NANYANG TECHNOLOGICAL UNIVERSITY

Automation of Scraping for Conflict of Interest in Webpages

Project Supervisor: A/P Sourav S Bhowmick

Examiner: A/P Prof Melanie Herschel

School of Computer Science and Engineering

Academic Year 2023/2024

NANYANG TECHNOLOGICAL UNIVERSITY

Automation of Scraping for Conflict of Interest in Webpages

Submitted in Partial Fulfilment of the Requirements
for the Degree of Bachelor of Computer Engineering
of the Nanyang Technological University
by
Lee Ming Jia

Supervisor: A/P Sourav S Bhowmick

School of Computer Science
Academic Year 2023/2024

Abstract

As the world advances in technology, researchers compete to submit their papers to gain recognition and to show advances in technology. However, just as these calls are held by people, some of these submissions could be submitted by those who had recent contact or relations with the organiser of the events. As a result, this could result in an unfair competition, leading to conflict of interest between the organisers and candidates. Hence, during the submission of a paper, the submission sites will request information about conflicts of interest of the paper's authors with program committee (PC) members.

This project presents the development of an automation python-based application that can be used to extract for information relating to conflict of interest related to an event. The goal of this project is to automate the process so that it can be done on multiple webpages at the same time, hence not requiring the user to individually type down every webpage to be scraped from.

Future work on the application will focus on optimising the code to prevent the code from extracting excessive information as well as improving its capabilities in scraping other information stored in the webpage.

Acknowledgements

I would like to express my sincere gratitude to those who supported me throughout my Final Year Project (FYP). I would also like to express my appreciation to my supervisor, Professor Sourav S Bhowmick, for his constant guidance, valuable insights with regards to my FYP. In addition, I would also like to thank both Professor Sourav S Bhowmick and Prof Melanie Herschel for their constructive feedback for my FYP report.

Table of Contents

1. Introduction

1.1 Motivation

As the world advances in technology, researchers compete to submit their papers to gain recognition and to show advances in technology. Such competitions include the call for paper that was held by NIST in 1997[1] to obtain new algorithms that is best used for block ciphers for standardization for international standards.

There is also an increase in demand for research to be done on different computer science related fields as well. These could be fields related to database management to create a more efficient and effective way to organize the database so that less computing power is needed to process information in the database. The people in these fields may also be able to work on computer security where they strengthen the security of databases and prevent identity theft of a person via leakage in database. Thus, for such cases, they are able to take part in multiple competitions of different fields.

However, as these calls are held by people, some of the submissions could be submitted by those who had recent contact or relations with the organiser of the events. As a result, this could result in an unfair competition, leading to conflict of interest between the organisers and candidates.

Hence, during the submission of a paper, the submission sites will request information about conflicts of interest of the paper's authors with program committee (PC) members. It is the full responsibility of all authors of a paper to identify all and only the PC members with whom they have a conflict of interest (COI).[2] Information relating to conflict of interest could also be stated in the event page for candidates to view or in the submission site where participants will submit their papers.

Thus, for people who wishes to take part in multiple different competitions and may have a wide network with potential committee members, there is a need to check for any conflict of interest as conflict could disqualify one from taking part in the call for papers event.

1.2 Project Objectives

This project aims to automate the challenge of identifying the different types of scenarios which constitute as Conflicts of Interest (COI) for competitions by coming up with an automation coding programme. The goal of this is to develop a programme that will automatically detect for the different types of scenarios constituted as conflicts of interest from different webpages using webpage scraping. For the purpose of this report, the action of detecting for the different types of scenarios of conflicts of interest will be phrased as detect for Conflicts of Interests. The information obtained will then be stored in a dataframe before being stored in a csv file for the user to use and detect the conflicts of interest that a competition has.

1.3 Project Scope

To achieve this project objective, there are 2 factors involved, the data required for webpage scraping and the web scraping tool that will be used.

Firstly, the data that is required for this will be obtained from csrankings.org, a website that contains information about various competition that is related to computer science fields. The plan is to scrape the website to pull out relevant data and analyse it to detect for potential conflicts of interest related to each competition.

Next, as some websites use JavaScript to load their webpage content dynamically, certain packages such as Scrapy and BeautifulSoup are not suitable to be used. Hence, Selenium [3] has been selected to perform this task. This project will develop a webpage scraping program that can automatically detect conflict of interest for different competition, allowing users to better understand any possible conflict that may occur.

For this project, Python in a Jupyter Notebook environment will be used to write and run the code.

2. Webpage scraping

2.1 Definition

Webpage scraping, also known as web scraping or data scraping, is the usage of software tools like Selenium to automate the process of extracting information from websites. This can help to obtain large amount of information efficiently from the internet for usage.

2.2 Purpose

Webpage Scraping plays an important role in many areas such as research, business intelligence and data analytics. With webpage scraping, individuals and organizations are able to access large amount of public data from websites, making a laborious task to be done easier and faster.

2.3 Applications in computer science

In computer science, webpage scraping can be used for a variety of applications. Some of these include machine learning, data science and search engines optimization. Since these applications require large amounts of data for training or analysis, the huge amount of data that can be obtained from webpage scraping makes it easier and faster for data collection.

2.4 Concerns

Even though webpage scraping can help to provide a large amount of data for a variety of usage, there are also concerns with regards to ethical and legal issues. Although most websites generally have terms of service that can help to regulate data extraction, there are some websites that prohibit it completely. For users scraping data from websites, there is a need to check robots.txt file of each webpage before doing so as it states what can or cannot be done. There are also some websites such as Reddit and Facebook that provide an API that can be used by users for scraping which on its own, help to limit what can be scraped to protect user privacy.

3. Conflicts of interest

From detecting for conflict of interest through the websites manually, the most common scenarios constituted as conflict of interest are as follows:

1. any program committee (PC) member who works/worked at the same organizational affiliation with any co-author in the last 24 months or reasonably expected within the next 12 months;
2. any PC member who has co-authored any book, article, report, abstract or paper with collaboration in the last 24 months or reasonably expected within the next 12 months;
3. any PC member who has collaborated on projects, such as funded grants, research or others with any co-author in the last 24 months or reasonably expected within the next 12 months;
4. any PC member who is/was the advisor or advisee of any co-author;
5. any PC member who graduated from the same research group and/or was under the same advisor/supervisor/manager in the last 48 months;
6. any PC member who is exposed to the full or partial authorship of the work being submitted here in any public or private channel (including but not limited to interview talks, campus visits, research discussion, conversations);
7. any PC member who has a close personal friendship or business relationship beyond professional settings that could affect the review process with any co-author;
8. any family member
9. deep personal animosity
10. anyone who is affiliated with a party that funds your research

Out of the 10 common COI mentioned above, points 1,2,3,4,5 and 8 are the conflicts of interest that are most commonly identified. In addition, even if the points for some of the conflicts are similar, there are also some differences based on the phasing of the words used as well as the time period. For example, when it comes to co authored and collaboration, there are differences between different competitions in terms of the period between the competition and collaboration before it is considered as conflict of interest or not. It could range from within 1 year to as long as four years for it to be classified as no conflict of interest. There are also some conflicts of interest that are not stated here as they only appear in one or two competitions only, making them to be the exception.

