# Smart Password Manager and Encryptor: Secure, Encrypted and Simple Password Management

Dr. M. K. Jayanthi Kannan[1], Swarali Shailesh Patankar[2], Tejal Sharma[3], Anvesha Agarwal[4], Akshat Mujmer[5]

[1]*Professor, School of Computing Science and Engineering, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh*

[2,3,4,5]*PG Students, School of Computing Science and Engineering, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh*

***Abstract*—The Smart Password Manager is a Python-based application designed to help users create, store, manage, and retrieve passwords efficiently. The system addresses common password-security problems such as weak passwords, password reuse, and unsafe credential storage. It uses Fernet symmetric encryption to protect stored password data and maintains encrypted credentials in a JSON-based local data store. The application provides secure master-password login, password generation, password-strength analysis, password search, credential update and deletion, and activity logging. It also includes login-attempt handling and security feedback to improve protection against unauthorized access. By combining password management, encryption, password generation, and strength analysis in a single desktop application, the prototype aims to provide a simple, user-friendly, and secure solution for everyday digital credential management.***

***Index Terms*—Password Manager, Fernet Symmetric Encryption, Encrypted JSON Database, Master Password, Password Generator, Password Strength Analysis, Authentication, Credential Vault, Activity Log, Python Tkinter.**

## I. INTRODUCTION

The increasing number of online accounts has made password management an important part of everyday digital security. Users are expected to maintain strong and unique credentials for different services, yet remembering many passwords can be difficult. This can lead to weak passwords, password reuse, or storage of credentials in insecure locations. A practical password-management system therefore needs to combine security mechanisms with convenient management features. The proposed Smart Password Manager is a secure Python-based desktop application that combines a graphical interface with encrypted local credential storage. Python and Tkinter are used to implement the application interface, while Fernet symmetric encryption is used to protect stored password data. Encrypted JSON storage provides a lightweight local data store for the prototype. The system provides master-password authentication, password generation, password-strength analysis, credential search, update and deletion, password retrieval, and activity logging. These features are integrated into a single application so that users can manage credentials without relying on separate tools for password creation, storage, and evaluation.

## II. PROBLEM STATEMENT AND OBJECTIVES

With the increasing number of online accounts, users need to manage multiple passwords for different applications and websites. Remembering many strong and unique passwords can be difficult, which often leads users to choose weak passwords, reuse the same password across multiple accounts, or store passwords in insecure locations. These practices increase the risk of unauthorized access and credential theft. Traditional methods may also fail to provide both secure storage and convenient management.

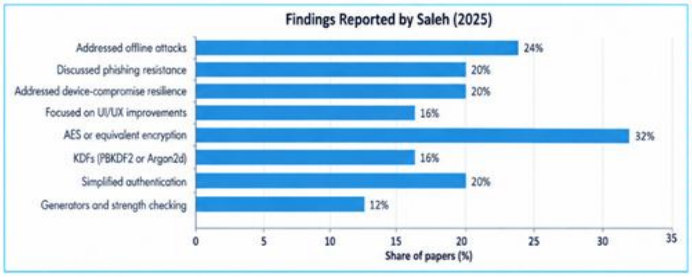
The Smart Password Manager addresses this need by providing encrypted password storage using Fernet encryption, secure master-password authentication, password generation, strength analysis, search, update and deletion features, and activity monitoring. Develop a secure password-management system for organized credential storage and management. Protect

stored passwords using Fernet symmetric encryption so that credentials are not maintained directly in readable form. Provide master-password authentication and login-attempt handling to reduce unauthorized access. Generate strong and unique passwords automatically. Analyze password strength and provide feedback to support stronger credentials. Provide add, retrieve, search, update, and delete operations for stored passwords. Maintain an activity log of important security and user actions. Provide a simple and user-friendly interface while maintaining the intended security features. Reduce risks associated with weak passwords, password reuse, and insecure credential storage.

## III. LITERATURE REVIEW OF EXISTING SYSTEMS

Saleh (2025) critically reviewed 25 password-manager studies from security, usability, and innovation perspectives. The reviewed systems considered threats including offline attacks, phishing, and device compromise, and investigated approaches such as encryption, key-derivation functions, OPRF, DE-PAKE, secret sharing, physically unclonable functions, biometrics, and distributed architectures. The review also considered user-interface improvements, simplified authentication, password generators, and password-strength mechanisms. The review used a critical and thematic analysis of 25 pre-selected research papers. The studies were grouped into four themes: security challenges; usability and adoption; innovative architectural designs; and cryptographic techniques and protocols. Findings were compared to identify common trends, differences, trade-offs, and research gaps. The review reported that 6 of the 25 papers (24%) addressed offline attacks, 5 (20%) discussed phishing resistance, 5 (20%) addressed device-compromise resilience, and 4 (16%) focused on UI/UX improvements. It also reported that 32% explicitly mentioned AES or equivalent encryption and 16% discussed KDFs such as PBKDF2 or Argon2d. The review highlighted continuing challenges involving usability, distributed-system scalability, synchronization, cryptographic development, and standardization/interoperability. These findings guided the design of the present prototype. The project combines encrypted credential storage, secure authentication, password generation, strength analysis, password search, update/delete operations, and activity logging in one user-oriented desktop application.

