VaultGuard

A Secure Password Manager Built with Python

Created by: Muhammad Daniyal Alam
Project Type: Python Desktop Application
Main Technologies: Python, Tkinter, Cryptography
Date: June 2025

Project Overview

VaultGuard is a secure, user-friendly password manager desktop application built in Python using the Tkinter GUI toolkit. It allows users to safely store, manage, and assess the strength of their passwords with a sleek, tabbed interface. The application is designed with a focus on security, usability, and functionality, making it ideal for personal use and a strong addition to any portfolio.

This project was developed as a learning experience to apply key programming concepts such as:

- File handling and data encryption
- Secure login and user authentication
- GUI design and event-driven programming
- Modular code structure and user experience design

VaultGuard ensures that all sensitive data is encrypted and user-specific, allowing individuals to create accounts, log in, and manage their passwords securely. It includes helpful features such as password strength rating, password hints, clipboard copying, and secure re-verification for viewing or editing data.

Features & Functionality

VaultGuard provides a comprehensive set of features focused on secure password management and user convenience. Below is a detailed breakdown of the core functionalities:

User Authentication

- **Sign-Up**: Users can register with a unique username, main password, and an optional password hint.
- **Login**: Secure login with verification of credentials.
- **Password Hint**: If login fails, a hint is shown (if provided) to assist the user.

Password Vault

- **Add Password**: Users can add entries for different platforms, including the site name, username/email, and password.
- **Show Passwords**: Requires re-verification of the main password before revealing saved entries.
- **Copy to Clipboard**: Once revealed, passwords can be copied with a single click (clipboard is automatically cleared after a short delay).
- **Delete Entry**: Users can delete saved password entries securely.

Edit Profile

- **Change Username or Password**: Allows secure updating of account credentials with password re-verification.
- All data is re-encrypted if the password is changed.

Password Strength Checker

- Users can enter any password to check its strength.
- Provides a strength rating (Weak, Moderate, Strong, etc.).
- Displays estimated time to crack the password using brute-force methods.

User Interface

- Dark-themed UI with a modern, clean layout.
- Tabbed navigation for seamless access to all sections.
- Responsive elements with visual feedback.

Security & Encryption

- All passwords and sensitive user data are encrypted using SHA-256 and secure hashing mechanisms.
- Each user’s data is stored in a separate, encrypted JSON file.
- Passwords are never stored in plain text.

Technology Stack

VaultGuard is developed using a combination of reliable and widely used tools and libraries, making the application both robust and easy to maintain. Below is a breakdown of the technology stack used:

Programming Language

- **Python 3**
  Chosen for its readability, simplicity, and wide support for GUI and cryptographic libraries.

Libraries & Modules

- **Tkinter**
  Used for creating the graphical user interface (GUI) with a dark-themed, tabbed layout.
- **ttk (Themed Tkinter Widgets)**
  Provides modern, styled widgets integrated with Tkinter.
- **Hashlib & Binascii**
  Used for secure password hashing and conversion of binary data to ASCII representations.
- **JSON**
  Stores encrypted user data and saved password entries in a structured format.

- **OS & SYS**
  Used for file handling, launching scripts, and managing application state during restarts.
- **Subprocess**
  Allows restarting the main interface (e.g., after changing user credentials) by calling the Python interpreter.

💻 Platform

- **Cross-platform (Windows/Linux/MacOS)**
  The application runs anywhere Python and Tkinter are supported

File Structure & Architecture

VaultGuard follows a modular structure, where each component of the application is separated by functionality. This not only improves readability and maintainability, but also allows easier testing and debugging.

🗀 Core Files:

`main.py`

Purpose: Acts as the primary entry point after the splash screen. Displays the login window and initiates the dashboard upon successful login.
Key Functions:

- Collects username and main password.
- Validates credentials via `user_manager.py`.
- Launches `dashboard.py` upon successful login.
- Displays error messages on failed attempts.
- Allows navigation to the sign-up screen.

`signup_screen.py`

Purpose: Handles user registration (sign-up process).
Key Functions:

- Collects username, main password, confirmation, and optional password hint.
- Checks if the username already exists.
- Hashes and stores user credentials securely.
- Creates an encrypted data file for the user.

`run.py`

Purpose: The actual launcher of the application. Displays the splash screen before routing the user to the login screen.
Key Functions:

- Shows the splash screen (`splash_screen.py`) for a few seconds.
- Launches the `main.py` login screen once loading is complete.

