import os

import pandas as pd

import time

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

from selenium.common.exceptions import NoSuchElementException, WebDriverException

# Function to extract views and likes from YouTube

def extract\_views\_and\_likes(driver):

try:

views = driver.find\_element(By.CSS\_SELECTOR, "span.view-count").get\_attribute("innerText")

likes = driver.find\_element(By.XPATH,

"(//div[@class='smartimation\_\_content'])[2]/div/like-button-view-model/toggle-button-view-model/button-view-model/button/div[2]").get\_attribute("innerText")

return views, likes

except NoSuchElementException:

return "Not Available", "Not Available"

# Function to get trailer information from YouTube

def get\_trailer\_info(movie\_name):

driver = webdriver.Chrome()

driver.get("https://www.youtube.com")

search\_box = driver.find\_element(By.CSS\_SELECTOR, "input#search")

search\_box.send\_keys(f"{movie\_name} trailer Telugu")

search\_box.submit()

time.sleep(3) # Wait for search results to load

try:

video\_link = driver.find\_element(By.CSS\_SELECTOR, "#video-title").get\_attribute("href")

driver.get(video\_link)

time.sleep(5) # Wait for the video page to load

views, likes = extract\_views\_and\_likes(driver)

except NoSuchElementException:

views, likes = "Not Available", "Not Available"

finally:

driver.quit()

return views, likes

# Extended list of Tollywood superstars (actors and actresses)

tollywood\_superstars = [

# Actors

'Chiranjeevi', 'Nagarjuna Akkineni', 'Venkatesh Daggubati', 'Balakrishna Nandamuri',

'Pawan Kalyan', 'Mahesh Babu', 'Prabhas', 'Jr. NTR', 'Ram Charan', 'Allu Arjun',

'Ravi Teja', 'Nani', 'Vijay Deverakonda', 'Nithiin', 'Ram Pothineni', 'Sai Dharam Tej',

'Varun Tej', 'Akhil Akkineni', 'Naga Chaitanya', 'Sharwanand', 'Sundeep Kishan',

'Raj Tarun', 'Adivi Sesh', 'Sumanth', 'Sudheer Babu', 'Sree Vishnu', 'Vishwak Sen',

'Kalyan Ram',

# Actresses

'Anushka Shetty', 'Kajal Aggarwal', 'Samantha Akkineni', 'Tamannaah Bhatia',

'Rashmika Mandanna', 'Pooja Hegde', 'Rakul Preet Singh', 'Nithya Menen', 'Keerthy Suresh',

'Sai Pallavi', 'Lavanya Tripathi', 'Rashi Khanna', 'Shruti Haasan', 'Eesha Rebba',

'Nivetha Thomas', 'Catherine Tresa', 'Regina Cassandra', 'Mehreen Pirzada', 'Payal Rajput',

'Pragya Jaiswal', 'Anupama Parameswaran', 'Shalini Pandey', 'Kriti Shetty',

'Priyanka Jawalkar', 'Aishwarya Rajesh'

]

# Open the browser

driver = webdriver.Chrome()

years = ['2024']

data = []

for year in years:

url = f"https://en.wikipedia.org/wiki/List\_of\_Telugu\_films\_of\_{year}"

driver.get(url)

# Wait for the table to be present

wait = WebDriverWait(driver, 10)

wait.until(EC.presence\_of\_element\_located((By.CSS\_SELECTOR, ".wikitable")))

# Find all movie rows

movie\_rows = driver.find\_elements(By.CSS\_SELECTOR, ".wikitable tbody tr")

# Loop through each movie row

for row in movie\_rows:

try:

# Ensure the row contains a movie link

link\_elements = row.find\_elements(By.CSS\_SELECTOR, "td:nth-child(2) a")

if not link\_elements:

continue

link = link\_elements[0]

movie\_url = link.get\_attribute("href")

movie\_name = link.text.strip()

# Open the movie page

driver.execute\_script("window.open('');")

driver.switch\_to.window(driver.window\_handles[1])

driver.get(movie\_url)

time.sleep(0.1) # Wait for the page to load

# Find and store each element using its XPath

elements = {

"Movie Name": "//h1[@id='firstHeading']",

"Directed by": "//th[contains(text(), 'Directed by')]/following-sibling::td",

"Starring": "//th[contains(text(), 'Starring')]/following-sibling::td",

"Music by": "//th[contains(text(), 'Music by')]/following-sibling::td",

"Production Company": "//th[contains(text(), 'Production') or contains(text(), 'Produced by')]/following-sibling::td",

"Country": "//th[contains(text(), 'Country')]/following-sibling::td",

"Language": "//th[contains(text(), 'Language')]/following-sibling::td",

"Budget": "//th[contains(text(), 'Budget')]/../td/span"

}

movie\_data = {}

# Loop through each attribute and store the value in the data dictionary

for title, xpath in elements.items():

try:

element = driver.find\_element(By.XPATH, xpath)

value = element.text.strip()

# Remove 'est. ', '₹', and '[]' for Budget

if title == "Budget":

value = value.replace('est. ', '').replace('₹', '').replace('[', '').replace(']', '')

movie\_data[title] = value

else:

movie\_data[title] = value

except NoSuchElementException:

movie\_data[title] = "Not Available"

# Special handling for Box Office as there are two cases

try:

box\_office\_element = driver.find\_element(By.XPATH, "//th[contains(text(), 'Box office')]/following-sibling::td/span")

box\_office = box\_office\_element.text.strip()

box\_office = box\_office.replace('est. ', '').replace('₹', '').replace('[', '').replace(']', '')

except NoSuchElementException:

try:

box\_office\_element = driver.find\_element(By.XPATH, "//th[contains(text(), 'Box office')]/following-sibling::td")

box\_office = box\_office\_element.text.strip()

box\_office = box\_office.replace('est. ', '').replace('₹', '').replace('[', '').replace(']', '')

except NoSuchElementException:

box\_office = "Not Available"

movie\_data["Box Office"] = box\_office

# Check if any superstar is in the Starring cast

if any(superstar in movie\_data.get("Starring", "") for superstar in tollywood\_superstars):

movie\_data["Superstar"] = "Yes"

else:

movie\_data["Superstar"] = "No"

# Append movie title and data to the list

movie\_data["Movie Title"] = movie\_name

# Get trailer views and likes from YouTube

views, likes = get\_trailer\_info(movie\_name)

movie\_data["Trailer Views"] = views

movie\_data["Trailer Likes"] = likes

data.append(movie\_data)

# Close the movie tab

driver.close()

# Switch back to the main tab

driver.switch\_to.window(driver.window\_handles[0])

except NoSuchElementException:

print(f"No movie link found for a row in the table.")

except WebDriverException as e:

print(f"WebDriverException encountered: {e}")

# After collecting all the data, close the browser

driver.quit()

# Convert the list of dictionaries to a pandas DataFrame

df = pd.DataFrame(data)

# Specify the file path to save the CSV file

csv\_file\_path = "C:/Users/SONY/Desktop/project4.csv"

# Save the DataFrame to a CSV file

df.to\_csv(csv\_file\_path, index=False)

print(f"CSV file saved to: {csv\_file\_path}")

# Print the DataFrame

print(df)