import requests

from bs4 import BeautifulSoup

from time import time

import urllib3

urllib3.disable\_warnings(urllib3.exceptions.InsecureRequestWarning)

CHECK\_TIME = time()

def create\_response(exit\_code, message, runtime):

return {

"exitcode": exit\_code,

"message": message,

"runtime": runtime

}

def get\_csrf\_token(session, login\_url):

response = session.get(login\_url, verify=False)

soup = BeautifulSoup(response.text, 'html.parser')

csrf\_token = soup.find('input', {'name': 'csrfmiddlewaretoken'})['value']

return csrf\_token

def login(session, login\_url, username, password, csrf\_token):

payload = {

'username': username,

'password': password,

'csrfmiddlewaretoken': csrf\_token

}

headers = {

'Referer': login\_url,

'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'

}

return session.post(login\_url, data=payload, headers=headers, verify=False)

def check\_login():

runtime = str(int(time() - CHECK\_TIME))

username = "#"

password = "#"

if not username or not password:

return create\_response(2, "Missing username or password in .env file", runtime)

login\_url = '#'

dashboard\_url = '#'

with requests.Session() as session:

try:

csrf\_token = get\_csrf\_token(session, login\_url)

login\_response = login(session, login\_url, username, password, csrf\_token)

except Exception as e:

return create\_response(2, f"Error during login: {str(e)}", runtime)

response = session.get(dashboard\_url, verify=False)

if "Log out" in response.text:

return create\_response(0, "Login successful", runtime)

else:

return create\_response(1, "Login failed", runtime)

if \_\_name\_\_ == "\_\_main\_\_":

result = check\_login()

print(result)