`splash_screen.py`

Purpose: A temporary screen that gives a polished look while the application is starting.
Key Features:

- Displays the VaultGuard logo or title with a loading bar or text.

- Automatically closes after a short delay, then loads the main interface.

`dashboard.py`

Purpose: The central hub of the application once the user logs in.
Key Features:

- **Saved Passwords Tab**: Lists saved accounts and services with decrypted details after re-verification.
- **Add New Password Tab**: Allows users to save new credentials with encryption.
- **Edit Profile Tab**: Lets users change their main password or username.
- **Password Strength Tab**: Rates the strength of any password and estimates crack time.
- Includes "copy to clipboard" and "show password" options.
- Handles encryption/decryption using the user's main password.

`user_manager.py`

Purpose: Manages user authentication, encryption, and data storage.
Key Functions:

- Hashes and verifies main passwords.

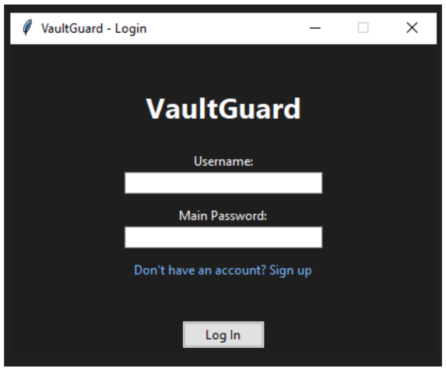
- Manages the creation of user files and encryption keys.
- Handles data saving and loading for each user's vault.
- Ensures password hints and re-verification mechanisms are secure.

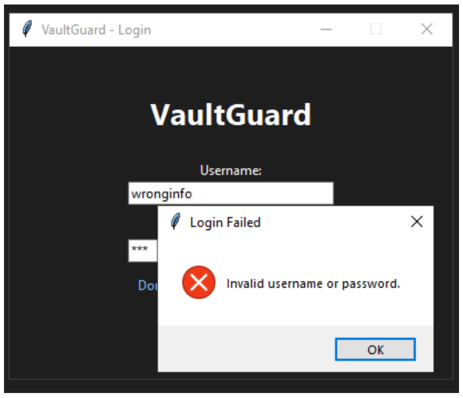
Key Features Overview

VaultGuard is designed to be a user-friendly, visually sleek, and highly secure desktop password manager. Below is a breakdown of its major features:

User Authentication

- Secure login system with username and main password.
- Invalid login attempts trigger helpful feedback.
- Password hint shown after incorrect login to assist users.



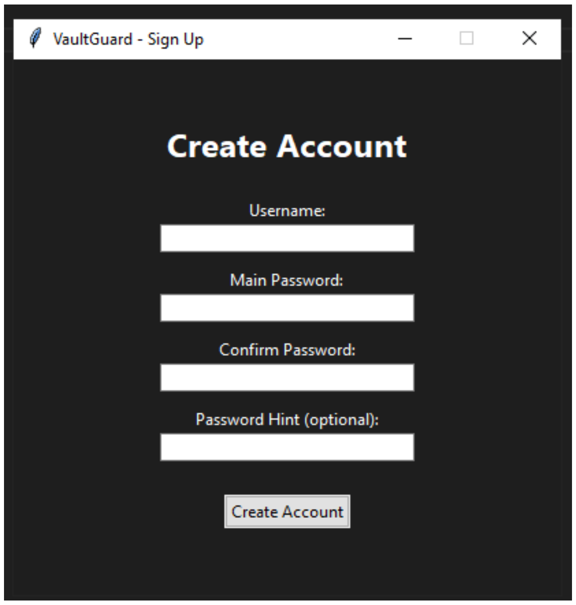


wronginfo

Invalid username or password.

🗒 User Registration

- New users can sign up with a username, password, and an optional password hint.
- All data is securely stored and uniquely encrypted per user.



