import pandas as pd from bs4 import BeautifulSoup import requests as r import streamlit as st st.markdown('

Search Engine Scraper

', unsafe_allow_html=True) query = st.text_input('', help='Enter the search string and hit Enter/Return') query = query.replace(" ", "+") #replacing the spaces in query result with + if query: #Activates the code below on hitting Enter/Return in the search textbox try:#Exception handling req = r.get(f"https://www.google.com/search?q={query}", headers = {"user-agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36"}) result_str = '' #Initializing the HTML code for displaying search results if req.status_code == 200: #Status code 200 indicates a successful request bs = BeautifulSoup(req.content, features="html.parser") #converting the content/text returned by request to a BeautifulSoup object search_result = bs.find_all("li", class_="b_algo") #'b_algo' is the class of the list object which represents a single result search_result = [str(i).replace("","") **for i in search_result] #removing the tag search_result = [str(i).replace("","") for i in search_result] #removing the** tag result_df = pd.DataFrame() #Initializing the data frame that stores the results for n,i in enumerate(search_result): #iterating through the search results individual_search_result = BeautifulSoup(i, features="html.parser") #converting individual search result into a BeautifulSoup object h2 = individual_search_result.find('h2') #Finding the title of the individual search result href = h2.find('a').get('href') #title's URL of the individual search result cite = f'{href[:50]}...' if len(href) >= 50 else href # cite with first 20 chars of the URL url_txt = h2.find('a').text #title's text of the individual search result #In a few cases few individual search results doesn't have a description. In such cases the description would be blank description = "" if individual_search_result.find('p') is None else individual_search_result.find('p').text #Appending the result data frame after processing each individual search result result_df = result_df.append(pd.DataFrame({"Title": url_txt, "URL": href, "Description": description}, index=[n])) count_str = f'**Bing Search returned {len(result_df)} results**' ## ########## HTML code to display search results ########## ## result_str += f'

{url_txt}

'+\ f'**{cite}**'+\ f'{description}'+\ f'' result_str += '

' #if the status code of the request isn't 200, then an error message is displayed along with an empty data frame else: result_df = pd.DataFrame({"Title": "", "URL": "", "Description": ""}, index=[0]) result_str = '' count_str = '**Looks like an error!!**' #if an exception is raised, then an error message is displayed along with an empty data frame except: result_df = pd.DataFrame({"Title": "", "URL": "", "Description": ""}, index=[0]) result_str = '' count_str = '**Looks like an error!!**' st.markdown(f'{count_str}', unsafe_allow_html=True) st.markdown(f'{result_str}', unsafe_allow_html=True) st.markdown('

Data Frame of the above search result

', unsafe_allow_html=True) st.dataframe(result_df)