Enhanced User Authentication and Management System

Introduction

The User Authentication and Management System is a robust application designed to handle user registration, login, password updates, and account deletion functionalities. It utilizes the SQLite3 database for data storage and management. The system ensures security by hashing passwords before storing them in the database. This document provides an overview of the system's functionality, its implementation details, and how users can interact with it. I used google docs.Code of User Authentication and Management System is provided in https://github.com/bhandariarun/KMC/blob/main/week3/day_02/login_process.py and further explanation in documentation:

Technologies Used

Programming Language: Python
Integrated Development Environments (IDEs): Visual Studio Code, PyCharm
Database: SQLite3
Documentation Tool: Google Docs

Code Repository

The code for the User Authentication and Management System can be found in the following GitHub repository: User Authentication and Management System - GitHub Repository
https://github.com/bhandariarun/KMC/blob/main/week3/day_02/login_process.py

System Components

The system primarily consists of four main functionalities:
Registration
Login
Update Password
Delete Account
Each of these functionalities is explained in detail below.

1. Registration

- Users are prompted to register their account by providing a unique username, a valid email address, and a password.
- The system checks if the username already exists in the database. If it does, registration fails, and an appropriate message is displayed to the user.
- Email addresses are validated to ensure they match the Gmail format.
- Passwords are securely hashed before being stored in the database.
- Upon successful registration, a confirmation message is displayed.

2. Login

- Users can log in to their accounts by entering their username and password.
- The system verifies the entered credentials against the stored data in the database.
- If the credentials match, the user is logged in successfully.
- If the credentials do not match or the user is already logged in, appropriate error messages are displayed.

3. Update Password

- Logged-in users have the option to update their passwords.
- Users are required to provide their username, old password, and new password.
- The system verifies the old password before updating it with the new password in the database.
- Upon successful password update, a confirmation message is displayed.

4. Delete Account

- Users can choose to delete their accounts.
- To delete an account, users must provide their username and password.
- The system validates the provided credentials before deleting the account from the database.
- Upon successful deletion, the user's account is permanently removed from the system.

Conclusion

The User Authentication and Management System provides a secure and user-friendly environment for managing user accounts. By leveraging Python, SQLite3, and robust error handling, the system ensures data integrity and confidentiality. Users can register, login, update their passwords, and delete their accounts with ease, enhancing their overall experience.