Proiect program Internship AROBS.

Topuzaru Catalin Andrei

1.INTRODUCERE:

Proiectul trebuie sa navigheze o pagina web pe youtube si sa redea un videoclip, inregistrand ecranul video si audio timp de 2 minute, dupa aceasta inregistrare, programul trebuie sa scrie intr-un fisier nivelul de decibeli din inregistrarea audio.

Sistemul pe care a fost creat acest program este:

* Laptop cu sistem operare: Windows 11;
* Compilator: Pycharm community edition;
* Python v3.12;

2. LIBRARII:

Librariile utilizate au fost cele din cerinta: Selenium, Pyaudio, Pyautogui, Numpy, Opencv;

Din nefericire am avut o eroare la instalarea librariei pyaudio, din cauza faptului ca nu aveam instalat Microsoft Visual C++ Build Tools, insa din nefericire asta nu mi a rezolvat problema, doarece pyaudio nu functioneaza cu versiunea de python 3.13 pe care o aveam la inceput, de acea am creat un mediu virtual care foloseste python 3.12.

Logul pentru instalarea acestora poate fi gasit mai jos:

(.venv) PS D:\proiect Arobs\Proiect NOU\AROBS\_Project> py --version

Python 3.12.0

(.venv) PS D:\proiect Arobs\Proiect NOU\AROBS\_Project> pip install selenium

Collecting selenium

Obtaining dependency information for selenium from https://files.pythonhosted.org/packages/91/08/10cff8463b3510b78f9e3dcef6b37c542b06d71ed1240a8940ba0c75d3bc/selenium-4.26.1-py3-none-any.whl.metadata

Using cached selenium-4.26.1-py3-none-any.whl.metadata (7.1 kB)

Collecting urllib3[socks]<3,>=1.26 (from selenium)

Obtaining dependency information for urllib3[socks]<3,>=1.26 from https://files.pythonhosted.org/packages/ce/d9/5f4c13cecde62396b0d3fe530a50ccea91e7dfc1ccf0e09c228841bb5ba8/urllib3-2.2.3-py3-none-any.whl.metadata

Using cached urllib3-2.2.3-py3-none-any.whl.metadata (6.5 kB)

Collecting trio~=0.17 (from selenium)

Obtaining dependency information for trio~=0.17 from https://files.pythonhosted.org/packages/3c/83/ec3196c360afffbc5b342ead48d1eb7393dd74fa70bca75d33905a86f211/trio-0.27.0-py3-none-any.whl.metadata

Using cached trio-0.27.0-py3-none-any.whl.metadata (8.6 kB)

Collecting trio-websocket~=0.9 (from selenium)

Obtaining dependency information for trio-websocket~=0.9 from https://files.pythonhosted.org/packages/48/be/a9ae5f50cad5b6f85bd2574c2c923730098530096e170c1ce7452394d7aa/trio\_websocket-0.11.1-py3-none-any.whl.metadata

Using cached trio\_websocket-0.11.1-py3-none-any.whl.metadata (4.7 kB)

Collecting certifi>=2021.10.8 (from selenium)

Obtaining dependency information for certifi>=2021.10.8 from https://files.pythonhosted.org/packages/12/90/3c9ff0512038035f59d279fddeb79f5f1eccd8859f06d6163c58798b9487/certifi-2024.8.30-py3-none-any.whl.metadata

Using cached certifi-2024.8.30-py3-none-any.whl.metadata (2.2 kB)

Collecting typing\_extensions~=4.9 (from selenium)

Obtaining dependency information for typing\_extensions~=4.9 from https://files.pythonhosted.org/packages/26/9f/ad63fc0248c5379346306f8668cda6e2e2e9c95e01216d2b8ffd9ff037d0/typing\_extensions-4.12.2-py3-none-any.whl.metadata

Using cached typing\_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)

Collecting websocket-client~=1.8 (from selenium)

Obtaining dependency information for websocket-client~=1.8 from https://files.pythonhosted.org/packages/5a/84/44687a29792a70e111c5c477230a72c4b957d88d16141199bf9acb7537a3/websocket\_client-1.8.0-py3-none-any.whl.metadata

Using cached websocket\_client-1.8.0-py3-none-any.whl.metadata (8.0 kB)

