

2025-12-08 - CSC 255 Group Assignment

Group members

- Fernando Pereira - fchiaver88
- La Min Aung - laung6186
- Quang Dang - ndang9904
- Audrey He - zhe7997
- Jiajun Wang - jwang8569

Github Repository

[Link](#)

Summary

Application Description

The Keeper Password Vault Python App is a secure, command-line-based password manager designed to replace unsafe paper lists. It enables users to store, retrieve, update, and delete password entries tied to unique names. All data is encrypted and stored centrally. Core features include password creation, retrieval, updates, deletion, and a report listing the top five most recently updated password entries. The app emphasizes security through encryption and efficient data handling using structures like hash tables and max heaps for retrieval and reporting.

Key Accomplishments

- Implemented full vault functionality: create, read, update, delete, and recent-history reporting.
- Added encryption and decryption for secure password storage.
- Designed interactive CLI flow for user operations.
- Built automated tests using unittest and unittest.mock to verify database and CLI behavior.
- Documented workflows, testing results, and README instructions.

Contributions Overview

- **Fernando Pereira:** Led project coordination, database functions, and testing.
- **Quang Dang:** Implemented encryption/decryption, structured main app entry point, and maintained documentation.
- **Audrey He:** Developed cli interface and user input flow.
- **Jiajun Wang:** Created and managed automated testing suites, bug fixes, and test documentation.
- **La Min Aung:** Contributed to summary documentation, and bug fixes.

Project Management

- Utilized a Kanban board to track progress across To-Do, In Progress, and Done tasks (more details about task management in the appendix). The team collaborated to successfully deliver 40 project items over a two week period.
- The team held regular sprint meetings (11/25, 12/01, 12/05) to plan iterations, delegate tasks, demo completed items, and review progress.
- The team performed a final delivery and submission meeting on 12/08.
- If the team had time for another iteration on the application, the following items would be prioritized:
 1. Move password name existence check to the client level vs calling it repeatedly inside each function
 2. Create an user account the first time someone uses the vault and provide a vault password to protect all password names, or implement 2FA protection
 3. Add feature to measure password strength to be called when creating or updating a password
 4. Add a feature to suggest a strong password to the user. Generate the password by using random number generation and establishing heuristics such as requiring lower and upper case letters, numbers, and special characters

Application Execution instructions

The user can interact with the application by executing the file [main.py](#) either on the computer's terminal or preferred IDE. In addition to application execution, the team has run a series of tests to ensure that features are working correctly. These tests can be executed through the terminal by using the commands below. More information about the application execution and tests performed is available in both the appendix below and in the accompanying READ ME file.

Database tests

```
python3 test_database.py
```

CLI tests

```
python3 -m unittest test_vault_cli.py -v
```

Appendix

Project Management

Project Kanban (Task list)

To-Do (Backlog)

- Nice to have (would be worked on in the next iteration):
 1. Dec 5, 2025 — App Dev - I think checking if the name exists should happen in the cli_input. The function delete() should just raise an error instead.
 2. Nov 23, 2025 — App Dev - Feature for the user to create an account and vault password when accessing for the first time OR create a master password to access the vault
 - Think about using 2FA with user e-mail (might be too complex)
 - Alternatively just create a separate file.
 3. Nov 25, 2025 — App Dev - Feature to measure password strength to be called when creating or updating a password
 4. Nov 25, 2025 — App Dev - Feature to suggest a strong password to the user
 - Maybe use random number generation? A-Z and 0-9
 - Consider rules such as having at least one capital letter, one lower case character, special character, and number character

Work in Progress

None at the moment.

Done

Date legend: The first date on the bullet point is the delivery date, and the second date is the task creation date

Week #1 - 11/25/2025 to 12/01/2025 - 16 items delivered!