🗂 Multi-Tab Dashboard

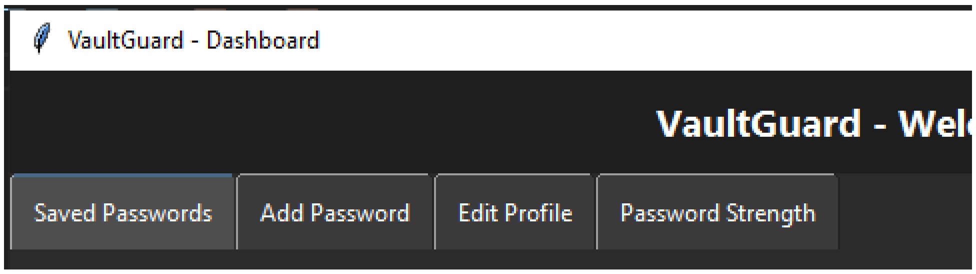
A dark-themed interface with the following tabs:

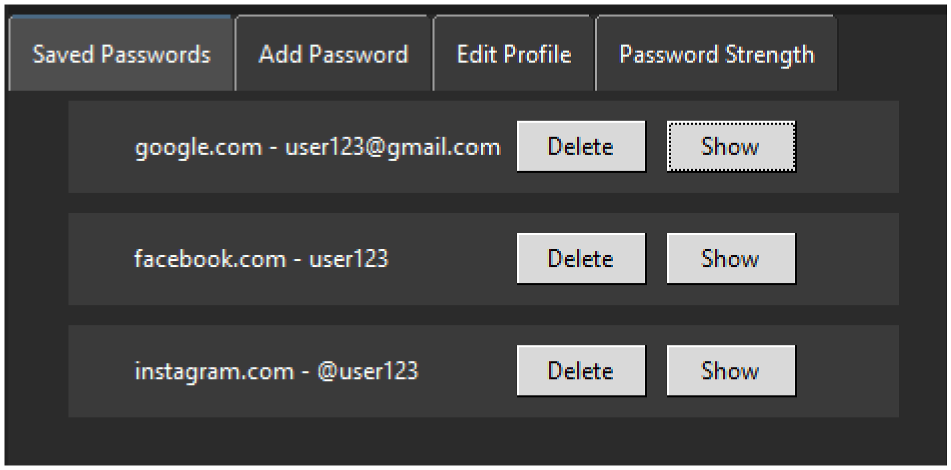
1. 🔎 **Saved Passwords**
   a. Shows user's stored passwords.
   b. Re-verification (entering main password again) required before revealing sensitive data.
   c. Passwords can be copied to the clipboard.
   d. Clean layout with scrollable list of entries.
2. ➕ **Add Password**
   a. Add a new password entry with website, username, and password fields.

b. Ensures secure encryption before saving.

3. 📝 **Edit Profile**
   a. Change username or main password securely.
   b. Updates re-encrypt user vault accordingly.

4. 🔍 **Password Strength Checker**
   a. Enter any password to get a security rating.
   b. Estimates how long the password would take to crack using brute force.
   c. Provides helpful suggestions if the password is weak.

