Name: Password Buddy
Github Link: https://github.com/ChowdhuryYeamin/emailEase/tree/main/email_drafter_swift

Secure Password Storage in Swift

This project is an iOS application built using Swift and SwiftUI. The application securely stores and retrieves user login data.

Main Features

1. Secure storage of website URL, email, and password.
2. Verification of a master password before retrieving data.
3. Checking password strength before data storage.
4. A clear and easy-to-use user interface.
5. Displays notifications for user interactions.

Implementation Details

The application is mainly composed of two SwiftUI views: `MainView` and `NextView`.

`MainView`

This is the primary view of the application, and it includes:

- Text fields for website URL, email, and the associated password.
- A "Store Data" button that stores the user's input in a secure and encrypted manner. Before storing data, the button action also checks the strength of the password, and will not store the data if the password is not strong enough.
- A text field for the master password and a "Retrieve Data" button that, when clicked, verifies the master password and then navigates to the `NextView` if the password is correct.

`NextView`

This view displays the stored login data if the correct master password is provided.

Password Strength Checking

This application checks the strength of the password input using a function called `checkPasswordStrength(password: String) -> String`. This function analyses the password based on its length, use of upper and lowercase characters, digits, and special characters. It returns a string representing the strength of the password ("Very Weak", "Weak", "Medium", "Strong", "Very Strong"). The application requires a password strength of "Medium" or stronger to store data.

Notifications

This application uses pop-up notifications to inform the user about the status of their interactions, such as incorrect master password or password strength status.

Security

The application uses the Swift package `KeychainSwift` to securely store and retrieve sensitive user data.

For complete code details, please refer to the code snippets provided in the previous conversation. This is a summary of the main aspects of the project. You might want to expand on some sections depending on your audience's knowledge level and the purpose of the documentation.