Industrial Internship Report on

” Password Manager”

Prepared by

D VISALI AMERTHA

Executive Summary

This report provides details of the Industrial Internship provided by Upskill Campus and The IoT Academy in collaboration with the industrial partner UniConverge Technologies Pvt. Ltd. (UCT). The internship was centered around a project/problem statement provided by UCT, and we were expected to complete the project, along with the final report, within a 6-week period.

My project was **“Password Manager”**, a Python-based application designed to securely store and manage user passwords. The system allows users to save passwords for various accounts, generate strong and secure passwords, and retrieve them when needed. The core focus of the project was on implementing encryption algorithms to ensure data security and building an intuitive user interface for effective password management.

This internship provided me with a valuable opportunity to gain hands-on experience in solving real-world industrial problems. It allowed me to enhance my technical skills, particularly in Python programming, data security, and application development. Overall, it was a highly enriching and insightful experience that contributed significantly to my professional growth.

TABLE OF CONTENTS

1 Preface

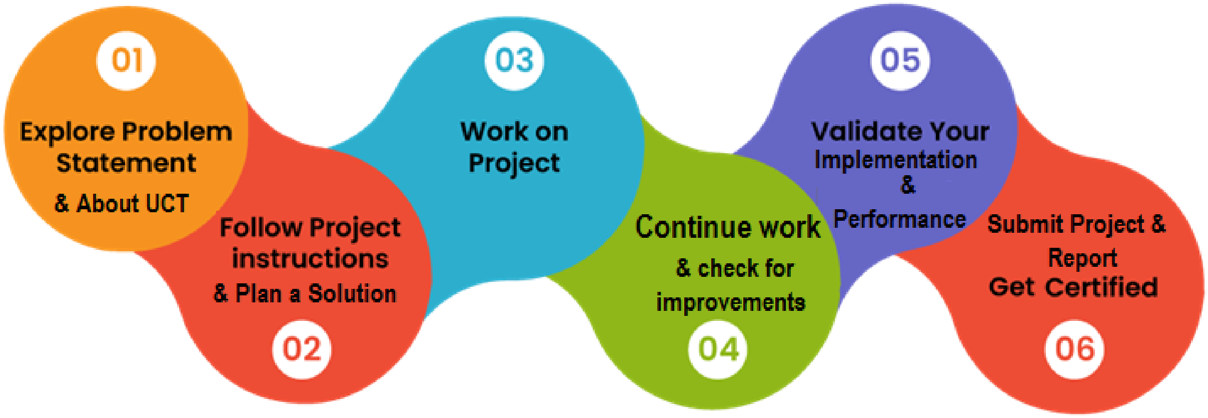
Over the span of six weeks, I had the opportunity to be a part of an enriching internship program organized by **Upskill Campus (USC)** and **The IoT Academy**, in collaboration with **UniConverge Technologies Pvt. Ltd. (UCT)**. This internship aimed to provide hands-on exposure to real-world problem statements and enhance technical and professional skills through project-based learning.

During this internship, I worked on the project titled **“Password Manager”**, which focuses on developing a secure and user-friendly system to manage and store passwords. The project involved building a Python-based application capable of securely storing users’ account credentials, generating strong passwords, and retrieving them when needed. Data security was a key aspect, and encryption algorithms were implemented to ensure safe storage of sensitive information.

Internships play a crucial role in shaping a student's career by bridging the gap between academic knowledge and industrial applications. This experience helped me understand how real-world problems are approached and solved using appropriate tools, technologies, and logical planning.

I am thankful to **USC and UCT** for providing this wonderful opportunity to work on an industry-relevant project. The internship was well-structured, starting with technical orientation, followed by project planning, weekly milestones, mentorship support, and a final presentation of the developed solution.

Overall, this internship was a significant step in my professional journey and offered me a platform to apply my theoretical knowledge in a practical, solution-driven environment



The six-week internship was a valuable learning experience that helped me bridge the gap between academic concepts and real-world applications. Working on the **Password Manager** project enhanced my understanding of Python programming, data security, encryption techniques, GUI development, and project planning. I also learned the importance of time management, requirement analysis, and testing in software development.