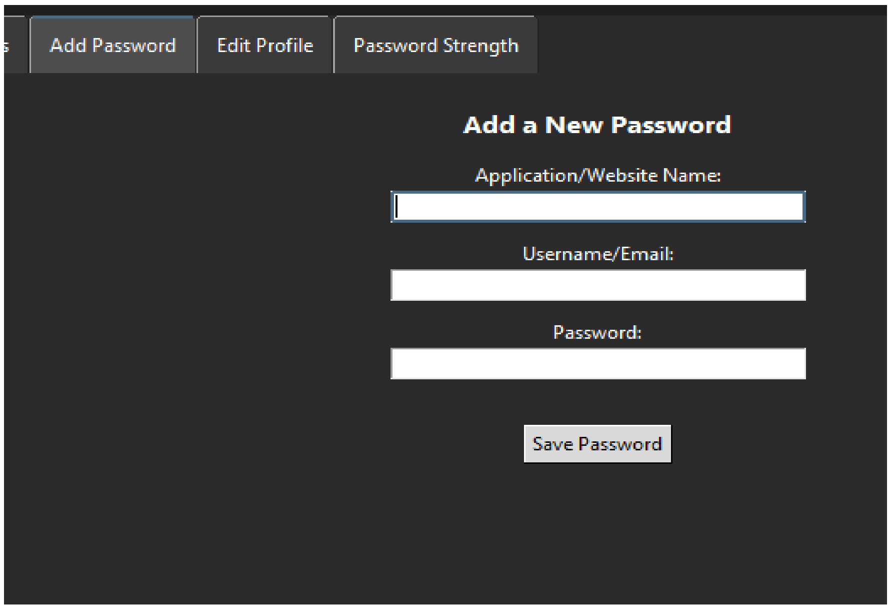




google.com - user123@gmail.com

facebook.com - user123

instagram.com - @user123

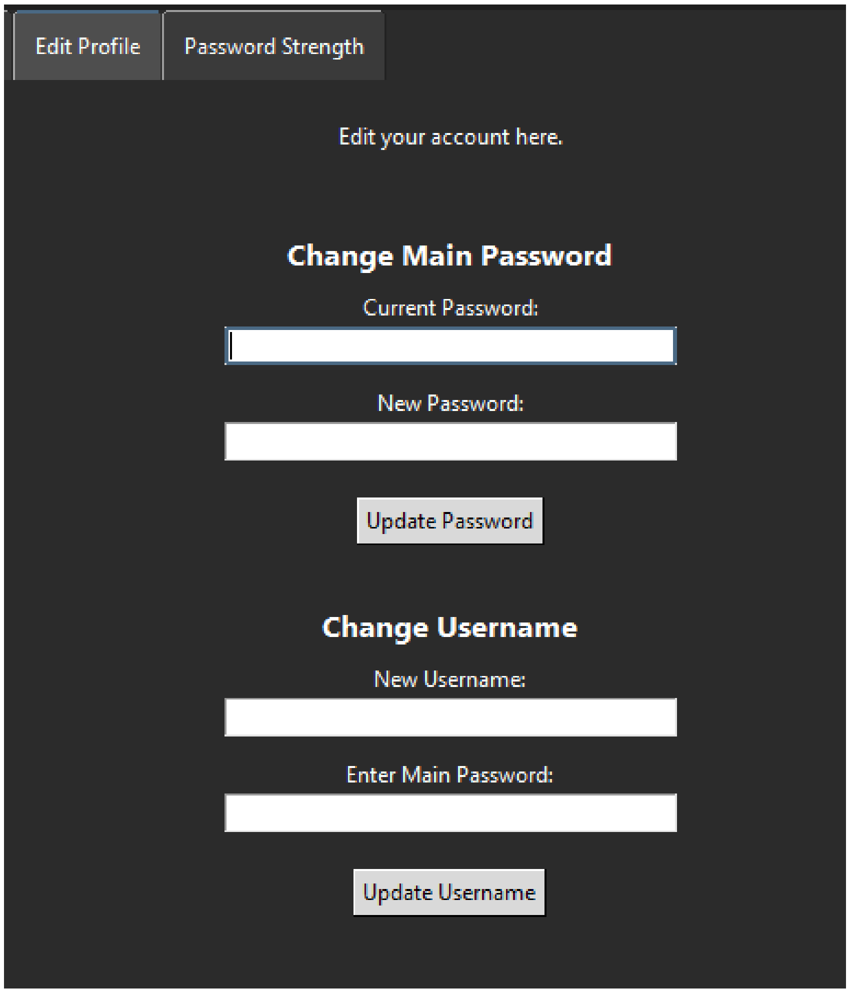


Add a New Password

Application/Website Name:

Username/Email:

Password:



Edit your account here.

Change Main Password

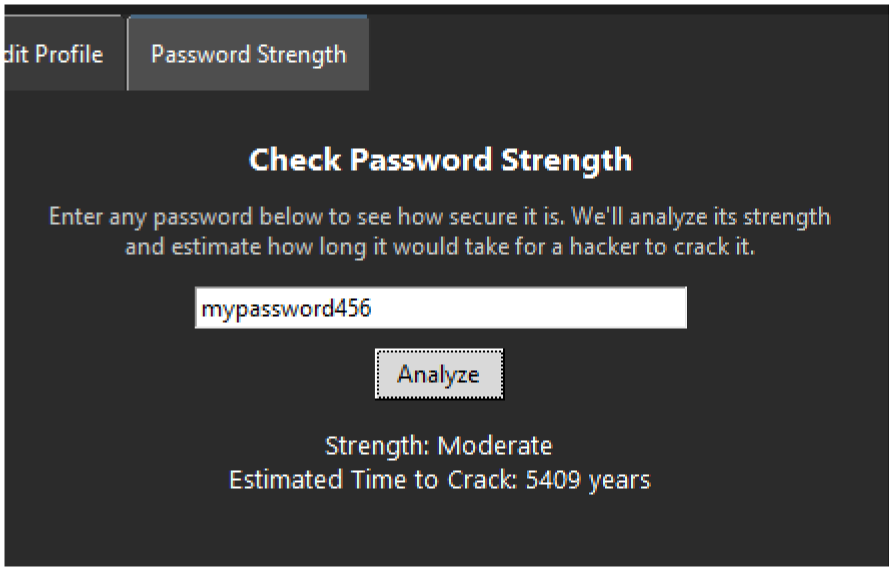
Current Password:

