Password Manager – Explanation Note

Introduction

The Password Manager is a secure application built in Python that allows users to safely store, retriev

Key Components

1. Database Management (SQLite):
- Uses sqlite3 to store service names, usernames, and encrypted passwords.
- Lightweight and file-based database, easy to use.

2. Encryption (Using cryptography):
- Uses Fernet symmetric encryption to protect passwords.
- Encryption key stored securely in a 'key.key' file.

3. Secure Input Handling:
- Uses getpass to input passwords securely.
- Ensures sensitive information is not displayed.

Working of the Application

1. Authentication: User enters a master password to access the application.
2. Adding a Password: Stores service credentials in encrypted format.
3. Viewing Passwords: Decrypts and displays stored entries.
4. Deleting a Password: Removes entries by service name.
5. Encryption Key Handling: Encrypts and decrypts data using a stored key.

Security Features

- Encryption: Protects stored passwords.
- Master Password Authentication: Ensures only authorized access.
- Secure Input: Passwords entered without being displayed.
- Database Protection: Local storage with backup options.

Possible Improvements

- Store master passwords using hashing algorithms.
- Add password strength validation.
- Encrypt the entire database for additional security.
- Build a graphical user interface (GUI).
- Integrate cloud storage for backup.

Conclusion

The Password Manager project is a simple yet powerful tool for managing passwords securely using