CloudVault

Password Manager with cloud feature

Work done by:

Ricardo Silva
Guillermo Lopéz De Arechavaleta
Saúl Sauca Torremocha

RUN-EU PYTHON 2024

1 Initial Ideas of the Project and the Goal for Making this App

1.1 Initial Ideas

The initial idea for this project was to create a secure and user-friendly password manager that would allow users to store, retrieve, and manage their login credentials for various websites efficiently. The motivation stemmed from the growing need to manage complex and unique passwords for multiple online services while maintaining the security of personal data.

1.2 Goal

The main goal was to develop an application that not only provides robust encryption for storing passwords but also ensures ease of use. The app aimed to address common issues such as password fatigue, where users struggle to remember numerous complex passwords, and security vulnerabilities arising from reusing passwords across different sites.

2 Main Features of the Program

2.1 Secure Storage

The program uses strong encryption algorithms to securely store user passwords in a local SQLite database, ensuring that sensitive information is protected from unauthorized access.

2.2 Password Management

- **Add Password**: Users can add new passwords for different websites. The program checks if a username for the given site already exists and prompts the user if they want to overwrite the existing password or add a new entry.
- **Update Password**: Users can update existing passwords if they change their login credentials for any site.
- **Delete Password**: Users can delete passwords they no longer need.
- **Search Passwords**: Users can search for stored passwords by site or username, making it easy to retrieve specific credentials.

2.3 User Interface

The program includes a simple and intuitive user interface, designed to make password management straightforward and accessible, even for users who are not tech-savvy.

3 How Usable is It?

The usability of the program has been a key focus throughout the development process. Key aspects of its usability include:

3.1 Ease of Use

The interface is designed to be clean and intuitive, minimizing the learning curve for new users.

3.2 Efficiency

Common tasks like adding, updating, and retrieving passwords can be done quickly and without unnecessary steps.

3.3 Feedback

The program provides clear feedback messages to the user, such as confirmations when passwords are added or updated and warnings when potential errors occur.

4 What We Couldn't Implement / Improve

4.1 Multi-Platform Support

The current version of the program is designed for a single operating system. Implementing multi-platform support would improve accessibility for users across different devices and operating systems.

4.2 Cloud Synchronization

A feature that allows users to sync their passwords across multiple devices via cloud storage was not implemented. This feature would greatly enhance the program's functionality by allowing users to access their passwords from anywhere.

4.3 User Authentication

Adding an additional layer of security, such as user authentication to access the password manager, was not included. This would ensure that only authorized users can view and manage the stored passwords.

4.4 Enhanced Encryption

While the current encryption is robust, implementing more advanced encryption techniques or integrating with established security frameworks could further enhance the security of stored passwords.

5 How It Works

5.1 Database Initialization

Upon the first run, the program initializes an SQLite database (`passwords.db`) and creates the necessary tables if they do not already exist. This ensures that the database structure is ready for storing passwords securely.

5.2 Adding a Password

When a user adds a new password, the program checks if an entry for the given site and username already exists. If it does, the user is prompted to decide whether to overwrite the existing password or add a new entry. The password is then encrypted and stored in the database.

5.3 Updating a Password

Users can update passwords by specifying the site and username. The program locates the existing entry and updates it with the new encrypted password.

5.4 Retrieving Passwords

Users can search for passwords by site or username. The program fetches the relevant entries from the database, decrypts the passwords, and displays them to the user.

5.5 Deleting a Password

Users can delete passwords by specifying the site and username. The program removes the corresponding entry from the database.

5.6 Executing with PyInstaller

To create an executable, the project uses PyInstaller. The setup ensures that `encryption.py` is executed first for any necessary encryption setup, followed by `main.py` to launch the application.