Collecting attrs>=23.2.0 (from trio~=0.17->selenium)

Obtaining dependency information for attrs>=23.2.0 from https://files.pythonhosted.org/packages/6a/21/5b6702a7f963e95456c0de2d495f67bf5fd62840ac655dc451586d23d39a/attrs-24.2.0-py3-none-any.whl.metadata

Using cached attrs-24.2.0-py3-none-any.whl.metadata (11 kB)

Collecting sortedcontainers (from trio~=0.17->selenium)

Obtaining dependency information for sortedcontainers from https://files.pythonhosted.org/packages/32/46/9cb0e58b2deb7f82b84065f37f3bffeb12413f947f9388e4cac22c4621ce/sortedcontainers-2.4.0-py2.py3-none-any.whl.metadata

Using cached sortedcontainers-2.4.0-py2.py3-none-any.whl.metadata (10 kB)

Collecting idna (from trio~=0.17->selenium)

Obtaining dependency information for idna from https://files.pythonhosted.org/packages/76/c6/c88e154df9c4e1a2a66ccf0005a88dfb2650c1dffb6f5ce603dfbd452ce3/idna-3.10-py3-none-any.whl.metadata

Using cached idna-3.10-py3-none-any.whl.metadata (10 kB)

Collecting outcome (from trio~=0.17->selenium)

Obtaining dependency information for outcome from https://files.pythonhosted.org/packages/55/8b/5ab7257531a5d830fc8000c476e63c935488d74609b50f9384a643ec0a62/outcome-1.3.0.post0-py2.py3-none-any.whl.metadata

Using cached outcome-1.3.0.post0-py2.py3-none-any.whl.metadata (2.6 kB)

Collecting sniffio>=1.3.0 (from trio~=0.17->selenium)

Obtaining dependency information for sniffio>=1.3.0 from https://files.pythonhosted.org/packages/e9/44/75a9c9421471a6c4805dbf2356f7c181a29c1879239abab1ea2cc8f38b40/sniffio-1.3.1-py3-none-any.whl.metadata

Using cached sniffio-1.3.1-py3-none-any.whl.metadata (3.9 kB)

Collecting cffi>=1.14 (from trio~=0.17->selenium)

Obtaining dependency information for cffi>=1.14 from https://files.pythonhosted.org/packages/50/b9/db34c4755a7bd1cb2d1603ac3863f22bcecbd1ba29e5ee841a4bc510b294/cffi-1.17.1-cp312-cp312-win\_amd64.whl.metadata

Downloading cffi-1.17.1-cp312-cp312-win\_amd64.whl.metadata (1.6 kB)

Collecting wsproto>=0.14 (from trio-websocket~=0.9->selenium)

Obtaining dependency information for wsproto>=0.14 from https://files.pythonhosted.org/packages/78/58/e860788190eba3bcce367f74d29c4675466ce8dddfba85f7827588416f01/wsproto-1.2.0-py3-none-any.whl.metadata

Using cached wsproto-1.2.0-py3-none-any.whl.metadata (5.6 kB)

Collecting pysocks!=1.5.7,<2.0,>=1.5.6 (from urllib3[socks]<3,>=1.26->selenium)

Obtaining dependency information for pysocks!=1.5.7,<2.0,>=1.5.6 from https://files.pythonhosted.org/packages/8d/59/b4572118e098ac8e46e399a1dd0f2d85403ce8bbaad9ec79373ed6badaf9/PySocks-1.7.1-py3-none-any.whl.metadata

Using cached PySocks-1.7.1-py3-none-any.whl.metadata (13 kB)

Collecting pycparser (from cffi>=1.14->trio~=0.17->selenium)

Obtaining dependency information for pycparser from https://files.pythonhosted.org/packages/13/a3/a812df4e2dd5696d1f351d58b8fe16a405b234ad2886a0dab9183fb78109/pycparser-2.22-py3-none-any.whl.metadata

Using cached pycparser-2.22-py3-none-any.whl.metadata (943 bytes)

Collecting h11<1,>=0.9.0 (from wsproto>=0.14->trio-websocket~=0.9->selenium)