4. Implementation

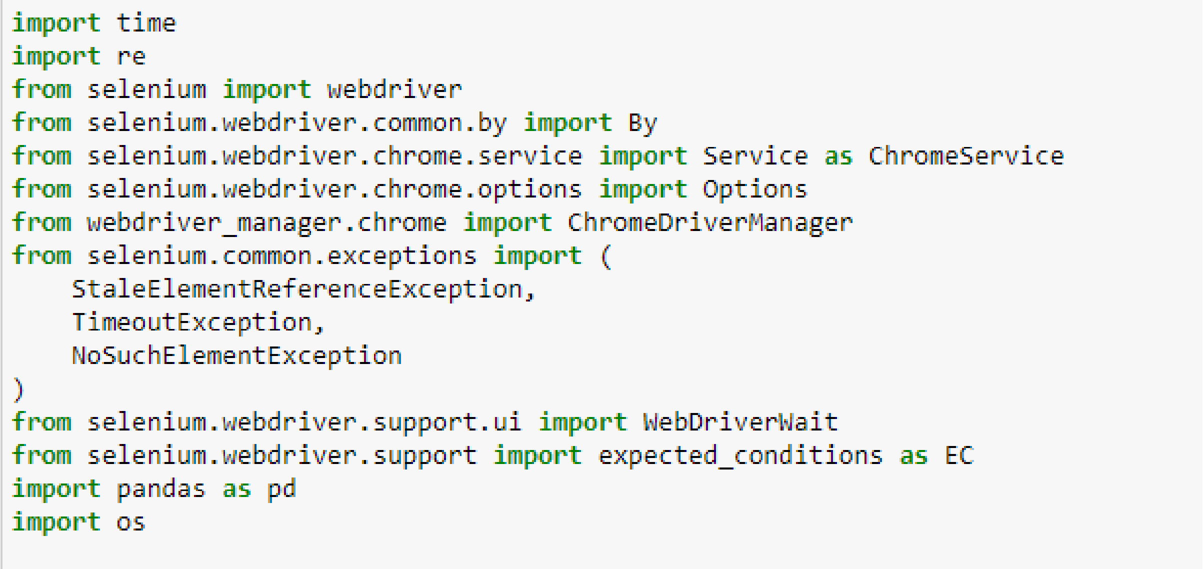
The implementation of the automation program consists of three parts as shown below.

1. Packages that are used to run the program
2. Functions that were defined to aid in the scraping of the webpage and for COI
3. The main code where most of the code will be running to scrape for COI

4.1 Packages

Before running any code, there is a need to first import all the different packages that are needed to run the code. This can be seen below.

- ‘time’ is needed to delay the running of the code so that there will be time for the webpage to load.
- the packages related to ‘selenium’ is needed for scraping as selenium does the automation of the scraping and extraction of the information from the webpage
- ‘webdriver_manager’ is used to allow for the code to access the web browser to access the information stored on the webpage.
- ‘pandas’ is used to store the data obtained from the scraping in a data frame
- ‘os’ is used to convert the data into a csv file on the desktop.
- ‘re’ is used to process the data collected into readable form to be stored in the dataframe before the csv file



```
import time
import re
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service as ChromeService
from selenium.webdriver.chrome.options import Options
from webdriver_manager.chrome import ChromeDriverManager
from selenium.common.exceptions import (
    StaleElementReferenceException,
    TimeoutException,
    NoSuchElementException
)
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import pandas as pd
import os
```

4.2 Scrape_links

The function ‘scrape_links’ is used to scrape the links. This function will first access the link. From there, it will look for link and urls that are stored on the webpage and extract them to be stored in a list.

In scrape_links, there is perform_scrape function. This function contains the part of the code for scraping and is later used in part 2 of scrape_links function to run scraping of the links again. This is done in cases where there are issues in scraping the links such as stale element exception.

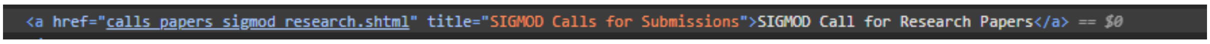
Perform_scrape has 3 parameters which are driver, url and retries.

- The driver is the chrome webdriver that will be defined later on in the main.
- Url is the links that will be scraped for further links from it and
- retries is number of times that the code will attempt to run perform_scrape should an error occur.

For example, the website may have the url (url.com/call or url.com/2024) or if the hyper link mentioned ‘call for paper’ or ‘KDD2024’. In such case, the ‘driver.get’ will get access to the webpage. Below is an example of how next part of the code will run.

For example, the tag/element is as follows:

<a href="calls_papers_sigmod_research.shtml" title="SIGMOD Calls for Submissions">SIGMOD Call for Research Papers</a>



```
<a href="calls_papers_sigmod_research.shtml" title="SIGMOD Calls for Submissions">SIGMOD Call for Research Papers</a> == $0
```

The ‘find_elements(BY.TAG_NAME, ‘a’)’ will look for items that are between ‘<a> item <a> ‘

If the inputs are as follows:

href = link.get_attribute('href')

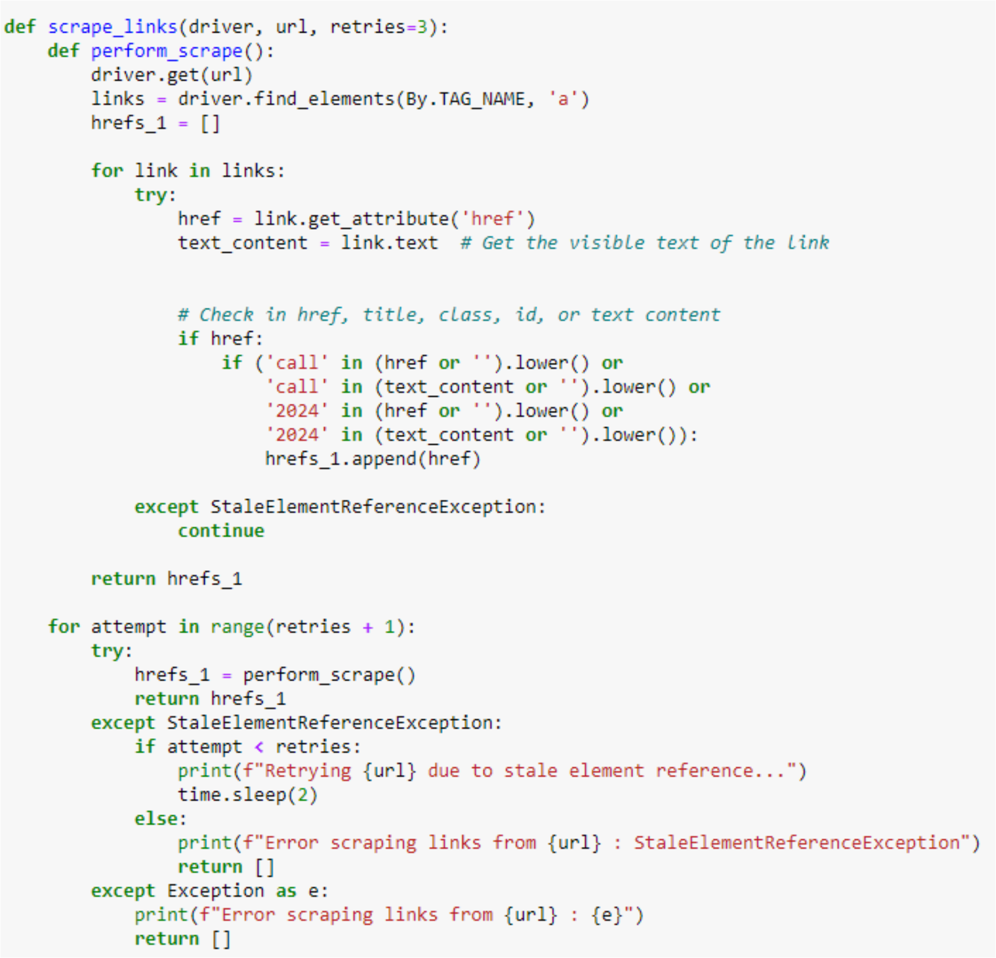
text_content = link.text

The return output will be:

