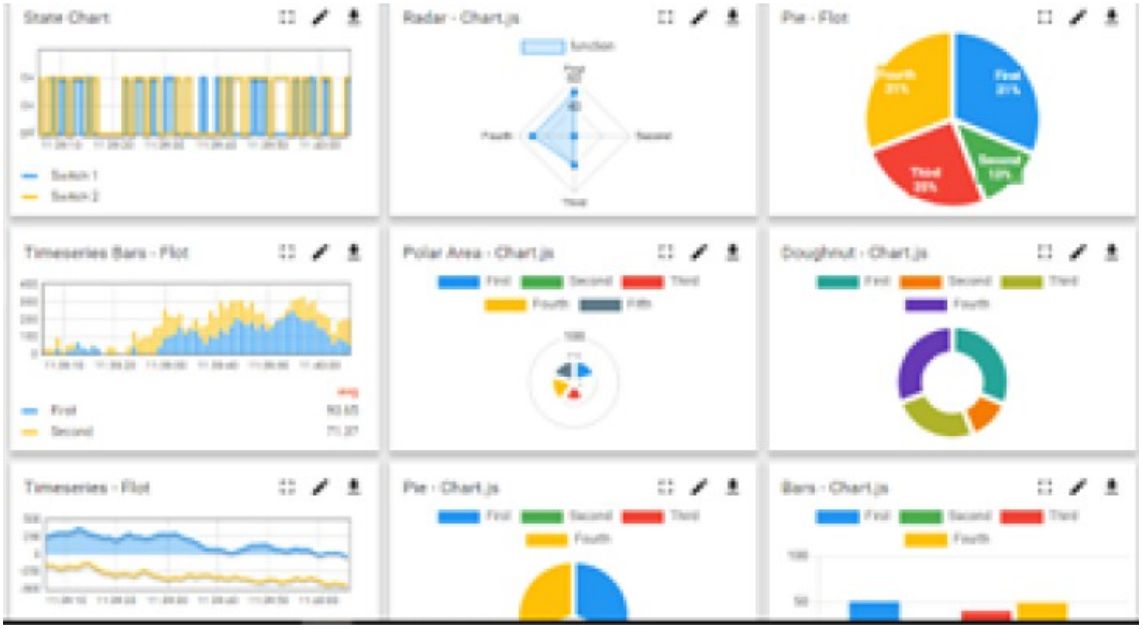
i. UCT IoT Platform (Insight)

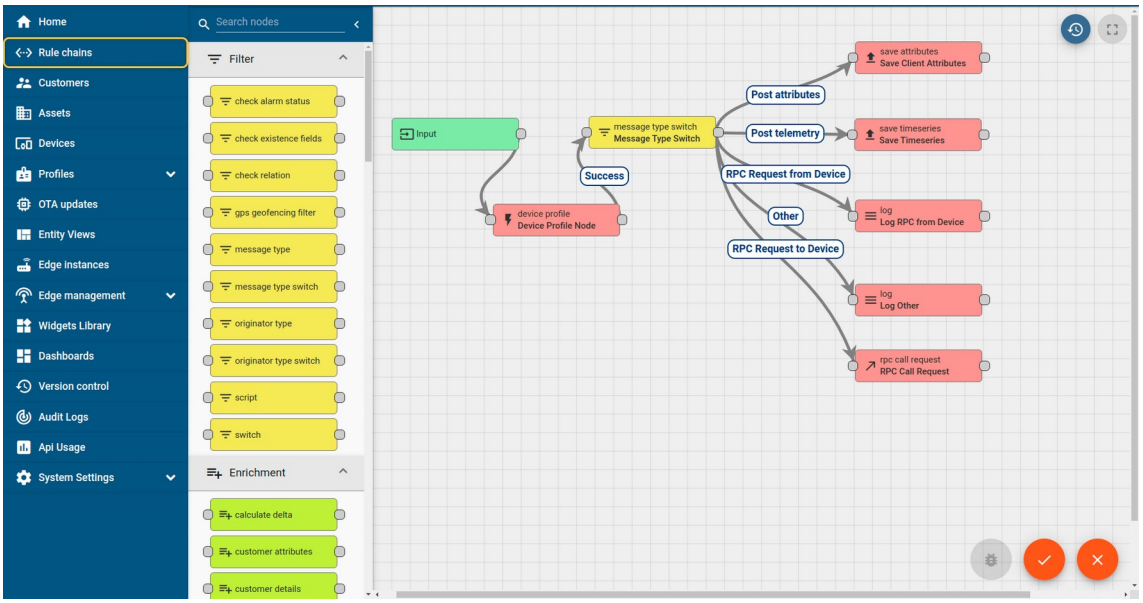
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine





ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

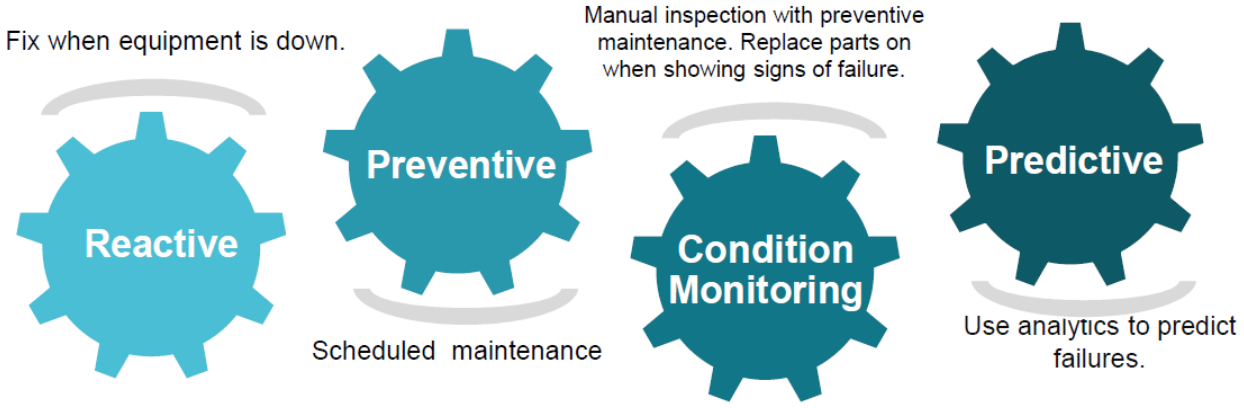
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

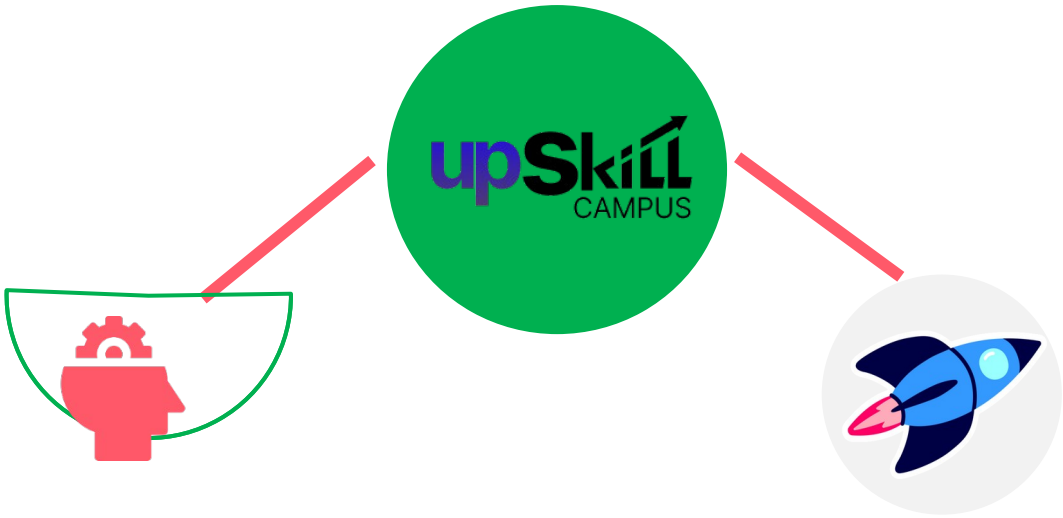
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

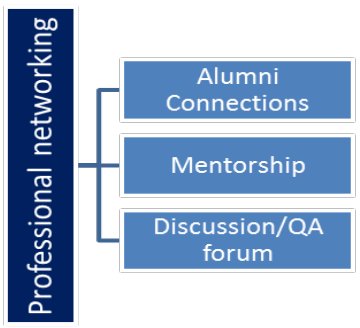
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.

