Password Manager - Version 1

This program was created by Jubeir Morad on December 15, 2024.
We developed it using Python version 3.11, adopting an object-oriented programming approach.

Overview

This program provides a secure way to store passwords through a non-graphical user interface.
Passwords are stored in a database file, offering greater security compared to text files or Excel sheets.

When the application is used for the first time, users are prompted to create a master password.
This master password serves as the key to the program and will be required every time the application is launched. It can be changed later if needed.

To save a password, the user must input a label (e.g., "email") and then the desired password to be saved under this label.
Each password must have a unique label that cannot be repeated.

When the user enters a command to display stored passwords, all saved passwords will appear next to their respective labels.
It is important to note that only one password can be saved under each label.

Important: The master password must be entered after every command, except for help and exit commands.

To view all the available commands, use the help command.
Remember to read the code comments for a better understanding of the implementation.

Components

Database Setup
The database is initialized in the database.py file.
This file contains the code to create the database. **Note:** Running this file directly will result in an error because the database tables already exist.
To recreate the database, delete the existing database file and then rerun this script.

Main Program File
The main program file, mainfile.py, is responsible for running the application.

Libraries: The required libraries are imported at the beginning of the file.

class function
Here, we define the core methods responsible for creating, modifying, and validating passwords in the database.

These methods do not include any print commands and do not take direct input from the user.

class Function

Another Function class includes methods with the same names as those in the previous class but focuses on user interaction.

For instance, examine the methods sinup and SinUp:

- The sinup method adds a password to the database.
- The SinUp method collects input from the user and passes it to the sinup method.
- A third method ensures that passwords meet the minimum security requirements (e.g., they must exceed seven characters).

This approach separates the program logic from user interaction, making the code easier to read, modify, and expand.

You can also add new methods to implement additional features.

Main Class

The main class executes functions based on user inputs. Each command triggers a corresponding function.

Text Coloring

In the colors.py file, a set of functions was created to add colors to the program's output. For example, errors are displayed in red.

Final Notes

Code comments have been added throughout the main file to help you understand its structure and functionality. We hope you find them useful.