Report

Introduction

A Python console program that asks the user to login with username and password, or allows them to create an account.

When creating an account:

- asks user for email, user name, two passwords
- confirms those passwords are identical
- stores that information in some kind of text file or database.

When logging in, validates every time they login against the text file.

Design

The main function controls the user interface – displays messages, captures user input. The complexity of this program does not call for classes.

The functions createUser() and Login() contain most of the logic.

Further validation is performed by the functions checkDetails() and askPassword().

Implementation

The main program captures user input by calling the AskUser() function.

The AskUser() function is as follows:

- A login prompt is displayed. 'LOGIN or CREATE?'
- The user input is read and uppercased.
- This is contained in a loop that only accepts LOGIN or CREATE as values for the user input, or it will repeat the same login prompt.
- The accepted output is returned to the main program.

An IF ELSE statement handles the user input;

- LOGIN calls the Login() function
- CREATE calls the CreateUser() function.

LOGIN

The Login() function contains a WHILE loop, where the only way to break out of it is for the details_check variable to have a value of 'pass'.

- the user is prompted to enter their email, username and password

- The CheckDetails() function is called.

The CheckDetails() function passes in the email, username and password variables.

- The CSV module reader function is called to compare these variables against each row in the CSV file.
- A FOR loop is used to go through the CSV file.
- If a row is found with 3 fields matching all 3 variables, then a ‘pass’ message is returned. Else, a ‘fail’ message is returned to the Login() function.

Back in the Login() function, if the message returned is ‘fail’, a “Details incorrect” message is displayed. Else, the user will break from the WHILE loop and a “Logged in” message is displayed.

CREATE

The CreateUser() function:

- Prompts user to enter their email and username
- The AskPassword() function is then called

The AskPassword() function:

- Prompts the user to enter their password and saves it as a variable
- Asks the user to re-enter their password and saves it as another variable
- If these two variables don’t match, this message is displayed: “Passwords don’t match. Re-enter matching passwords.”
- The above is contained within a WHILE loop. The escape condition is for the two variables to match.
- The password is then returned to the CreateUser() function.

Back in the CreateUser() function:

- The SYS module open function is called in Append Mode to open the CSV
- The CSV writer function creates an object called user_writer
- This writerow() function is applied to the user_writer object to pass in the variables email, username and password, and write it to the CSV file.
- The message “User created ” is displayed.

Evaluation

Features

- Input validation is performed on initial user input, checking login details, and on account creation to check the passwords are correct.
- User info is stored in a CSV file.

Limitations

- It's possible for users to create multiple accounts with the same email and/or username.
- Email can be entered in incorrect format i.e. without @ or a . character
- Password is visible when typing into console
- Passwords in CSV file are not encrypted

These limitations were not addressed as they were outside the scope of the specification.

Improvements

- When creating a user account, check if there is already a user with the same email or username. If so, tell the user that and suggest they choose another value.
- Check email address is in correct format (i.e. has at least 1 @ and 1 . character)
- Hide password when typing in to console.
- Encrypt password in CSV file

Conclusion

This program demonstrates the main functionality outlined in the specification.