Industrial Internship Report on

”Password Manager”

Prepared by

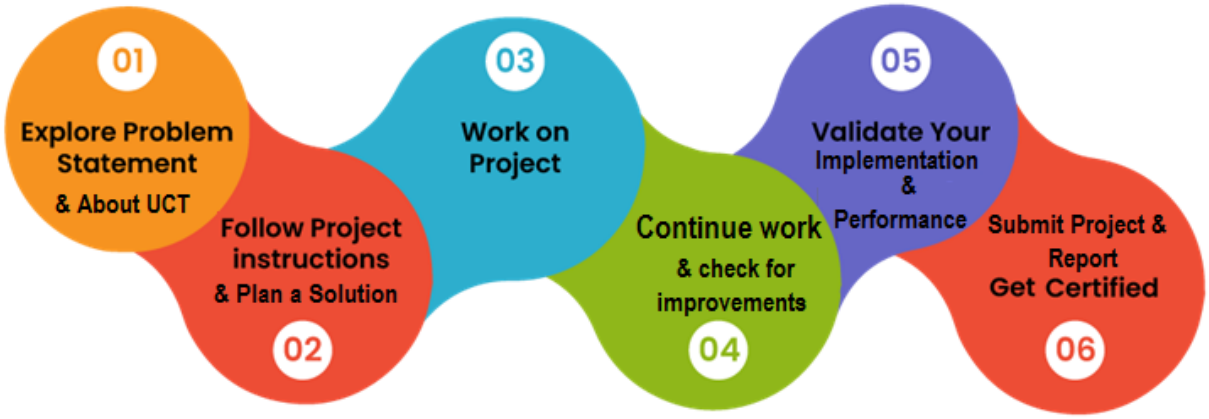
GAGAN DHANAPUNE

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 creating a project manager, which helps the user to save and recall the old password which were saved earlier. 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

In this whole 6-week internship I got to learn a lot of things related with technical terms as well as terms related with industrial experience too. In this project of creating password manager, we have created a user-friendly interface which helps the user to save the password foe different websites and different E-mails, which we can get after recalling with the input of website and E-mail. We have completed this project with the help of tinker. We have submitted every week's progress of this project where you will find the detailed slow and steady progress of our project. We are really grateful to get this opportunity from UCT/USC. There is a massive need of relevant Internship in career development, which is fulfilled by this opportunity.



I really felt overwhelmed when I got this amazing opportunity at this stage, where a learner needs such more opportunities for inspiration and increase industrial as well as learning experiences.

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 (uct Insight)

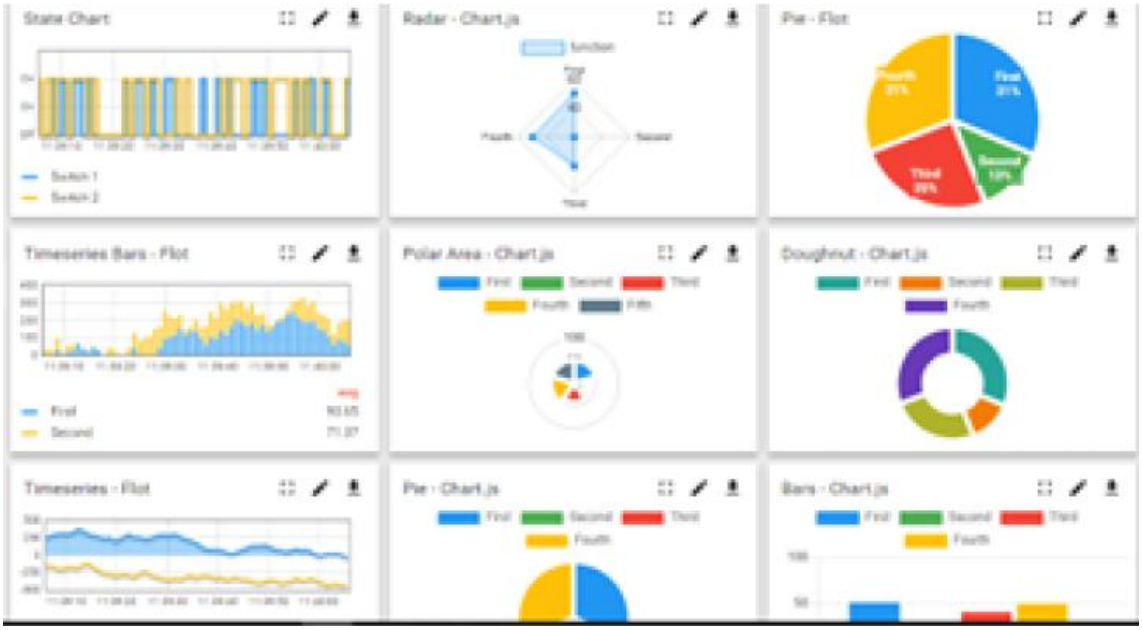
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

