Algorithm Design

LAB 02 : AUTHENTICATION

Due Saturday at 5:00 PM MST

This week, we will continue our review of the Python language. We will build a simple Python application to read a list of names from a file and then see if a user-provided name is in our list.

Program Description

Write a program to read the contents of the file into a list. The program will then prompt the user for a username and password. Finally, we will tell the user whether the user is authenticated.

Read From a File

The first part of the program will be to read the contents of a file into a list. This data will be in JSON. For example, the contents of the file Lab02.json might be:

```
{
    "username": [
    "King Arthur",
    "Lancelot",
    "Sir Robin, the Not-Quite-So-Brave",
    "Black Knight",
    "Sir Bedivere",
    "Roger the Shrubber",
    "Brother Maynard",
    "Bridgekeeper",
    "French Soldier",
    "Knight of Ni",
    "Dead Collector",
    "Dennis",
    "King of Swamp Castle"
  ],
  "password": [
    "Run away!",
    "She turned me into a newt!",
    "That's enough singing for now",
    "None shall pass",
    "How do you know so much about swallows?",
    "Oh, what sad times are these when passing ruffians can say Ni at will to old ladies.",
    "Armaments, chapter two, verses nine through twenty-one",
    "What... is the air-speed velocity of an unladen swallow?",
    "I fart in your general direction",
    "You must cut down the mightiest tree in the forest... WITH... A HERRING!",
    "Bring out your dead!",
    "Help, I'm being oppressed",
    "Let's not bicker and argue over who killed who"
  ]
}
```

The program will open Lab02.json (which you will have to download onto your computer). If the file does not exist or the program is otherwise unable to open it, then present the user with a user-friendly error:

```
Unable to open file Lab02.json.
```

Covert to List

The next step is to convert the JSON data into a list. This is accomplished in a three-step process:

1. Read the data from the file into a single string.
2. Convert the string into a JSON object.
3. Convert the `username` and `password` components of the JSON object into two lists.

Find the Username and Password

The final part of the exercise is to prompt the user for a username and password. You will then search the `usernames` and `passwords` lists to see if the `username` and `password` are on the lists. If both are present and at the same location in the two lists, then the user is considered authenticated.

Example

The following example is a run-through of the program. First, a demonstration of an incorrect username:

```
Username: John Cheese
Password: None shall pass
You are not authorized to use the system.
```

Next, a demonstration of an incorrect password:

```
Username: Black Knight
Password: Tis but a scratch.
You are not authorized to use the system.
```

Finally, a demonstration of the correct username and password:

```
Username: Black Knight
Password: None shall pass
You are authenticated!
```

Assignment

To submit this assignment, two things are needed: your source code and a demonstration video.

Source Code

As with Lab 01 last week, submit your source code as a file attachment in I-Learn. It must be possible to open this file in Python and execute it without any additional work. At the top of your program, include a comment answering these five questions:

```
# 1. Name:
#      -your name-
# 2. Assignment Name:
#      Lab 02: Authentication
# 3. Assignment Description:
#      -describe what this program is meant to do-
# 4. What was the hardest part? Be as specific as possible.
#      -a paragraph or two about how the assignment went for you-
# 5. How long did it take for you to complete the assignment?
#      -total time in hours including reading the assignment and submitting the program-
```

Question 4 is the reflection part of the assignment. Here, please answer questions such as these:

- Was the syntax of Python the hardest part? If so, what part?
- Was there some aspect of the problem that was particularly difficult to solve?
- Was there an especially difficult bug? If so, how did you resolve it?
- Was there some difficulty with the instructions or any part of the problem definition?

Demonstration Video

As with Lab 01 last week, record a short video demonstrating the execution of your program. As the video is recorded, mention the test case you are covering. The video must be no longer than one minute in length.

Your demonstration video must cover the following test cases:

Case	Username	Password	Result
Incorrect username	John Cheese	None shall pass	Not authenticated
Incorrect password	Black Knight	Tis but a scratch.	Not authenticated
Both incorrect	John Cheese	Tis but a scratch.	Not authenticated

Wrong Index	King Arthur	Bring out your dead!	Not authenticated
Valid Black Knight	Black Knight	None shall pass	Authenticated
Valid King Arthur	King Arthur	Run away!	Authenticated
Valid French Soldier	French Soldier	I fart in your general direction	Authenticated

Assessment

Your grade for this activity will be according to the following rubric:

	Exceptional 100%	Good 90%	Acceptable 70%	Developing 50%	Missing 0%
Code Quality 40%	Perfection! The code is extremely easy to understand and very straightforward	Professional and efficient	A few obvious mistakes were made	Readable	Little effort was spent on style. The code looks thrown together or is missing
Functionality 40%	All the test cases execute perfectly	Everything works but there are minor cosmetic defects	One test case fails to execute as expected	At least one test case works as expected	Code does not run, is missing, or does not resemble a working solution
Reflection 20%	The reflection component of the assignment concisely describes what went well and what went poorly	It is clear that thought went into the reflection component of the assignment	Each reflection question is addressed, but there is no evidence of introspection	At least one reflection question is answered	There is no reflection or the reflection fails to address any of the required points