Link href: calls_papers_sigmod_research.shtml

Link text: SIGMOD Call for Research Papers

The next part of the code will then detect if there is a href. If there is, it will run [‘call’ in (href or ‘’).lower or] for all the different possible cases of where ‘call’ or ‘2024’ can be. If either of these 2 can be found, then the href will be appended to ‘href_1’ and the link will be stored in the list, or else the link is not appended, and it will continue onto the next item that is in links.



```python
def scrape_links(driver, url, retries=3):
    def perform_scrape():
        driver.get(url)
        links = driver.find_elements(By.TAG_NAME, 'a')
        hrefs_1 = []

        for link in links:
            try:
                href = link.get_attribute('href')
                text_content = link.text  # Get the visible text of the link


                # Check in href, title, class, id, or text content
                if href:
                    if ('call' in (href or '').lower() or
                        'call' in (text_content or '').lower() or
                        '2024' in (href or '').lower() or
                        '2024' in (text_content or '').lower()):
                        hrefs_1.append(href)

            except StaleElementReferenceException:
                continue

        return hrefs_1

    for attempt in range(retries + 1):
        try:
            hrefs_1 = perform_scrape()
            return hrefs_1
        except StaleElementReferenceException:
            if attempt < retries:
                print(f"Retrying {url} due to stale element reference...")
                time.sleep(2)
            else:
                print(f"Error scraping links from {url} : StaleElementReferenceException")
                return []
        except Exception as e:
            print(f"Error scraping links from {url} : {e}")
            return []
```

4.3 Search_conflict_of_interest

Next, how the function 'search_conflict_of_interest' detect for conflict of interest in the webpage will be explained. This function will check if there is the text 'conflict' found on the webpage. If yes, it will continue to look at the items that can be found in a list underneath it and extract those items into a list. If it fails to do that or found a text 'conflict' in the webpage but cannot extract the items, it will return the error that occurred.

It has 3 parameters, driver, url and retries which work similarly to parameters in scrape_links.

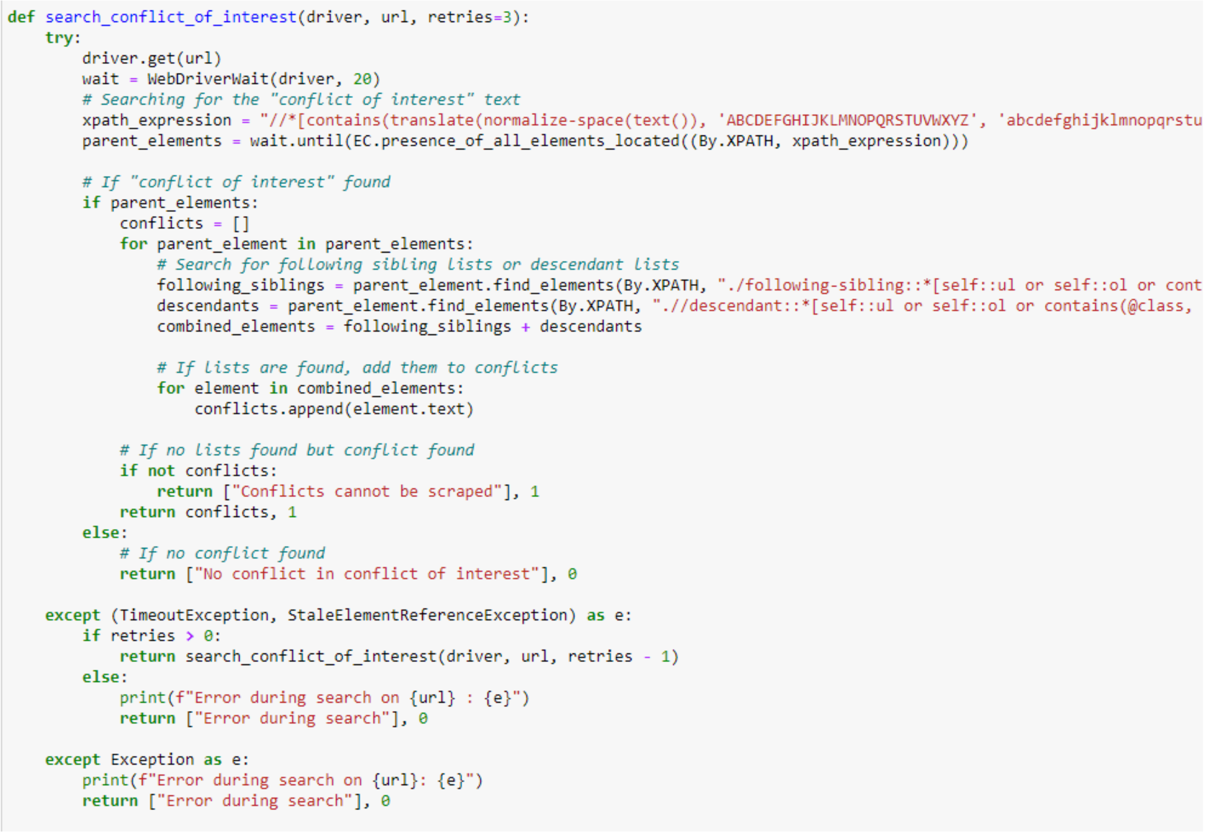
In addition, in the code:

- 'wait' , will be used in parent_elements to give the code time to load up and to ensure that that the function only proceeds once the elements are available.
- 'Xpath_elements' contain the item that the code will search for. It has many different functions like translate, normalize-space and 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz' to account for different combinations of the word conflict such as Conflict, CONFLICT and such. The code will thus, look for the text 'conflict' in the webpage and considering different combinations of it. From there, it will begin to scrape the information below the text that is related to conflicts.

‘If parent_element’ will check if there is any conflict detected. If a conflict is detected, it will return a positive and create a list conflict. It will then iterate through the items in ‘parent_elements’ to look for items that are self::ul, self::ol, class ‘ul’ or class ’ol’ . ‘self’ is used to look at item own tag or element.