| Paper 1 | Objective used | Technology used | Methodology used | Efficiency | Issues |
|---|---|---|---|---|---|
| **Name:** Swarali Shailesh Patankar<br>**Registration No.:** 25MIP10026<br>**Title:** Password Managers: A Critical Review of Security, Usability, and Innovative Designs<br>**Author:** Hussein Abdulkhaleq Saleh<br>**Journal:** Journal of Computer Sciences Institute<br>**Volume:** 36<br>**Year:** 2025<br>**Pages:** 328–335<br>**DOI:** 10.35784/jcsi.7725<br>The paper synthesizes findings from 25 research papers and examines password managers from Security, Usability, and Innovation perspectives. | • Evaluates password managers from security, usability, and innovation perspectives.<br>• Examines offline attacks, phishing, and device compromise.<br>• Studies interface and authentication improvements.<br>• Examines innovative architectures and cryptographic techniques.<br>• Identifies research gaps and future directions.<br>• Explores the balance between security and usability. | • AES/AES-equivalent encryption and AES-256.<br>• PBKDF2 and Argon2d for key derivation.<br>• OPRF and DE-PAKE.<br>• Shamir's Secret Sharing.<br>• PUFs and USB keys.<br>• RSA-based hardware protection.<br>• Biometrics and privacy-preserving biometric templates.<br>• Cloud, local, distributed and multi-device storage.<br>• Password generators and strength checkers.<br>• UPL/domain verification.<br>• APIs for website integration.<br>• Blowfish + LSB steganography.<br>• Decentralized file formats. | • Critical/thematic review of 25 pre-selected papers.<br>• No external database search or PRISMA-based systematic search.<br>• Information was systematically extracted from each paper, including the problem, proposed solution, methodology, findings, and security/usability implications.<br>• Studies were grouped into four themes: Security Challenges, Usability and Adoption, Innovative Architectural Designs, and Cryptographic Techniques and Protocols.<br>• Findings were compared to identify common trends, differences, trade-offs, and research gaps. | • 6/25 papers (24%) addressed offline attacks.<br>• 5/25 (20%) discussed phishing resistance.<br>• 5/25 (20%) addressed device-compromise resilience.<br>• 4/25 (16%) focused on UI/UX improvements.<br>• 8/25 (32%) explicitly mentioned AES/equivalent encryption.<br>• 4/25 (16%) discussed KDFs such as PBKDF2/Argon2d.<br>• Generative systems reduce vault-compromise risk; distributed and hardware approaches improve resilience. | • Offline brute-force attacks against stolen vaults.<br>• Phishing and fraudulent websites.<br>• Device theft/malware compromise.<br>• Master-password dependency in traditional managers.<br>• Cognitive load and additional information requirements.<br>• User trust and adoption barriers.<br>• Secure synchronization across devices.<br>• Distributed-system complexity and network dependency.<br>• Hardware solutions can reduce portability and convenience.<br>• Limited real-world usability testing.<br>• Distributed-system scalability underexplored.<br>• Need for stronger cryptographic protocols and standardization/interoperability. |




| Finding | Papers | Share |
|---|---|---|
| Addressed offline attacks | 6 of 25 | 24% |
| Discussed phishing resistance | 5 of 25 | 20% |
| Addressed device-compromise resilience | 5 of 25 | 20% |
| Focused on UI/UX improvements | 4 of 25 | 16% |
| Explicitly mentioned AES or equivalent encryption | 8 of 25 | 32% |
| Discussed KDFs such as PBKDF2 or Argon2d | 4 of 25 | 16% |
| Explored simplified authentication | Not specified | 20% |
| Encouraged secure behaviour through generators and strength checking | Not specified | 12% |

Fig. 1. Literature Review of Smart Password Manager and Encryptor.

Saleh (2025) critically reviewed 25 password-manager studies focusing on security, usability, and innovative architectures. The reviewed systems addressed threats such as offline attacks, phishing, and device

compromise using encryption, KDFs, OPRF, DE-PAKE, secret sharing, PUFs, biometrics, and distributed architectures. The study found that password managers must balance strong security with usability and identified research gaps in usability testing, scalability, cryptographic development, and standardization. These findings guided our project in implementing encrypted credential storage, secure authentication, password generation, strength analysis, and user-friendly password-management features. The findings of this literature review directly support the design of our Smart Password Manager. Based on the security and usability challenges identified in the reviewed studies, our project combines encrypted password storage, secure authentication, password generation, password-strength analysis, password search, update/delete operations, and activity logging in a single user-friendly application. Thus, the project applies important concepts identified in existing password-manager research while providing a practical implementation suitable for everyday credential management.

Saleh's 2025 study presents a critical review of 25 research papers related to password managers, focusing on security, usability, and innovative designs. The review examines how different password-manager systems address security threats such as offline attacks, phishing, and device compromise, while also considering usability and user adoption. The reviewed studies use a variety of technologies and approaches, including AES and equivalent encryption techniques, PBKDF2 and Argon2d, OPRF, DE-PAKE, Shamir's Secret Sharing, Physically Unclonable Functions (PUFs), biometrics, cloud-based and distributed architectures, and password-generation mechanisms. Some systems focus on avoiding password storage completely through generative approaches, while others distribute sensitive information across multiple devices or combine software with hardware-based security.

The methodology of the paper is based on a critical and thematic analysis of 25 pre-selected research papers. The studies are examined according to four major areas: security challenges, usability and adoption, innovative architectural designs, and cryptographic techniques and protocols. The review identifies the strengths and limitations of different approaches rather than proposing a single password-manager implementation.