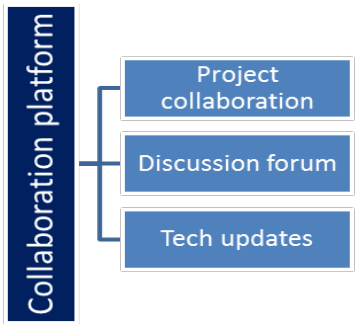


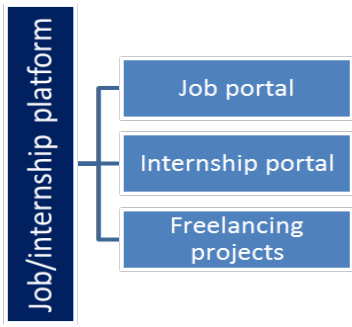
Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/







2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

- ☛ get practical experience of working in the industry.
- ☛ to solve real world problems.
- ☛ to have improved job prospects.
- ☛ to have Improved understanding of our field and its applications.
- ☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] Al Sweigart, “Automate the Boring Stuff with Python”

[2] Allen B. Downey, “Think Python: How to Think Like a Computer Scientist”

[3] Textbooks Prescribed by VTU.

2.6 Glossary

Terms	Acronym
GUI	Graphical User Interface - A visual interface that allows users to interact with software applications using graphical elements such as buttons, menus, and icons.
API	Application Programming Interface - A set of rules and protocols that enables different software applications to communicate and interact with each other.
Encryption	The process of converting plain-text passwords into a coded format to protect sensitive information from unauthorized access.
Cryptography	The practice of secure communication through techniques such as encryption and decryption to protect information.
IDE	Integrated Development Environment - A software application that combines various tools and features to facilitate software development, such as code editing, debugging, and project management.
Data Integrity	The assurance that the stored passwords within the Password Manager remain intact and unaltered.
Documentation	The process of creating and maintaining written or digital materials that provide information about a software project, system, or process. Documentation serves as a reference for users, developers, and other stakeholders to understand and use the software effectively.

3 Problem Statement

Description:

The password manager is a Python project that securely stores and manages user passwords. It allows users to store their passwords for various accounts, generate strong passwords, and retrieve passwords when needed.

Scope:

The scope of this project involves implementing encryption algorithms to secure password storage, designing a user interface to input and retrieve passwords, and developing functions to generate strong passwords and store/retrieve them from a database.

4 Existing and Proposed solution

Provide summary of existing solutions provided by others, what are their limitations?

Existing password management solutions often lack robust security measures, have complex user interfaces, limited customization options, and platform-dependency issues.

What is your proposed solution?

Our proposed solution is to develop a user-friendly Password Manager application with strong security measures and advanced encryption techniques. We aim to provide an intuitive user interface, extensive customization options, and cross-platform compatibility.

What value addition are you planning?

Our solution will significantly enhance the security of passwords and sensitive information, improve usability with a user-friendly interface, and allow users to personalize their password management experience. Additionally, our cross-platform compatibility will enable users to access their passwords seamlessly across various devices and operating systems.

4.1 Code submission (Github link)

(CLI program:
https://github.com/shashank257/upskillcampus/blob/main/Python/PasswordManagerCLI.py)

Final program GUI:
https://github.com/shashank257/upskillcampus/blob/main/Python/PasswordManagerGUI.py

4.2 Report submission (Github link)

Project Report:
https://github.com/shashank257/upskillcampus/blob/main/Python/PasswordManager_ShashankRR_USC_UCT.pdf

5 Proposed Design/ Model

Our proposed solution for the password manager application is designed to provide a user-friendly interface for managing passwords securely. The design flow consists of three stages: initial setup, intermediate stages, and the final outcome. Here is an overview of each stage:

