**A MINI-PROJECT REPORT**

**ON**

**“PASSWORD MANAGER”**

BACHELOR OF TECHNOLOGY COMPUTER SCIENCE & ENGINEERING

Submitted By

Mayank Hete – 23070122135

Sreehari Nair – 23070122144

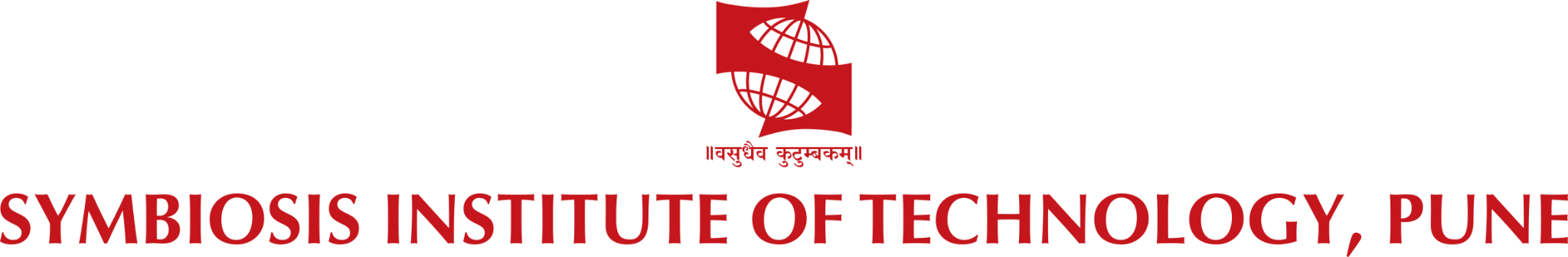
Parth Damle - 23070122161

Pratik Lakra - 23070122166

UNDER THE GUIDANCE OF

Mrs. Shwetambari Chiwhane

Professor, SIT, Pune

****

Pune – 412115, Maharashtra State, India  
<https://www.sitpune.edu.in/>

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**AY 2024-25**

## ACKNOWLEDGEMENT

I want to express my sincere gratitude to everyone who supported me throughout this mini-project. First and foremost, I would like to thank my project guide, **Mrs. Shwetambari Chiwhane**, for his valuable guidance, encouragement, and feedback. She has been a constant source of inspiration and motivation for me.

I would also like to thank the head of the department, **Dr. Deepali R. Vora**, for providing me with the necessary facilities and resources for conducting this project. I am grateful to her for her constant support and advice.

I want to acknowledge the contribution of my project team members, Mayank, Sreehari, Parth and Pratik, who have worked hard and cooperated with me in every project stage. They have been accommodating and supportive throughout this journey.

I would also like to thank my family and friends for their love, care, and support. They have always been there for me in times of need and stress. They have encouraged me to pursue my passion and achieve my goals.

Lastly, I thank **Symbiosis Institute of Technology, Pune** for allowing me to work on this project and enhance my skills and knowledge. I am proud to be a part of this prestigious institution.

**ABSTRACT**

The Password Manager Project, developed using C++, aims to provide an efficient, secure, and user-friendly solution for managing credentials associated with websites and applications. This command-line-based application enables users to store, search, update, and delete passwords through a structured menu interface. It employs key object-oriented principles such as inheritance, polymorphism, and dynamic casting to manage different types of credentials, encapsulating them in ‘*WebsitePassword*’ and ‘*AppPassword*‘ classes that extend the base ‘*Password*‘ class.

The project leverages file handling mechanisms to ensure persistent storage through CSV files, allowing the system to reload previously stored data at startup and save any changes made during execution. Shared pointers from the C++ Standard Library *(std::shared\_ptr)* are used to manage memory efficiently and ensure no data is lost when objects go out of scope. The polymorphic behaviour of the ‘*Password*‘ class allows seamless access to specific implementations such as retrieving URLs or application names without compromising code modularity.