| Paper 1 | Objective | Technology Used | Methodology Use | Efficiency | Issues |
|---|---|---|---|---|---|
| **Tejal Sharma**<br>**Reg. No.: 25MIP10009**<br>**Title:**<br>SECURITY ANALYSIS ON PASSWORD MANAGERS APPLICATIONS<br>**Journal:**<br>International Journal of Computer Science and Information Technology Research<br>**Vol. 8, Issue 3, pp. 264-267**<br>**Year: 2020**<br>**Authors: Abdulaziz Alrushaid, Reem Algarawi** | 1. Analyze password manager applications from a security perspective.<br>2. Examine how different password managers store confidential data.<br>3. Study Master Key derivation and user authentication security.<br>4. Compare web browser and native application-based password managers.<br>5. Present security assessment results and recommendations. | **Password Manager Categories:**<br>• Web Browser Password Managers<br>• Native Application-based Password Managers<br>**Technologies / Algorithms:**<br>• SQLite database<br>• Windows DPAPI<br>• Triple-DES (3DES)<br>• SHA-1<br>• SHA-256<br>• AES-256<br>• PBKDF2<br>• HMAC-SHA-256<br>• CBC mode<br>• SSL<br>• HKDF<br>• PAKE / SRP | Security Analysis: The paper examines popular password managers and their security mechanisms.<br>Web Browsers: Chrome, Firefox and Internet Explorer are analyzed for credential storage and encryption.<br>Native Managers: LastPass, KeePass and 1Password are examined.<br>Master Key Derivation: The study analyzes how master passwords, salts, hashing and key derivation functions protect stored credentials.<br>Authentication: User authentication mechanisms and protection of encryption keys are examined.<br>Encryption: The study compares encryption techniques such as AES-256, 3DES and CBC-based protection. | 1. Password managers reduce password fatigue, mental load and login time.<br>2. They can facilitate better password quality and reduce password reuse.<br>3. Native application-based managers provide strong security mechanisms.<br>4. KeePass stores and protects the complete database locally using AES-256 and integrity protection.<br>5. LastPass performs MasterKey derivation locally using PBKDF2 with SHA-256.<br>6. 1Password uses two-secret key derivation and authenticated authentication mechanisms. | 1. Browser-based Security: The paper concludes that most browser-based password managers are not secure enough against many attacks.<br>2. Password Dependency: Security can depend on the strength of the user's master or system password.<br>3. Cloud Storage Risk: Synchronization with servers can introduce privacy and unauthorized-access concerns.<br>4. Key Management: Weak protection of master keys can affect overall security.<br>5. Security Variation: Security levels vary between password managers depending on their features and implementation. |

Fig. 2. Literature Review of Smart Password Manager and Encryptor.

The review highlights that password managers can significantly improve credential security, but there is a continuing trade-off between security and usability. Generative approaches can reduce the risks associated with storing passwords, while distributed and hardware-based approaches can provide additional protection against device compromise. However, these approaches may introduce complexity,

synchronization problems, or reduced convenience. The study also emphasizes the importance of user-friendly interfaces, simplified authentication, password generators, and password-strength mechanisms in encouraging secure user behavior. The paper identifies several important research gaps, including real-world usability testing, scalability of distributed password-management systems, stronger and evolving cryptographic techniques, and standardization/interoperability. It suggests that future password managers may benefit from hybrid approaches that combine the strengths of generative, distributed, biometric, and hardware-assisted techniques.

| Paper 1 | Objective | Technology Used | Methodology Use | Efficiency | Issues |
|---|---|---|---|---|---|
| **Anvesha Agrawal**<br>**25MIP10003**<br>**Title:**<br>DESIGN AND DEVELOPMENT OF A SECURE PASSWORD MANAGER USING END-TO-END ENCRYPTION.<br>**Journal:**<br>International Journal of Innovative Research in Technology (IJIRT)<br>**Year: 2026**<br>**Volume: 12, Issue 8**<br>**Paper ID: IJIRT 190156**<br>**Pages: 2432-2435** | 1. Develop a secure password manager using end-to-end encryption.<br>2. Protect users' credentials from unauthorized access.<br>3. Encrypt sensitive information on the client side before storage.<br>4. Reduce risks caused by password reuse and storing passwords in plaintext.<br>5. Provide a simple interface to add, view, and delete credentials securely. | **Technologies:**<br>• AES-256 encryption<br>• End-to-end encryption<br>• Client-side encryption<br>• HTML<br>• CSS<br>• JavaScript<br>• Web technologies<br>• Master-password-based key generation<br>**Security:**<br>• Local credential storage<br>• Authentication<br>• Encrypted backup storage | End-to-End Encryption: Sensitive credentials are encrypted on the client side before storage.<br>AES-256: The master password is used to derive a cryptographic key for encryption and decryption.<br>Encryption Process: Plaintext credential data D is converted into encrypted data C using C = E(D,K).<br>Decryption Process: Encrypted data is converted back using the correct key K.<br>Local Storage: Encrypted credentials are stored locally rather than on a server.<br>User Interface: A responsive interface allows users to securely add, view and delete credentials.<br>Authentication: Access to passwords and system settings is protected. | 1. Strong Security: AES-256 protects stored credentials.<br>2. Privacy: Client-side encryption reduces exposure of sensitive data.<br>3. No Readable Password Storage: Passwords are not persisted in readable format.<br>4. Reduced Server Risk: The system avoids server-side credential storage.<br>5. Usability: The interface is simple, responsive and mobile-friendly.<br>6. Performance: The paper reports that AES-256 provides strong security while keeping performance efficient. | 1. Master Password Dependency: The correct master password is required to decrypt credentials.<br>2. User Responsibility: The master password should be strong and must not be stored or shared.<br>3. Device Risk: Users should avoid accessing the password manager on public or shared devices.<br>4. Backup Security: Additional backups must use encrypted storage.<br>5. Input Security: Input validation is required to prevent incorrect or harmful data entry.<br>6. Future Extensions: Biometric authentication, encrypted cloud synchronization and multi-factor security are suggested for further development. |

Fig. 3. Literature Review of Smart Password Manager and Encryptor.