Items in list are usually placed in either ul or ol or under class ul or ol. Since the list can be either descendant or sibling, ‘following_sibling’ will look for those items that are sibling after conflict and ‘descendants’ for those that are descendant of conflict. ‘combined_elements’ will then take the 2 lists and merge them together. After which, the items in the list ‘combined_elements’ are appended to the list ‘conflicts’. There is a focus on scraping list as scraping those items that is in not a list would result in excessive scraping of information which will be mentioned in Chapter 6 - Issues.

If there is conflict but no list can be found, it will be appended as ‘Conflicts cannot be scraped’. On the other hand, if there is no conflict, then it will be appended as ‘No conflict in conflict of interest’. The second part of the code is to account for other errors in the scraping such as stale element exception and return the error that occurs. Should an error occur, the retry function will be activated to do a second check to ensure that the error is not erroneous.



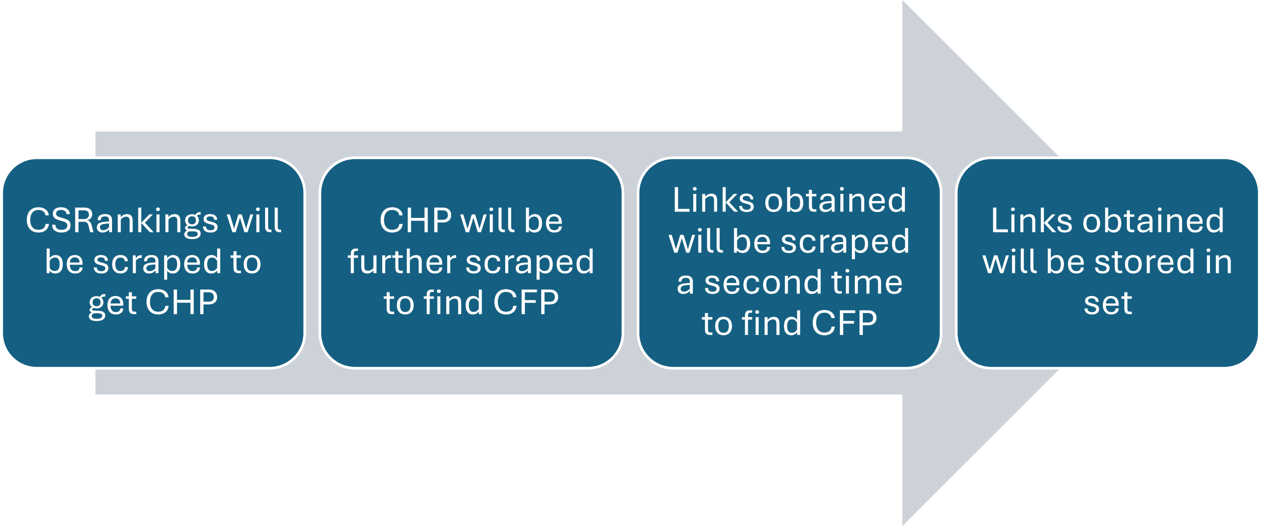
```
def search_conflict_of_interest(driver, url, retries=3):
    try:
        driver.get(url)
        wait = WebDriverWait(driver, 20)
        # Searching for the "conflict of interest" text
        xpath_expression = "//*[contains(translate(normalize-space(text()), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstu
        parent_elements = wait.until(EC.presence_of_all_elements_located((By.XPATH, xpath_expression)))

        # If "conflict of interest" found
        if parent_elements:
            conflicts = []
            for parent_element in parent_elements:
                # Search for following sibling lists or descendant lists
                following_siblings = parent_element.find_elements(By.XPATH, "./following-sibling::*[self::ul or self::ol or cont
                descendants = parent_element.find_elements(By.XPATH, ".//descendant::*[self::ul or self::ol or contains(@class,
                combined_elements = following_siblings + descendants

                # If lists are found, add them to conflicts
                for element in combined_elements:
                    conflicts.append(element.text)

            # If no lists found but conflict found
            if not conflicts:
                return ["Conflicts cannot be scraped"], 1
            return conflicts, 1
        else:
            # If no conflict found
            return ["No conflict in conflict of interest"], 0

    except (TimeoutException, StaleElementReferenceException) as e:
        if retries > 0:
            return search_conflict_of_interest(driver, url, retries - 1)
        else:
            print(f"Error during search on {url} : {e}")
            return ["Error during search"], 0

    except Exception as e:
        print(f"Error during search on {url}: {e}")
        return ["Error during search"], 0
```

4.4 Main

4.4.1 Scraping CSRankings

The webpage CSRankings which is a webpage containing the links to the home page of the different competitions (competition home page, (CHP)) will be scraped to find the links to the different competition home page. Next, these competition home pages will be further scraped to find the page containing information related to the competition page such as call for paper page (CFP page).

As first round of scraping of the CHP may lead to just a list of competitions that are happening, another round of scraping is needed to obtain the CFP page, leading us to undergo 2 rounds of scraping of the competition home page to find the CFP page. This will be further explained in Chapter 4.4.2.


CSRankings will be scraped to get CHP

CHP will be further scraped to find CFP

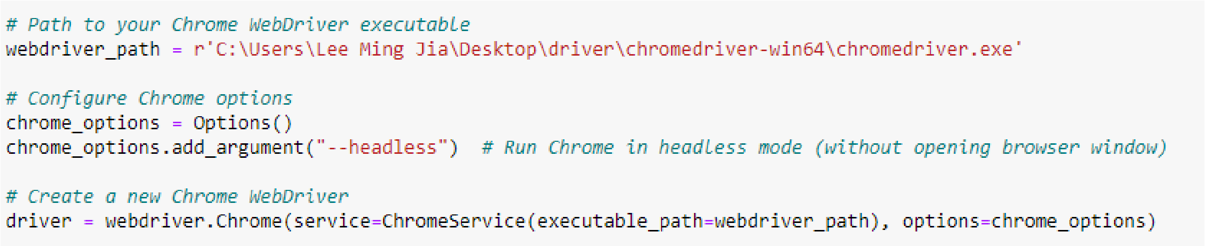
Links obtained will be scraped a second time to find CFP

Links obtained will be stored in set

First, the webdriver path to the chrome driver will be stated as webdriver_path.

This is due to the usage of a chrome driver to enable the code to access the browser. Thus, the chrome driver is left on the computer and the path to it needs to be stated so that the code is able to find it and use it to access the browser to access the webpage.

