Designing a Secure Password Management System: The SaveAPass Architecture

Author: Keirelys Marlén De Jesús Aponte
Advisor: Alfredo Cruz PhD
Electrical & Computer Engineering and Computer Science Department

Abstract

SaveAPass is a secure and user-friendly web application for managing passwords. The project's goal was to protect user credentials using strong encryption while maintaining an intuitive interface. It uses client-side AES encryption with a secret key tied to the user's password, ensuring only the user can access their data. Sensitive actions require re-authentication, and custom dialogs replace insecure browser prompts. Built with React and Firebase, SaveAPass encrypts all data before it reaches the database, safeguarding information even in case of a breach. The result is a practical, real-world password manager that demonstrates how strong security and usability can coexist.

Introduction

In today's digital landscape, individuals manage dozens of online accounts, often leading to weak, reused, or forgotten passwords. This widespread issue poses serious security risks, making secure password management tools essential. The SaveAPass project was developed to address this challenge by providing a secure, user-friendly web application for storing and managing login credentials. It aims to empower users to maintain strong, unique passwords without compromising usability or privacy. By combining strong encryption, intuitive design, and modern authentication practices, SaveAPass offers a practical solution to a common cybersecurity problem.

Background

With the rising reliance on digital services, password security has become a critical issue. Studies like Security Analysis on Password Managers Applications by Alrushaid and Algarawi [1] highlight that users often reuse passwords or store them insecurely. Password Managers: Attacks and Defenses by Stanford researchers [2] exposes vulnerabilities in autofill mechanisms, stressing the need for stronger safeguards. Meanwhile, Shinde and Deshpande's A Study for an Ideal Password Management System [3] emphasizes balancing security and usability to increase adoption among non-technical users.Informed by these findings, SaveAPass was developed as a user-friendly yet secure password manager. Built with React 18 and Firebase, it uses AES encryption via CryptoJS, Firebase Authentication, and encrypted user-specific keys. The interface, designed with Material UI, ensures accessibility across devices. SaveAPass bridges gaps identified in the literature, addressing both usability and security for a broad audience.

Problem

Many users struggle with managing strong, unique passwords, often resorting to insecure practices that lead to data breaches. This project addresses the need for a secure and user-friendly password manager by developing SaveAPass, which combines strong encryption with an intuitive interface to improve password security for non-technical users.

Methodology

SaveAPass was developed using a flexible, iterative approach focused on delivering one feature at a time. This feature-based progression allowed for continuous testing and refinement, enabling quick adjustments to ensure security and usability.

System Architecture

SaveAPass employs a frontend-centric architecture with Firebase handling backend services, emphasizing data security and user privacy.

- Frontend (React + Material UI): Responsive UI for password management, including vault, forms, authentication, and password encryption handled client-side before data transmission.
- Backend (Firebase Authentication + Firestore): Secure user authentication and session management; encrypted passwords and metadata stored with strict access control.
- Encryption Layer (AES via CryptoJS): Client-side AES encryption ensures plaintext passwords never leave the user device; stored data remains unreadable without user-specific keys.

Figure 1 visually shows the secure, modular flow between the frontend and backend, highlighting client-side encryption and protected Firestore storage.

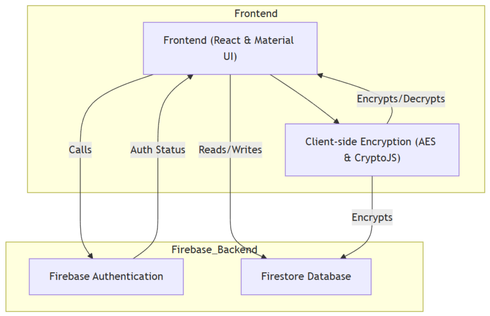

Figure 1 Frontend and Backend Interaction

During user registration, a unique secret key is generated and encrypted with the user's password before being stored. Whenever a password is added, viewed, edited, or deleted, the frontend prompts the user for their password to decrypt the secret key, which is then used for encryption or decryption of saved credentials. When the user changes their account password, the secret key is decrypted using the old password and re-encrypted with the new one, ensuring continuity and security.

Key UI/UX Principles Followed

SaveAPass features a clean, intuitive interface using Material UI for consistency and responsiveness. It emphasizes accessibility, simple navigation, and user-friendly interactions, ensuring secure and easy password management for all users.

Results and Discussion

The SaveAPass web application successfully met its objective of providing a secure and user-friendly platform for managing login credentials. Using Firebase Authentication, users can securely log in and access a personalized dashboard featuring the Password Vault. This core feature allows users to add, view, edit, or delete encrypted credentials, with all sensitive data protected by client-side AES encryption using CryptoJS.