| Paper 1 | Objective | Technology Used | Methodology Use | Efficiency | Issues |
|---|---|---|---|---|---|
| **Tejal Sharma**<br>**Reg. No.: 25MIP10009**<br>**Title:**<br>SECURITY ANALYSIS ON PASSWORD MANAGERS APPLICATIONS<br>**Journal:**<br>International Journal of Computer Science and Information Technology Research<br>**Vol. 8, Issue 3, pp. 264-267**<br>**Year: 2020**<br>**Authors: Abdulaziz Alrushaid, Reem Algarawi** | 1. Analyze password manager applications from a security perspective.<br>2. Examine how different password managers store confidential data.<br>3. Study Master Key derivation and user authentication security.<br>4. Compare web browser and native application-based password managers.<br>5. Present security assessment results and recommendations. | **Password Manager Categories:**<br>• Web Browser Password Managers<br>• Native Application-based Password Managers<br>**Technologies / Algorithms:**<br>• SQLite database<br>• Windows DPAPI<br>• Triple-DES (3DES)<br>• SHA-1<br>• SHA-256<br>• AES-256<br>• PBKDF2<br>• HMAC-SHA-256<br>• CBC mode<br>• SSL<br>• HKDF<br>• PAKE / SRP | Security Analysis: The paper examines popular password managers and their security mechanisms.<br>Web Browsers: Chrome, Firefox and Internet Explorer are analyzed for credential storage and encryption.<br>Native Managers: LastPass, KeePass and 1Password are examined.<br>Master Key Derivation: The study analyzes how master passwords, salts, hashing and key derivation functions protect stored credentials.<br>Authentication: User authentication mechanisms and protection of encryption keys are examined.<br>Encryption: The study compares encryption techniques such as AES-256, 3DES and CBC-based protection. | 1. Password managers reduce password fatigue, mental load and login time.<br>2. They can facilitate better password quality and reduce password reuse.<br>3. Native application-based managers provide strong security mechanisms.<br>4. KeePass stores and protects the complete database locally using AES-256 and integrity protection.<br>5. LastPass performs MasterKey derivation locally using PBKDF2 with SHA-256.<br>6. 1Password uses two-secret key derivation and authenticated authentication mechanisms. | 1. Browser-based Security: The paper concludes that most browser-based password managers are not secure enough against many attacks.<br>2. Password Dependency: Security can depend on the strength of the user's master or system password.<br>3. Cloud Storage Risk: Synchronization with servers can introduce privacy and unauthorized-access concerns.<br>4. Key Management: Weak protection of master keys can affect overall security.<br>5. Security Variation: Security levels vary between password managers depending on their features and implementation. |

Fig. 4. Literature Review of Smart Password Manager and Encryptor.

## IV. RESEARCH GAP

The reviewed literature shows a continuing trade-off between security and usability. Strong protection may increase complexity, while convenient systems can introduce additional attack surfaces. Real-world usability testing remains limited in the reviewed studies.

Distributed and multi-device architectures require further work on scalability and interoperability. Cryptographic standardization and stronger future-ready mechanisms remain open areas.

## V. PROPOSED SYSTEM DESIGN

The proposed system is organized into three principal areas: authentication, secure credential management, and security utilities. Authentication controls access through a master-password login, login confirmation, show/hide password support, and logout confirmation. The secure vault provides encrypted credential storage together with add, retrieve, update, delete, and search operations. Security utilities provide password generation, password-strength analysis, activity logging, and secure local storage.

| | | | | |
|---|---|---|---|---|
| User | Tkinter GUI | Authentication | Password Modules | Fernet Encryption |

Fig. 5. Simplified system architecture: User Encrypted local storage.

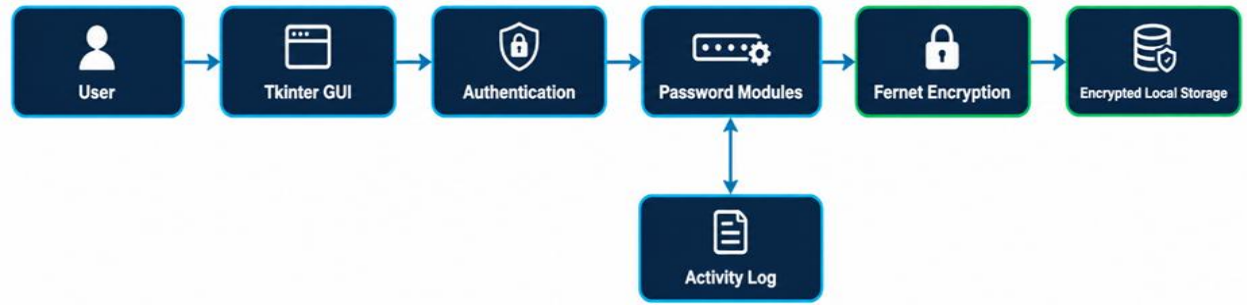



Fig. 6. Simplified System Architecture of Encrypted

## VII. PROJECT FUNCTIONAL MODULES AND IMPLEMENTATION

Login and Authentication. Provides master-password access control, show/hide password support, and secure login behavior. Dashboard. Provides centralized navigation, quick actions, and security information. Add Password. Accepts account name, username/email, and password information for storage. Password Vault. Allows account selection and password retrieval.

Password Generator. Provides selectable password length and generated credentials. Strength Analyzer. Provides a score and strength classification for an entered password. Activity Log. Maintains timestamped records of important vault operations. Update, Delete, and Search. Supports credential maintenance and account searching. Prototype credential Management: Add Password, Account Name, Username/ Email, Password Save Password, Success confirmation.

