

Password Manager Documentation

GitHub Link

https://github.com/matthewcao/Password_Manager

Overview

Password Manager is a program written in Python designed to securely store passwords. Users can create their own accounts and log in to add, view, and remove a list of personal passwords. All user information including log in data and stored passwords are encrypted. To manage their passwords, users need their username and password for the Password Manager as well as a physical decryption key, which is stored on a separate storage device such as a USB stick.

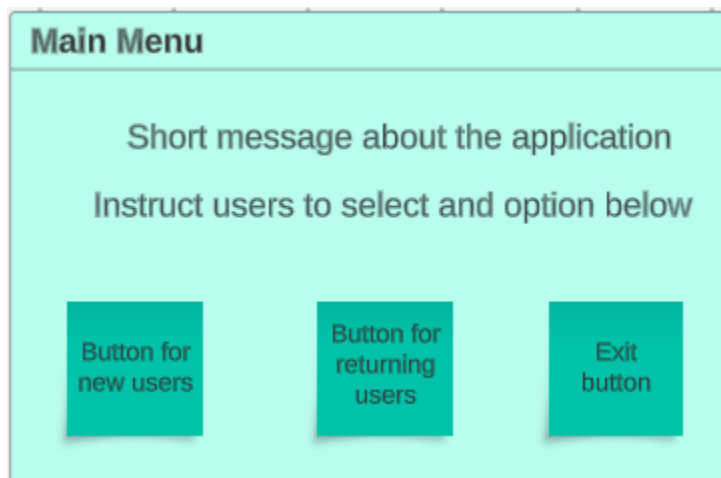
User Stories

- As Harry, I want my passwords to be stored somewhere with clear labels stating what account each password is associated with so that I don't get confused about what my passwords are for.
- As Darryl, I want my passwords to be in a safe location so that only I can access them.
- As Caleb, I want to be able to have a large number of passwords stored somehow so that I don't forget them.
- As James, I want multi-factor authentication to keep my passwords safe so that I don't have to worry too much about my password manager credentials being stolen.
- As Jimmy, I want to make sure my passwords are encrypted instead of in plain text so that my passwords aren't completely visible even if my password manager is compromised.
- As Sarah, I want a user interface for a password manager that is really easy to figure out without needing an instruction manual so that I can save time and use it efficiently.

- As Liam, I want a password manager that works on my desktop computer so that I can secure my accounts on my home computer.
- As Evan, I want my account for my password manager to have certain password requirements to so that I know what password would be secure for my password manager.
- As George, I to be able to easily add new passwords to a password manager with the click of a button so that the new accounts I create frequently can be immediately secured.

Lo-fi Mockups

- Main Menu
 - The main menu screen is the first thing that is visible when the application runs.
There are three buttons with specific purposes. The first button is for new users to set up an account for using the application. Another button allows returning users to access existing accounts. The last button is for exiting the application, but I may end up removing that button because it isn't necessary.



- New User

- Clicking the “New Users” button on the main menu causes a new window to pop up. New users must create an account to use the application. The new user menu asks for a username and password.

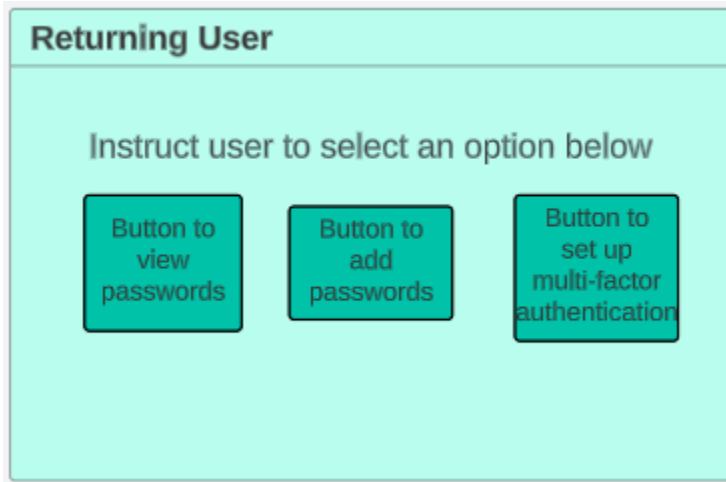
The diagram shows a light blue rectangular box with a title bar at the top that reads "Returning User". Below the title bar, the text "Instruct users to fill out the fields below to se up a new account" is displayed. Centered within the box are three stacked, rounded rectangular buttons. The top button is labeled "Username field", the middle button is labeled "Password field", and the bottom button is labeled "Finish button".

