Google Image Scrapping

from flask import Flask, render\_template, request,jsonify

from flask\_cors import CORS,cross\_origin

import requests

from bs4 import BeautifulSoup

from urllib.request import urlopen as uReq

import logging

import pymongo

logging.basicConfig(filename="scrapper.log" , level=logging.INFO)

import os

app = Flask(\_\_name\_\_)

@app.route("/", methods = ['GET'])

def homepage():

    return render\_template("index.html")

@app.route("/review" , methods = ['POST' , 'GET'])

def index():

    if request.method == 'POST':

                try:

                    # query to search for images

                    query = request.form['content'].replace(" ","")

                            # directory to store downloaded images

                    save\_directory = "images/"

                            # create the directory if it doesn't exist

                    if not os.path.exists(save\_directory):

                        os.makedirs(save\_directory)

                            # fake user agent to avoid getting blocked by Google

                    headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36"}

                            # fetch the search results page

                    response = requests.get(f"https://www.google.com/search?q={query}&sxsrf=AJOqlzUuff1RXi2mm8I\_OqOwT9VjfIDL7w:1676996143273&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiq-qK7gaf9AhXUgVYBHYReAfYQ\_AUoA3oECAEQBQ&biw=1920&bih=937&dpr=1#imgrc=1th7VhSesfMJ4M")

                            # parse the HTML using BeautifulSoup

                    soup = BeautifulSoup(response.content, "html.parser")

                            # find all img tags

                    image\_tags = soup.find\_all("img")

                            # download each image and save it to the specified directory

                    del image\_tags[0]

                    img\_data=[]

                    for index,image\_tag in enumerate(image\_tags):

                                # get the image source URL

                                image\_url = image\_tag['src']

                                #print(image\_url)

                                # send a request to the image URL and save the image

                                image\_data = requests.get(image\_url).content

                                mydict={"Index":index,"Image":image\_data}

                                img\_data.append(mydict)

                                with open(os.path.join(save\_directory, f"{query}\_{image\_tags.index(image\_tag)}.jpg"), "wb") as f:

                                    f.write(image\_data)

                    return "image laoded"

                except Exception as e:

                    logging.info(e)

                    return 'something is wrong'

            # return render\_template('results.html')

    else:

        return render\_template('index.html')

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

    app.run(debug=True)