Obtaining dependency information for h11<1,>=0.9.0 from https://files.pythonhosted.org/packages/95/04/ff642e65ad6b90db43e668d70ffb6736436c7ce41fcc549f4e9472234127/h11-0.14.0-py3-none-any.whl.metadata

Using cached h11-0.14.0-py3-none-any.whl.metadata (8.2 kB)

Using cached selenium-4.26.1-py3-none-any.whl (9.7 MB)

Using cached certifi-2024.8.30-py3-none-any.whl (167 kB)

Using cached trio-0.27.0-py3-none-any.whl (481 kB)

Using cached trio\_websocket-0.11.1-py3-none-any.whl (17 kB)

Using cached typing\_extensions-4.12.2-py3-none-any.whl (37 kB)

Using cached websocket\_client-1.8.0-py3-none-any.whl (58 kB)

Using cached attrs-24.2.0-py3-none-any.whl (63 kB)

Downloading cffi-1.17.1-cp312-cp312-win\_amd64.whl (181 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 182.0/182.0 kB 998.5 kB/s eta 0:00:00

Using cached PySocks-1.7.1-py3-none-any.whl (16 kB)

Using cached sniffio-1.3.1-py3-none-any.whl (10 kB)

Using cached wsproto-1.2.0-py3-none-any.whl (24 kB)

Using cached idna-3.10-py3-none-any.whl (70 kB)

Using cached outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)

Using cached sortedcontainers-2.4.0-py2.py3-none-any.whl (29 kB)

Using cached urllib3-2.2.3-py3-none-any.whl (126 kB)

Using cached h11-0.14.0-py3-none-any.whl (58 kB)

Using cached pycparser-2.22-py3-none-any.whl (117 kB)

Installing collected packages: sortedcontainers, websocket-client, urllib3, typing\_extensions, sniffio, pysocks, pycparser, idna, h11, certifi, attrs, wsproto, outcome, cffi, trio, trio-websocket, selenium

Successfully installed attrs-24.2.0 certifi-2024.8.30 cffi-1.17.1 h11-0.14.0 idna-3.10 outcome-1.3.0.post0 pycparser-2.22 pysocks-1.7.1 selenium-4.26.1 sniffio-1.3.1 sortedcontainers-2.4.0 trio-0.27.0 trio-websocket-0.11.1 typing\_extensions-4.12.2 urllib3-2.2.3 websocket-client-1.8.0 wsproto-1.2.0

[notice] A new release of pip is available: 23.2.1 -> 24.3.1

[notice] To update, run: python.exe -m pip install --upgrade pip

(.venv) PS D:\proiect Arobs\Proiect NOU\AROBS\_Project> pip install webdriver-manager

Collecting webdriver-manager

Obtaining dependency information for webdriver-manager from https://files.pythonhosted.org/packages/b5/b5/3bd0b038d80950ec13e6a2c8d03ed8354867dc60064b172f2f4ffac8afbe/webdriver\_manager-4.0.2-py2.py3-none-any.whl.metadata

Using cached webdriver\_manager-4.0.2-py2.py3-none-any.whl.metadata (12 kB)

Collecting requests (from webdriver-manager)

Obtaining dependency information for requests from https://files.pythonhosted.org/packages/f9/9b/335f9764261e915ed497fcdeb11df5dfd6f7bf257d4a6a2a686d80da4d54/requests-2.32.3-py3-none-any.whl.metadata

Using cached requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)

Collecting python-dotenv (from webdriver-manager)

Obtaining dependency information for python-dotenv from https://files.pythonhosted.org/packages/6a/3e/b68c118422ec867fa7ab88444e1274aa40681c606d59ac27de5a5588f082/python\_dotenv-1.0.1-py3-none-any.whl.metadata

Using cached python\_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)

Collecting packaging (from webdriver-manager)

Obtaining dependency information for packaging from https://files.pythonhosted.org/packages/88/ef/eb23f262cca3c0c4eb7ab1933c3b1f03d021f2c48f54763065b6f0e321be/packaging-24.2-py3-none-any.whl.metadata

Using cached packaging-24.2-py3-none-any.whl.metadata (3.2 kB)

Collecting charset-normalizer<4,>=2 (from requests->webdriver-manager)