1. Nov 21, 2025 – Nov 21, 2025 – Management - Reach out to group members - DRI: Fernando Pereira
2. Nov 25, 2025 - Nov 23, 2025 – Management - Set date for first meeting - DRI: Fernando Pereira
 - Meeting goal: define topic of interest
 - Meet with your group to brainstorm and deliberate on your topic, and agree on the general idea of the project. Then use the chatbot some more to finetune your inquiries on the project idea before formulating your project thesis
 - Define backlog and assign tasks
 - Define how to work and collaborate on code
3. Nov 25, 2025 - Nov 23, 2025 – Management - Create WhatsApp group chat - DRI: Fernando Pereira
 - [Link](#)
4. Nov 25, 2025 - Nov 23, 2025 – Management - Set up Github environment - DRI: Fernando Pereira
 - [Link](#)
5. Nov 25, 2025 – Nov 23, 2025 – Meeting - Define topic from the list below - DRI: Group
 - The group aligned on developing a Password Vault app that uses cryptography as encryption
6. Nov 25, 2025 – Nov 25, 2025 – App Dev - Create [main.py](#) and application template - DRI: Quang
7. Nov 26, 2025 - Nov 25, 2025 – App Dev - Implement feature to the **create** (write) the password and its details to the file database - DRI: Fernando Pereira
8. Nov 26, 2025 - Nov 25, 2025 – App Dev - Implement feature to **lookup (read)** the password name and its details in the database- DRI: Fernando Pereira
 - **Additional details:** used a hash table to retrieve the password name and the password
9. Nov 26, 2025 - Nov 25, 2025 – App Dev - Implement feature to edit (update) the password in the file database - DRI: Fernando Pereira
10. Nov 26, 2025 - Nov 25, 2025 – App Dev - Implement feature to delete the password name and its details - DRI: Fernando Pereira

11. Nov 26, 2025 - Nov 25, 2025 – App Dev - Implement feature to report Top 5 most recent password names updated - DRI: Fernando Pereira
 - o Additional details: used a max heap data structure to fetch results from the database
12. Nov 26, 2025 - Nov 25, 2025 – App Dev - Implement test function to add, update, delete, and top 5 most recent updated password names report look up - DRI: Fernando Pereira
13. Nov 26, 2025 - Nov 25, 2025 – App Dev - Create encryption and decryption method - DRI: Quang
14. Nov 28, 2025 – Nov 25, 2025 – App Dev - Intro menu with options to what you can do in the app (create a password, or get a password) - DRI: Audrey
 - o Use CLI terminal framework
 - o UI - Prompt user for choice (save password) and get their input (password name, password) (Audrey)
 - o UI - Display confirmation message (Audrey)
 - o UI - Prompt user for choice (get password), get their input (password name) (Audrey)
 - o Display password to user (Audrey)
15. Nov 28, 2025 – Nov 25, 2025 – Testing - App Dev – Create, run, and report test on implemented features – DRI: Jiajun Wang
 - o Created automated test suite (`test_vault.py`) to validate core vault features
 - o Implemented tests for create, lookup, update, and recent-password reporting functionality (Jiajun)
 - o Ensured tests run on an isolated test vault file without modifying real data (Jiajun)
 - o Verified all tests pass successfully on local environment (Jiajun)
 - o Documented testing process and results in `TEST_REPORT.md` (Jiajun)
16. Nov 26, 2025 – Nov 25, 2025 – App Dev - Create readme - DRI: Quang