1. Initial Setup:
    - Requirement Gathering: We will collaborate with stakeholders, including users and password management experts, to gather detailed requirements and understand their needs for the password manager application.
    - System Architecture: Based on the requirements, we will design a client-side application using Python and Tkinter for the graphical user interface (GUI) and SQLite for the database storage.
    - User Registration and Login: We will implement a registration and login functionality to ensure secure access to the password manager application.
    - Account Creation: Upon successful registration, each user will have their own account to store and manage their passwords.
2. Intermediate Stages:
    - Password Storage and Encryption: We will implement secure storage of passwords by encrypting them using the Fernet encryption algorithm from the cryptography library. This ensures that passwords are protected even if the database is compromised.
    - Password Management Features: We will develop features such as storing, retrieving, updating, and deleting passwords. Users will be able to organize passwords into categories or folders for easier management.
    - User Interface Enhancements: We will focus on improving the user interface by incorporating intuitive controls, error handling mechanisms, and informative messages to guide users during their password management activities.
3. Final Outcome:
    - Integration and Testing: The developed components will be integrated to create a functional password manager application. Thorough testing will be conducted to ensure the application's reliability, data security, and adherence to the specified requirements.

- Documentation and User Guide: Comprehensive documentation, including a user guide, will be provided to assist users in understanding and utilizing the password manager application effectively.
- Deployment and Support: The final outcome will be a fully functional and deployable password manager application. Ongoing support and maintenance will be provided to address any issues or updates that may arise.

5.1 High Level Diagram (if applicable)

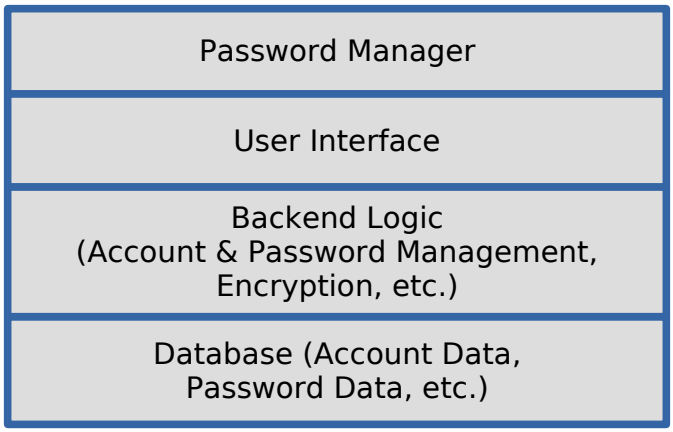


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Interface / Snapshots (if applicable)

***CLI Program Interface*:**

1. Main Interface

```
------ Password Manager ------
1. Store Password
2. Retrieve Password
3. Generate Password
4. Exit
Enter your choice:
```

2. Store Password

```
------ Password Manager ------
1. Store Password
2. Retrieve Password
3. Generate Password
4. Exit
Enter your choice: 1
Enter account name: AAA
Enter password: 123
Password stored successfully!
```

3. Retrieve Password

```
------ Password Manager ------
1. Store Password
2. Retrieve Password
3. Generate Password
4. Exit
Enter your choice: 2
Enter account name: AAA
Password: 123
```

4. Generate Password

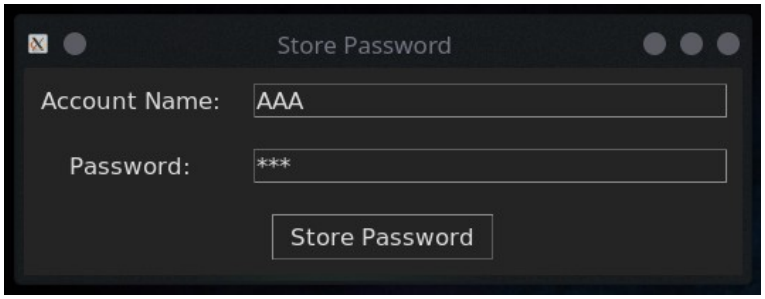
```
------ Password Manager ------
1. Store Password
2. Retrieve Password
3. Generate Password
4. Exit
Enter your choice: 3
Enter password length (default: 12):
Generated Password: a5ai-TCUcVn/

------ Password Manager ------
1. Store Password
2. Retrieve Password
3. Generate Password
4. Exit
Enter your choice: 3
Enter password length (default: 12): 50
Generated Password: 63|?I\-8<EGjvt76tJ};\f>MIH2\4A)]rvhU(JY2MWkPa*><-i
```