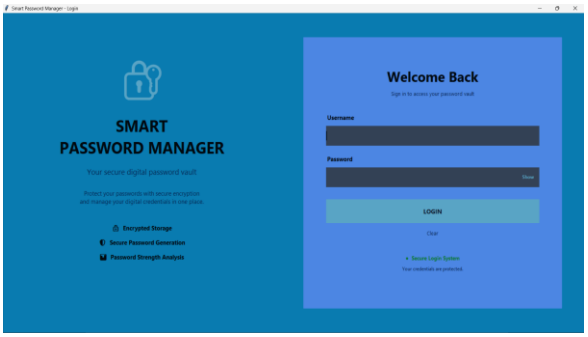



Fig.7: Login page of Password Manager and Encryptor

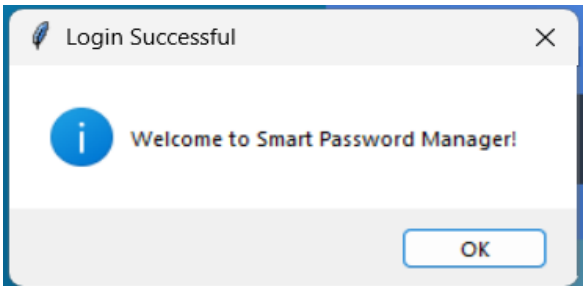



Fig.8: Welcome page

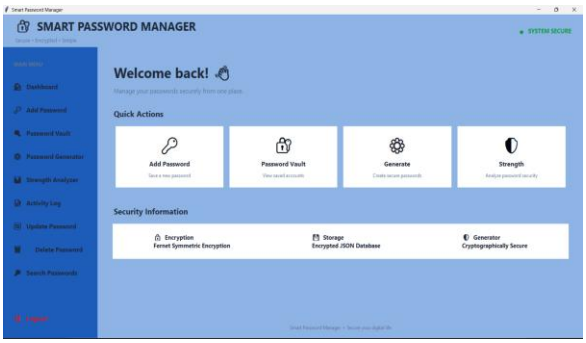



Fig.9: Dashboard of Smart Password Manager and Encryptor




Fig.10: Add password

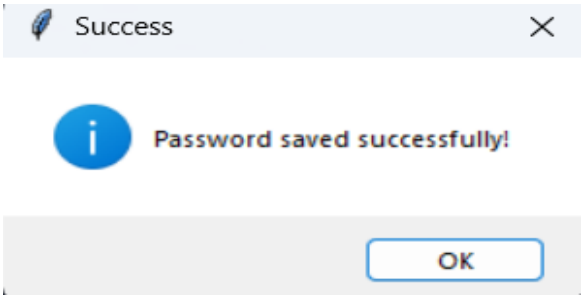



Fig.11: Window showing password is added successfully

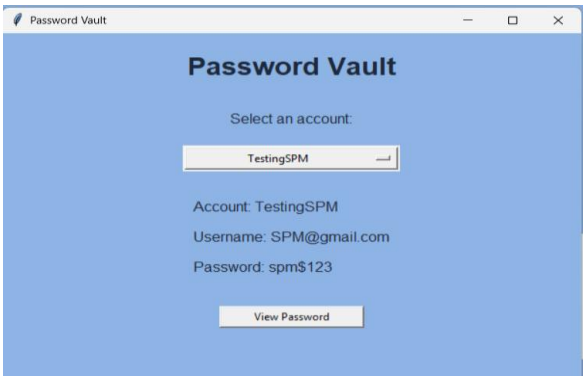



Fig.12: Password Vault showing all the credentials for a selected account

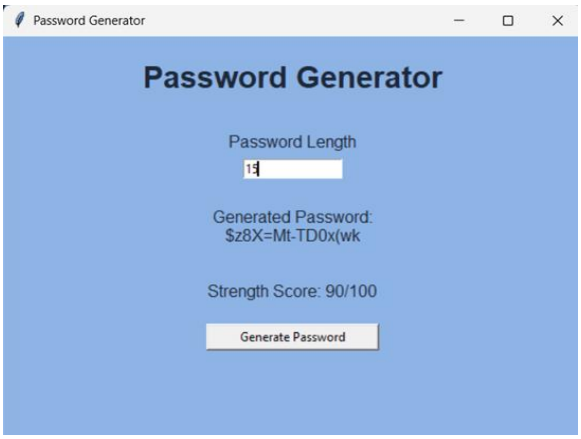






Fig.13 Password generator based on length and strength score

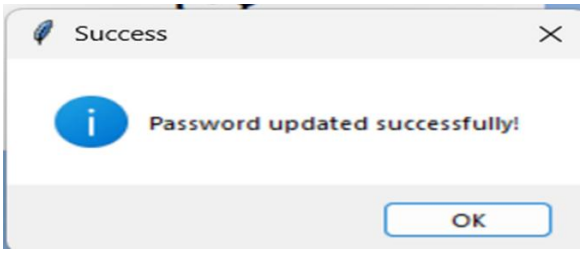



Fig.14: Updating an existing password

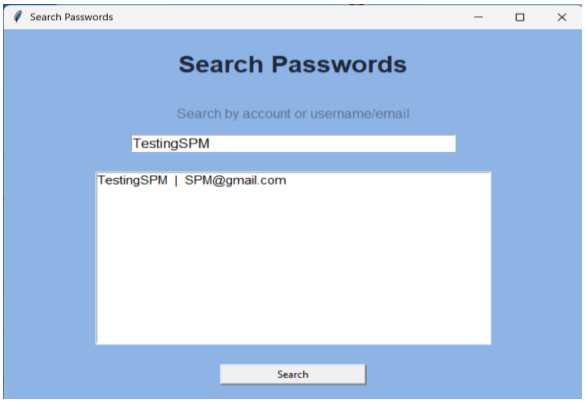
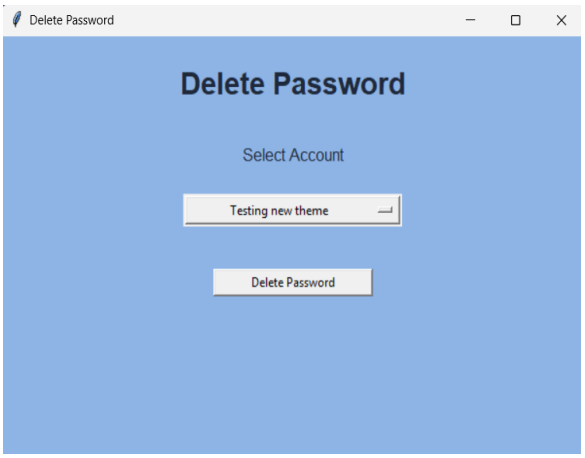