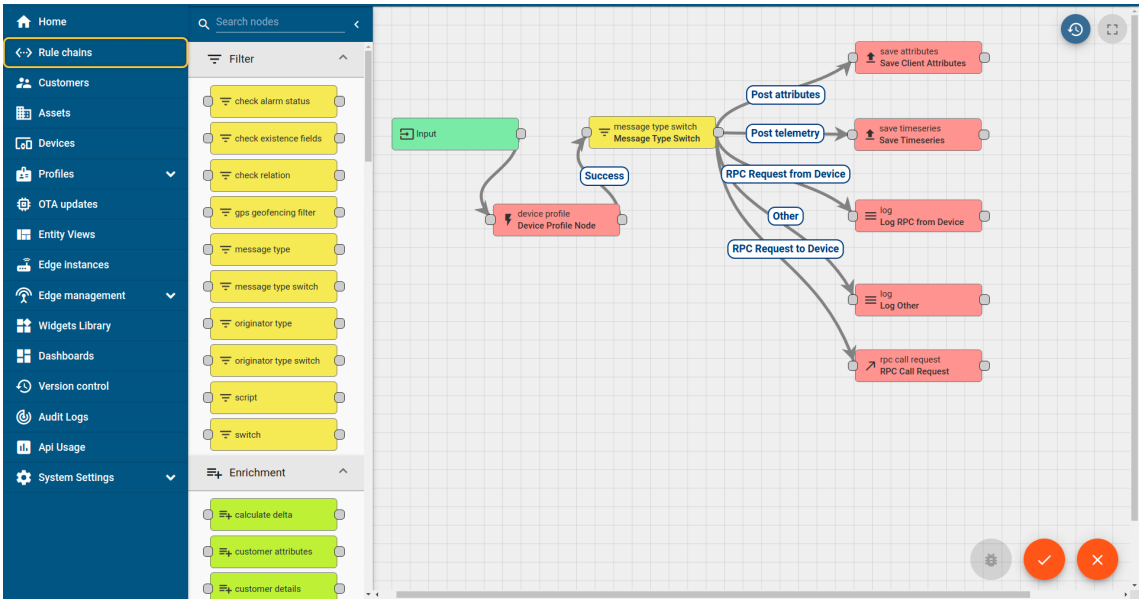
- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA

- 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.

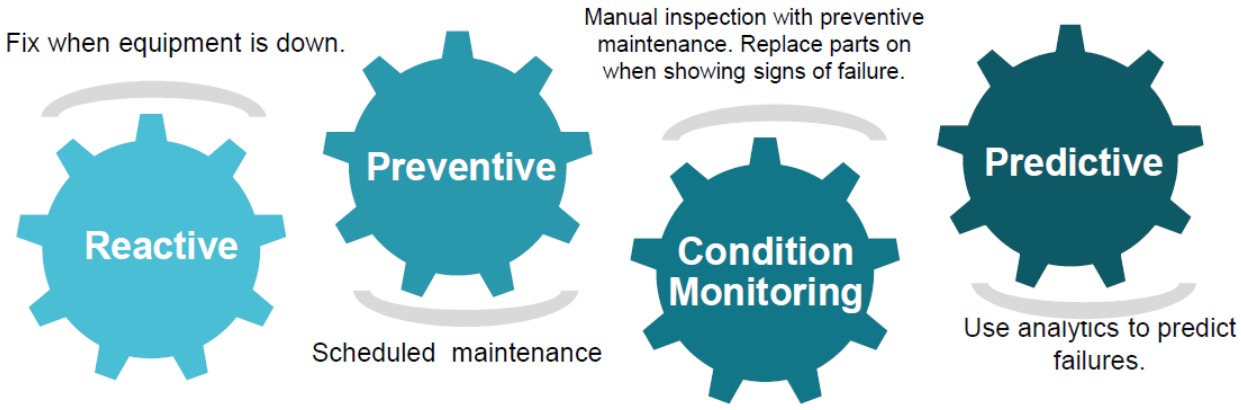
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1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