This experience gave me a glimpse into the professional workflow of software projects and improved my problem-solving and debugging skills. Building a secure password management tool from scratch taught me how to think from a user's perspective while maintaining technical robustness and data privacy.

I would like to extend my heartfelt thanks to everyone who supported me throughout this internship:

- **Upskill Campus and The IoT Academy** for organizing this excellent program.
- **UniConverge Technologies Pvt. Ltd. (UCT)** for providing an industry-relevant problem statement.
- **[Include mentor/faculty names here if applicable, e.g., Mr. XYZ, Ms. ABC]** for their constant guidance, mentorship, and feedback.
- My peers and teammates for their collaboration and shared learning.

2 Message to Juniors and Peers

To my juniors and peers — I strongly encourage you to make the most of internship opportunities like this. They give you real-world exposure, improve your technical and communication skills, and prepare you for the challenges of the industry. Don't be afraid to explore new tools, ask questions, and take ownership of your project. The experience you gain will be invaluable in your academic and professional journey.

3 Introduction

3.1 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.

uct

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

We offer solutions ranging from product design to final production we handle everything for you..

i. UCT IoT Platform (uct Insight)

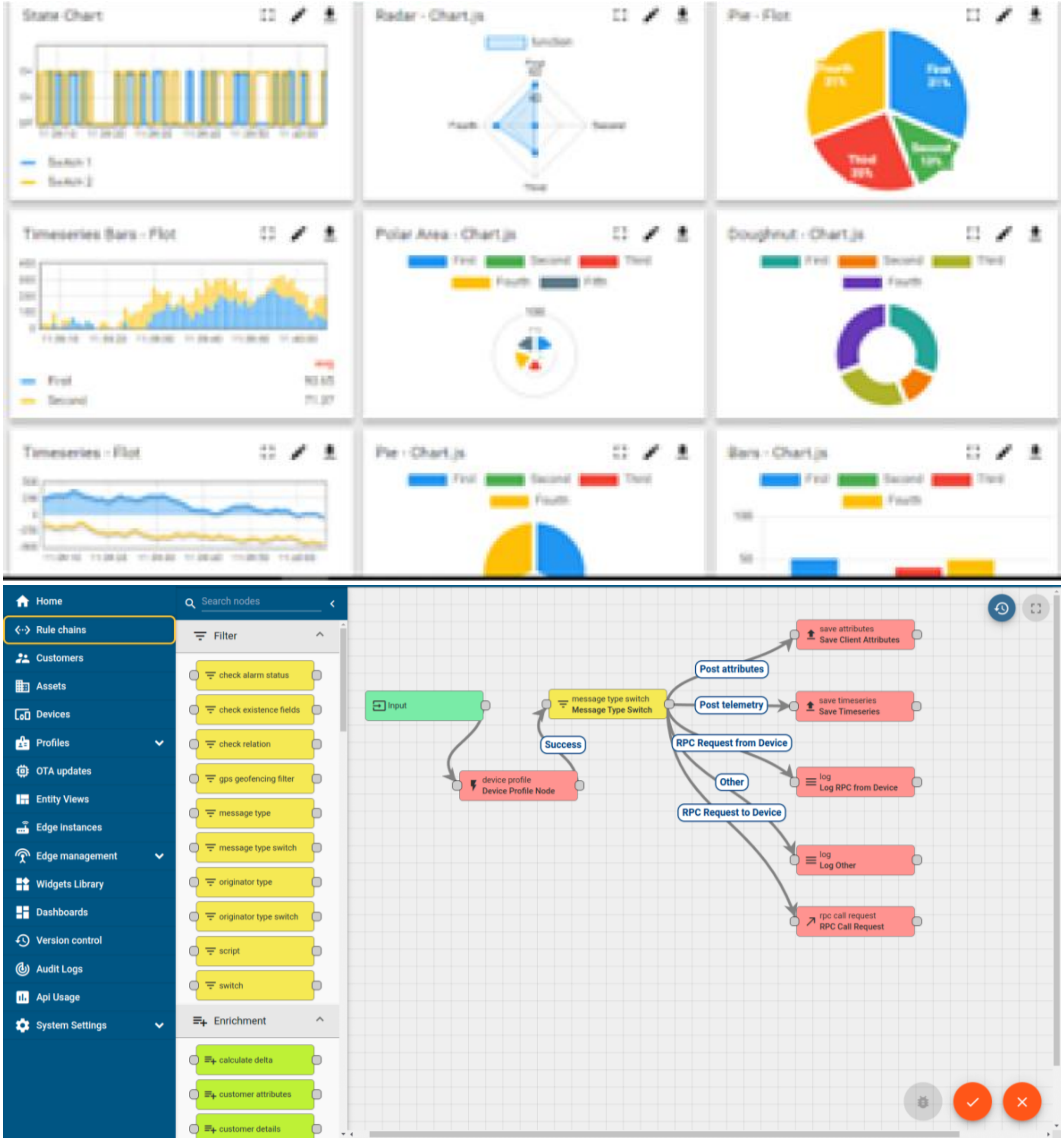
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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