- Returning User Login
 - A returning user of the application who already has an account set up can log into their account. There is a username and password field. I have not yet set up multi-factor authentication, but plan to do so by having a small pop-up message request a verification code after a username and a password are entered.

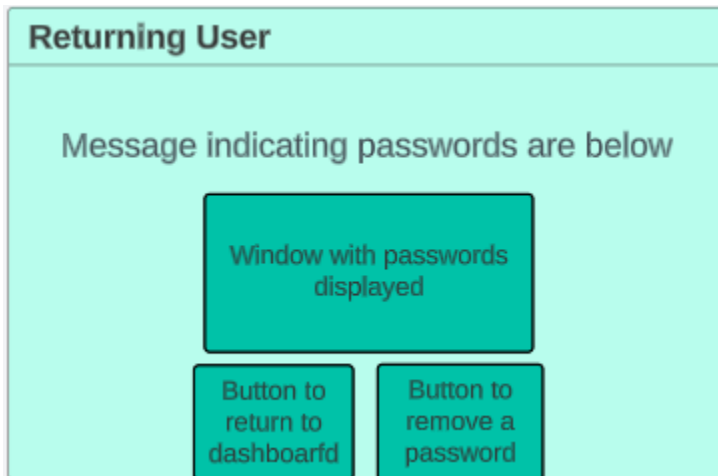
The diagram shows a light blue rectangular box with a title bar at the top that reads "Returning User". Below the title bar, the text "Welcome back returning users" is displayed. Centered within the box are three stacked, rounded rectangular buttons. The top button is labeled "Username field", the middle button is labeled "Password field", and the bottom button is labeled "Button to complete login".

- Returning User Dashboard

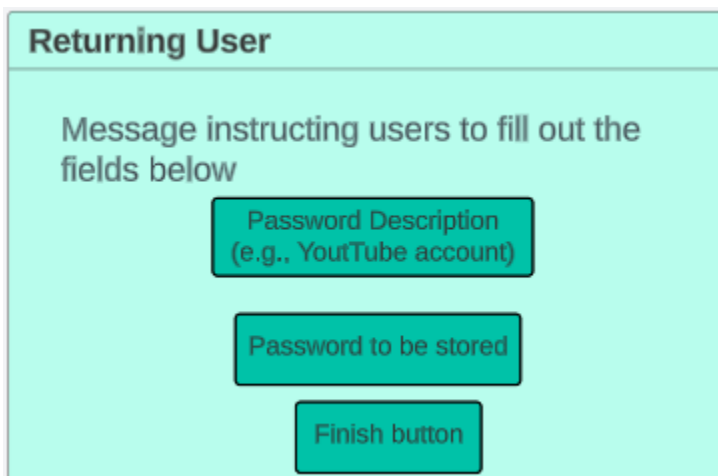
- After returning users of the application log in to their accounts, they will see a dashboard to manage their passwords. There are four buttons on the dashboard. The first one is to view passwords that are stored in the password manager. The second button is to add new passwords to be stored. The third button is to set up multi-factor authentication.



- View Passwords
 - Users can view the passwords saved in the password manager. The passwords are displayed in a window within the main screen. Additionally, there is a button to return to the dashboard and another button to remove a password. If the button to remove a password is clicked, a pop-up message will be displayed that will ask for the password to be removed. After the password is entered, it will be deleted from the password manager.



- Add Passwords
 - Returning users can add a password to be stored in the password manager. There is a fields requesting a description of what account the password is for and another field for entering the password itself. Lastly, there is a button to finish adding a password.



Program Structure

- Import libraries
 - tkinter (messagebox, askstring), cryptography (fernet), os (path, pyperclip)
- Globally set up a key file for encrypting and decrypting files using fernet

- Main window class
 - Initialize function
 - Background color, title, window size, prevent resizing, center window, header label
- Main buttons class
 - Initialize function
 - New users button, returning users button, usage information button, exit button
 - New user window function
 - Set window as top level
 - Call instance of new user window class
 - Returning user window function
 - Set window as top level
 - Call instance of returning user window class
 - Usage information window function
 - Set window as top level
 - Call instance of usage information window class
 - Exit window function
 - Close window
- Get username input class
 - Initialize function
 - Username input function
 - If user data file doesn't exist, create file and write username