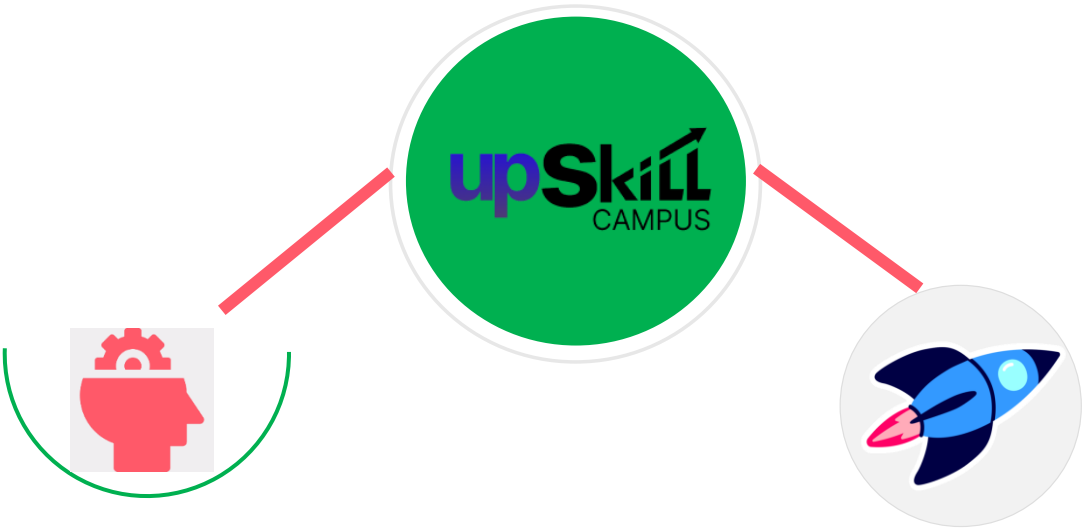
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

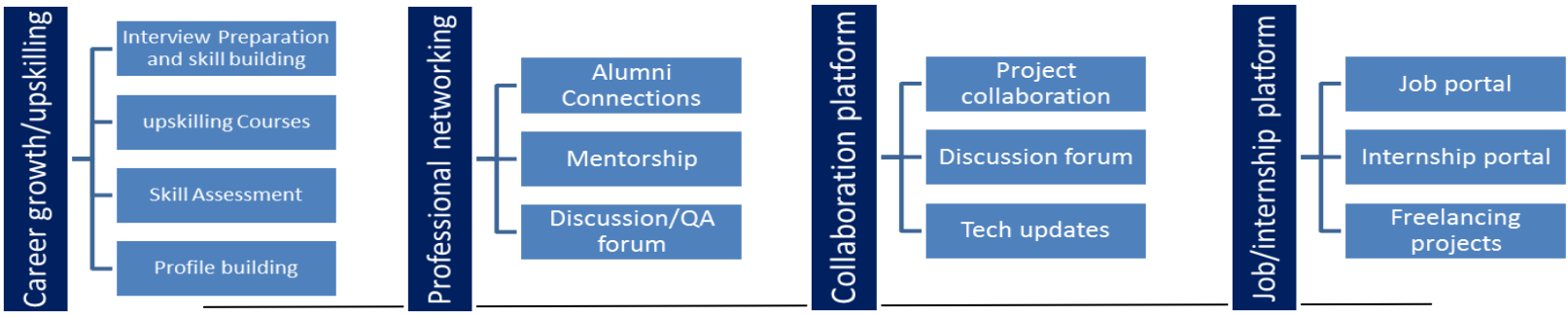
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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] GitHub

[2] YouTube

[3] AI platform

3 Problem Statement

- Password Manager:

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

Existing solutions for password managers in Python vary in complexity and features. Some basic implementations use simple file-based storage with encryption, while more advanced solutions may incorporate databases and additional security measures.

However, these solutions often have limitations. For example, some basic implementations may not adequately protect passwords if the encryption method used is weak or improperly implemented. File-based storage can also be prone to corruption or loss of data if not handled properly.