Week #2 - 12/02/2025 to 12/08/2025 - 24 items delivered

1. Dec 1, 2025 - Nov 23, 2025 – Meeting - Sprint Review #1 + Sprint Plan #2 - 12/01 @ 9pm EST- DRI: Fernando Pereira
2. Dec 4, 2025 - Dec 1, 2025 – App Dev - Merge `cli_input.py` and `Main.py` - DRI: Audrey He & Quang
3. Dec 4, 2025 - Dec 1, 2025 – App Dev - `cli_input.py` - Iterate on user flow and function calls- DRI: Audrey He
 - o Suggested flow:
 - Open the app
 - Get welcome message
 - Get a menu with the following options:
 - Store a new password name and password
 - o calls `create_password_name(name, encrypt(pwd))`
 - Retrieved a stored password name and password
 - o calls `lookup_password_name(name)`
 - Update a stored password name and password
 - o calls `update_password_name(name, encrypt(pwd))`
 - Delete a stored password name and password
 - o calls `delete_password_name(name, encrypt(pwd))`
 - Get recent password details
 - o calls `get_most_recent_password_names()` - call is OK in cli
 - Exit
4. Dec 4, 2025 - Dec 1, 2025 – App Dev - `cli_input.py` - Ensure decryption when retrieving the password back to the user. - DRI: Quang
 - o `cli_input.py`
 - 1) Issue:
 - Retrieved passwords are being printed with encryption
 - Suggested fix: connect the functions to change and to remove password
5. Dec 4, 2025 - Dec 1, 2025 – App Dev - `main.py` - Move database functions to `database.py` and make `main.py` the entry point of the app - DRI: Quang
6. Dec 4, 2025 - Dec 1, 2025 – Summary Doc - Update read me file. Emphasize required modules needed to use the app - DRI: Quang
7. Dec 4, 2025 - Dec 3, 2025 – App Dev - Database- Handle listing passwords when the vault is empty (or not created yet) - DRI: Quang(?)
8. Dec 5, 2025 - Nov 25, 2025 – App Dev - Database- Implement feature to avoid duplicate password names to be created - DRI: Jiajun
9. Dec 5, 2025 - Dec 2, 2025 -Database-Fix Update password: got “The password NAME {name} was not found in order to be updated.” need to add f-string to `update_password_name()` for this print. - DRI: La Min Aung
10. Dec 5, 2025 - Dec 3, 2025 – App Dev - Database- Bug: When vault is empty and the user chooses to retrieve an item, error is raised because there is no file to read. The behavior should be notifying the user that the vault does not exist - DRI: Fernando Pereira
11. Dec 5, 2025 - Dec 3, 2025 – App Dev - Database- Bug: When vault is empty and the user chooses to update an item, it is created. The behavior should be notifying the user that the item does not exist - DRI: Jiajun

12. Dec 5, 2025 – Testing - test_vault.py - Need to test from main.py, simulating user inputs using “from unittest.mock import patch”. - DRI: Jiajun
13. Dec 5, 2025 – Testing - test_database.py - Need to test database logic (create, lookup, update, delete, recent) using an isolated vault file. — DRI: Jiajun
14. Dec 5, 2025 - Dec 5, 2025 – Summary Doc - TEST_REPORT.md-Need to document testing coverage & results. — DRI: Jiajun
15. Dec 5, 2025 - Dec 5, 2025 – Summary Doc - README Testing Section-Need to add instructions for running tests. — DRI: Jiajun
16. Dec 5, 2025 - Dec 1, 2025 – App Dev -Database-delete_password_name() confirmation loop needs to compare encrypted data- DRI: Fernando Pereira
 - o delete_password_name() might need some adjustment, because there is a password confirmation step that will compare decrypted password with encrypted password. Data points need to be aligned and compared at the same state.
17. Dec 5, 2025 - Dec 2, 2025 – App Dev -Database-Fix Deletion of non-existing password name gave:
 - o “Password name Fernando not found. Please, create a new password name for this name to be able to use it.
 - o
 - o CANNOT DELETE PASSWORD. Wrong password entered. Try again, or type 'EXIT' to go back to the main menu.
 - o Enter the password:exit” - DRI: Fernando Pereira
18. Dec 5, 2025 - Dec 3, 2025 – App Dev -cli_input.py- Bug: When updating password, but there is no vault, it gives credentials captured. DRI: Fernando Pereira
 - o Removed the printing below as it is already being communicated by database methods.
 - Credentials captured!
 - Website: Fernando
 - Password: ****
19. Dec 5, 2025 - Dec 2, 2025 – App Dev -Database-Password names report: updated to list all password names instead of only the top 5- DRI: Fernando Pereira
20. Dec 5, 2025 - Dec 1, 2025 – Meeting -Final checkpoint meeting on 12/05- DRI: Group
 - o Finishing touches, check on project report document
21. Dec 7, 2025 – Dec 7, 2025 – App Dev - test_vault.py - Need to test from main.py, simulating user inputs using “**from unittest.mock import patch**”. DRI:Jiajun Wang
 - o Finished fix the Test.vault_cli.py
22. Dec 7, 2025 – Nov 23, 2025 – Testing -Create, test, and document application test - DRI: Jiajun Wang
23. Dec 7, 2025 – Nov 23, 2025 – Summary Doc -Complete final project summary. Add it to the top of the document. - DRI: La Min
 - o Add an app description
 - o Comment on features we would like to add if we had more time
 - o Summarize the contributions
 - o Summarize Features
24. Dec 8, 2025 – Nov 23, 2025 – Meeting -Delivery meeting on 12/08 - DRI: Group
 - o Submit the work