While the current version focuses on essential CRUD operations (Create, Read, Update, Delete), the project demonstrates a foundation that can be extended to incorporate advanced features like password encryption, secure hashing, and graphical user interfaces (GUI). The design ensures flexibility, enabling future integration with cloud-based storage for remote access and automatic password generation for enhanced security. This report discusses the design, development process, results, and potential improvements to elevate the software to a production-ready tool.

**Introduction**

Password management has become a critical aspect of digital security, as users handle multiple accounts across websites and applications. Remembering unique and strong passwords can be challenging, leading to compromised credentials. This project offers a C++-based solution to store, search, update, and delete passwords efficiently.

Our Password Manager is a simple command-line tool that securely manages both website and app passwords. It ensures persistent data storage through CSV files and provides users with easy access to their credentials whenever required.

**Literature Review**

Several password management solutions exist, such as LastPass, Bitwarden, and KeePass, which offer encryption and cloud sync features. However, many of these solutions are proprietary and require paid subscriptions. This project aims to provide a lightweight, open-source alternative using C++.

Research on password management systems emphasizes the importance of encryption, secure storage, and usability. The use of object-oriented programming ensures that future improvements and additional features can be easily integrated into the system [1][2].

**Problem Statement**

Users often struggle to manage passwords for multiple online services, leading to poor password practices. A need exists for a user-friendly, local password manager that provides essential password management operations with potential for further enhancements.

**Objectives**

The primary objectives of the project are:

1. To develop a local password manager using C++.
2. To implement CRUD operations for both website and app passwords.
3. To ensure persistent data storage using CSV files.
4. To provide a foundation for future improvements, including encryption and GUI integration.

**Design and Implementation**

**5.1 Design Overview**

The system follows an object-oriented architecture, consisting of three primary classes:

* **Password:** Abstract base class with email and password attributes.
* **WebsitePassword:** Derived class to manage website-related passwords.
* **AppPassword:** Derived class to manage app-related credentials.

Polymorphism is employed to treat both website and app passwords uniformly, while dynamic casting helps retrieve specific class implementations.

**5.2 Key Functionalities**

* **Add Password:** Stores a new password along with its email and identifier (URL or app name).
* **Search Password:** Finds a stored password by matching the email and identifier.
* **Update Password:** Replaces the existing password with a new one.
* **Delete Password:** Removes a password entry from the system.

**5.3 Code Implementation**

Below is a brief overview of the code modules:

* ***Password* Class:** Provides a virtual interface for derived classes and common attributes like email and password.
* ***WebsitePassword* Class:** Extends *Password* to handle website credentials using a URL identifier.
* ***AppPassword* Class:** Handles app passwords with the app name as the identifier.
* ***PasswordManager* Class:** Manages all CRUD operations and file handling, ensuring data persistence with CSV files.

**Results and Discussion**

The password manager successfully performs all core functions. It reads and writes data to the *passwords.csv* file, ensuring passwords are available even after the program terminates. User testing confirmed that the search, update, and delete operations work as expected. However, some potential challenges include:

* Lack of password encryption, making stored data vulnerable.
* The current command-line interface, which may not be user-friendly for all users.

**Conclusion**

This project demonstrates the development of a lightweight password manager using C++. It implements essential password management features and ensures data persistence through file handling. The modular design allows easy future enhancements, such as encryption and the introduction of a GUI. Although functional, the system can be further improved to meet modern security standards.

**Future Scope**

Future developments of the password manager can focus on the following areas:

1. **Password Encryption:** To secure stored credentials using hashing algorithms.
2. **GUI Development:** To enhance usability with a graphical interface.
3. **Cloud Integration:** To allow remote access and synchronization.
4. **Automatic Password Generation:** To help users generate secure passwords effortlessly.

**References**

[1] L. F. Cranor and S. Garfinkel, *Security and Usability: Designing Secure Systems that People Can Use*. O'Reilly Media, 2005.  
[2] M. Jakobsson and R. Perrig, *The Future of Passwords: Security and Convenience*. Springer, 2014.  
[3] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.