Obtaining dependency information for charset-normalizer<4,>=2 from https://files.pythonhosted.org/packages/3e/67/7b72b69d25b89c0b3cea583ee372c43aa24df15f0e0f8d3982c57804984b/charset\_normalizer-3.4.0-cp312-cp312-win\_amd64.whl.metadata

Downloading charset\_normalizer-3.4.0-cp312-cp312-win\_amd64.whl.metadata (34 kB)

Requirement already satisfied: idna<4,>=2.5 in d:\proiect arobs\proiect nou\arobs\_project\.venv\lib\site-packages (from requests->webdriver-manager) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in d:\proiect arobs\proiect nou\arobs\_project\.venv\lib\site-packages (from requests->webdriver-manager) (2.2.3)

Requirement already satisfied: certifi>=2017.4.17 in d:\proiect arobs\proiect nou\arobs\_project\.venv\lib\site-packages (from requests->webdriver-manager) (2024.8.30)

Using cached webdriver\_manager-4.0.2-py2.py3-none-any.whl (27 kB)

Using cached packaging-24.2-py3-none-any.whl (65 kB)

Using cached python\_dotenv-1.0.1-py3-none-any.whl (19 kB)

Using cached requests-2.32.3-py3-none-any.whl (64 kB)

Downloading charset\_normalizer-3.4.0-cp312-cp312-win\_amd64.whl (102 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 102.2/102.2 kB 1.2 MB/s eta 0:00:00

Installing collected packages: python-dotenv, packaging, charset-normalizer, requests, webdriver-manager

Successfully installed charset-normalizer-3.4.0 packaging-24.2 python-dotenv-1.0.1 requests-2.32.3 webdriver-manager-4.0.2

[notice] A new release of pip is available: 23.2.1 -> 24.3.1

[notice] To update, run: python.exe -m pip install --upgrade pip

(.venv) PS D:\proiect Arobs\Proiect NOU\AROBS\_Project> pip install numpy

Collecting numpy

Obtaining dependency information for numpy from https://files.pythonhosted.org/packages/a6/84/fa11dad3404b7634aaab50733581ce11e5350383311ea7a7010f464c0170/numpy-2.1.3-cp312-cp312-win\_amd64.whl.metadata

Using cached numpy-2.1.3-cp312-cp312-win\_amd64.whl.metadata (60 kB)

Using cached numpy-2.1.3-cp312-cp312-win\_amd64.whl (12.6 MB)

Installing collected packages: numpy

Successfully installed numpy-2.1.3

[notice] A new release of pip is available: 23.2.1 -> 24.3.1

[notice] To update, run: python.exe -m pip install --upgrade pip

(.venv) PS D:\proiect Arobs\Proiect NOU\AROBS\_Project> pip install pyautogui

Collecting pyautogui

Using cached PyAutoGUI-0.9.54-py3-none-any.whl

Collecting pymsgbox (from pyautogui)

Using cached PyMsgBox-1.0.9-py3-none-any.whl

Collecting pytweening>=1.0.4 (from pyautogui)

Using cached pytweening-1.2.0-py3-none-any.whl

Collecting pyscreeze>=0.1.21 (from pyautogui)

Using cached PyScreeze-1.0.1-py3-none-any.whl

Collecting pygetwindow>=0.0.5 (from pyautogui)

Using cached PyGetWindow-0.0.9-py3-none-any.whl

Collecting mouseinfo (from pyautogui)

Using cached MouseInfo-0.1.3-py3-none-any.whl

Collecting pyrect (from pygetwindow>=0.0.5->pyautogui)

Using cached PyRect-0.2.0-py2.py3-none-any.whl

Collecting pyperclip (from mouseinfo->pyautogui)

Using cached pyperclip-1.9.0-py3-none-any.whl

Installing collected packages: pytweening, pyscreeze, pyrect, pyperclip, pymsgbox, pygetwindow, mouseinfo, pyautogui

Successfully installed mouseinfo-0.1.3 pyautogui-0.9.54 pygetwindow-0.0.9 pymsgbox-1.0.9 pyperclip-1.9.0 pyrect-0.2.0 pyscreeze-1.0.1 pytweening-1.2.0

[notice] A new release of pip is available: 23.2.1 -> 24.3.1

[notice] To update, run: python.exe -m pip install --upgrade pip

(.venv) PS D:\proiect Arobs\Proiect NOU\AROBS\_Project>

(.venv) PS D:\proiect Arobs\Proiect NOU\AROBS\_Project> pip install opencv-python