Moreover, some solutions may lack features such as password generation, secure sharing of passwords, or two-factor authentication, which are increasingly important for modern password managers.

Overall, while there are many existing solutions for password managers in Python, they often have limitations in terms of security, scalability, and features compared to more robust commercial password managers.

4.1 Code submission (Github link) :

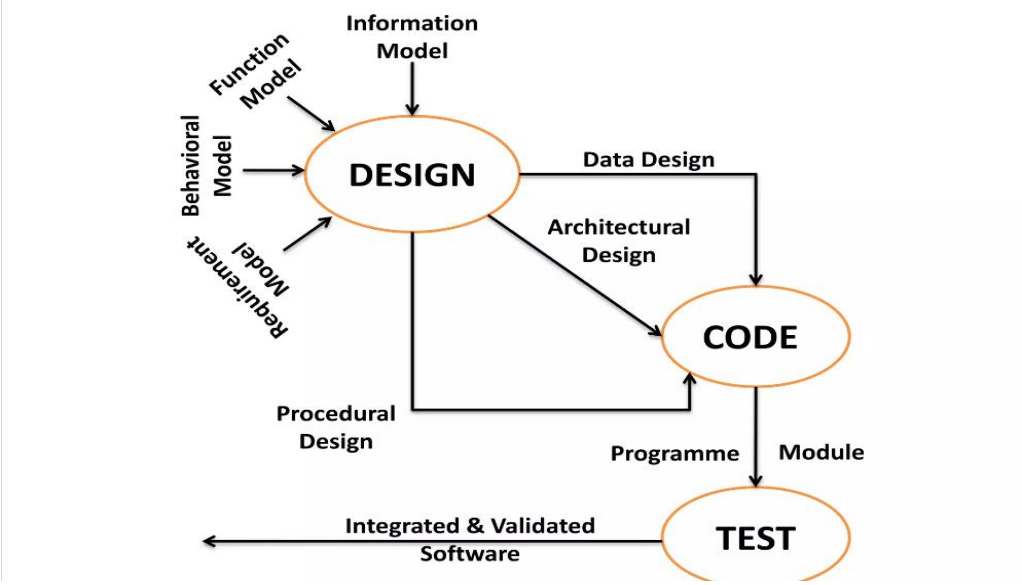
link

4.2 Report submission (Github link) :

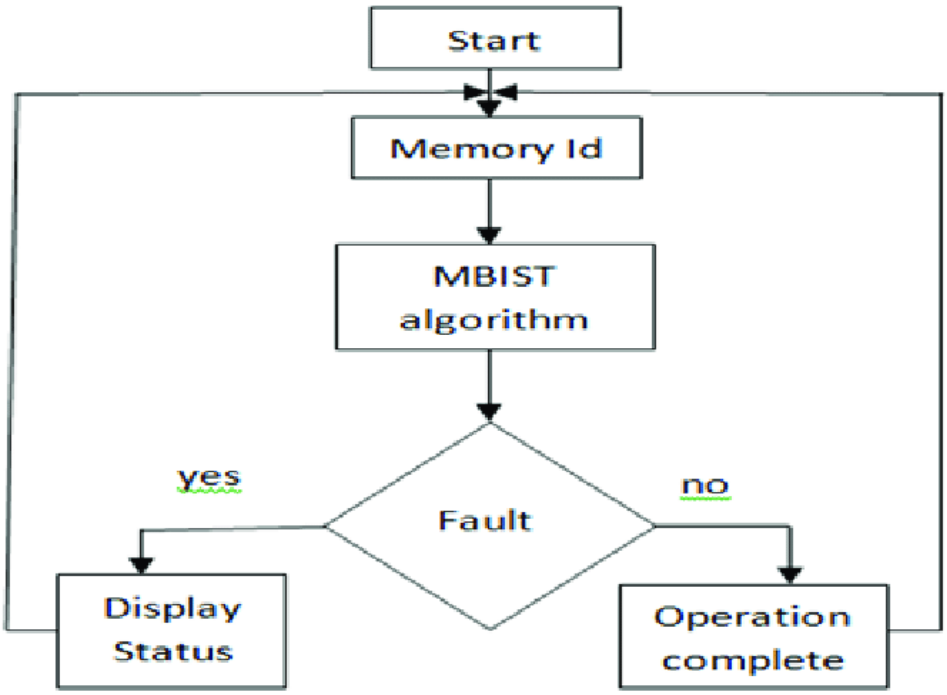
link

5 Proposed Design/ Model

5.1 High Level Diagram (if applicable)



5.2 Low Level Diagram (if applicable)



6 Performance Test

This is very important part and defines why this work is meant of Real industries, instead of being just academic project.

