

Practice Python Exam: Log-In System

This assignment is entirely optional and designed to give you practice writing code and applying lessons and topics for the Python exam.

The Assignment

This Python exam will involve implementing a *mini log-in system*. You will download the skeleton of the program, then implement the functions and write the unit tests. The design of the program has been set up for you.

In this system, users will be able to sign up, log in, change their password, delete their accounts, and display general statistics about sign-ups and log-ins. The program will initially load a collection of usernames and passwords from a .csv file and store them in a dictionary.

Note: We know that storing plaintext passwords in a file is not a good practice in real life applications, but for the sake of this exercise, we'll keep it simple.

Steps

1. Complete the required functions
 - a. Implement all of the functions defined in the *login_system.py*
 - i. Docstrings have already been provided
 - ii. You can create any number of helper functions (with docstrings)
 - iii. Add *brief* comments to all non-trivial lines of code
 - b. Complete the unit testing in *login_system_tests.py*
 - i. Write at least 3 test cases for each unit testing function
 - c. Run the full suite of unit tests in *login_system_tests_full.py* to evaluate your program
2. Complete the program by calling the required functions in the *main* function
3. Make sure your program and the testing file run without errors!

Evaluation

1. Does your program work as expected?
 - a. Can you log in to the system with one of the usernames and passwords in the .csv file?
 - b. Can you sign-up with a new username and password?

2. Did you implement the functions correctly?
 - a. Does your program successfully load and parse the .csv file and store the user credentials in a dictionary database?
 - b. Do you accurately test for a valid password?
3. Did you write good unit tests?
 - a. Did you write at least 3 test cases for each function?
 - b. Do you test both typical examples and edge cases?
 - c. Does your program pass all of your tests?
4. Coding Style
 - a. Appropriate naming of variables
 - b. Naming of helper functions (with docstrings)
 - c. Clear comments in your code