Fig.15. Searching for a password



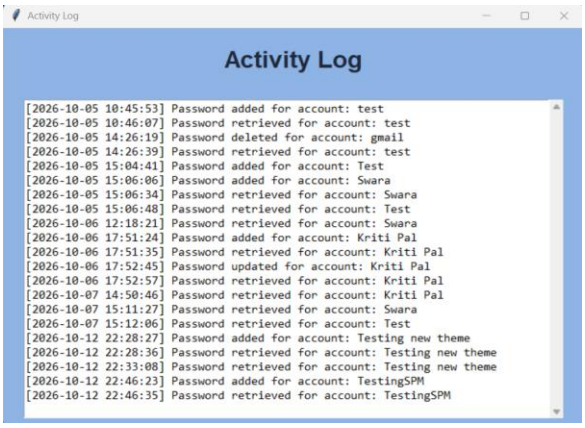




Fig.19: Activity log with details Date and Time

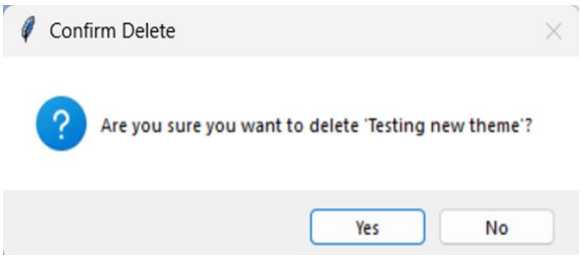



Fig.16: Confirm password

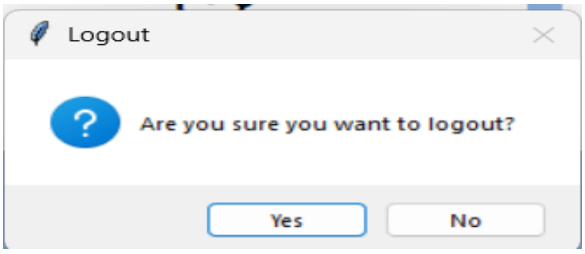



Fig.20: Logout window

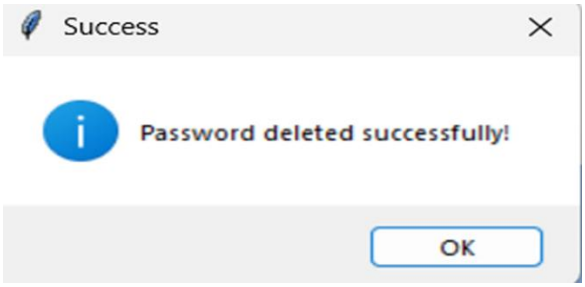



Fig.17: Password deleted successfully

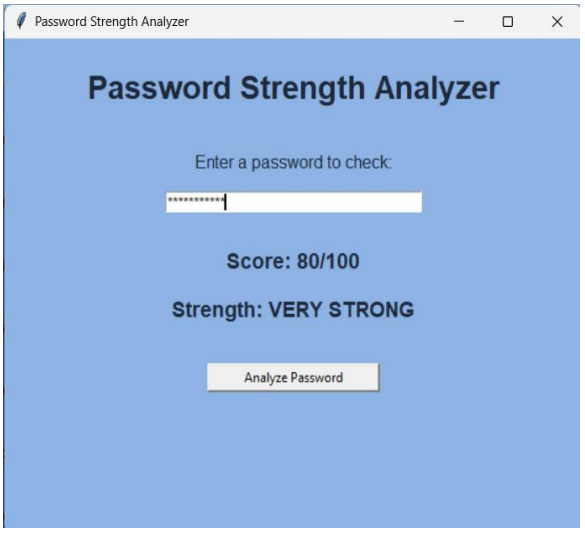



Fig.18: strength analyzer

## VIII. METHODOLOGY FOR DEVELOPING THE APPLICATION FOR SMART PASSWORD MANAGER AND ENCRYPTOR

Requirements. The required capabilities were identified as secure storage, authentication, password generation, password analysis, and credential management. Design. GUI navigation, modules, data flow, and encryption/storage components were planned. Development. The desktop interface and password-management functions were implemented using Python and Tkinter. Integration. Authentication, encryption, JSON storage, password modules, and activity logging were connected into the application workflow. Testing. Login, CRUD operations, password generation, strength analysis, navigation, and logout behavior were tested. Maintenance. The application can be refined through issue fixing, usability improvements, and additional security features.

## IX. ALGORITHM

1. User Authentication Algorithm

The authentication algorithm verifies the user's login credentials before allowing access to the dashboard.

Steps:
1. Accept username and master password from the user.
2. Compare the entered credentials with the stored authentication information.
3. If the credentials are correct, allow access to the dashboard.
4. If the credentials are incorrect, display a login-failure message.
5. Track failed login attempts and apply the defined security handling.
6. Allow the user to retry the login process.