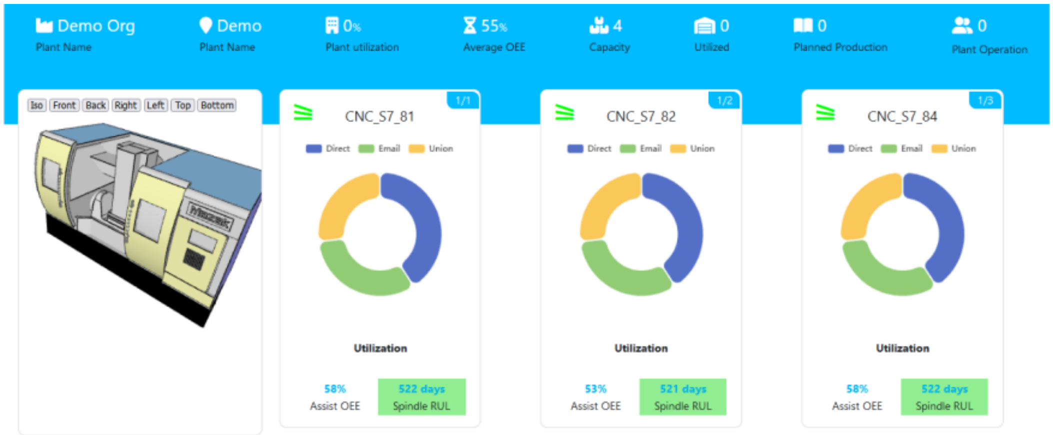
ii. Smart Factory Platform (FACTORY WATCH)

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.