Here we need to first find the constraints.

How those constraints were taken care in your design?

What were test results around those constraints?

Constraints can be e.g. memory, MIPS (speed, operations per second), accuracy, durability, power consumption etc.

In case you could not test them, but still, you should mention how identified constraints can impact your design, and what are recommendations to handle them.

6.1 Test Plan/ Test Cases

Encryption Test:

Test Case 1: Verify that passwords are encrypted before being stored.

Test Case 2: Verify that passwords are decrypted correctly when retrieved.

Storage Test:

Test Case 3: Verify that passwords are stored securely and cannot be accessed without proper authentication.

Test Case 4: Verify that passwords are stored in a format that is not easily readable.

User Interface Test:

Test Case 5: Verify that the user interface is intuitive and easy to use.

Test Case 6: Verify that users can add, edit, and delete passwords easily.

Security Test:

Test Case 7: Verify that the password manager is protected against common security threats such as SQL injection and cross-site scripting (XSS).

Test Case 8: Verify that the password manager does not store passwords in plain text.

Compatibility Test:

Test Case 9: Verify that the password manager works correctly on different operating systems (Windows, macOS, Linux).

Performance Test:

Test Case 10: Verify that the password manager performs well under heavy load, with a large number of passwords stored.

Password Generator Test:

Test Case 11: Verify that the password generator creates strong, random passwords based on user-defined criteria.

6.2 Performance Outcome

The performance of a password manager in a Python project can be evaluated through several key metrics. Response time is crucial, as users expect quick access to their passwords. Scalability is also important, ensuring the manager can handle a growing number of passwords without a significant drop in performance. Efficient use of system resources, such as CPU and memory, is essential to prevent the manager from being overly resource-intensive. The ability to handle concurrent user access is another key factor, ensuring the manager remains responsive under load. Database performance is critical if the manager uses a database, as slow database queries can significantly impact overall performance. Additionally, network performance and the speed of encryption and decryption algorithms can affect user experience. Overall, performance testing is vital to ensure the password manager meets performance requirements and provides a fast, reliable, and secure user experience.

7 My learnings

In this Python project, I have focused on developing a password manager. Through this project, I have enhanced my problem-solving skills and deepened my understanding of Python programming. I have learned about various concepts such as data encryption, file handling, and user interface design.

This project has not only improved my technical skills but also my ability to manage and organize complex information securely. As I progress in my career, these skills will be invaluable in roles that require software development, cybersecurity, or data management.

Overall, this project has been instrumental in my learning journey, equipping me with the skills and knowledge necessary for my career growth in the field of technology.

8 Future work scope

Certainly! Here are some ideas that you could not work on due to time limitations but could be added in the future to enhance your Python project report:

1. **Multi-platform Support:** Extend the password manager to support multiple platforms such as Windows, macOS, and Linux. This could involve adapting the user interface and file handling to be compatible with different operating systems.
2. **Cloud Integration:** Integrate cloud services like Google Drive, Dropbox, or iCloud to securely sync and backup passwords across devices. This would involve learning about APIs and authentication mechanisms for these services.
3. **Advanced Encryption:** Implement more advanced encryption algorithms such as AES (Advanced Encryption Standard) for enhanced security and to comply with industry standards.
4. **Password Strength Checker:** Add a feature that checks the strength of passwords entered by the user and provides suggestions for stronger passwords.
5. **Two-Factor Authentication (2FA):** Implement two-factor authentication for an added layer of security, requiring users to verify their identity using a second method such as a code sent to their phone.
6. **Password Generator:** Include a password generator feature that creates strong, random passwords based on user-defined criteria such as length and character types.
7. **Password Expiry Reminder:** Add a feature that reminds users to change their passwords regularly to maintain security.
8. **Integration with Browser Extensions:** Develop browser extensions that can autofill passwords saved in the password manager, improving user convenience and security.