from bs4 import BeautifulSoup

import requests

import random

import openpyxl

def get\_user\_input():

# Use of While

while True:

a = input("Enter a year (between 1900 and 2023): ")

year = int(a)

# Validate the User Input

if 1900 <= year <= 2100:

return a

else:

print("Invalid input. Please enter a valid year (between 1900 and 2023).")

def scrape\_movies\_by\_year(year):

# Define a list of User Agents for different browsers

user\_agents = [

'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36',

'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:53.0) Gecko/20100101 Firefox/53.0',

'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; Trident/5.0; Trident/5.0)',

'Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; Trident/6.0; MDDCJS)',

'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.79 Safari/537.36 Edge/14.14393',

'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)'

]

# Get the data from the Wikipedia page and store it in the source variable

source = requests.get('https://en.wikipedia.org/wiki/List\_of\_highest-grossing\_films', headers={'User-Agent': random.choice(user\_agents)})

source.raise\_for\_status()

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

tbody = soup.find('tbody')

# Exception handling

try:

# List to store movies by year

movies\_by\_year = []

# Iterate through rows in the table, skipping the header row.

for row in tbody.find\_all('tr')[1:]:

columns = row.find\_all('td')

# Get rank, peak, title, gross, and year.

title\_tag = row.find('a')

Title = title\_tag.text.strip()

rank = columns[0].text.strip()

peak = columns[1].text.strip()

Gross = columns[2].text.strip()

Year = columns[3].text.strip()

# Check if the year matches user input

if Year == year:

movies\_by\_year.append({

'Rank': rank,

'Peak': peak,

'Title': Title,

'Gross': Gross,

'Year': Year

})

# Return the list of movies by year

return movies\_by\_year

except requests.exceptions.RequestException as req\_error:

print(f"Request failed: {req\_error}")

except Exception as e:

print(f"An unexpected error occurred: {e}")

# Function to display movies to the user

def display(movies):

print()

print(f"Movies released in {year\_input}:")

print()

for movie in movies:

print(f"Rank: {movie['Rank']}")

print(f"Peak: {movie['Peak']}")

print(f"Title: {movie['Title']}")

print(f"Gross Revenue: {movie['Gross']}")

print()

# Function to write data to Excel

def write\_to\_excel(movies, filename='movies\_data.xlsx'):

wb = openpyxl.Workbook()

sheet = wb.active

# Write header row

header = ['Rank', 'Peak', 'Title', 'Gross Revenue']

sheet.append(header)

# Write data rows

for movie in movies:

row\_data = [movie['Rank'], movie['Peak'], movie['Title'], movie['Gross']]

sheet.append(row\_data)

wb.save(filename)

print(f"Data written to {filename}")

# Get year of user input and scrape results for that year

year\_input = get\_user\_input()

movies = scrape\_movies\_by\_year(year\_input)

# Check if movies are found for the specified year

if movies:

# Display movies to the user

display(movies)

# Create Excel file and write data

write\_to\_excel(movies, filename=f'movies\_by\_year\_{year\_input}.xlsx')

else:

print(f"No movies found for the specified year {year\_input}.")