Meetings Log

Dec 5, 2025 | Oakton CS Group Assignment - Sprint Review + Sprint Planning

Attendees: quang.dang.n42@gmail.com jwang8569-02wangjiajun@gmail.com Fernando Pereira zh657@nyu.edu 02wangjiajun@gmail.com ammon134@gmail.com laung6186@oakton.edu

Notes

- Agenda
 - o Team check
 - o Application Demo
- Retro
 - o Challenges with the master password
- Demo
 - o Tests:
 - Test all options without password in the vault
 - Retrieve - OK
 - Update - OK
 - Delete - OK
 - Recent passwords report - OK
 - Improvement: call password_lookup() form the cli (avoid asking for password if the password is not in the vault)

- Test adding a password - OK
 - Test Updating a password - OK
 - Test Deletion - Check if the password confirmation is working - OK
 - Test getting the lists of passwords - OK
 - Test_database.py:
 - Fix: assert result == False
 - Test.vault_cli.py:
 - Need to fix the call to main()?
- Schedule submission meeting:
 - Schedule at 8pm CST

Action items



Dec 1, 2025 | 🗓 Oakton CS Group Assignment - Sprint Review + Sprint Planning

Attendees: Fernando Pereira zh657@nyu.edu Quang Jiajun

Agenda

- Mini sprint retrospective
- Application Sprint Review
- Prioritize items for this week's sprint
- Schedule:
 - Final check point on Friday 12/05 @ 9pm EST (7pm CST)
 - Submission meeting on Monday 12/07 @ 5pm

Discussion

- Mini sprint retrospective:
 - week #1 (16 deliveries!)
 - What could have been better? What went well?
 - AH: Felt good about it.
 - JW: Also feels good about it, having worked on the testing.
- Application Feedback:
 - Demo
 - FP:
 - **cli_input.py**
 - **1) Issue:**
 - Retrieved passwords are being printed with encryption
 - Suggested fix: connect the functions to change and to remove password
 - **2) Current state** - Flow and Menu need to be iterated further:
 - When you open the app, it gives you the ability to add a password, but, then, you are only able to retrieve the password, not able to add additional passwords. However, previous passwords are still available in the file and can be retrieved
 - **Suggested State and Flow:**
 1. Open the app
 2. Get welcome message
 3. If it is the first time using the app, create an account - **Nice to Have**
 1. Ask for a login
 2. Ask for a password
 3. Store those in a file
 4. Next time the user opens the app, he or she needs to add those credentials to access the app menu.
 4. Get a menu with the following options: - **Must Have**
 - a. Store a new password name and password
 - i. calls create_password_name(name, encrypt(pwd))
 - b. Retrieved a stored password name and password
 - i. calls lookup_password_name(name)
 - c. Update a stored password name and password
 - i. calls update_password_name(name, encrypt(pwd))
 - d. Delete a stored password name and password
 - i. calls delete_password_name(name, encrypt(pwd))
 1. delete_password_name() might need some adjustment, because there is a password confirmation step that will compare decrypted password with encrypted password. Data points need to be aligned and compared at the same state.
 - e. Get recent password details
 - i. calls get_most_recent_password_names() - call is OK in cli
 - f. Exit
 - **test_vault.py** - Need to test from cli, simulating user inputs using "from unittest.mock import patch"

- [Example](#)

- Prioritize items for this week's sprint
- Schedule:
 - Final check point on Friday 12/05 @ 5pm
 - Submission meeting on Monday 12/07 @ 5pm

Action items



Nov 25, 2025 | Oakton CS Group Lab

Attendees: Audrey He, Nhat Quang Dang, Fernando Pereira

Agenda

- Intros
- Lay the Groundwork
 - **Communication:** WhatsApp group ([link](#))
 - **Documentation:** This doc will be the source used for the project documentation. ([link](#))
- Define scope