- If user data file does exist, decrypt file, add username to existing data, and rewrite to file
- Get password input class
 - Initialize function
 - Password input function
 - Get password and write to file
- New user window class
 - Initialize function
 - Save username and password inputs, title, window size, prevent window resizing, background color,
 - Header label, username label and entry, password label and entry
 - Save profile button
 - Call instances of username input and password input classes to save inputs to file
 - Get inputs function
 - Validate username and password
 - Get username and password inputs
 - Encrypt user data file
 - Display success message if account is created
 - Close window
- Get new password description input class
 - Initialize function
 - New description input function

- Write password description to file
- Get new password input class
 - Initialize function
 - New password input function
 - Write password to file
- Returning user window class
 - Initialize function
 - Get username and password inputs
 - Title, window size, prevent window resizing
 - Set up frames, set up default frame to login
 - Save password description and password inputs
 - Login frame function
 - Set up frame
 - Background color
 - Header label, username label and entry, password label and entry
 - Button to complete login
 - Display login frame
 - Check close login frame function
 - Get username and password inputs
 - Open user data file to verify inputs
 - Verify inputs and close frame
 - Dashboard frame function
 - Set up frame

- Background color
- Header label
- View passwords button, add passwords button, exit button
- Display dashboard frame
- View passwords frame function
 - Set up frame
 - Background color
 - Header label
 - Set up a scrollbar to display passwords
 - Try display passwords; except FileNotFoundError Error then pass
 - Back to dashboard button, remove password button
 - Display view passwords frame
- Remove password function
 - Pop-up message asking which password to remove
 - If password exists, remove password and associated description
 - Write new data with password removed back to file
- Add password function
 - Set up frame
 - Background color
 - Header label, new password description label and entry, new password label and entry
 - Back to dashboard button, finish button

- Call new password description input and new password input classes to save inputs
 - Display the frame
- Get inputs function
 - Decrypt saved passwords file, write new password, encrypt file
 - Close add passwords frame and return to dashboard frame
 - Display success message
- Return to dashboard frame function
 - Return to dashboard when button is clicked
- Exit function
 - Exit window
- Info window class
 - Initialize function
 - Title, window size, prevent window resizing, background color
 - Call show info function
 - Show info function
 - Header label, usage information labels
 - Exit window button
- Main function
 - Run the program
 - Remove decryption key file when program ends
 - Clear clipboard when program ends
- Script checker

Resources

Set up a user interface with Tkinter	https://www.geeksforgeeks.org/create-first-gui-application-using-python-tkinter/
Change font size of labels	https://www.geeksforgeeks.org/how-to-change-the-tkinter-label-font-size/
Create buttons	https://www.geeksforgeeks.org/python-creating-a-button-in-tkinter/
Set windows as top level	https://stackoverflow.com/questions/48278853/tk-toplevel-and-winfo-toplevel-difference-between-them-and-how-and-when
Encrypt and decrypt files in Python	https://www.geeksforgeeks.org/encrypt-and-decrypt-files-using-python/
New window pop-up when a button is clicked	https://www.geeksforgeeks.org/open-a-new-window-with-a-button-in-python-tkinter/
Validating passwords	https://www.geeksforgeeks.org/password-validation-in-python/
Find text in files	https://www.geeksforgeeks.org/python-how-to-search-for-a-string-in-text-files/
Create an exit button	https://www.tutorialspoint.com/how-to-exit-from-python-using-a-tkinter-button
Input validation	https://www.sololearn.com/Discuss/1582033/how-to-check-if-input-is-empty-in-python

	https://www.sololearn.com/Discuss/1582033/how-to-check-if-input-is-empty-in-python
Pack() method with Tkinter	https://tinyurl.com/98ydf9nd
Check if a file exists in directory	https://tinyurl.com/mvzu9zhv
Remove a file from directory	https://www.w3schools.com/python/python_file_remove.asp
Clear clipboard	https://stackoverflow.com/questions/53226110/how-to-clear-clipboard-in-using-python
Change background color of windows	https://www.tutorialspoint.com/how-to-change-the-background-color-of-a-tkinter-canvas-dynamically
Tkinter colors	https://tinyurl.com/3enkbu7d
Change text color	https://www.tutorialspoint.com/python/tk_text.htm
Center a window on screen	https://tinyurl.com/3tnctpaf
Convert a Python script to an executable	https://www.geeksforgeeks.org/convert-python-script-to-exe-file/
Code troubleshooting	https://chat.openai.com/