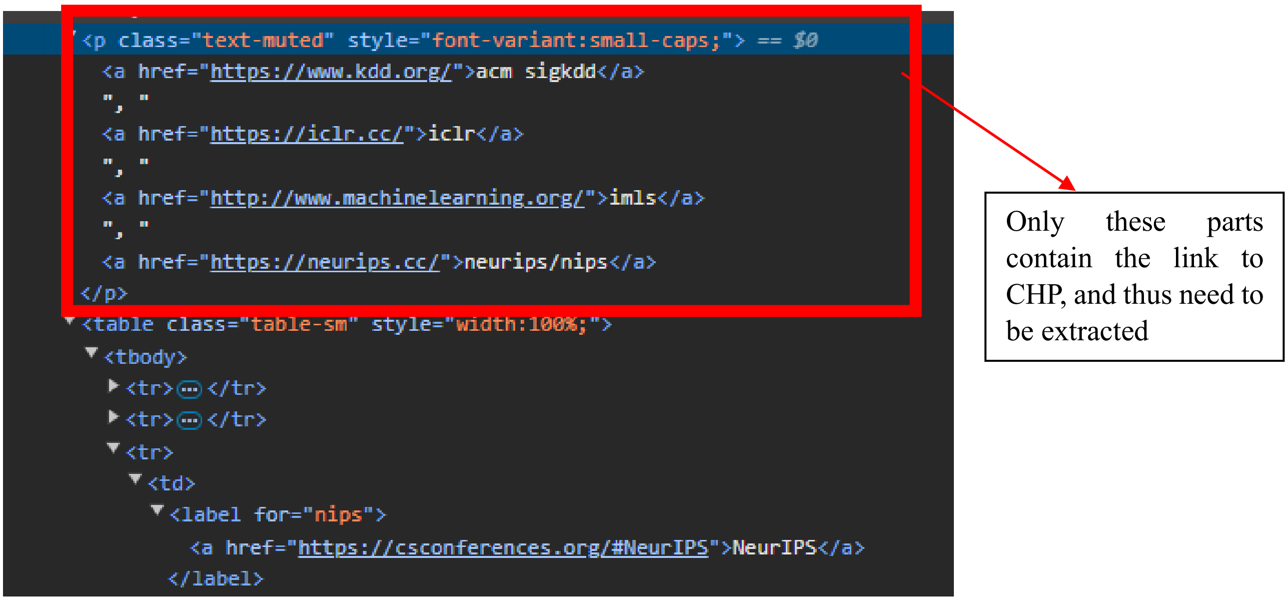
Next, using Options from Selenium, condition for the driver will be set such as “–headless”. This is to prevent excessive pop ups of separate browser while the code is running. Driver will then be created using these conditions that have been set. The webdriver_path give the code the path to the driver which will be used to access the web browser.


```
# Path to your Chrome WebDriver executable
webdriver_path = r'C:\Users\Lee Ming Jia\Desktop\driver\chromedriver-win64\chromedriver.exe'

# Configure Chrome options
chrome_options = Options()
chrome_options.add_argument("--headless")  # Run Chrome in headless mode (without opening browser window)

# Create a new Chrome WebDriver
driver = webdriver.Chrome(service=ChromeService(executable_path=webdriver_path), options=chrome_options)
```

Lastly, css_selector is used to find specific elements from webpage that is wanted and stored as div_element.

This is further filtered into p_element a second time via css_seletor again. This is done so that the link from just specific parts of the webpage CSRanking can be extracted.

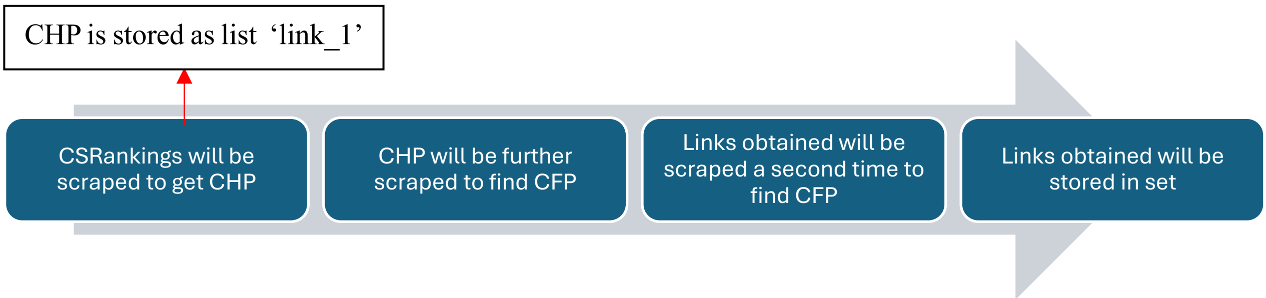
This can be seen below where the links to be extracted are those under ‘p class= “text-muted” and not those links under ‘label for = ‘nips” and so forth.

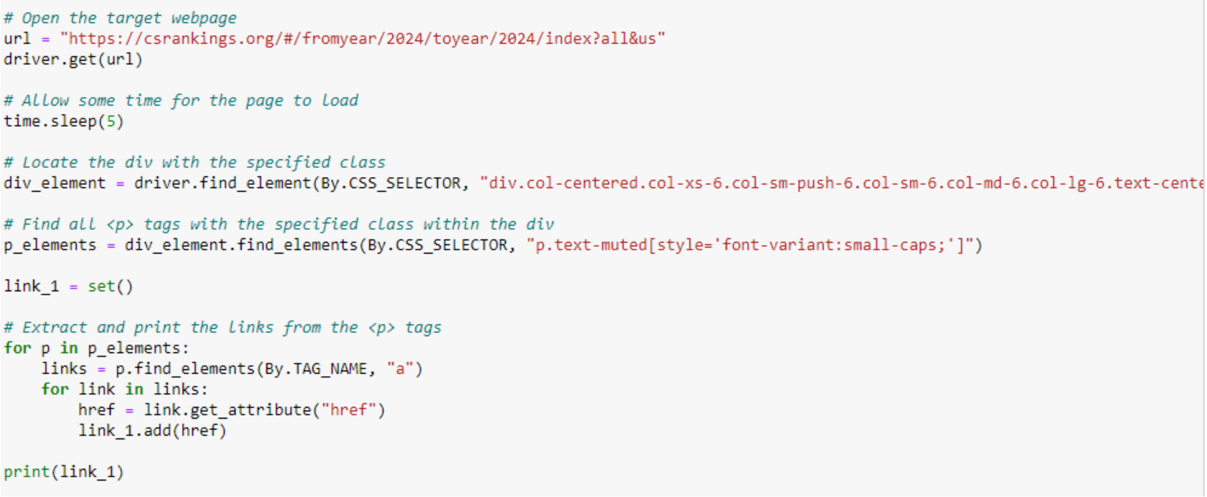


```
<p class="text-muted" style="font-variant:small-caps;"> == $0
  <a href="https://www.kdd.org/">acm sigkdd</a>
  ", "
  <a href="https://iclr.cc/">iclr</a>
  ", "
  <a href="http://www.machinelearning.org/">imls</a>
  ", "
  <a href="https://neurips.cc/">neurips/nips</a>
</p>
▼<table class="table-sm" style="width:100%;">
  ▼<tbody>
    ▶<tr>⋯</tr>
    ▶<tr>⋯</tr>
    ▼<tr>
      ▼<td>
        ▼<label for="nips">
            <a href="https://csconferences.org/#NeurIPS">NeurIPS</a>
          </label>
```

Only these parts contain the link to CHP, and thus need to be extracted

List ‘link_1’ is created to store the links (CHP) that will be extracted from CSRankings. The next part will then take elements in p_elements, find the hyper link via tag name ‘a’ and append the links into ‘link_1’.