2. Fernet Encryption Algorithm
The system uses Fernet symmetric encryption to protect stored password information.
Steps:
1. Obtain the password entered by the user.
2. Convert the password into an appropriate encoded format.
3. Apply the Fernet encryption mechanism using the encryption key.
4. Store the resulting encrypted value instead of the original password.
5. When the password is requested, decrypt the stored value using the corresponding key.
6. Display the original password only to an authenticated user.

Flow:
Plain Password → Fernet Encryption → Encrypted Password → JSON Storage
and during retrieval: Encrypted Password → Fernet Decryption → Original Password

3. Password Generation Algorithm
The password generator creates strong passwords automatically.
Steps:
1. Select the required password length.
2. Select character categories such as uppercase letters, lowercase letters, numbers, and special characters.
3. Randomly select characters from the selected categories.
4. Combine the selected characters to create the password.
5. Display the generated password to the user.

This reduces dependence on manually created passwords and helps users create stronger credentials.

4. Password Strength Analysis Algorithm
The strength analyzer evaluates a password according to different characteristics.
Steps:
1. Accept a password from the user.
2. Check its length.
3. Check for uppercase letters.
4. Check for lowercase letters.
5. Check for numbers.
6. Check for special characters.
7. Calculate a strength score based on the implemented criteria.
8. Display the corresponding strength level and feedback.

Flow:
Password Input → Character Checks → Strength Score → Strength Level → Feedback

5. Password Management Algorithm
The system provides CRUD-based password management operations.
Steps:
1. Authenticate the user.
2. Select the required operation from the dashboard.
3. For Add, accept account details and encrypt the password before storage.
4. For Retrieve/Search, locate the required encrypted credential and decrypt it for the authenticated user.
5. For Update, modify the selected credential and securely store the updated password.
6. For Delete, identify the selected credential and remove it from the stored data.
7. Record the important operation in the activity log.

6. Activity Logging Algorithm
The activity-log mechanism records important actions performed by the user.
Steps:
1. Detect an important system or user action.
2. Record the action and relevant event information.
3. Store the activity information in the log.
4. Display the recorded activities through the Activity Log module when required.

## X. SYSTEM WORKFLOW

The workflow of the Smart Password Manager begins when the user launches the application and enters the

master login credentials. The system validates the credentials before providing access to the main dashboard. After successful authentication, the user can select the required module according to the task to be performed. The user can add, retrieve, search, update, or delete stored credentials, generate new passwords, analyze password strength, or view the activity log.

During password-management operations, sensitive password information is processed using Fernet symmetric encryption and stored in the encrypted JSON-based local storage. When an authorized user requests a stored password, the encrypted information is decrypted for retrieval. Important operations are recorded in the activity log to provide visibility into password-management activities. After completing an operation, the user can return to the dashboard and continue using another feature or choose to log out. The workflow therefore provides a structured process from authentication to password management, secure data processing, activity logging, and logout.

Launch Application → Master Login → Dashboard → Select Module → Process Request → Encrypt/Decrypt → Read/Write Storage → Log Activity

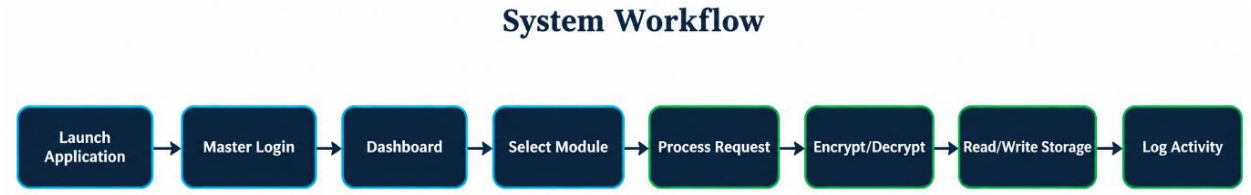



Fig.13: system workflow

## XI. CONTRIBUTION AND FINDINGS

The project combines password storage, generation, strength analysis, and credential management in one GUI. Fernet-based encryption protects stored password data in the implemented prototype. The dashboard provides centralized access to security and management functions.

Password generation and strength analysis encourage stronger password practices. Activity logging improves visibility into important vault operations. The design balances security features with a simple desktop interface.

## XII. ADVANTAGES AND LIMITATIONS

Advantages. Centralized password management; encrypted local credential storage; password generation and strength analysis; CRUD and search functionality; activity logging; and a simple desktop GUI. Limitations.

The prototype is primarily local rather than cloud-synchronized. Master-password security remains important. Multi-device synchronization, biometric or hardware integration, and wider real-world usability testing were not demonstrated in the implemented prototype.

## XIII. FUTURE SCOPE

Optional biometric authentication and hardware-backed security. Secure cloud synchronization with carefully designed encryption and key management. Multi-device support and interoperability. Stronger password-generation policies and configurable security rules. Expanded audit/activity monitoring and recovery mechanisms. Further usability testing with real users. Exploration of newer cryptographic and privacy-preserving techniques.

## XIV. CONCLUSION

The Smart Password Manager provides an integrated approach to secure password storage and management. The implemented prototype combines master-password authentication, Fernet encryption, encrypted JSON storage, password generation, password-strength analysis, credential management, and activity logging. The system is designed to address both security and usability requirements for a local password-management application. The literature review indicates that password managers must continue balancing security, convenience, scalability, and interoperability. Future enhancements can extend the prototype toward stronger authentication, secure

synchronization, broader interoperability, and additional privacy-preserving mechanisms.

## REFERENCES


[1] H. A. Saleh, "Password Managers: A Critical Review of Security, Usability, and Innovative Designs," *Journal of Computer Sciences Institute*, vol. 36, pp. 328–335, 2025, doi: 10.35784/jcsi.7725.

[2] A. Alrusaid and R. Algarawi, "Security Analysis on Password Managers Applications," International Journal of Computer Science and Information Technology Research, vol. 8, no. 3, pp. 264–267, 2020.