Decision: use Python

Password Vault Web App

- There is a User Interface
- Use cryptography
- Open the terminal, pvault add name for the password (ex: OaktonPass), add to a file
 - Create a pvault.list that shows the passwords, their categories
 - Last password entered
 - To make it more secure, you request a vault password
- Encryption method

- Roles and responsibilities
 - Everybody will code
 - Coding Areas:
 - Database (potential file handling)
 - Data structures
 - User experience interface (option menus)
 - Simulation and Testing (run a unittest.mock patch script, generate statistics)
 - Documentation (complexity analysis)
- Lab pace
 - Set a cadence for Check ins -
 - Suggestion: Async stand ups @ 5pm EST every two days
 - Schedule other meetings
 - Sprint Plan #1 - Today
 - Sprint Review #1 + Sprint Plan #2 - 12/01 @ 9pm EST
 - Sprint Review #2 - 12/05
 - Submission meeting - 12/08 - 8pm EST

Application Testing

How to Run the Tests

Note

Running tests directly inside an IDE may cause the **real interactive menu** to appear. This is expected because the IDE runs the test as a script instead of running it as a module, bypassing the execution of `unittest` and not applying mocks. In order to trigger the automated testing behavior, always run tests from the **terminal** for correct automated behavior.

Run database tests

```
python3 test_database.py
```

Run CLI tests

```
python3 -m unittest test_vault_cli.py -v
```

Testing Overview

To ensure the reliability of both our database layer and CLI interface, we implemented two automated test suites: **database tests** and **CLI tests**.

The database tests (`test_database.py`) validate all core storage operations, including create, retrieve, update, delete, and recent-password reporting. These tests run against a temporary vault file, ensuring that the real application data is never modified. The CLI tests (`test_vault_cli.py`) verify the menu-driven application by simulating user interactions with `unittest.mock.patch`.

Instead of requiring manual input, these tests automatically initiate menu selections, password inputs, and system responses, allowing us to ensure proper behavior for storing, retrieving, updating, deleting, and quitting the program.

These tests provide comprehensive functional coverage of the password vault system, from low-level file operations to high-level user activities. This ensures that our application acts consistently across different execution pathways and guards against regressions as new features are implemented.

