Report-1(week 1)

Password manager using Python

By Sai Harika .P , Poojitha .V , Sai Lavanya .K , Rizwana . D , Revathi.M

Project Overview:

The Password Manager Python project aims to develop a secure and user-friendly application that helps users store and manage their passwords. The project utilizes Python programming language to implement various features, including password encryption, storage, retrieval, and password generation.

Progress Made During Week 1:

During the first week of the project, significant progress has been made in the following areas:

Project Planning:

* Defined the project scope, objectives, and requirements.
* Identified the target user base and their needs.
* Established a timeline and milestones for the project.

User Interface Design:

* Created wireframes and mock-ups for the password manager application.
* Gathered feedback from potential users and made necessary design iterations.
* Finalized the visual layout and user flow for the application.

Database Design:

* Designed the database schema to store user account information securely.
* Identified the required fields, such as username, password, website, and additional notes.
* Determined the appropriate encryption algorithms and methods to store passwords.

Encryption Algorithm Selection:

* Conducted research on various encryption algorithms.
* Selected a suitable algorithm to securely encrypt user passwords.
* Implemented the encryption functionality in Python for testing and validation.

Password Generation:

* Developed a password generation module that creates strong and random passwords.
* Implemented customizable parameters such as length, character types, and complexity.
* Verified the generated passwords' strength and tested their compatibility with various websites.

User Account Creation:

* Implemented a registration module to allow users to create an account.
* Added validation checks for username availability and password strength.
* Stored the encrypted user credentials securely in the database.

Goals for Week 2:

Based on the progress made during Week 1, the following goals have been set for Week 2:

User Authentication:

Implement a login module to authenticate registered users.

Verify the entered credentials against the stored encrypted passwords.

Implement password hashing and salting techniques to enhance security.

Password Storage and Retrieval:

Develop functionality to store user passwords securely in the database.

Implement encryption and decryption methods to protect sensitive information.

Allow users to retrieve their stored passwords when needed.

User Interface Implementation:

Begin implementing the finalized user interface design using Python GUI libraries.

Create the necessary screens and forms for login, registration, and password management.

Ensure a user-friendly and intuitive interface for smooth navigation.

Password Management Features:

Enable users to add, edit, and delete passwords for different accounts.

Implement search and filtering functionalities to facilitate easy password retrieval.

Incorporate additional features, such as password expiration reminders and category-based organization.

Error Handling and Exceptional Cases:

Identify potential error scenarios and implement appropriate error handling mechanisms.

Ensure robustness and graceful handling of exceptions to prevent crashes or data loss.

Conduct thorough testing and debugging to resolve any issues.

Challenges Faced:

During Week 1, a few challenges were encountered:

Selection of Encryption Algorithm:

* User Choosing the appropriate encryption algorithm that balances security and efficiency was challenging.
* Extensive research and testing were conducted to ensure a reliable and secure solution.

Interface Design Iterations:

Gathering feedback and making design iterations was time-consuming.

Balancing aesthetics, functionality, and usability posed certain challenges.