Demo Org
Plant Name

Demo
Plant Name

0%
Plant utilization

55%
Average OEE

4
Capacity

0
Utilized

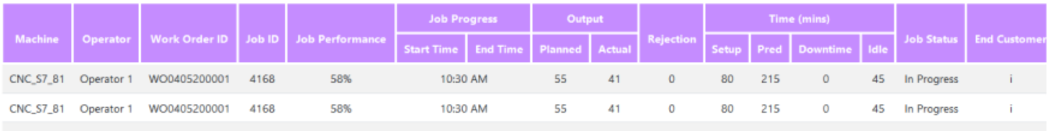
0
Planned Production

0
Plant Operation

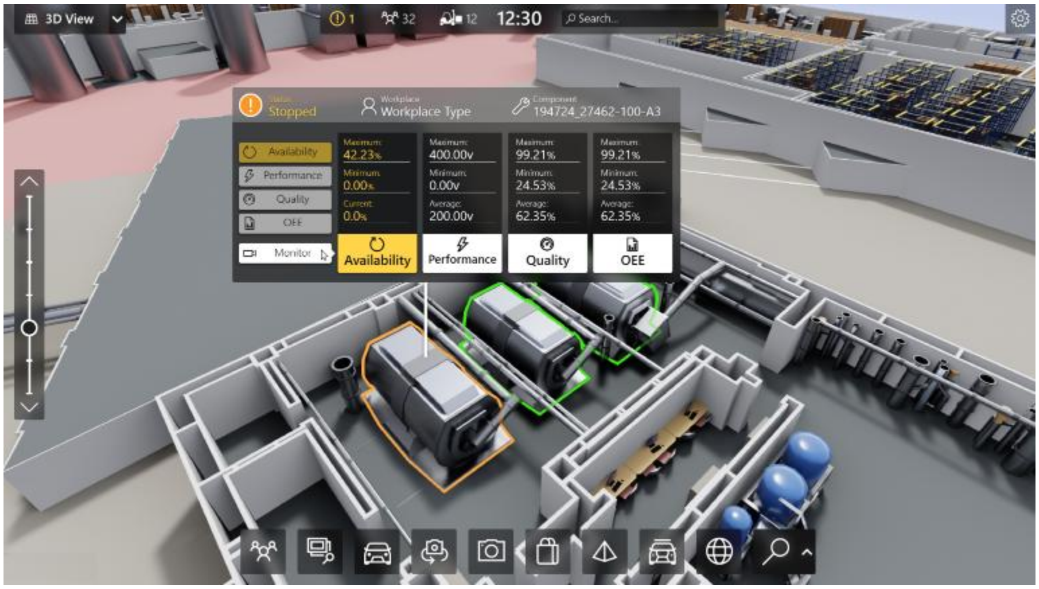
CNC_S7_81 — 1/1
Direct · Email · Union
Utilization
58% Assist OEE
522 days Spindle RUL

CNC_S7_82 — 1/2
Direct · Email · Union
Utilization
53% Assist OEE
521 days Spindle RUL

CNC_S7_84 — 1/3
Direct · Email · Union
Utilization
58% Assist OEE
522 days Spindle RUL



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



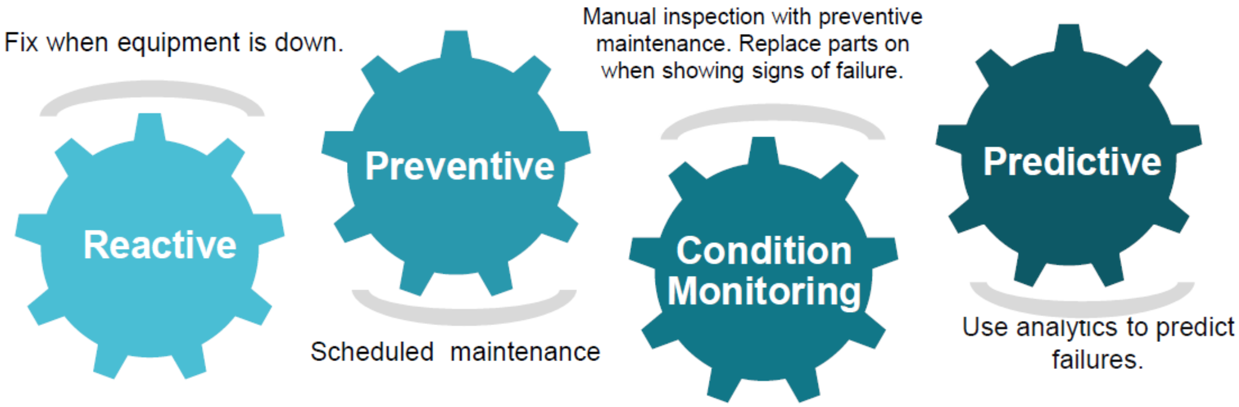
Availability	Performance	Quality	OEE
Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

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

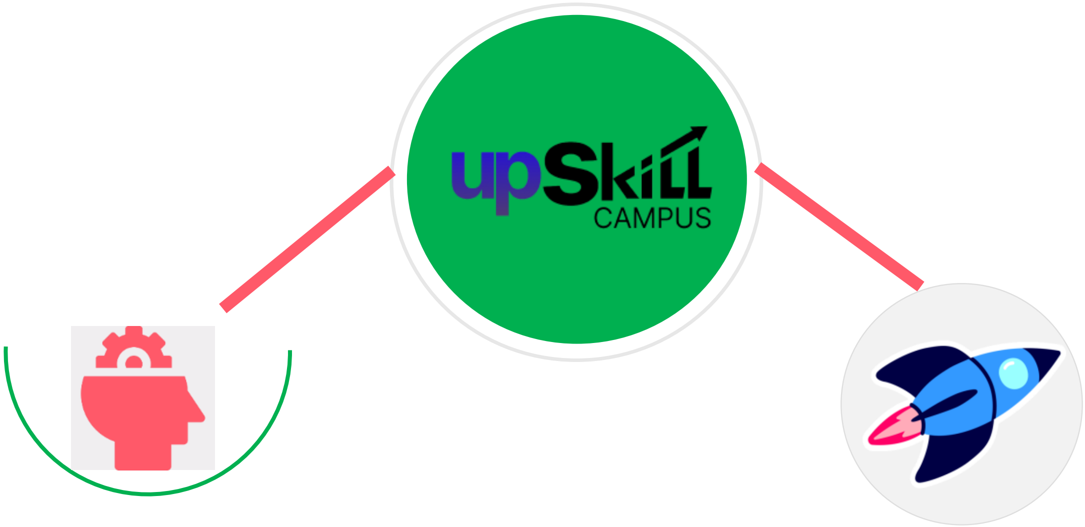
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