Automated Test Results

```
● jiajunwang@Jiajuns-MacBook-Air CSC-255-Oakton-Group-Lab-main % python3 test_database.py
Running database tests...

Password stored successfully for gmail
test_create_password passed

Password stored successfully for github
test_lookup_password passed

Password stored successfully for uiuc

Password for uiuc successfully updated
test_update_password passed

Password stored successfully for gmail

Password for gmail successfully deleted.
test_delete_password passed

Password stored successfully for site1

Password stored successfully for site2

Password stored successfully for site3

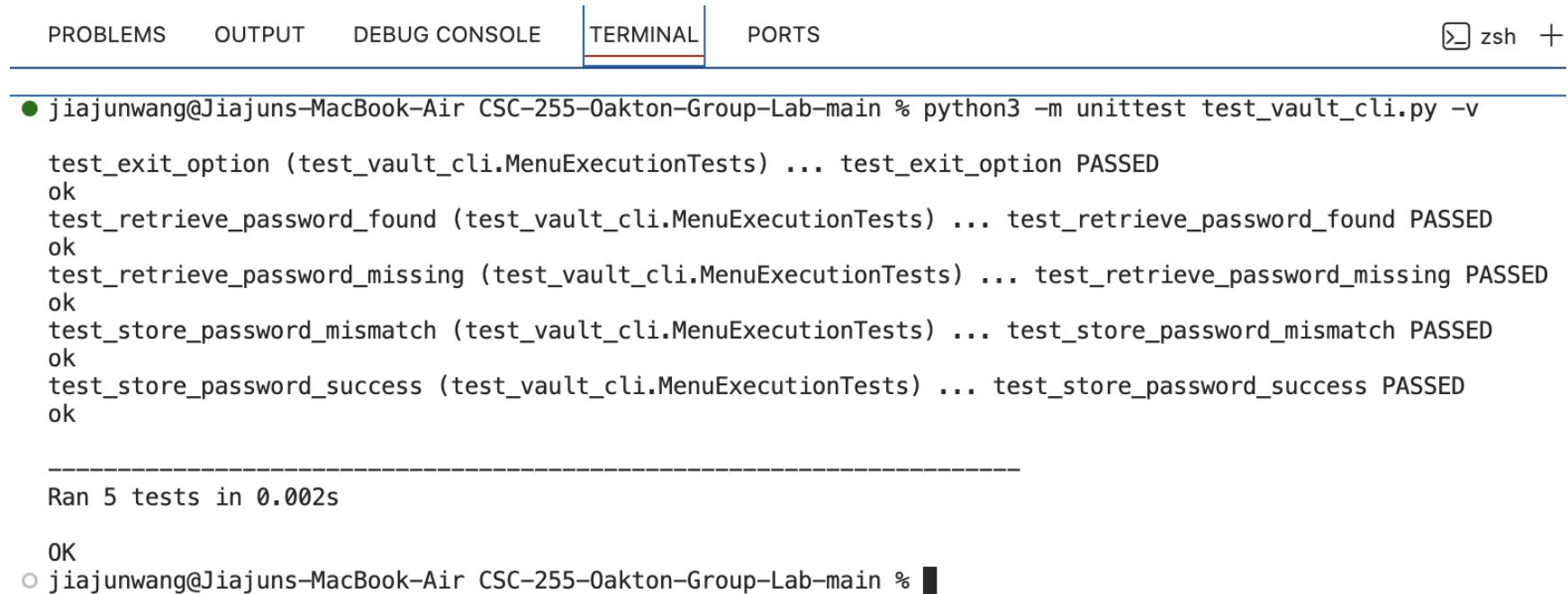
Password stored successfully for site4

Password stored successfully for site5

Password stored successfully for site6

Password stored successfully for site7
Running get_most_recent_password_names() (no crash test)
=====
= PASSWORD NAMES REPORT (ORDERED BY CREATION TIME DESCENDING =
=====
# | Password name      | Most Recent Update
1 | site7             | 2025-12-08 19:17:48.727383
2 | site6             | 2025-12-08 19:17:48.714570
3 | site5             | 2025-12-08 19:17:48.701803
4 | site4             | 2025-12-08 19:17:48.689255
5 | site3             | 2025-12-08 19:17:48.678596
6 | site2             | 2025-12-08 19:17:48.665885
7 | site1             | 2025-12-08 19:17:48.654495
-----
test_top5_most_recent passed

ALL DATABASE TESTS PASSED ✓
```



The screenshot shows a terminal window with the following content:

```
● jiajunwang@Jiajuns-MacBook-Air CSC-255-Oakton-Group-Lab-main % python3 -m unittest test_vault_cli.py -v
test_exit_option (test_vault_cli.MenuExecutionTests) ... test_exit_option PASSED
ok
test_retrieve_password_found (test_vault_cli.MenuExecutionTests) ... test_retrieve_password_found PASSED
ok
test_retrieve_password_missing (test_vault_cli.MenuExecutionTests) ... test_retrieve_password_missing PASSED
ok
test_store_password_mismatch (test_vault_cli.MenuExecutionTests) ... test_store_password_mismatch PASSED
ok
test_store_password_success (test_vault_cli.MenuExecutionTests) ... test_store_password_success PASSED
ok

Ran 5 tests in 0.002s

OK
○ jiajunwang@Jiajuns-MacBook-Air CSC-255-Oakton-Group-Lab-main %
```

The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. There is also a zsh icon and a '+' button.

