FITNESS APP PROJECT

PART 1.

Write a program that allows me to log exercises and store them to a file. I should be able to access logs by date.

- what is included in a log?

> date

> user select [ push day / pull day / legs day ]

> user input (new set: exercises: reps / weights)

> write to file along with DATE ADDED ( import datetime )

- Error checking

> ~~empty groups~~

> missing input

> no numbers in group, reps/weight

> ~~no comma in between group and reps/weight~~

> non strings in type, exercise

- Access logs by date

> access logs

> error message if log does not exist

~~> more efficient search algorithm (works under the presumption that there no is more than one workout on a given day~~

- Edit logs

- Delete logs

PART 2.

Write a program that allows me to create an account (username, password). The password should have requirements and use regex to enforce those conventions. The account information should be stored to a file with the encrypted password. (email password reset?)

- Making an account

> (use regex) username, password, email, and an id (provided automatically)

> should not allow same usernames/emails

> username and encrypted password are stored to a file

key = Fernet.generate\_key()

file = open('encrypt\_key.key', 'wb')

file.write(key)

file.close()

> logging in (what happens when checkInfo returns true?)

- Using an account

> each account should have a unique workout log

> account should be able to calculate calories (using weight, activity, etc.)

~~> display data trends?~~

~~> provide workouts?~~

> logging out

PART 3.

Command line interface for using program

> create account, log in, log out

> log workout, lookup workout

> view/edit profile

> edit log

> help menu, exit

PART 4.

Use the tkinter module to create a GUI for the program.

- gui pages / options (follow draft image)

> home page

> create account

**email requirements:** r”[\w]+[\.\_]?[\w]+[@][a-z]+[\.]+[a-z]{3}$”

@, character after and before @, three letter finish with a period (.com, .edu, etc.)

**Username reqs:** r “^(?=.{6,20}$)(?!.[@.])[\w!#$%^&\*()]+$”

\* username could be 123456 or !!!!!! (no minimum letters/nums/chars) \*

Between 6 – 20 characters,

letters, numbers, and special characters (except for “@” and “.”) are allowed

**Password reqs:**

**8 - 30 characters; numbers, letters, and special characters are allowed**

> log in, log out

~~> forgot password~~

> profile page

> edit profile

> calculate calories

> back button

> log workouts

> view / edit logs

PART 5.

Clean everything up and make it look nicer

> files need to work without hardcoded paths (if someone downloads it on their comp, it should work. However, they would need the relevant modules [cryptography, …] as well as python)

> add back buttons and fix ‘close window’ buttons

> reorganize buttons

> window should initialize in the middle of the user’s screen

> windows should either expand and contract properly or be a fixed size

> new windows should appear where the last one was positioned

> reformat awkward looking text, add ridges, edit colors, edit wording, etc.

**Color Palette:**

Dark green - “#386641"

Lime Green - "#A7C957"

Green - "#6A994E"

Eggshell - "#F2E8CF"

Notes / To Do: (**Last time**: )

- \* write a script to update github \*

- make “workout.py” a class

- add docstrings

- editing a log only edits the first log on a given date

- *During the first time the program runs*:

account\_data file needs to be initialized (delete account\_data and create an account in order to test this)

a “users” folder should be initialized as well (in the current\* directory)

* **files need to work without hard-coded file path, replace each hardcoded instance with a more dynamic method (probably use os / sys modules)**
* **create\_account (GUI) [imports]**
* **view\_profile (GUI) [innit]**
* **log\_in (GUI) [imports]**
* **create\_account (GUI) [imports]**
* **account [importing cryptography, will need to move user info to users folder]** 
  + **check\_info, store\_info, has\_user\_or\_email, innit workout\_file\***