Collecting opencv-python

Obtaining dependency information for opencv-python from https://files.pythonhosted.org/packages/ec/6c/fab8113424af5049f85717e8e527ca3773299a3c6b02506e66436e19874f/opencv\_python-4.10.0.84-cp37-abi3-win\_amd64.whl.metadata

Using cached opencv\_python-4.10.0.84-cp37-abi3-win\_amd64.whl.metadata (20 kB)

Requirement already satisfied: numpy>=1.21.2 in d:\proiect arobs\proiect nou\arobs\_project\.venv\lib\site-packages (from opencv-python) (2.1.3)

Using cached opencv\_python-4.10.0.84-cp37-abi3-win\_amd64.whl (38.8 MB)

Installing collected packages: opencv-python

Successfully installed opencv-python-4.10.0.84

[notice] A new release of pip is available: 23.2.1 -> 24.3.1

[notice] To update, run: python.exe -m pip install --upgrade pip

(.venv) PS D:\proiect Arobs\Proiect NOU\AROBS\_Project> pip install pyaudio

Collecting pyaudio

Obtaining dependency information for pyaudio from https://files.pythonhosted.org/packages/b0/6a/d25812e5f79f06285767ec607b39149d02aa3b31d50c2269768f48768930/PyAudio-0.2.14-cp312-cp312-win\_amd64.whl.metadata

Using cached PyAudio-0.2.14-cp312-cp312-win\_amd64.whl.metadata (2.7 kB)

Using cached PyAudio-0.2.14-cp312-cp312-win\_amd64.whl (164 kB)

Installing collected packages: pyaudio

Successfully installed pyaudio-0.2.14

[notice] A new release of pip is available: 23.2.1 -> 24.3.1

[notice] To update, run: python.exe -m pip install --upgrade pip

(.venv) PS D:\proiect Arobs\Proiect NOU\AROBS\_Project>

3. CODUL PROGRAMULUI

Pentru o prima versiune a codului am incercat sa realizez navigarea pe internet, intra pe youtube, cauta un videoclip si il reda pentru doua secunde, si deja intampinam prima problema, de fiecare data cand rulam acest program se deschide pagina de youtube cu un pop-up cu accepta cookies. Asa ca inainte sa cautam un videoclip programul cauta daca primeste acest pop-up si apasa pe accept, daca nu, trece mai departe. Mai jos atasez si prima varianta a codului:

from selenium import webdriver  
from selenium.common import NoSuchElementException  
from selenium.webdriver.chrome.service import Service  
from selenium.webdriver.common.by import By  
from webdriver\_manager.chrome import ChromeDriverManager  
  
import time  
  
# Configureaza serviciul Chrome cu WebDriverManager  
service = Service(ChromeDriverManager().install())  
driver = webdriver.Chrome(service=service)  
  
# Navigheaza catre YouTube  
driver.get("https://www.youtube.com")  
  
# Asteapta ca pagina sa se incarce  
time.sleep(5)  
  
# Gaseste bara de cautare si tasteaza numele unui videoclip  
search\_box = driver.find\_element(By.NAME, "search\_query")  
search\_box.send\_keys("test video")  
  
# Apasa pe butonul de cautare  
search\_button = driver.find\_element(By.ID, "search-icon-legacy")  
search\_button.click()  
  
# Așteapta rezultatele  
time.sleep(5)  
  
# Deschide primul rezultat  
first\_result = driver.find\_element(By.XPATH, '//\*[@id="video-title"]')  
first\_result.click()  
  
# Așteapta sa înceapa redarea videoclipului  
time.sleep(10)  
  
# Inchide browserul  
driver.quit()

La care am adaugat si secventa pentru acceptarea cookie-urilor:

try:  
 accept\_button = driver.find\_element(By.XPATH, '//\*[@aria-label="Accept the use of cookies and other data for the purposes described"]')  
 accept\_button.click()  
except NoSuchElementException:  
 pass  
  
time.sleep(3)

Pentru asta am cautat in folosit inspect page intr-un tab nou youtube, unde am gasit elementul cu accept cookies. Aceeasi metoda a fost folosita pentru cautarea tuturor elementelor din acest cod.

