Assignment T1A3

Lewis Hardie

Presentation Outline

Overview

Walk Through

Overview of code

Structure

Review

Overview

Project features and Structure

Main Features

Database Creation for users and passwords

User login password hashing

Login to account management

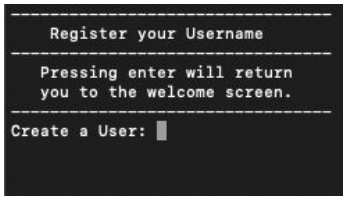
Database creation for user accounts

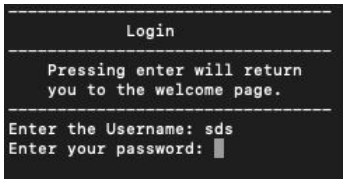
Account Management

Password generation

Features


Welcome
Select what you would like to do:
[1] Register
[2] Login
[3] Exit Application
Enter your choice:


Register your Username
Pressing enter will return
you to the welcome screen.
Create a User:


Login
Pressing enter will return
you to the welcome page.
Enter the Username: sds
Enter your password:

Features

```
---------------------------------
        Account options
---------------------------------
Select from the options below,
by inputing the revelant number.
---------------------------------
---------------------------------

    [1] List Accounts
    [2] Add Accounts
    [3] Get Password
    [4] Remove Account
    [5] Log Out
    [Enter anything else to exit..]
---------------------------------
Enter your selection:
```

```
---------------------------------
        Add Account
---------------------------------
    Enter no value to return
---------------------------------
---------------------------------
What account is this for?
```

```
---------------------------------
        Add Account
---------------------------------
    Enter no value to return
---------------------------------
---------------------------------
What account is this for?
You must enter a Website to continue
What account is this for?
You must enter a Website to continue
What account is this for? sda
What is your username for the account? sadsa
What is the email associated with the account? asdsa.com
Invalid Email
What is the email associated with the account? asd@.com
---------------------------------------------
What is the minimum length of the password? 8
---------------------------------------------
What is the maximum length of the password? 20
---------------------------------
        New account added

---------------------------------
            SDA
---------------------------------
Your Username is : sadsa
Your  Email   is : asd@.com
---------------------------------
Press enter to continue
```

```
---------------------------------
        Get Password
---------------------------------
---------------------------------
    Enter no value to return
---------------------------------
--------------
        [1] SDA
--------------
Which account password would you like to retrieve?: 1
The account 'SDA' password has been copied to your clipboard.
Which account password would you like to retrieve?:
```

```
---------------------------------
        Remove Account
---------------------------------
    Enter no value to return
---------------------------------
---------------------------------
[1] SDA: sadsa, asd@.com
---------------------------------
Enter the number of the account to remove:
```

```
---------------------------------
        Your Accounts
---------------------------------
---------------------------------
      Account List Empty
---------------------------------
Press enter to continue
```

```
-------------------------------
        Log out of account
-------------------------------
Are you sure you want to log out?[y/n]:
```