CHP is stored as list ‘link_1’
CSRankings will be scraped to get CHP
CHP will be further scraped to find CFP
Links obtained will be scraped a second time to find CFP
Links obtained will be stored in set



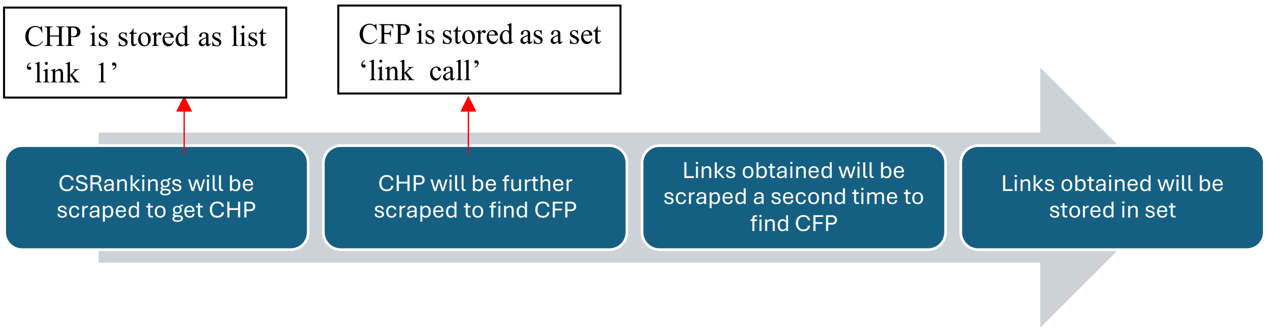
```
# Open the target webpage
url = "https://csrankings.org/#/fromyear/2024/toyear/2024/index?all&us"
driver.get(url)

# Allow some time for the page to load
time.sleep(5)

# Locate the div with the specified class
div_element = driver.find_element(By.CSS_SELECTOR, "div.col-centered.col-xs-6.col-sm-push-6.col-sm-6.col-md-6.col-lg-6.text-cente

# Find all <p> tags with the specified class within the div
p_elements = div_element.find_elements(By.CSS_SELECTOR, "p.text-muted[style='font-variant:small-caps;']")

link_1 = set()

# Extract and print the links from the <p> tags
for p in p_elements:
    links = p.find_elements(By.TAG_NAME, "a")
    for link in links:
        href = link.get_attribute("href")
        link_1.add(href)

print(link_1)
```

4.4.2 scraping for links from link_1

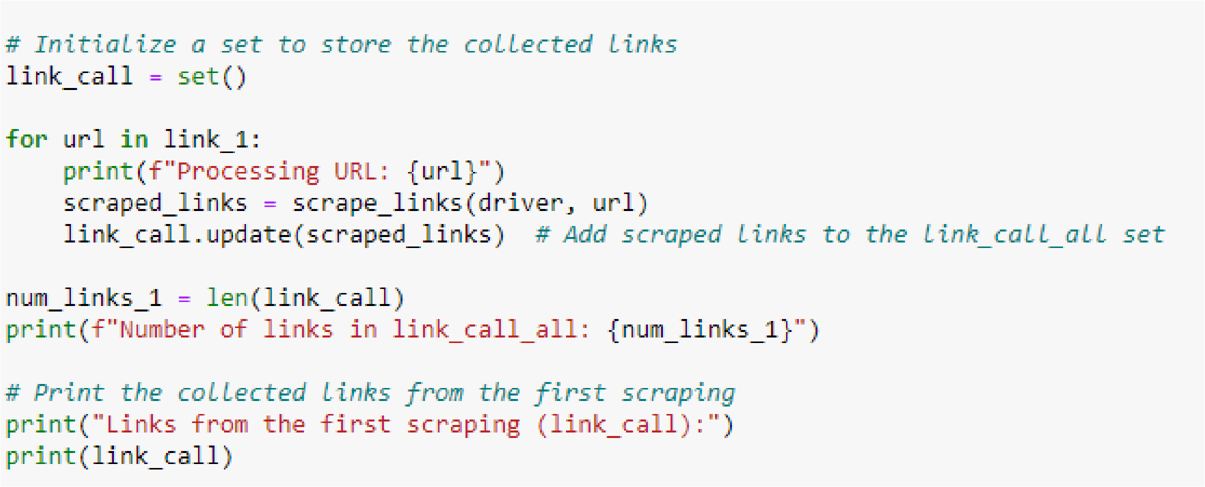
As mentioned in section 4.4.1, further scraping will be done on ‘link_1’ to get CFP page from the CHP. As CHP does not contain information about the competition, there is a need to obtain the CFP page that contains the desired information which is COI.

Links and url from ‘link_1’ will be used to try and find the page containing information about the competition. To do this, links will be extracted from ‘link_1’ with ‘scrape_links’ to obtain the first round of CFP pages. The set ‘link_call’ will be created via ‘link_call = set()’. This will be used to store the links obtained from this scraping. A set is used here instead of a list to prevent same links from being updated into ‘link_call’.

By iterating through the items in ‘link_1’, ‘scrape_links’ will be run and return ‘href_1’ from the function. This ‘href_1’ will then be updated in the ‘link_call’. This will continue on until all the links in ‘link_1’ has been done.