Each user has a unique encryption key, secured with their password and stored in Firestore. Sensitive actions require password re-entry to decrypt this key, ensuring identity verification before data access. The app also supports secure password changes and account deletion, with proper re-authentication and data handling. Figure 2 illustrates the entry screen where users can sign up for a new account or log in with their credentials.

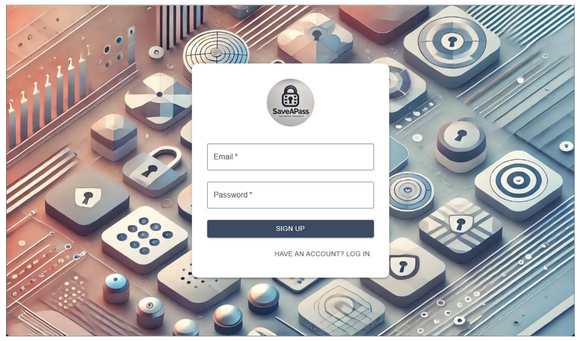

Figure 2 SaveAPass User Sign Up Screen

Figure 3 illustrates the dashboard where the password vault is displayed, showing the stored entries and available actions.

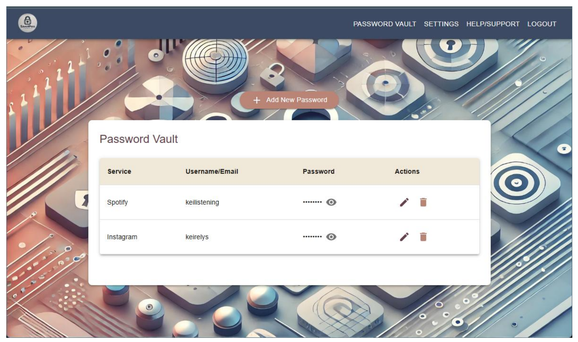

Figure 3 SaveAPass Password Vault Screen

The interface, built with React and Material UI, replaces standard browser prompts with styled dialog components for a smoother, more secure experience. All encryption operations are done on the client side, ensuring that no decrypted data is stored.

SaveAPass balances strong encryption with intuitive design, making secure password management accessible. It also lays the groundwork for future features like 2FA and backups, showcasing best practices in secure software development.

Conclusions

SaveAPass provides a secure and intuitive solution for managing login credentials, addressing the growing need for personal cybersecurity tools. By combining AES encryption, user-specific secret keys, and Firebase Authentication, the app ensures that sensitive data remains private and accessible only to verified users. Its use of React and Material UI offers a clean, responsive interface that simplifies complex security tasks, encouraging stronger digital habits without overwhelming users. The project demonstrates how effective security and user-friendly design can coexist, proving that privacy-first development is both achievable and essential. Features like password-based encryption, re-authentication for sensitive actions, reinforce its strong security model. SaveAPass not only fulfills current password management needs but also lays the groundwork for future enhancements such as two-factor authentication and cloud backups. Overall, SaveAPass reflects best practices in secure software development and highlights the importance of empowering users to take control of their digital safety in a connected world.

Future Work

Future enhancements for SaveAPass aim to strengthen security and improve usability. Key additions include Two-Factor Authentication (2FA) for added login protection, user activity logging to monitor and detect suspicious behavior, and encrypted data backup for recovery in case of device loss. Other improvements include a password strength checker to promote strong credentials and user-centered features like password grouping, sharing, and expiration reminders. These upgrades will expand SaveAPass into a more robust and versatile password management solution.

Acknowledgements

I would like to express my sincere gratitude to Professor Alfredo Cruz, Ph.D., for his invaluable guidance and support throughout the development of this project. His mentorship played a crucial role in shaping the direction and quality of this work.

References

[1] A. Alrushaid and R. Algarawi, "Security Analysis on Password Managers Applications," Research Publish, 2020. [Online]. Available: https://www.researchpublish.com/upload/book/paperpdf-1600434399.pdf. [Accessed: Dec. 11, 2024].

[2] D. Silver, S. Jana, E. Chen, C. Jackson, and D. Boneh, "Password Managers: Attacks and Defenses," Stanford University, 2014. [Online]. Available: https://crypto.stanford.edu/~dabo/pubs/papers/pwdmgrBrowser.pdf. [Accessed: Dec. 20, 2024].

[3] S. K. Shinde and M. V. Deshpande, "A Study for an Ideal Password Management System," International Journal of Research in Applied Science & Engineering Technology, vol. 8, no. 4, pp. 123–130, Apr. 2020. [Online]. Available: https://doi.org/10.22214/ijraset.2022.39970. [Accessed: Jan. 18, 2025].