DSA Project **Review-2**

Register no.-> **18BCE0557** Name-> **Kushal**

Progress: Command line interface has been developed and tested thoroughly.

Application of stack ->

- *Usage History:* When a movie is watched then its history is created which is “push” of that particular title and when a history of a client is asked then a “pop” is made from history, therefore it follows “first in last out” order.

Application of queue ->

- *Uploading movie:* When a movie is uploaded by server then it behaves in an “enqueue” manner of queue, where every movie is placed at the top of enqueued one after another. Queue is our database.
- *Available movie:* When the available movies option is used then a “dequeue” is made from database.
- *Movie recommendation:* A client can recommend a movie which is “enqueue” in recommendations and then after when server checks for recommendations those movies are “dequeue”, so it follows “first in first out” order.

Link to GitHub repository: https://github.com/D-E-F-E-A-T/D-Netflix

Code:

---

d_netflix.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-

import start_file

start_file.greet()
start_file.side()
```

---

start_file.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-

import server_file
import client_file

def greet():
    print("Hi! This is D-Netflix, A simple application based on Stack and queue.")
    print("New user may refer to User_guide firstly.\n\n")

def side():
    print("Are you a Client or Server?")
    print("Press 'C' or 'c' for client and hit enter.")
    print("Press 'S' or 's' for server and hit enter.")
    side_var = input()
    print()
    if side_var in ('S', 's'):
        server_file.start()
        # Server will be started
    elif side_var in ('C', 'c'):
        client_file.start()
        # Client will be started
    else:
        print("Invalid choice\n")
        side()
```

Client_file.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-

import webbrowser
#import os
import server_file
import start_file
import time
import list_extractor as liex
from collections import Counter

global server_id
global server_pass
server_id = "dsap"
server_pass = "92det"
global collective
collective = "collective.txt"

def login():
    global client_id
    global client_pass
    client_id = input("Give your id: ")
    client_pass = input("Give your pass: ")

    try:
        client_access = open("clients/"+client_id+'_'+client_pass+'.txt', "r")
        print("Welcome "+client_id+".\n")
        client_access.close()
        return True
```

```
    except:
        return False


def signup():
    global client_id
    global client_pass
    client_id = input("Give your id: ")
    client_pass = input("Create a password for id: ")
    try:
        exsist = open("clients/"+client_id+'_'+client_pass+'.txt', "r")
        if exsist:
            print("Sorry, this username is already registered.")
            print()
            exsist.close()
            signup()

    except:
        client_sign = open("clients/"+client_id+'_'+client_pass+'.txt', "w")
        client_sign.close()
        print("Your database has been created successfully.")
        print()


def search():
    m_name = input('Give name of film: ')
    file = open(server_id+'_'+server_pass+'.txt', "r")
    file_text = file.readlines()
    file.close()
    file_r_text = liex.cleaner(file_text)
    for x in file_r_text:
```

```
        if x[1]==m_name:
            print("Yes, we have "+m_name)
            break
    else:
        print("Sorry, We don't have this film.")
    print()

def watch():
    title = input("Give title of movie to watch: ")

    file = open(server_id+'_'+server_pass+'.txt', "r")
    file_text = file.readlines()
    file.close()
    file_r_text = liex.cleaner(file_text)
    for line in file_r_text:
        if line[1]==title:
            print("Now watching "+title+' at '+time.ctime())
            file = open("clients/"+client_id+'_'+client_pass+'.txt', "a+")
            file.write(time.ctime()+'\t '+title+'\n')
            collect = open(collective, "a+")
            collect.write(time.ctime()+'\t '+title+'\n')
            collect.close()
            file.close()
            webbrowser.open(line[0])
            print()
            break
    else:
        print("This film is currently unavilable.\n")
```

```
def recommend(): # queue-> When server see it, it falls in first in first out
    file = open("recommend.txt", "a+")
    file.write(input("Suggest the film name: ")+'\n')
    file.close()
    print("We will act in shortly!\n")

def history(): # first in last out
    file = open("clients/"+client_id+'_'+client_pass+'.txt', "r")
    file_text = file.readlines()
    file.close()
    file_text.reverse()
    print("DateTime                \tFilm:::")
    for line in file_text:
        print(line, end='')
    print()

def available_movies(): #first in first out
    file = open(server_id+'_'+server_pass+'.txt', "r")
    file_text = file.readlines()
    file.close()
    file_r_text = liex.cleaner(file_text)
    print("Following "+str(len(file_r_text))+" movies we have -->")
    count = 1
    for line in file_r_text:
        print(str(count)+'.)\t'+line[1]+'\t'+'Genre: '+line[2])
        count += 1
    print()
```