Manual Testing Results

Test 01: Retrieve, Update, Delete, or get password names report when there is no password

Log Test 01

```
Welcome to Keeper, a simple and secure solution for managing your credentials right from the command line!
All you need to do is to provide website name and it's password.
Keeper will encrypt your password using industry-standard protocols, ensuring your sensitive information remains
private and protected.

=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 2
Please enter the website name: Facebook

Password name Facebook not found. Please, create a new password name for this name to be able to use it.

=====

Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 3
Please enter the website name you want to update: Facebook
Please enter the password (characters will show as *) : *****
Please confirm the password: *****

The password NAME Facebook was not found in order to be updated.

=====

Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 4
Please enter the website name you want to delete: Facebook
Please enter the password (characters will show as *) : *****

The password NAME Facebook does not exist and cannot be deleted. Try another password name.

=====

Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 5

  Your vault is empty!
  Start by creating your first password entry.
```

Test 02: Store, retrieve, update, get report, delete a new Passwords, and exit the app

Log Test 02

Storing a new password:

```
=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 1
Please enter the website name: Facebok
Please enter the password (characters will show as *) : ****
Please confirm the password: ****

Password stored successfully for Facebok
```

```
=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 2
Please enter the website name: Facebok
The password for Facebok is 1234
```

Updating passwords:

```
=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 3
Please enter the website name you want to update: Facebok
Please enter the password (characters will show as *) : *****
Please confirm the password: *****

Password for Facebok succesfully updated
```

```
=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 2
Please enter the website name: Facebok
The password for Facebok is 12345
```

Picture of the password stored in encrypted form:

```
vault.txt
password_name,encrypted_password,created_at,last_updated_at
Facebook,<;h8=,1765243893.4465652,1765244075.6961029
```

Password Names Report:

```
=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 1
Please enter the website name: Google
Please enter the password (characters will show as *) : *****
Please confirm the password: *****
```

```
Password stored successfully for Google
```

```
=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 1
Please enter the website name: Microsoft
Please enter the password (characters will show as *) : *****
Please confirm the password: ****
```

```
Password stored successfully for Microsoft
```

```
=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 5
=====
= PASSWORD NAMES REPORT (ORDERED BY CREATION TIME DESCENDING =
=====
# | Password name          | Most Recent Update |
1 | Microsoft             | 2025-12-08 20:36:51.465912 |
2 | Google                 | 2025-12-08 20:36:31.326650 |
3 | Facebook               | 2025-12-08 20:34:35.696103 |
```

Deleting a password

```
=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 4
Please enter the website name you want to delete: Facebook
Please enter the password (characters will show as *) : *****
Password for Facebook successfully deleted.
```

Exiting the app:

```
=====
Please select from the following menu:
1 - Store a new password
2 - Retrieve a stored password
3 - Update a stored password
4 - Delete a stored password
5 - Get recent password
6 - Exit
Please enter your choice: 6
Thank you for using Keeper! Goodbye!
```

Glossary

DRI - Designated Responsible Individual: the person responsible for completing the task

GitHub tutorials

- [How To Use GitHub For Beginners](#)
- [Master GitHub: From Beginner to Expert in 46 min](#)

Meeting times

(add an "Y" to the table below with times you could meet, or leave a note in this section about your availability)

Date	Time (US Eastern Time)	Fernando	La Min	Quang	Audrey	Jiajun
11/24	Morning					
	Afternoon	Y				
	Evening	Y				
11/24	Morning					
	Afternoon					
	Evening					
11/25	Morning	Y				
	Afternoon	BEFORE 3PM		Y		
	Evening	Y		Y	Y	
11/26	Morning					
	Afternoon					
	Evening					
11/27	Morning				Y	
	Afternoon	Y			Y	

Date	Time (US Eastern Time)	Fernando	La Min	Quang	Audrey	Jiajun
	Evening					
11/28	Morning	Y				Y
	Afternoon	Y			Y	Y
	Evening	Y			Y	Y
11/29	Morning					Y
	Afternoon					Y
	Evening				Y	Y
11/30	Morning					Y
	Afternoon					Y
	Evening				Y	Y
12/01	Morning					Y
	Afternoon	Y				
	Evening	Y				Y
12/02	Morning					Y
	Afternoon					
	Evening					Y
12/03	Morning					Y
	Afternoon					
	Evening	MAYBE				Y
12/04	Morning					Y
	Afternoon					
	Evening					Y
12/05	Morning	Y				Y
	Afternoon	Y				
	Evening	Y				Y
12/06	Morning					Y
	Afternoon					
	Evening					Y
12/07	Morning					Y
	Afternoon					
	Evening					Y
12/08	Morning	Y				Y
Last	Afternoon	Y				Y
Day!	Evening	Y				Y