CloudVault

Password Manager

Work Done by :

Ricardo Silva
Guillermo Lopéz De Arechavaleta
Saúl Sauca Torremocha

RUN-EU SAP PYTHON 2024

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

Chapter 2

Namespace Documentation

2.1 database Namespace Reference

Functions

- connect_db ()
- create_db ()
- add_password (site, username, encrypted_password)
- update_password (site, username, encrypted_password)
- get_passwords ()
- **search_passwords_by_site_user** (filter_text)
- delete_password (site, username)

2.1.1 Detailed Description

```
@file database.py
@brief Functions for database operations in the Password Manager.

Authors:
- Ricardo Silva
- Guillermo
- Saúl

@date 2024
@version 1.0
@note SAP: PYTHON 2024

This module provides functions to connect to the database, create tables, add, search, and update passwords.
```

2.1.2 Function Documentation

2.1.2.1 add_password()

```
database.add_password (
              site,
              username,
              encrypted_password)
```

```
@brief Adds a new password to the database.

@param site The site associated with the password.
@param username The username associated with the password.
@param encrypted_password The encrypted password.
```

2.1.2.2 connect_db()

```
database.connect_db ()
```

```
@brief Connects to the SQLite database.

@return A connection object or None if connection fails.
```

2.1.2.3 create_db()

```
database.create_db ()
```

```
@brief Creates the passwords table if it does not exist.
```

2.1.2.4 delete_password()

```
database.delete_password (
              site,
              username)
```

```
@brief Deletes an existing password from the database.

@param site The site associated with the password.
@param username The username associated with the password.
```

2.1.2.5 get_passwords()

```
database.get_passwords ()
```

```
@brief Retrieves all passwords from the database.

@return A list of tuples containing all passwords.
```

2.1.2.6 update_password()

```
database.update_password (
              site,
              username,
              encrypted_password)
```

```
@brief Updates an existing password in the database.

@param site The site associated with the password.
@param username The username associated with the password.
@param encrypted_password The new encrypted password.
```

2.2 encryption Namespace Reference

Functions

- generate_key ()
- load_key (key_path="key.key")
- encrypt_password (password)
- decrypt_password (encrypted_password)

2.2.1 Detailed Description

```
@file encryption.py
@brief Functions for encrypting and decrypting passwords using cryptography.fernet.

Authors:
- Ricardo Silva
- Guillermo
- Saúl

@date 2024
@version 1.0
@note SAP: PYTHON 2024

This module provides functions to generate encryption keys, encrypt passwords, and decrypt passwords
using the cryptography.fernet library.
```

2.2.2 Function Documentation

2.2.2.1 decrypt_password()

```
encryption.decrypt_password (
              encrypted_password)
```

```
@brief Decrypts an encrypted password using the loaded encryption key.

@param encrypted_password The encrypted password to decrypt (as bytes).
@return The decrypted password as a string.
```

2.2.2.2 encrypt_password()

```
encryption.encrypt_password (
              password)
```

```
@brief Encrypts a password using the loaded encryption key.

@param password The password to encrypt.
@return The encrypted password as bytes.
```

2.2.2.3 generate_key()

```
encryption.generate_key ()
```

```
@brief Generates an encryption key and saves it to a file named 'key.key'.
```

2.2.2.4 load_key()

```
encryption.load_key (
              key_path = "key.key")
```

```
@brief Loads the encryption key from a specified file.

@param key_path The path to the file containing the encryption key (default is 'key.key').
@return The encryption key loaded from the file.
```

2.3 main Namespace Reference

Functions

- hash_password (password)
- save_master_password (password_hash)
- load_master_password ()
- setup_master_password ()
- authenticate_on_start ()
- generate_custom_password ()
- add_password_ui ()
- show_passwords_in_list ()
- show_filtered_passwords (event=None)
- show_user_and_password (event)
- add_example_passwords ()
- clear_all_data ()

Variables

- str **MASTER_PASSWORD_FILE** = "master_password.txt"
- **stored_master_password** = load_master_password()
- **app** = ctk.CTk()
- **label** = ctk.CTkLabel(app, text="Password Manager")
- **pady**
- **frame_buttons** = ctk.CTkFrame(app)
- **padx**
- **side**
- **LEFT**
- **fill**
- **btn_add_password** = ctk.CTkButton(frame_buttons, text="Add Password", command=add_password_ui)
- **btn_generate_password** = ctk.CTkButton(frame_buttons, text="Generate Password", command=generate_custom_password)
- **btn_add_example_passwords** = ctk.CTkButton(frame_buttons, text="Add Example Passwords", command=add_example_passwords)
- **entry_search** = ctk.CTkEntry(frame_buttons, placeholder_text="Search by site or user")
- **frame_list** = ctk.CTkFrame(app)
- **RIGHT**
- **BOTH**
- **expand**
- **scrollbar** = ctk.CTkScrollbar(frame_list)
- **listbox_passwords** = tk.Listbox(frame_list, yscrollcommand=scrollbar.set, width=30, font=('Arial', 14))
- **command**

2.3.1 Detailed Description

```
@file password_manager.py
@brief Password Manager application for securely storing and managing passwords.

Authors:
- Ricardo Silva
- Guillermo López de Arechavaleta
- Saúl Sauca

@date 2024
@version 1.0
@note SAP: PYTHON 2024

This program allows users to securely store passwords for different websites,
generate custom passwords, and manage them securely using encryption techniques.
```

2.3.2 Function Documentation

2.3.2.1 add_example_passwords()

```
main.add_example_passwords ()
```

```
@brief Adds example passwords to the database for testing purposes.
```

2.3.2.2 add_password_ui()

```
main.add_password_ui ()
```

```
@brief Prompts the user to add a new password to the database.

Encrypts the password and adds it to the database.
```

2.3.2.3 authenticate_on_start()

```
main.authenticate_on_start ()
```

```
@brief Authenticates the user at the start of the program.

Prompts the user to enter the master password and verifies it against the stored hash.
```

2.3.2.4 clear_all_data()

```
main.clear_all_data ()
```

```
@brief Clears all data from the database.
```

2.3.2.5 generate_custom_password()

```
main.generate_custom_password ()
```

```
@brief Generates a custom password based on user preferences.

Allows users to specify password length and select character types (uppercase, numbers, symbols).
```

2.3.2.6 hash_password()

```
main.hash_password (
              password)
```

```
@brief Hashes the given password using SHA-256.

@param password The password to hash.
@return The hashed password as a hexadecimal string.
```

2.3.2.7 load_master_password()

```
main.load_master_password ()
```

```
@brief Loads the hashed master password from a file.

@return The hashed master password as a string, or None if the file doesn't exist.
```

2.3.2.8 save_master_password()

```
main.save_master_password (
              password_hash)
```

```
@brief Saves the hashed master password to a file.

@param password_hash The hashed master password to save.
```

2.3.2.9 setup_master_password()

```
main.setup_master_password ()
```

```
@brief Guides the user through setting up the master password.

@return The hashed master password once successfully set up.
```

2.3.2.10 show_filtered_passwords()

```
main.show_filtered_passwords (
              event = None)
```

```
@brief Filters and displays passwords based on search criteria.

@param event The event (typically a key release in the search entry).
```

2.3.2.11 show_passwords_in_list()

```
main.show_passwords_in_list ()
```

```
@brief Retrieves and displays all passwords in the listbox.
```

2.3.2.12 show_user_and_password()

```
main.show_user_and_password (
              event)
```

```
@brief Displays the username and decrypted password when an item in the listbox is double-clicked.

@param event The event triggered by double-clicking on an item in the listbox.
```

Index