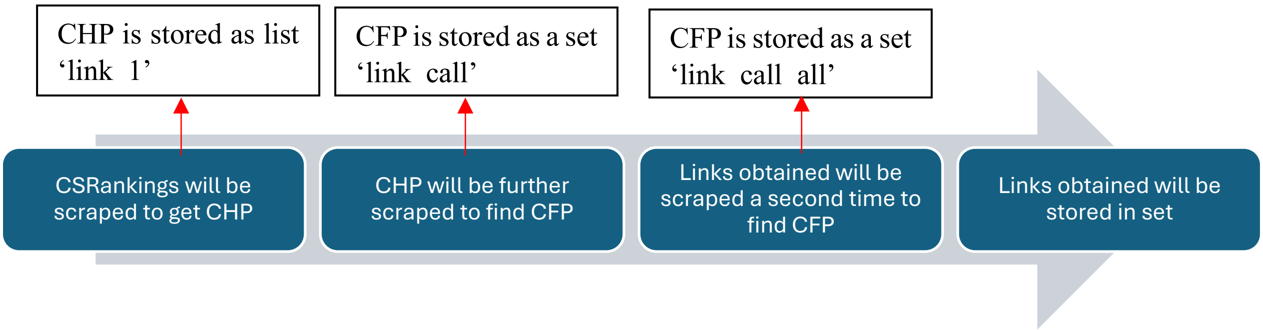


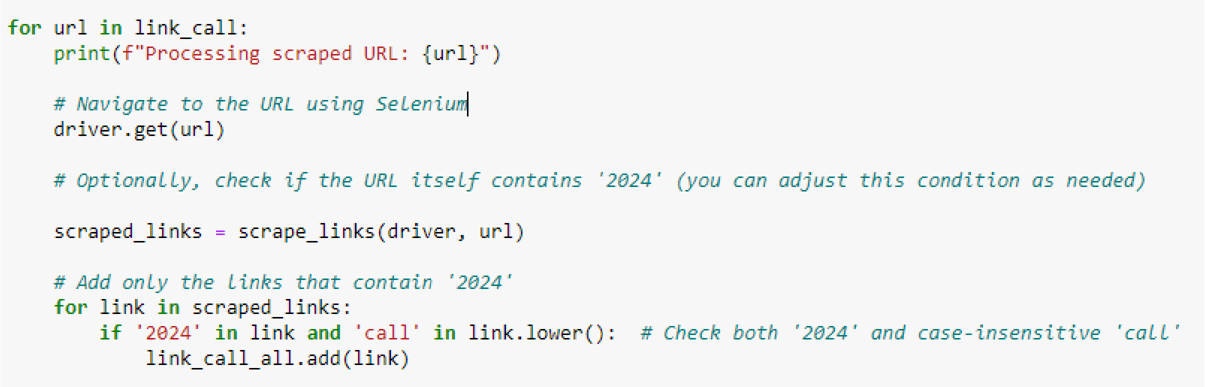
CHP is stored as list ‘link 1’
CFP is stored as a set ‘link call’
CSRankings will be scraped to get CHP
CHP will be further scraped to find CFP
Links obtained will be scraped a second time to find CFP
Links obtained will be stored in set



```
# Initialize a set to store the collected links
link_call = set()

for url in link_1:
    print(f"Processing URL: {url}")
    scraped_links = scrape_links(driver, url)
    link_call.update(scraped_links)  # Add scraped links to the link_call_all set

num_links_1 = len(link_call)
print(f"Number of links in link_call_all: {num_links_1}")

# Print the collected links from the first scraping
print("Links from the first scraping (link_call):")
print(link_call)
```

A second round of the scraping will be done to increase the chances of getting the links that have the CFP. This is done via a second set ‘link_call_all’. In this set, there is a further filtering of the links that will be updated into ‘link_call_all’ where it will check if ‘call’ and ‘2024’ is in the link before adding it into ‘link_call_all’. There are checks for ‘call’ and ‘2024’ in the link as those CFP have a high probability of having the word ‘call’ in the url. CFP is also generally called ‘call for papers’ with some variations such as using just the word ‘call’ and so forth. There is also the limiting of urls to contain the number ‘2024’ so that only the CFP that happen in the year 2024 is scraped. By using both, the url scraped can be restricted to only CFP and the year 2024.


CHP is stored as list ‘link 1’
CFP is stored as a set ‘link call’
CFP is stored as a set ‘link call all’
CSRankings will be scraped to get CHP
CHP will be further scraped to find CFP
Links obtained will be scraped a second time to find CFP
Links obtained will be stored in set



```
for url in link_call:
    print(f"Processing scraped URL: {url}")

    # Navigate to the URL using Selenium
    driver.get(url)

    # Optionally, check if the URL itself contains '2024' (you can adjust this condition as needed)

    scraped_links = scrape_links(driver, url)

    # Add only the links that contain '2024'
    for link in scraped_links:
        if '2024' in link and 'call' in link.lower():  # Check both '2024' and case-insensitive 'call'
            link_call_all.add(link)
```

4.4.3 Searching for conflict of interest

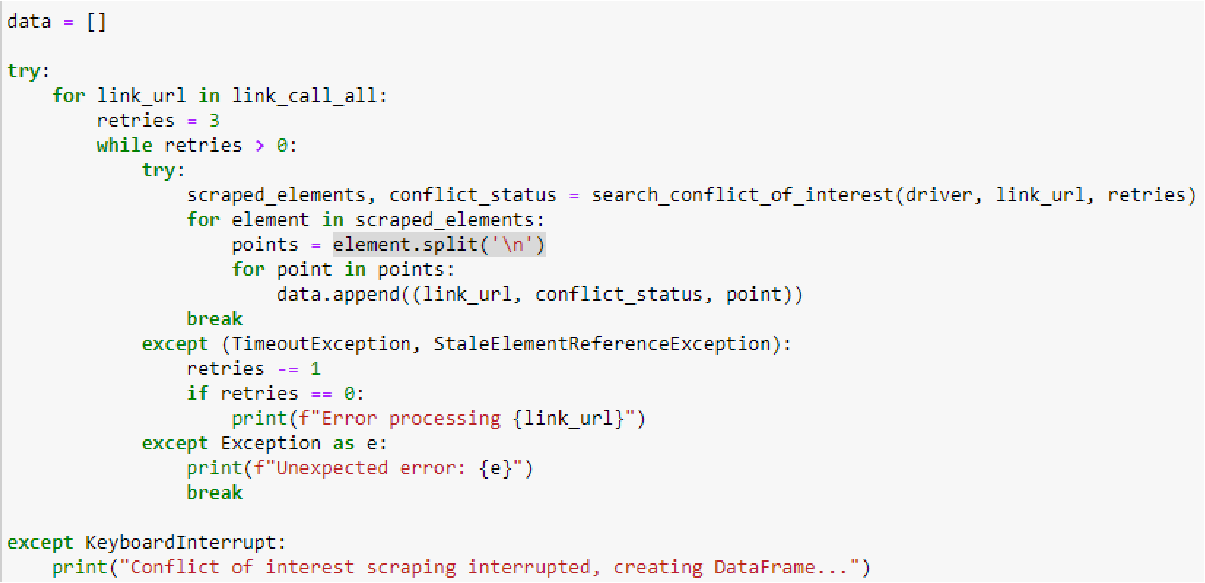
The links that are stored in ‘link_call_all’ will then be scraped to detect if there is COI. Since the links in ‘link_call_all’ have been filtered earlier, it would reduce the number of webpages that needs to be scraped by the function ‘search_conflict_of_interest’. From this scraping, information about COI will be extracted from the webpage and stored as a list in a format ready to be extracted out and stored in a dataframe.

List ‘data’ is created to store the conflict of interest obtained from the function. ‘search_conflict_of_interest’ will return ‘scraped_elements’ and ‘conflict_status’

- ‘scraped elements’ is a list of COI or error for those that cannot be scraped.
- ‘conflict_status’ is either 1 or 0 depending on if any conflict is found where 1 mean conflict exists in webpage and 0 for no conflict found in webpage.

The item in ‘scraped_elements’ is processed by using split(‘\n’) to split up the words in the item. These will then be appended into data as

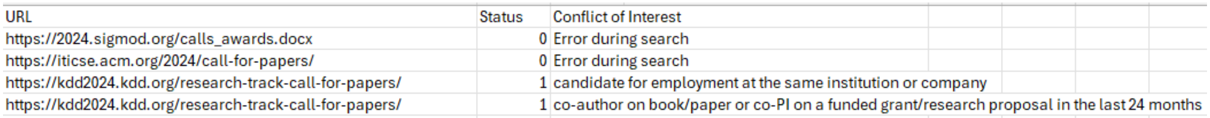
- ‘link_url’ which contains the link,
- ‘conflict_status’
- ‘point’ which contains the items in ‘scraped_item’.