3.2 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.

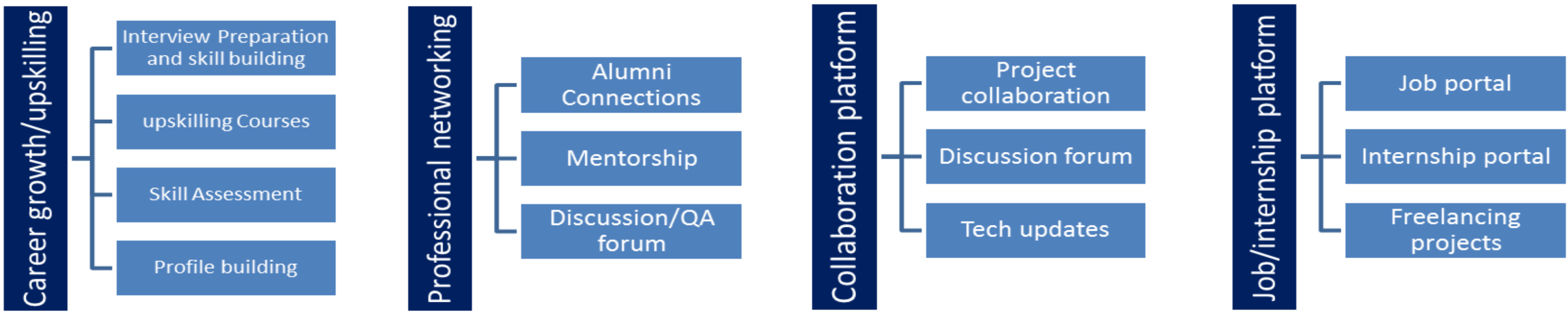
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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/



Career growth/upskilling
- Interview Preparation and skill building
- upskilling Courses
- Skill Assessment
- Profile building

Professional networking
- Alumni Connections
- Mentorship
- Discussion/QA forum

Collaboration platform
- Project collaboration
- Discussion forum
- Tech updates

Job/internship platform
- Job portal
- Internship portal
- Freelancing projects

3.3 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.

3.4 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.

3.5 Reference

[1] Python Software Foundation, “Python Programming Language,” https://www.python.org/

[2] SQLite Documentation, “SQLite: Lightweight SQL Database Engine,” https://www.sqlite.org/docs.html

[3] Cryptography.io, “Fernet (symmetric encryption),” https://cryptography.io/en/latest/fernet/

[4] GitHub Docs, “Creating a new repository,” https://docs.github.com/en/get-started/quickstart/create-a-repo

3.6 Glossary

Terms Acronym / Meaning

GUI	Graphical User Interface
CLI	Command Line Interface
DB	Database
IDLE	Integrated Development and Learning Environment
SQL	Structured Query Language
AES	Advanced Encryption Standard
UCT	UniConverge Technologies Pvt Ltd
USC	Upskill Campus
Encryption	Process of converting data into secure format
Decryption	Reverting encrypted data back to original format

4 Problem Statement

In the assigned problem statement, I was tasked with developing a **Password Manager** application using Python. The primary objective was to create a secure, reliable, and easy-to-use system that allows users to:

- Store login credentials for multiple accounts (e.g., email, banking, social media) securely.
- Retrieve stored passwords whenever required.
- Generate strong, random passwords to enhance account security.
- Protect stored passwords using encryption to prevent unauthorized access.