```
def watch_frequency():
    file = open("clients/"+client_id+'_'+client_pass+'.txt', "r")
    file_text = file.readlines()
    file.close()
    text = [x.split('\t') for x in file_text]
    frequent = [x[1].replace('\n', '') for x in text]
    print("Frequency\t Film:::")
    for x in Counter(frequent):
        print(str(Counter(frequent)[x])+'\t\t'+x)
    print()


def thorough_description(c_id="client_id", c_pass="client_pass"):
    global client_id
    global client_pass
    if __name__ == "__main__":
        pass
    else:
        client_id = c_id
        client_pass = c_pass
    print()
    print("Client id: "+client_id+"\t\t"+"Client Password: "+client_pass+'\t\t'+"DateTime: "+time.ctime())
    print()
    f = open("clients/"+client_id+'_'+client_pass+'.txt', "r")
    ff = f.readlines()
    f.close()
    print("Total "+str(len(ff))+" films have been watched so far.")
    text = [x.split('\t') for x in ff]
    frequent = [x[1].replace('\n', '') for x in text]
    counter = Counter(frequent).most_common(3)
```

```
    print()
    print(client_id+" favourate films:")
    for x in counter:
        print(x[0])
    print("\n")

def movie_stats():
    print("Key:\tOperation:")
    print("1\tTo see available films.")
    print("2\tTo see movies watch frequency.") # Will develope in last
    print("3\tTo see a thorough description.") # Will add once I add date, genre etc in database.
    print("4\tTo search for a film.")
    choice = input()
    print()
    if choice == '1' :
        available_movies()
    elif choice == '2' :
        watch_frequency()
    elif choice == '3':
        thorough_description()
    elif choice== '4':
        search()
    else:
        print("Invalid choice!")

def choice():
    choice = input("Enter your choice: ")
    print()
    while choice not in ('1', '2', '3', '4', '5', '6'):
```

```
        print("Invalid choice.")
        print() #formating is done
        choice = input("Enter your choice again: ")
        if choice in ('1', '2', '3', '4', '5', '6'): # some lazy issue I had here, so this.
            print()
            break

    if choice == '1':
        watch()
    elif choice == '2':
        movie_stats()
    elif choice == '3':
        recommend()
    elif choice == '4':
        history()
    elif choice == '5':
        print("Closing database")
        print("Exiting server.")
        print()
        start_file.side()
    elif choice == '6':
        print("Exiting D-Netflix")
        exit()

global count
count = 0
def start():
    global count
    count += 1
```

```
print("New users press 'yes' or 'YES' or '1'")
print("Existed users may press Enter.")
user_type = input()
if user_type in ('yes', 'YES', '1'):
    signup()
    login_stat = True
else:
    login_stat = login()

if login_stat:
    print("Key:\tOperation:")
    print("1\tTo Watch a film.")
    print("2\tTo See film stats.")
    print("3\tRecommend a film.")
    print("4\tTo view your watch history") #stack
    print("5\tTo exit the client.")
    print("6\tTo exit D-Netflix.")
    print()
else:
    print("Invalid login credentials, please try again!\n")
    if count%2==0:
        change = input("Do you want to get back? Press 1 or 'yes', 'YES' if yes. ")
        print()
        if change in ('1', 'yes', 'YES'):
            print()
            start_file.side()
    start()

while True:
```

```
        choice()



