T1A3 - Terminal App

Jonas Manalang

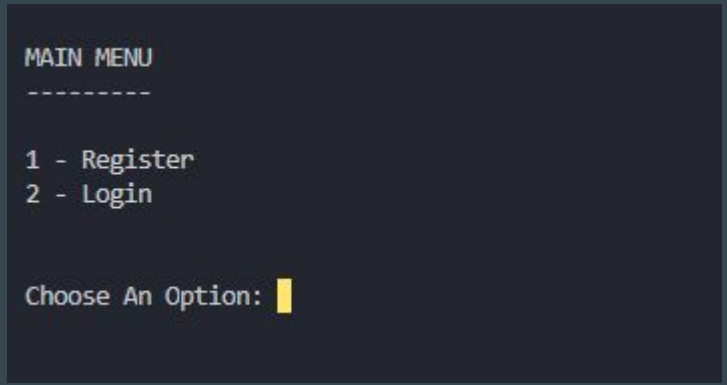
Say Hello to:

Description

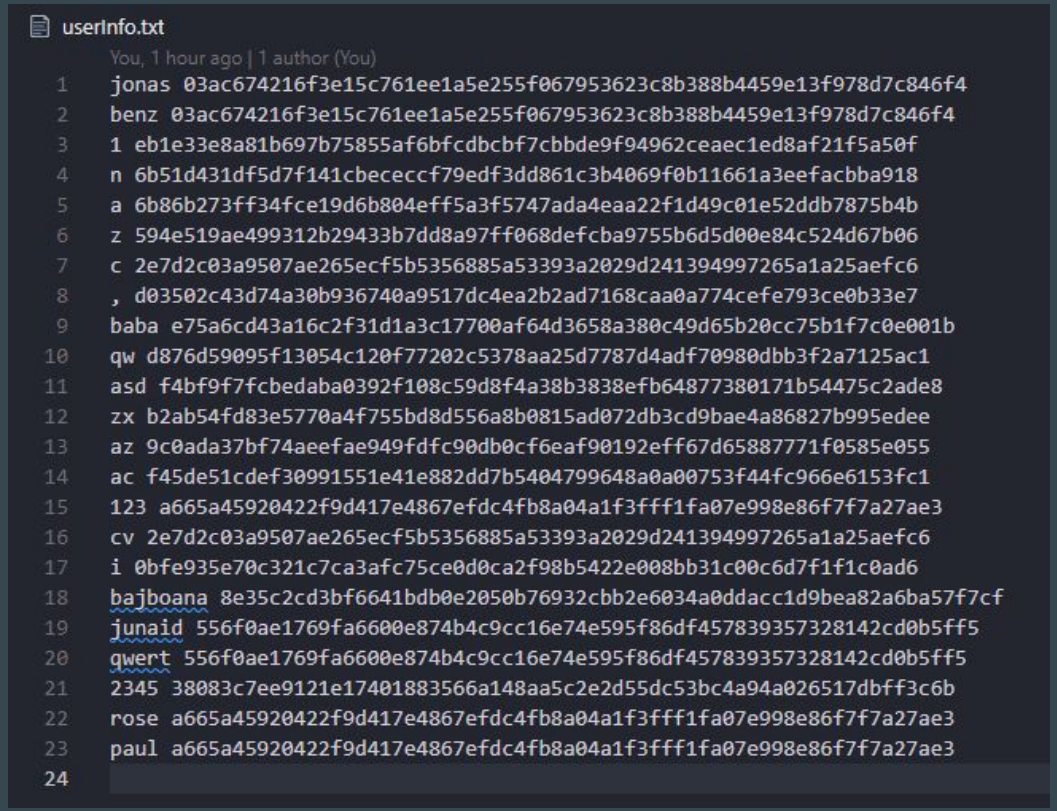
I created a terminal app that checks on the app user's wellbeing and provides the user positive affirmations.

How does it work?

User is prompted to register or existing user can simply log in.



Feature of the app is that user passwords are encrypted.



```
1  from stdiomask import getpass
2  import hashlib
```

Stdiomask is used to secure the password entry into **** and getpass to read the entry.