New Password:

Change Username

New Username:

Enter Main Password:



Check Password Strength

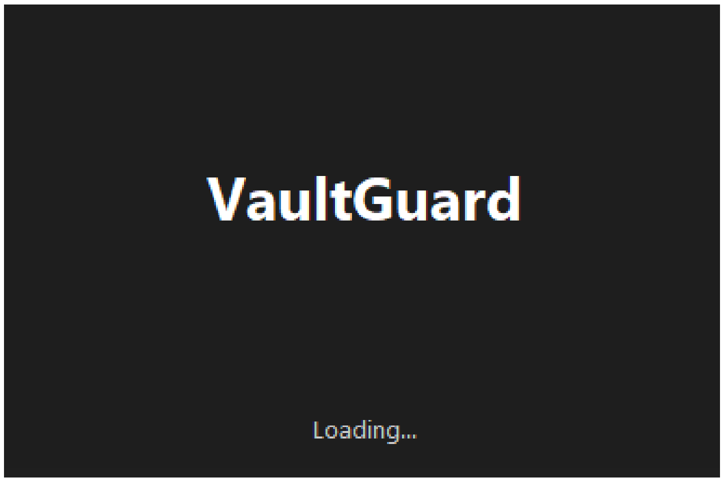
Enter any password below to see how secure it is. We'll analyze its strength and estimate how long it would take for a hacker to crack it.

mypassword456

Strength: Moderate
Estimated Time to Crack: 5409 years

✦ Splash Screen

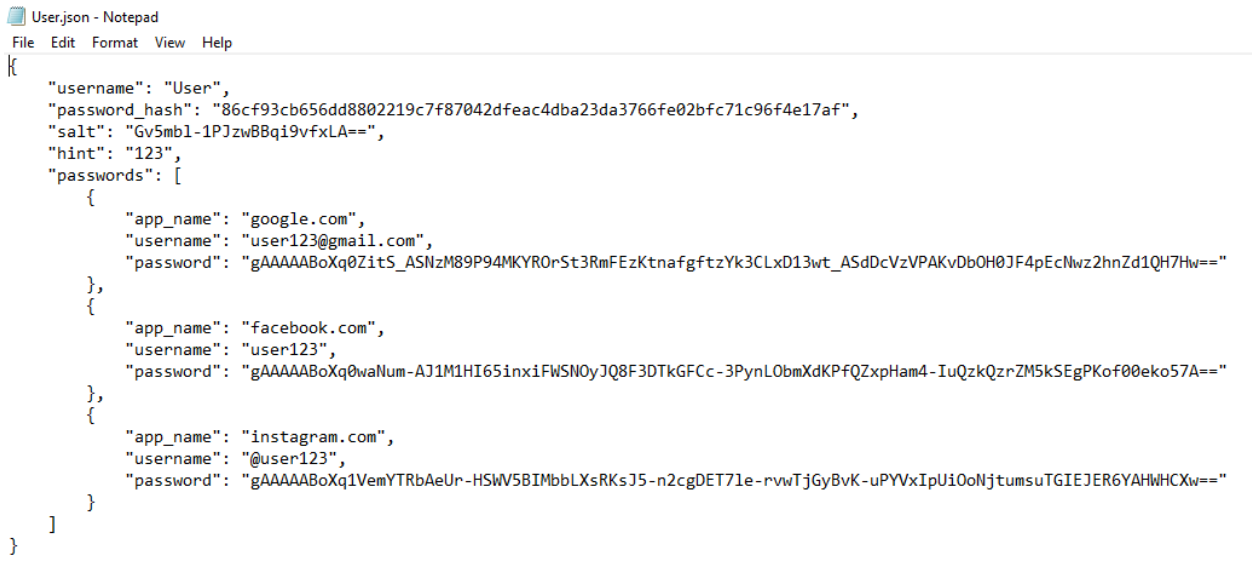
- Attractive animated splash screen shown at launch.
- Enhances user experience with professional UI.



Encryption and Security

- All saved data is encrypted using a key derived from the main password.
- Vaults are unreadable without correct credentials.
- Uses SHA-256 hashing, PBKDF2 key derivation, and AES encryption.