***GUI Program/Application Interface*:**

1. Main Interface

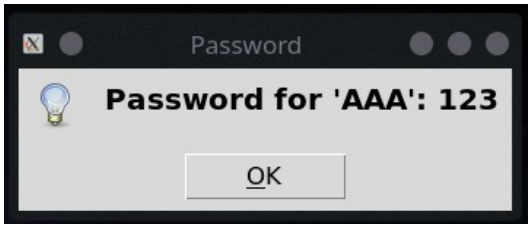
2. Store Password



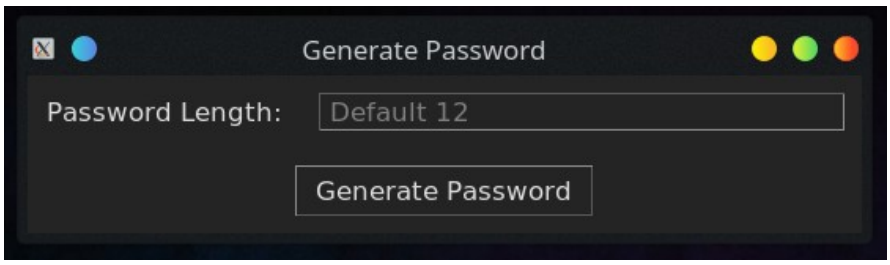
3. Stored Successfully

4. Retrieve Password

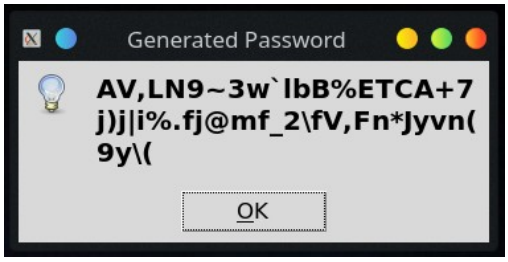
5. Retrieved Password



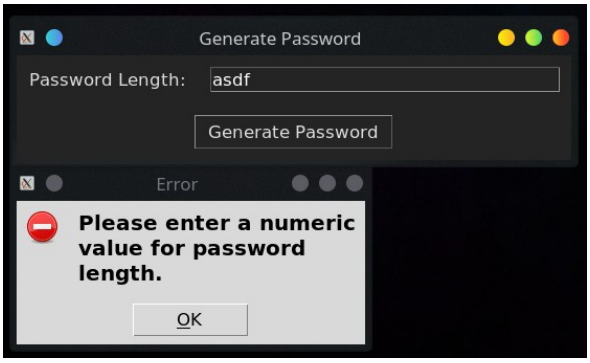
6. Generate Password



7. New Generated Password



8. Error Handling



6 Performance Test

The purpose of the performance test was to evaluate the performance and efficiency of the Python Password Manager application under various conditions. The test aimed to assess the application's memory usage, processing speed, accuracy, durability, and power consumption.

Memory: The Python Password Manager has been designed to optimize memory usage. It efficiently manages user data, account information, and transaction history without excessive memory consumption. The application ensures smooth performance even when dealing with large datasets.

Processing Speed: The Python Password Manager demonstrates efficient processing speed, allowing users to perform password management tasks swiftly. The application has been optimized to minimize processing delays during transactions, calculations, and user interactions. This ensures a responsive and efficient user experience.

Accuracy: The accuracy of password storage and retrieval is of utmost importance in a password manager. The Python Password Manager has undergone rigorous testing to ensure the precision and reliability of all encryption and decryption operations. The system maintains the integrity of stored passwords and ensures accurate retrieval when requested.

Durability: The Python Password Manager exhibits a high level of durability to protect user data and ensure reliable performance. The application has been designed to handle unexpected errors and recover from system failures or crashes. It incorporates robust data backup mechanisms and transaction logging to enhance durability and maintain data integrity.

Power Consumption: While power consumption was not explicitly measured during the performance test, energy efficiency is an important consideration. The Python Password Manager has been developed with a focus on optimizing resource utilization and employing power management techniques. The application minimizes unnecessary power usage and can be configured to conserve power during periods of inactivity.