Walk Through of app

Step by step of the app

Overview of Code

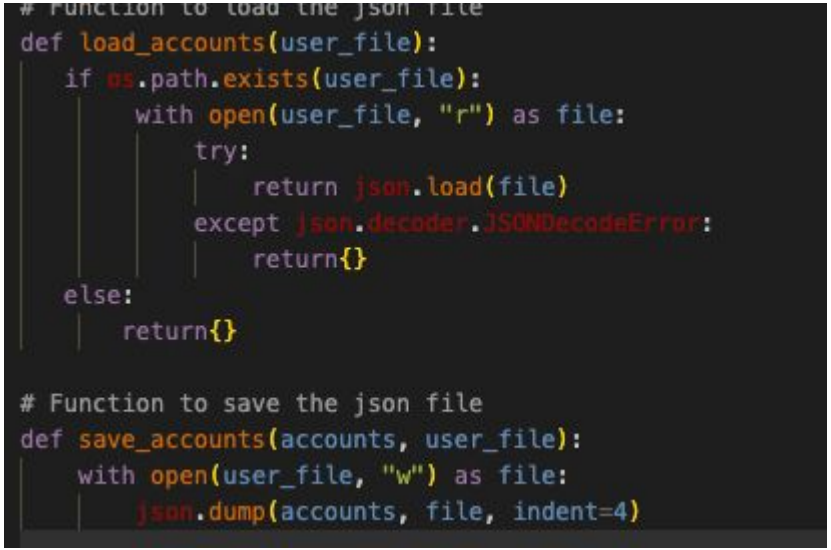
Password Generation

```python
def generate_password():
    # setting variables for the strings that are being imported
    letters = string.ascii_letters
    digits = string.digits
    special_characters = string.punctuation
    characters = letters + digits + special_characters

    # password length
    # Ask the user for the websites specific length requirements guidelines
    # Most websites asking for a minimum of 8, so I'll set it at 10 to ensure a more secure password from the get go
    while True:
        # loop for min password length input, check correct input and error handling
        try:
            print("-----------------------------------------------")
            min_password_length = int(input("What is the minimum length of the password? "))
            # if password doesn't have a value, then return to begining screen
            # if password is 8 or less characters then break loop and start again
            if min_password_length >= 8:
                break
            else:
                # while min password length < 8 print error message and continue loop
                print("""-----------------------------------------------------------------
The minimum password length must be 8 or more characters long!
""")
        # value error for when a user enters a letter or non digit instead of a digit
        except ValueError:
            print("""-----------------------------------
You must enter a number to proceed!
""")
```

```python
    # Ask user for the maximum password length allowed
    while True:
        # loop for max password length input, check correct input and error handling
        try:
            print("-----------------------------------------------")
            max_password_length = int(input("What is the maximum length of the password? "))
            # if max password length is <= the min password length break loop and start again
            if min_password_length < max_password_length:
                break
            else:
                # while min password legnth is > then max password length print error message and continue loop
                print("""-----------------------------------------------------------------
The maximum password must be greater than the minimum password to proceed!
""")
        except ValueError:
            print("""-----------------------------------
You must enter a number to proceed!
""")

    # password length is randomly selected through secrets module in the range from min_password_length to max_password_leng
    password_length = secrets.choice(range(min_password_length, max_password_length + 1))
    # password is set to empty
    password = ""

    while True:
        # for loop to add characters to the password up to the password length
        for i in range(password_length):
            # each character added is randomised through sercrets module
            password += "".join(secrets.choice(characters))
        # check to make sure there is atleast one character of each type in the password
        if (any(c.islower for c in password)
                and any(c.isupper for c in password)
                and sum(c in digits for c in password)
                and any(c in special_characters for c in password)):
            break

    # print(password)
    return password
```

Using the secrets module to arrange characters in a truly random order, and only creating a password if it contains all requires characters

JSON files



```
# Function to load the json file
def load_accounts(user_file):
    if os.path.exists(user_file):
        with open(user_file, "r") as file:
            try:
                return json.load(file)
            except json.decoder.JSONDecodeError:
                return{}
    else:
        return{}

# Function to save the json file
def save_accounts(accounts, user_file):
    with open(user_file, "w") as file:
        json.dump(accounts, file, indent=4)
```

```
    # Store the user and hashed password in the database (JSON file)
    with open(DATABASE_FILE, "r") as db_file:
        data = json.load(db_file)

    data[user] = hashed_password

    with open(DATABASE_FILE, "w") as db_file:
        json.dump(data, db_file, indent=4)

    # Create a user-specific database for the password manager to access
    user_accounts_file = f"{user}_accounts.json"
    with open(user_accounts_file, "w") as user_file:
        json.dump({}, user_file, indent=4)

    clear_terminal()
    print("You've successfully registered")
    print(f"Username: {user}")

def user_exists(user):
    with open(DATABASE_FILE, "r") as db_file:
        data = json.load(db_file)
        return user in data
```

Working with json files,

- the image are the two functions created for loading and saving accounts.
- The image on the rightis used to add users and the passwords to the user database file

Review

It didn’t go so well

What I did well

Hashing with bcrypt

Unique Users through json

Password generation

What I can improve on

Time management

Test based coding

Objects and Classes

Password encryption