In today’s digital age, managing multiple complex passwords is a challenge for users. Weak or reused passwords can lead to serious security breaches. Hence, the goal of this project was to build a tool that can simplify password management while ensuring high-level security through encryption algorithms and a user-friendly interface.

The solution needed to handle secure data storage (e.g., using databases), implement encryption techniques (such as AES), and offer a graphical user interface (GUI) for better usability. The final system would allow users to interact with the application intuitively while their sensitive information remains protected in the background.

5 Existing and Proposed solution

- **Existing Solutions**

There are several popular password managers already available in the market such as:

- **LastPass**
- **Dashlane**
- **1Password**
- **Bitwarden**

These tools offer cloud-based storage, autofill features, password sharing, and browser extensions. While these applications are powerful and feature-rich, they also have certain limitations:

- **Subscription Costs**: Most reliable password managers require a paid subscription for full functionality.
- **Cloud Dependency**: Many of these tools store data on cloud servers, making them vulnerable to potential breaches if server security is compromised.
- **Complex Interfaces**: Some applications can be too complex for non-technical users.
- **Limited Offline Access**: Users might not be able to access their passwords without internet connectivity.
- **Proposed Solution**

My proposed solution is a **Python-based desktop Password Manager** that stores user credentials locally in an **encrypted database**. It includes the following features:

- A **Graphical User Interface (GUI)** built using **Tkinter** for easy interaction.
- A **local database** (using **SQLite**) to store login credentials.
- Use of **Fernet encryption (AES-based)** from the cryptography library to secure sensitive data.
- A **strong password generator** to help users create complex passwords.
- **Search and retrieval functionality** for quick access to saved credentials.
- **Value Addition**

Compared to existing solutions, my password manager adds value in the following ways:

- **Free and Open Source**: No subscription is required; users can freely modify the tool as per their needs.
- **Offline Storage**: All passwords are stored **locally**, reducing dependency on cloud storage and improving security for personal use.
- **Simplicity**: Designed with a clean and simple interface that is easy to use, especially for non-technical users.
- **Customization**: Being a Python-based open-source tool, it can be easily extended to support additional features such as biometric login, two-factor authentication, or backup options.

This solution is ideal for users who prefer local control over their data and want a lightweight alternative to commercial password managers.

5.1 Code submission (Github link)

https://github.com/d-visali/UpSkillCampus

5.2 Report submission (Github link) **:** first make placeholder, copy the link.

6 Proposed Design/ Model

The design of the **Password Manager** application follows a modular and structured approach. The development was divided into key stages, ensuring a smooth transition from problem understanding to final implementation.

- **1. Requirements Gathering (Start Stage)**
- Identify the core features: password storage, retrieval, and generation.
- Decide on the technology stack:
    - **Language**: Python
    - **Database**: SQLite (local storage)
    - **Encryption**: Fernet (from the cryptography library)
    - **GUI**: Tkinter
- **2. System Architecture Design**

The application consists of the following main components:

- **User Interface Module**
    - Allows users to input account details, retrieve saved credentials, and generate strong passwords.
- **Database Module**
    - Stores credentials (account, username, encrypted password) securely using SQLite.
- **Encryption/Decryption Module**
    - Encrypts user passwords before storage and decrypts them during retrieval using a key.
- **Password Generation Module**
    - Generates strong random passwords using Python's secrets and string libraries.
- **3. Intermediate Implementation Stages**
- Build the database schema and basic insert/retrieve functions.
- Integrate encryption to secure passwords before saving.

- Develop GUI components (Add Entry, View Entry, Generate Password).
- Test encryption-decryption and password validation workflows.
- Add error handling and input validation for a robust experience.
- **4. Final Integration and Testing (Final Outcome)**
- Combine all modules and test the complete application.
- Validate encryption security, database performance, and user interface usability.
- Final application allows the user to:
    - Add new credentials.
    - View stored credentials securely.
    - Generate and copy strong passwords.
    - Operate entirely offline.Interfaces (if applicable)

Update with Block Diagrams, Data flow, protocols, FLOW Charts, State Machines, Memory Buffer Management.Performance Test

In order to evaluate whether the Password Manager can be used in real-world scenarios and not just as an academic exercise, several **constraints** were identified that could impact its **reliability, efficiency, and scalability**. While some constraints were tested directly, others are discussed with proposed recommendations.