Pentru o functionare mai buna a programului am inlocuit time.sleep dintre comenzi care erau plasate pentru a astepta ca pagina sa se incarce, cu o comanda gasita pe internet care chiar asteapta cat este nevoie ca pagina sa se incarce, singura problema este ca nu putea fi folosita decat o data pentru ca pana nu se schimba pagina, pagina a fost incarcata practic, chiar daca elementele nu sunt afisate, asa ca pentru urmatoarele instante ex: asteptarea incarcarii rezultatelor, am adaptat putin comanda pentru a astepta pana cand elementele sunt afisate pe ecran, aceste comenzi asteapta pana la 10 secunde, iar daca pana atunci nu se incarca nimic, ies din program.

4. PROBLEMA PRINCIPALA A CODULUI

Cea mai mare problema pe care am avut-o in crearea acestui cod a fost trecerea de reclamele videoclipului. In prima faza am incercat sa analizez tipurile de reclame pe care youtube le afiseaza la inceperea unui videoclip si am constatat urmatoarele aspecte:

* Durata reclamelor variaza intre 6 secunde si cateva minute;
* Butonul de skip ad poate aparea dupa 3 sau 6 secunde;
* La inceput pot intra una sau doua reclame;

Ca o prima solutie am incercat sa ma leg de bara de stare a reclamelor, deoarece aceasta are o culoare diferita fata de cea a videoclipului in sine, insa nu a dat roade. Deoarece sunt acelasi element si astfel trebuia sa imi complic existenta cu mai multe conditii.

Am descoperit mai apoi ca fiecare reclama contine un div cu numele si logo-ul companiei careia ii se face reclama. Insa am avut probleme la gasirea selectorilor, ulterior am aflat de ce dupa gasirea unei alte solutii ce credeam ca ma va duce spre succes.

Pentru o perioada mare de timp am incercat sa ma leg de textul de sponsored care apare, indiferent de reclama. Insa codul imi dadea erori, chiar si atunci cand iesea din reclama, acesta citea videoclipul ca si reclama.

Dupa multe incercari mi-am dat seama ca textul sponsored, poate aparea in mai multe locuri in pagina, chiar daca foloseam un class name, acesta era acelasi, asa ca m am folosit de un div specific reclamei de pe videoclip, acest div inglobeaza textul sponsored cu numarul de reclame.

In sfarsit codul capta reclamele, insa, mi se parea ca era foarte ingramadit si greu de schimbat. Fapt pentru care am schimbat abordarea codului, si pentru asta am inceput sa folosesc functii si diferite fisiere python pentru cele trei “sub-programe” daca le pot numi asa.

5. SRUCTURAREA PROGRAMULUI IN MAI BINE

In momentul acela nu aveam inca cele 3 sub-programe. Dar am incercat sa ma focusez pe ce am. In interiorul acestui program de navigare pe youtube, am creat 4 functii:

* O functie care cauta daca sunt reclame prezente pe videoclip;
* O functie care cauta si apasa butonul de skip ad dupa 6 secunde;
* O functie care actioneaza in cazul in care apar reclame fara buton de skip ad;

Logica fiecarei functii:

Prima functie care detecteaza daca avem reclame prezente, este una simpla

* Cauta prezenta elementului din reclame;
* In cazul in care elementul este vizibil returneaza True, altfel False
* Daca nu este detectat elementul returneaza False

Cea dea a doua functie, care cauta butonul de skip ad functioneaza astfel:

* Cauta elementul buton skip ad
* Daca acesta este vizibil si interactiv
* Asteapta 5 secunde, apasa butonul, returneaza True
* Altfel returneaza False
* Daca nu gaseste elementul returneaza False

Functia pentru reclame fara buton skip lucreaza practic in tandem cu functia de buton skip, astfel:

* Reclamele au o durata afisata pe ecran, asa ca m am folosit de asta, din durata totala scadem durata deja parcursa din reclama si aflam cat timp mai avem pana la finalizarea reclamei. Asa ca primul pas a fost sa luam **valorile** acestor doua elemente
* Dupa care le scoatem separatorii deoarece durata este exprimata astfel “00:30”
* Si separam minutele de secunde si folosim expresia minut\*60 + secunde, astfel primi durata in secunde care mai este de parcurs din reclame.
* Dupa aflarea acestei durate cautam butonul de skip apeland functia anterioara, daca returneaza true, iesim din functie; daca returneaza false atunci activeaza timerul care scurge secunda cu secunda pana cand se incheie reclama