[3] G. D. K. Singh and M. K. Jayanthi, Eds., Network Security Attacks and Countermeasures. IGI Global, 2016, doi: 10.4018/978-1-4666-8761-5.

[4] M. K. J. Kannan, M. K. Aaijaz, K. G. Mani, and V. Tewari, The Future of Innovation and Technology in Education: Trends and Opportunities. S&M Publications, 2025.

[5] A. Agrawal, "Design and Development of a Secure Password Manager Using End-to-End Encryption," International Journal of Innovative Research in Technology (IJIRT), vol. 12, no. 8, Paper ID IJIRT190156, pp. 2432–2435, 2026.

[6] A. Mujmer, "A Secure Password Manager," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 4, no. 3, Paper ID JETIR1703043, 2017.

[7] B. R. M. and J. K. M. K., "Intrusion Detection on AWS Cloud through Hybrid Deep Learning Algorithm," Electronics, vol. 12, Art. no. 1423, 2023, doi: 10.3390/electronics12061423.

[8] S. K. Shukla, U. Dwivedi, M. K. J. Kannan, and C. Sarvani, Python for Data Analytics: Practical Techniques and Applications. JSR Publications, 2024.

[9] M. K. J. Kannan, S. Kumar, P. T. Kalaivaani, and G. Tripathi, Artificial Intelligence and Blockchain Technology for Human Resource Management. Scientific International Publishing House, 2025.

[10] M. K. Jayanthi, "Strategic Planning for Information Security—DID Mechanism to Befriend the Cyber Criminals to Assure Cyber Freedom," in Proc. 2nd Int. Conf. Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 142–147, doi: 10.1109/Anti-Cybercrime.2017.7905280.

[11] P. Jain, I. Rajvaidya, K. K. Sah, and J. Kannan, "Machine Learning Techniques for Malware Detection—A Research Review," in Proc. IEEE Int. Students' Conf. Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2022, pp. 1–6, doi: 10.1109/SCEECS54111.2022.9740918.

[12] B. R. M., M. M. V., and J. K. M. K., "Performance Analysis of Bag of Password Authentication Using Python, Java and PHP Implementation," in Proc. 6th Int. Conf. Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 1032–1039, doi: 10.1109/ICCES51350.2021.9489233.

[13] M. K. J. Kannan and T. R. Shree Nee, "Qubits Unveiled: A Deep Dive into Quantum Computing and Its Revolutionary Potential for Supply Logistics," in Qubits Unveiled: Quantum Computing Solutions for Efficient Supply Logistics, P. Gaba, A. Panwar, V. Jain, and R. Kannan, Eds. Nova Science Publishers, 2025, pp. 273–293, doi: 10.52305/WSXW8884.

[14] Python, Tkinter, and Cryptography/Fernet documentation used during project implementation.

[15] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The Tangled Web of Password Reuse," in Proc. 21st Annu. Network and Distributed System Security (NDSS) Symp., 2014, pp. 23–26, doi: 10.14722/ndss.2014.23357.

[16] T. R. Shree Nee, M. K. J. Kannan, and K. Mariyappan, "Digital Health and Medical Tourism Innovations for Digitally Enabled Care for Future Medicine: The Real-Time Project's Success Stories," in Navigating Innovations and Challenges in Travel Medicine and Digital Health. IGI Global Scientific Publishing, 2025, pp. 325–344, doi: 10.4018/979-8-3693-8774-0.ch016.

[17] H. Naik and M. K. J. Kannan, "A Survey on Protecting Confidential Data Over Distributed Storage in Cloud," SSRN, Dec. 1, 2020. [Online]. Available: https://ssrn.com/abstract=3740465.

[18] T. Safaie, ByPass: Reconsidering the Usability of Password Managers. M.A.Sc. thesis, Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada, 2021.

[19] F. Alodhyani, G. Theodorakopoulos, and P. Reinecke, "Password Managers—It's All About Trust and Transparency," Future Internet, vol. 12, no. 11, Art. no. 189, 2020, doi: 10.3390/fi12110189.

[20] H. A. Saleh, "Leveraging Social Engineering Techniques for Ethical Purposes: An Approach to Develop Fake Android App for Collecting Valuable Data Discreetly," Wasit Journal of Computer and Mathematics Science, vol. 3, no. 3, pp. 15–59, 2024, doi: 10.31185/wjcms.268.

[21] C. Rahalkar and D. Gujar, "A Secure Password Manager," International Journal of Computer Applications, vol. 178, no. 44, pp. 5–10, 2019, doi: 10.5120/ijca2019919323.

[22] E. Kavitha, R. Tamilarasan, A. Baladhandapani, and M. K. J. Kannan, "A Novel Soft Clustering Approach for Gene Expression Data," Computer Systems Science and Engineering, vol. 43, no. 3, pp. 871–886, 2022, doi: 10.32604/csse.2022.021215.

[23] Q. Guo, J. Ye, B. Li, Y. Hu, X. Li, Y. Lan, and G. Zhang, "PUFPass: A Password Management Mechanism Based on Software/Hardware Codesign," Integration, vol. 64, pp. 173–183, 2019, doi: 10.1016/j.vlsi.2018.10.003.

[24] B. Yang, H. Chu, G. Li, S. Petrovic, and C. Busch, "Cloud Password Manager Using Privacy-Preserved Biometrics," in Proc. IEEE Int. Conf. Cloud Engineering, 2014, pp. 610–614, doi: 10.1109/IC2E.2014.41.

[25] Y. Kumarakalva, V. S. G. S., S. K. K., S. P. H., and A. G. R., "A Secure Password Manager," *Journal of Emerging Technologies and Innovative Research*, vol. 4, no. 3, pp. 204–206, 2017.