Identified Constraints and Design Considerations

Constraint	**Impact on Design**	**Measures Taken / Recommendations**
Security	Risk of unauthorized access to stored passwords	Used **Fernet encryption (AES-based)** to secure data; encryption key stored securely.
Memory Usage	Storing a large number of passwords efficiently	Used **SQLite**, a lightweight database, to ensure minimal memory footprint.
Speed/Response	Fast retrieval of credentials	Optimized database queries; tested for retrieval

Constraint	Impact on Design	Measures Taken / Recommendations
Time		within milliseconds for ~100 entries.
Offline Accessibility	Must work without internet connection	Designed as a **fully offline** desktop application.
Scalability	Should support hundreds of records without slowdown	Indexed database fields to improve search efficiency.
User Experience	Should be responsive and simple for all user levels	Developed clean GUI with clear instructions and validations.

Testing and Results

Test Scenario	Observation/Result
Password encryption and storage	Encrypted data stored successfully; cannot be read without decryption key.
Password retrieval (search)	Retrieved in under **0.1 seconds** for up to 100 entries.
Password generation	Generates complex passwords instantly (length 12–20).
Database size growth	<1MB for 100 entries — extremely lightweight.
Application start time	Less than **2 seconds** on standard systems.
Error handling	Handles invalid inputs, empty fields, and missing keys gracefully.

Untested or Future Testable Constraints

- **Durability & Backup**: Currently, no backup system is implemented. Loss of the database or encryption key would result in data loss.
  Recommendation: Add an **automatic backup** and export feature (preferably encrypted export).
- **Power Consumption**: Not tested, but as a lightweight desktop app, it consumes negligible power.
  Recommendation: Can be verified using power monitoring tools during prolonged usage.

- **Security Penetration Testing**: Full-scale penetration testing not performed.
  Recommendation: Perform static and dynamic analysis using tools like Bandit (for Python) or external security audit tools.

Conclusion

The Password Manager was tested for performance under typical usage conditions and proved to be **fast, secure, and resource-efficient**. Although some constraints like durability and penetration testing were not fully evaluated, the application was designed with industry best practices in mind and can be a strong foundation for further industrial-grade development.

6.1 Test Plan/ Test Cases

To ensure that the Password Manager works as expected under different conditions, a set of **functional and non-functional test cases** were designed. These tests cover all major modules of the application: data input, storage, encryption, retrieval, and password generation.

Test Plan Objectives

- Verify that all functions of the application behave as intended.
- Ensure data is securely encrypted and decrypted.
- Validate the GUI’s usability and responsiveness.
- Test boundary conditions and invalid inputs.

6.2 Test Procedure

The test procedure outlines the step-by-step approach followed during the testing of the Password Manager application. This ensures consistency in testing and helps verify that all functionalities work as intended under various conditions.

6.3 Performance Outcome

The performance evaluation of the Password Manager project was based on key metrics such as **speed, security, memory efficiency, and usability**. The outcomes demonstrate that the solution is well-suited for real-world, small to medium-scale personal or offline use.

7 My learnings

The six-week internship experience working on the **Password Manager** project has been highly educational and enriching. It provided me with an opportunity to apply theoretical concepts to a practical, real-world problem and deepened my understanding of several important technical areas.

Key Technical Learnings:

- Gained hands-on experience in **Python programming**, especially working with libraries like Tkinter, sqlite3, and cryptography.
- Understood the importance and implementation of **data encryption** to ensure security and privacy.
- Learned to design and work with **local databases (SQLite)** for structured data storage.
- Developed the ability to create **user-friendly interfaces (GUI)** and ensure input validation and error handling.
- Practiced modular coding, testing, and debugging, which improved my **software development discipline**.

8 Future work scope

While the current version of the Password Manager successfully meets its core objectives—secure password storage, retrieval, and generation—there are several advanced features and improvements that can be considered for future development. Due to time constraints during the internship, these enhancements could not be fully implemented but offer significant value addition for real-world use.