Ca sa trec de problema cu mai multe reclame per videoclip, am initiat un i care creste cat timp exista reclame, si reia acest ciclu de cautare reclama, cautare buton si tot asa de cate ori gasim prezenta unei reclame. Nu am intalnit niciodata sa am mai mult de 2 reclame dar niciodata nu poti fi prea sigur.

Intr-un final mai avem o functie care ruleaza tot programul pentru navigare youtube ca mai apoi sa fie apelat in alt fisier python.

Fisierul principal numit main.py care mai tarziu va apela cele 3 subprograme de care va ziceam mai devreme.

Cel de al doilea sub-program se ocupa de inregistrarea ecranului. Avand la randul lui 3 functii principale

* Functie pentru inregistrare audio
* Functie pentru inregistrare video
* Functie pentru inregistrare simultan a celor doua

Prima functie care creaza o inregistrare audio si o salveaza sub numele final\_audio.wav si functioneaza astfel:

Alocam urmatoarele variabile

Chunk = 1024 care reprezinta dimensiunea blocurilor de date citite din microfon, adica 1024 mostre per citire.

Channels = 2 pentru stereo audio

Format = pyaudio.paInt16 formatul audio pe 16 biti

Rate = 44100 rata de esantionare 44.1 kHz, care este rata standard audio

Cream o instanta pentru captarea sunetului

Cream un flux de intrare pentru captarea sunetului

Folosim o bucla pentru inregistrare si adaugam fiecare bloc in lista frames

Salvam datele intr-un fisier wav

Cea de a doua functie pentru inregistrarea video

Declara fourcc care reprezinta codul pentru compresia video XVID fiind un codec popular pentru fisiere AVI

Creaza un obiect pentru salvarea videoclipului cu 20 cadre per secunda si dimensiunile ecranului

Initiem o bucla ca cea pentru inregistrarea audio care captureaza o imagine a ecranului, converteste captura intr-un array numpy, convertese culorile din format BGT in format RGB si adauga cadrul intr-im fisier video

A treia functie foloseste un thread pentru a rula ambele functii simultan

Creaza un thread pentru inregistrarea audio si il proneste

Incepe inregistrarea video pe threadul principal

Ultimul sub-program al acestui proiect este reprezentat de anilza datelor din inregistrarea audio si sa scrie intr-un fisier nivelul minim, maxim si mediu de decibeli. Acest program functioneaza in felul urmator:

Detine o functie care:

* Deschide fisierul audio
* Citeste datele acestui fisier
* Verifica daca exista mai multe canale si face o medie intre cele doua canale
* Calculeaza cele trei niveluri de amplitudine
* Si scrie rezultatele intr-un output file numit audio\_analysis.txt

Toate aceste sub-programe sunt apelate in programul main al acestui proiect care:

* Configureaza serviciul chrome in prima instanta folosind WebDriverManager, motivul pentru care configuram in main.py acest serviciu si nu in youtube\_navigation.py este pentru ca vrem ca acest serviciu sa raman activ si dupa gestionarea paginii web.
* Navigheaza pe youtube apeland programul youtube\_navigation.py
* Realizeaza cele doua inregistrari, audio si video apeland screen\_recording.py
* Analizeaza fisierul audio si creaza un alt fisier de tip text pentru interpretarea datelor apeland audio\_analysis.py
* Dupa terminarea acestor pasi inchide browserul.

Nu am ales sa intru in detalii cu posibila problema a conexiunii de internet deoarece am fost presat de timp, plus ca programul deja face print daca apari erori neprevazute si inchide programul daca trec 10 secunde de la asteptarea incarcarii anumitor pagini.

Motivul pentru care am folosit foarte multe print-uri in cadrul acestui program este pentru ca am vrut sa mi se afiseze fiecare pas pe ecran in timpul rulari pentru a-mi fi mult mai usor pentru diagnosticarea si testarea acestuia.