Import hashlib is used to encryp the data.

```
223  def hashPassword(password):
224      return hashlib.sha256(str.encode(password)).hexdigest()
225
226  def checkPasswordHash(password, hash):
227      return hashPassword(password) == hash
```

```
Registered!
How are you today?
Choose from the following: Happy, Sad, Angry, Depressed, Anxious
Please enter how your are feeling:
```

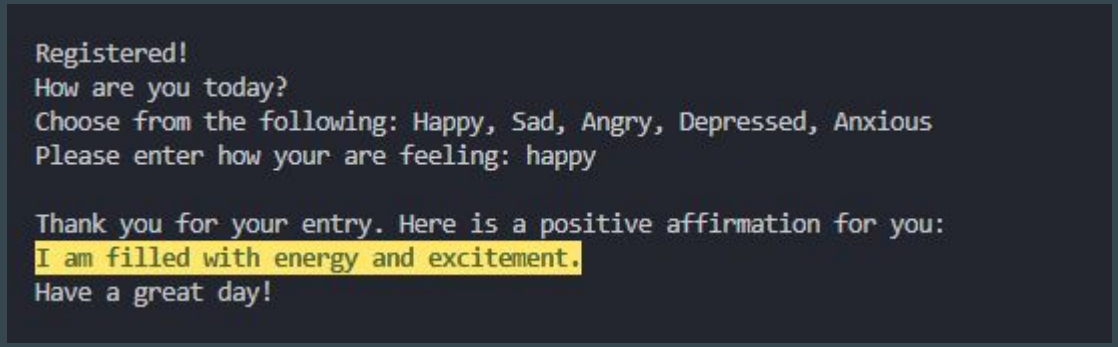
A function is used to get user input and to call on the corresponding positive affirmation from an external file.

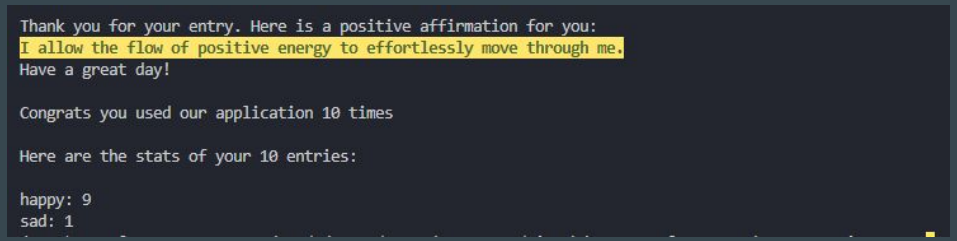
```python
def ask():
    initialQuestion = input("How are you today?\nChoose from the following: Happy, Sad, Angry, Depressed, Anxious\nPlease enter how your are feeling: ")
    sanitizeVariable = initialQuestion.lower().strip()

    if sanitizeVariable == happy or sad or angry or depressed or anxious:
        quote = random.choice(happy or sad or angry or depressed or anxious)
        print("\nThank you for your entry. Here is a positive affirmation for you:")
        print(Style.BRIGHT + Back.YELLOW + Fore.GREEN + f"{quote}")
        print("Have a great day!\n")
        return sanitizeVariable

    else:
        print(Style.BRIGHT + Back.YELLOW + Fore.RED + "**Invalid Entry**")
        print(Style.BRIGHT + Back.YELLOW + Fore.RED + "Please enter a valid response.")
        ask()
```

User mood is captured so that user has insights into their last entry.

```
247 def write_file(user_mood):
248     """function to write the user mood data back to the file"""
249
250     file = open('usermood.txt', 'w') # open text file in write mode
251
252     # iterate through each value in our dictionary
253     for user in user_mood:
254         file.write(user + " " + " ". join(user_mood[user]) + "\n") # write the data in specific format
255
256     file.close()
257
258
259 def read_file():
260     """function to read the previous mood of all the users and add them to the dictionary"""
261
262     file = open('usermood.txt', 'r') # open the file in read mode
263     user_mood = {} # dictionary to store the user mood for their last entries if any
264      # iterate though all the lines in the file
265     for line in file:
266         temp = line.split() # split the lines on the spacing
267         user_mood[temp[0]] = temp[1:] # add the user mood plus their previous entry number in the dictionary
268
269     file.close() # close the file
270     return user_mood # return the data
```

Below screenshot shows the positive affirmation being displayed to the user.





Previous user entries are also collated up to 10 entries to provide the user their stats after every 10th use of Friendster.

```
mood = ask()

count_mood = {} # dictionary to store the mood stats of this specific user

# if the application has been used 10 times, we calculate the stats
if app_used + 1 >= 10:
    print("Congrats you used our application 10 times")
    for prev_mood in user_mood[userName][1:]:
        if prev_mood in count_mood:
            count_mood[prev_mood] += 1
        else:
            count_mood[prev_mood] = 1
    if mood in count_mood:
        count_mood[mood] += 1
    else:
        count_mood[mood] = 1

    print("\nHere are the stats of your 10 entries:\n")
    for myMood in count_mood:
        print(f"{myMood}: {count_mood[myMood]}")


if len(mood_lst) == 0:
    user_mood[userName] = mood_lst
    user_mood[userName].append(str(app_used + 1))
else:
    user_mood[userName][0] = str(app_used + 1)
user_mood[userName].append(mood)

write_file(user_mood)
```