Week 2 Report

Report Date: 26-june-2023

Password manager using Python

By K.Ramya Sree , Y.Sai Rohitha

**Project Overview:**

The password manager Python project aims to create a secure and user-friendly application for managing passwords. The password manager will allow users to store their passwords securely, generate strong passwords, and retrieve passwords when needed.

**Progress Made:**

During the second week of development, we the team of 5 have a significant progress was made in the following areas:

User Interface Design:

* Incorporated necessary input fields, buttons, and labels for password management functionalities. Developed a basic graphical user interface (GUI) using a Python GUI library such as Tkinter or PyQt.
* Implemented a clean and intuitive design to provide a user-friendly experience.

Database Integration:

* Set up a database management system (DBMS) to store and retrieve password records securely.
* Created the necessary database tables and columns to store user credentials.
* Implemented SQL queries to handle database operations, such as inserting, retrieving, and updating passwords.

Password Encryption and Hashing:

* Implemented a strong encryption mechanism to securely store user passwords in the database.
* Utilized cryptographic libraries, such as bcrypt or hashlib, to hash the passwords.
* Ensured that the passwords are securely stored and cannot be easily decrypted.

Password Generation:

* Developed a password generator function that can generate strong and random passwords.
* Allowed users to specify the length and complexity requirements for the generated passwords.
* Integrated the password generator into the user interface for easy accessibility.

Challenges Faced:

During the second week of development, the following challenges were encountered:

GUI Development Complexity:

* Designing an intuitive and aesthetically pleasing user interface required careful consideration of user experience (UX) principles.
* Handling user input validation and error handling for various fields presented a challenge in maintaining a smooth user flow.

Security Considerations:

* Ensuring the highest level of security for password storage demanded rigorous testing and implementation of encryption and hashing algorithms.
* Striking a balance between usability and security when generating strong passwords required thoughtful decision-making.

Goals for Next Week:

The upcoming week will focus on the following tasks:

Password Retrieval:

* Implement functionality to retrieve stored passwords for a selected account or website.
* Display the retrieved passwords securely, adhering to best security practices.

Account Management:

* Develop features to manage user accounts, such as creating new accounts, modifying existing accounts, and deleting accounts.
* Implement necessary input validation and error handling to ensure data integrity.

Testing and Bug Fixes:

* Conduct comprehensive testing to identify and fix any potential bugs or vulnerabilities.
* Gather feedback from testers to improve the user experience and address any issues.

Documentation:

* Create detailed documentation describing the application's features, installation instructions, and usage guidelines.
* Provide clear instructions on how to run the program, set up the database, and navigate the user interface.

**Conclusion:**

The second week of development for the password manager Python project was productive, with significant progress made in GUI design, database integration, password encryption, and password generation. Challenges were encountered but successfully addressed. Next week's focus will be on implementing password retrieval, account management functionalities, conducting testing, and finalizing the documentation. The project is on track, and efforts will be made to ensure a robust and secure password manager for users.

By K.Ramya Sree

Mail : koneniramya@gmail.com