if __name__ == "__main__":
    start()
```

Server_file.py

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-

import start_file
import list_extractor as liex
import datetime as dt
import client_file
import os
from collections import Counter

global collective
collective = "collective.txt"

def login():
    global server_id
    global server_pass
    server_id = input("Give server id: ")
    server_pass = input("Give server pass: ")

    try:
        database = open(server_id+'_'+server_pass+'.txt', "r")
```

```
        print("Database access granted.\n")
        database.close()
        return True
    except:
        return False


def choice():
    choice = input("\nEnter your choice: ")
    print()
    while choice not in ('1', '2', '3', '4', '5'):
        print("Invalid choice.")
        print()
        choice = input("Enter your choice again: ")
        if choice in ('1', '2', '3', '4', '5'):
            print()
            break


    if choice == '1':
        upload()
    elif choice == '2':
        remove()
    elif choice == '3':
        requests()
    elif choice == '4':
        print("Closing database")
        #database.close()
        print("Exiting server.")
        print()
        # break from here and take to start again
```

```
        start_file.side()
    elif choice == '5':
        #database.close()
        print("Exiting D-Netflix")
        exit()

def upload():
    global server_id
    global server_pass
    link = input("Give Link of film: ")
    title = input("Give Title of film: ")
    genre = input("Give Genre of film: ")
    print("Movie has been uploaded successfully.")
    print()
    database = open(server_id+'_'+server_pass+'.txt', "a+")
    td = dt.datetime.now()
    database.write(str([link, title, genre, td.year, td.month, td.day, td.hour, td.minute, td.second])+'\n')
    database.close()

def remove():
    title = input("Give title of movie to be deleted: ")
    database = open(server_id+'_'+server_pass+'.txt', "r")
    lines = database.readlines()
    o_lines = lines # lines is not a nested list so no trouble
    database.close()

    database = open(server_id+'_'+server_pass+'.txt', "w")
    lines = liex.cleaner(lines)
    #print(lines)
```

```
    for x in range(len(lines)):
        #print(title, line[1])
        if title!=lines[x][1]:
            database.write(str(o_lines[x]))
        else:
            print(title+" is successfully removed.\n")
    database.close()


def client_info():
    files = []
    for file in os.listdir("clients/"):
        if file.endswith(".txt"):
            files.append(file)
    for file in files:
        client = file.split("_")
        client_id = client[0]
        client_pass = client[1].split('.txt')[0]
        client_file.thorough_description(client_id, client_pass)


def available_movies():
    file = open(server_id+'_'+server_pass+'.txt', "r")
    file_text = file.readlines()
    file.close()
    file_r_text = liex.cleaner(file_text)
    print("Following "+str(len(file_r_text))+" movies we have -->")
    count = 1
    for line in file_r_text:
        print(str(count)+'.)\t'+line[1]+'\t\t'+'Genre: '+line[2])
        count += 1
```

```
    print()



def watch_frequency():
    file = open(collective, "r")
    file_text = file.readlines()
    file.close()
    text = [x.split('\t') for x in file_text]
    frequent = [x[1].replace('\n', '') for x in text]
    print("Frequency\t Film:::")
    for x in Counter(frequent):
        print(str(Counter(frequent)[x])+'\t\t'+x)
    print()

def thorough_description():
    f = open("collective.txt", "r")
    ff = f.readlines()
    f.close()
    print("Total "+str(len(ff))+" films have been played so far.")
    text = [x.split('\t') for x in ff]
    frequent = [x[1].replace('\n', '') for x in text]
    counter = Counter(frequent).most_common(5)
    print()
    print("5 Most popular films:")
    for x in counter:
        print(x[0])
    print("\n")

def requests():
```

```
        print("To get client info--> press 'CI' or 'ci' and hit enter.")
        print("To get Movie stats--> press 'MS' or 'ms' and hit enter.")
        choice = input()
        if choice in ('CI', 'ci'):
            client_info()
        elif choice in ('MS', 'ms'):
            movie_stats()
        else:
            print("Invalid choice!")
            print("Taking back to main server menu.")



def movie_stats():
    print("Key:\tOperation:")
    print("1\tTo see available movies.")
    print("2\tTo see movies watch frequency.") # Will develope in last
    print("3\tTo see a thorough description.") # Will add once I add date, gener etc in database.
    choice = input()
    if choice == '1' :
        available_movies() # queue-> first in first out
    elif choice == '2' :
        watch_frequency()
    elif choice == '3':
        thorough_description()
    else:
        print("Invalid choice!")

global count
count = 0
```

```
def start():
    global count
    login_stat = login()
    count += 1
    if login_stat:
        print("Key:\tOperation:")
        print("1\tTo upload a movie.")
        print("2\tTo remove a movie.")
        print("3\tTo make special requests.")
        print("4\tTo exit the server.")
        print("5\tTo exit D-Netflix.")
    else:
        print("Invalid login credentials, please try again!\n")
        if count%2==0:
            change = input("Do you want to get back? Press 1 or 'yes', 'YES' if yes. ")
            if change in ('1', 'yes', 'YES'):
                print()
                start_file.side()
        start()

    while True:
        choice()



if __name__ == "__main__":
    start()
```

List_extractor.py

```
#! /usr/bin/env python

# -*- coding: utf-8 -*-


"""Firstly readlines from file and store it in a list/tuple simply pass the list/tuple to the below cleaner function and it returns clean_data, which is a nested list of your data."""

# Your data should be like below only, or else modification maybe needed.

# data = ["['https://www.youtube.com/watch?v=I0TinhhUwKQ', 'kush']\n",
"['https://www.youtube.com/watch?v=I0TinhhUwKQ', 'unknown']\n"]


def cleaner(data):
    clean1 = []
    clean2 = []
    clean3 = []
    for x in data:
        clean1.append(x[1:len(x)-2])
    for x in clean1:
        clean2.append(x.split(', '))
    for x in clean2:
        clean = [x[0][1:len(x[0])-1], x[1][1:len(x[1])-1], x[2][1:len(x[2])-1]]
        clean3.append(clean)
    return clean3


if __name__ == "__main__":
    file_name = input("Give file name to be parsed, with proper format: ")
    file_open = open(file_name, "r")
    data = file_open.readlines()
    clean_data = cleaner(data)
    print(clean_data)
*******************************************************************************
```