How a profile data is stored:



```
{
    "username": "User",
    "password_hash": "86cf93cb656dd8802219c7f87042dfeac4dba23da3766fe02bfc71c96f4e17af",
    "salt": "Gv5mbl-1PJzwBBqi9vfxLA==",
    "hint": "123",
    "passwords": [
        {
            "app_name": "google.com",
            "username": "user123@gmail.com",
            "password": "gAAAAABoXq0ZitS_ASNzM89P94MKYROrSt3RmFEzKtnafgftzYk3CLxD13wt_ASdDcVzVPAKvDbOH0JF4pEcNwz2hnZd1QH7Hw=="
        },
        {
            "app_name": "facebook.com",
            "username": "user123",
            "password": "gAAAAABoXq0waNum-AJ1M1HI65inxiFWSNOyJQ8F3DTkGFCc-3PynLObmXdKPfQZxpHam4-IuQzkQzrZM5kSEgPKof00eko57A=="
        },
        {
            "app_name": "instagram.com",
            "username": "@user123",
            "password": "gAAAAABoXq1VemYTRbAeUr-HSWV5BIMbbLXsRKsJ5-n2cgDET7le-rvwTjGyBvK-uPYVxIpUiOoNjtumsuTGIEJER6YAHWHCXw=="
        }
    ]
}
```

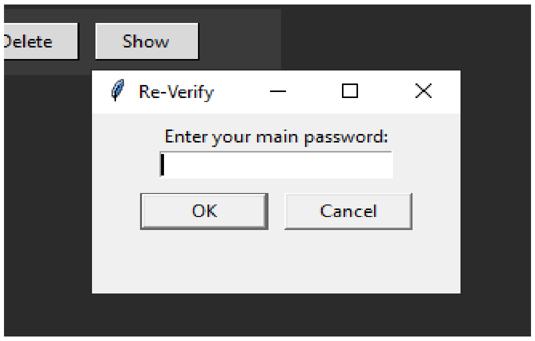
The main password, website passwords, and salt are encrypted and stored securely to make it difficult for attackers to decrypt and steal the data.

User Verification & Password Reveal

In VaultGuard, security is a top priority. To prevent unauthorized access to sensitive data, the application enforces **re-verification** of the user's main password before displaying any saved passwords. Here's how it works:

✓ Step-by-Step Verification Process:

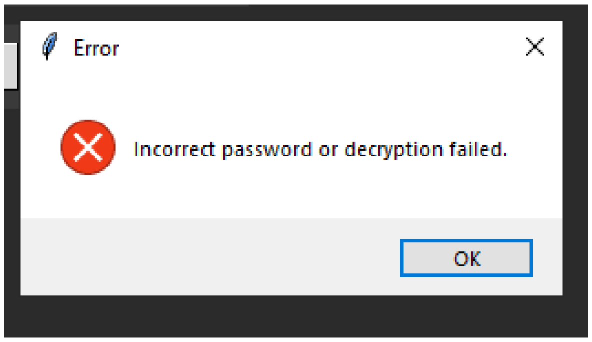
1. **User logs into their account** using the correct username and main password. If the login is successful, the user is taken to the main dashboard.
2. In the **"Saved Passwords" tab**, the user sees a list of entries with masked (hidden) password fields.
3. To **view a specific password**, the user must:
    a. Click the **"Show"** button next to that entry.
    b. A **popup window** will appear asking them to re-enter their **main password**.



4. If the entered password **matches** the one used at login:
    a. The application decrypts the password entry using a key derived from that password.

    b. The **actual password is displayed** in the GUI.

5. If the entered password is **incorrect**:
    a. The popup closes or shows an error.
    b. The password remains hidden to protect sensitive data.



Incorrect password or decryption failed.

Why This Matters

- This double-verification adds an **extra layer of security** in case someone gains access to a logged-in session.
- Even if the main GUI is open, passwords can’t be revealed or copied without proving identity again.
- It mirrors security behavior used in professional-grade password managers.

Conclusion

VaultGuard was designed with simplicity and functionality in mind, while incorporating real-world security practices such as encryption, password strength checking, and user re-verification.

Through features like:

- **Encrypted password storage**
- **Multi-tab GUI with dark mode**
- **Profile editing capabilities**
- **Real-time password strength analysis**
- **Password hint system**
- **Copy-to-clipboard and re-verification mechanism**

VaultGuard demonstrates the core principles of secure application development, usability-focused design, and practical implementation of cryptographic techniques.

This project not only showcases technical proficiency in Python and GUI development, but also reflects an understanding of user experience and cybersecurity fundamentals. Whether used as a portfolio piece, educational tool, or foundation for future projects, VaultGuard stands as a meaningful step toward mastering secure software development.