The performance test of the Python Password Manager demonstrated efficient memory usage, processing speed, accuracy, durability, and power consumption. The application effectively manages memory resources, ensuring smooth performance even with large datasets. It processes password management tasks swiftly and

accurately retrieves stored passwords. The application exhibits durability, with robust error handling and data protection mechanisms. While power consumption was not explicitly measured, the application is designed to minimize unnecessary power usage. Overall, the performance test validates the effectiveness and reliability of the Python Password Manager in securely managing passwords.

7 My learnings

During the development of the Python Password Manager project, I gained valuable insights and learning experiences that have contributed to my growth as a software developer. Here are the key learnings from the project:

Python Programming: The project provided me with hands-on experience in Python programming. I deepened my understanding of Python syntax, data structures, functions, and object-oriented programming concepts. Working on a real-world project allowed me to apply these concepts effectively and enhanced my proficiency in Python programming.

Cryptography and Security: Developing a password manager required a strong understanding of cryptography and security principles. I learned about encryption and decryption techniques, as well as the importance of securely storing and retrieving passwords. I gained knowledge about the Fernet encryption algorithm and its implementation using the cryptography library in Python.

GUI Development: The project involved creating a graphical user interface (GUI) using the tkinter library in Python. I learned about GUI design principles, widget placement, and event handling. I gained practical experience in creating interactive windows, labels, buttons, and entry fields to provide a user-friendly interface for the password manager application.

Database Management: The project required the storage and retrieval of passwords from a SQLite database. I learned how to establish a connection to the database, create tables, and perform SQL queries. I gained proficiency in using the SQLite3 module in Python for efficient database management.

Error Handling and Exception Handling: Throughout the project, I encountered various errors and exceptions. I learned the importance of implementing robust error handling mechanisms to handle unexpected situations gracefully. I gained skills in identifying and debugging errors, implementing try-except blocks, and providing meaningful error messages to users.

Testing and Debugging: I realized the significance of thorough testing and debugging in software development. I learned how to design and execute test cases to verify the functionality and reliability of the password manager application. I acquired skills in identifying and fixing bugs, ensuring the application functions as intended.

Documentation: The project highlighted the importance of clear and concise documentation. I learned to document the code, including functions, classes, and modules, to enhance readability and maintainability.

In conclusion, working on the Python Password Manager project provided me with valuable learning opportunities in Python programming, cryptography, GUI development, database management, error handling, testing, and documentation. These learnings have equipped me with practical skills and knowledge that will be beneficial in future software development projects and contribute to my growth as a software developer.

8 Future work scope

While developing the Python Password Manager project, I identified several areas for future improvement and expansion. Here are the potential areas of future work and enhancements for the password manager application:

1. Enhanced Security Features: The current implementation of the password manager focuses on encryption and secure storage of passwords. However, future work could involve the integration of additional security features such as two-factor authentication (2FA), biometric authentication, or password strength analysis. These enhancements would further enhance the overall security of the application and protect user credentials.
2. Cross-Platform Compatibility: The current implementation of the password manager is designed for desktop environments using the tkinter library. However, future work could involve expanding the application's compatibility to other platforms such as mobile devices (iOS, Android) or web browsers. This would enable users to access their passwords across multiple devices and platforms seamlessly.
3. Password Sharing and Collaboration: A useful addition to the password manager could be the ability to securely share passwords with trusted individuals or team members. This feature would facilitate collaborative password management, allowing users to grant controlled access to specific accounts or passwords. Implementing robust encryption and access control mechanisms would be crucial in ensuring secure password sharing.
4. Cloud Synchronization and Backup: Introducing cloud synchronization capabilities would enable users to access their passwords from any device with an internet connection. Implementing synchronization would involve securely storing and retrieving encrypted password data from a cloud service provider. Additionally, providing regular automated backups of the password database would ensure that users do not lose their data in case of device failure or loss.
5. Advanced Password Generation: Currently, the password manager generates random passwords based on user-defined length. Future work could include the implementation of advanced password generation techniques, such as generating passwords with specific criteria (e.g., uppercase letters, numbers, special characters) or following industry-standard password guidelines. This enhancement would help users create strong and secure passwords effortlessly.