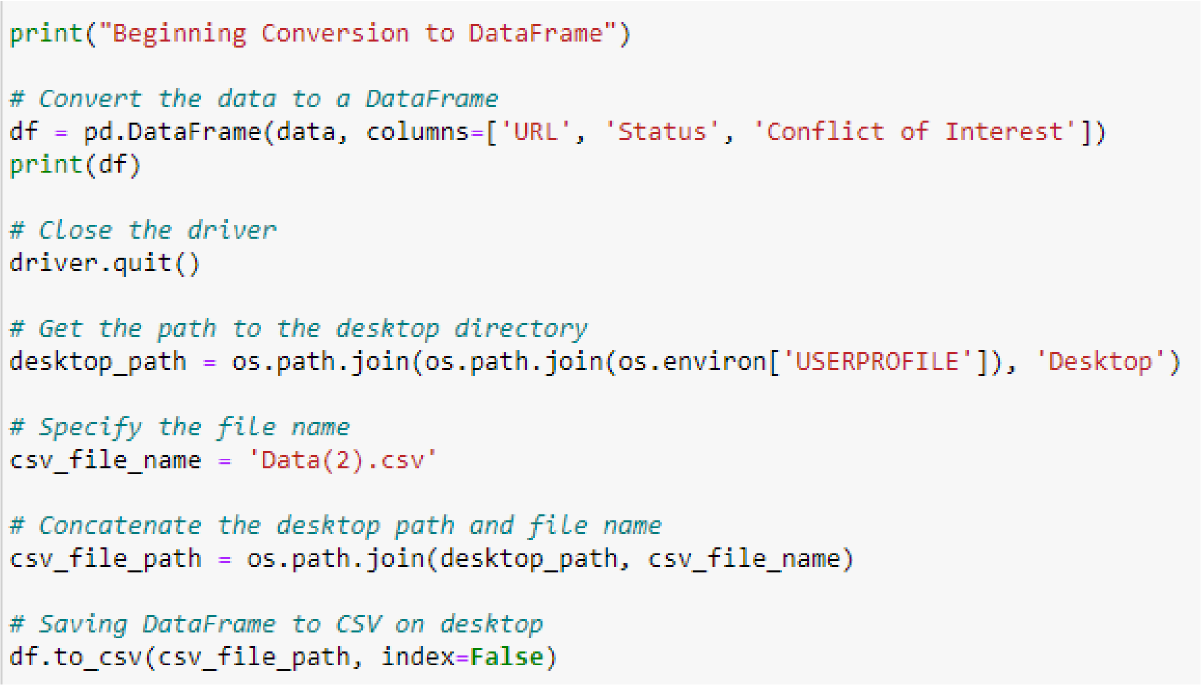
```
data = []

try:
    for link_url in link_call_all:
        retries = 3
        while retries > 0:
            try:
                scraped_elements, conflict_status = search_conflict_of_interest(driver, link_url, retries)
                for element in scraped_elements:
                    points = element.split('\n')
                    for point in points:
                        data.append((link_url, conflict_status, point))
                break
            except (TimeoutException, StaleElementReferenceException):
                retries -= 1
                if retries == 0:
                    print(f"Error processing {link_url}")
            except Exception as e:
                print(f"Unexpected error: {e}")
                break

except KeyboardInterrupt:
    print("Conflict of interest scraping interrupted, creating DataFrame...")
```

4.4.4 Dataframe

The information in the list ‘data’ will then be converted into dataframe with the columns ‘URL’, ‘Status’, ‘Conflict of Interest’. This dataframe will then be converted into csv file for our easy view and access.



URL	Status	Conflict of Interest
https://2024.sigmod.org/calls_awards.docx	0	Error during search
https://iticse.acm.org/2024/call-for-papers/	0	Error during search
https://kdd2024.kdd.org/research-track-call-for-papers/	1	candidate for employment at the same institution or company
https://kdd2024.kdd.org/research-track-call-for-papers/	1	co-author on book/paper or co-PI on a funded grant/research proposal in the last 24 months



```
print("Beginning Conversion to DataFrame")

# Convert the data to a DataFrame
df = pd.DataFrame(data, columns=['URL', 'Status', 'Conflict of Interest'])
print(df)

# Close the driver
driver.quit()

# Get the path to the desktop directory
desktop_path = os.path.join(os.path.join(os.environ['USERPROFILE']), 'Desktop')

# Specify the file name
csv_file_name = 'Data(2).csv'

# Concatenate the desktop path and file name
csv_file_path = os.path.join(desktop_path, csv_file_name)

# Saving DataFrame to CSV on desktop
df.to_csv(csv_file_path, index=False)
```

5. Analysis

The below tables list some of the events obtained from the code and the CFP page that was manually found and checked for COI in different CFP pages. In this chapter, the CFP page will be classified under event here. The differences between the conflicts that was successfully scraped via the code versus the conflicts that were manually scraped will be analyzed to identify the biggest issue that resulted in the differences between the data from code and the data from manual. With this, it will enable the user to understand what are the problems that were faced and need to be resolved for improvement of the code for higher accuracy.

Table 1. Comparing manual checks vs code checks

	Manual[2]	Code[3]
No. of event[1]	74	169
No. of events with COIs detected	31	21
Event in manual but not in code (CFP not found)	69	
Webpage not in manual but in code (Non-CFP event)		164
Successfully scraped COI from CFP		5

Table 2. Issue with scraping COI in manual

	Manual
No of event (CFP)	74
No. of events with COIs detected	31
- No issue	5
- COI not under ul or ol element or in p element	11
- Issue to COI being on another page (code can scrape if in the right page)	4
- COI will return extra items than what is required	4
- Others[4]	7

[1] *'Event' in this context refer to webpages containing information about different types of activities such as competitions, conference, award etc.*

[2] *'Manual' in this context refer to one manually checking every event (CFP) one by one to see if there is COI mentioned*

[3] *'Code' in this context refer to the events that are found by the code via webpage scraping.*

[4] *Refer to events that has different issues from the others such as conflict in pdf file and such*

Table 3. Common links found between manual checks and code

Common	5
- Success scraped and event is in manual (CFP event that COI was found)	3
- Success scraped but event not in manual (non-CFP event that COI was found)	1
- Common event found but failed scraped (CFP event but COI not found)	1

Table 1 shows the differences between events obtained by checking manually and events that are obtained via the code. From Table 1, through manual checks, 31 out of 74 events (CFP) were identified to contain COI. From the data that was scraped from the code, 21 out of 169 events were identified to contain COI. From this, we can observe a huge difference in the number of events that is obtained via manual check and code scraping.

This is due to the code scraping obtaining events that are not related to CFP. This can be seen in table 1 where:

- Out of the 74 events in manual check which are CFP page, the code was unable to scrape and find 69 of them.
- Out of the 169 events in code scraping, 164 of the events are events not listed in manual check.
- Only 5 of the events obtained from code scraping were CFP event listed in manual check.
- For the 5 events that are successfully scraped for COI from code, 3 of them are CFP event listed in manual while the last 2 are not CFP events listed in manual check

Out of the 5 common events between code and manual, as shown in Table 3, 3 of the events were able to be successfully scraped (CFP with COI), another event was successfully scraped but event was not found in manual (non CFP event with COI) and the last event did not manage to scrape any COI (CFP event but no COI) .

Table 2 shows the breakdown of the 31 events from manual check that has COI. The most common issue is that COI is not in ul or ol element or in p element which is usually used for paragraph. This issue will be further explained in Chapter 6 - Issues. From table 2, it can also be observed that there are also many other issues with regards to scraping for COI that need to be resolved as well.

For the items that is scraped from the code, the below COI are the most common ones to be mentioned:

1. The PC member is a co-author of the paper
2. The PC member has been a co-worker in the same company or university
3. The PC member has been a collaborator
4. The PC member is or was the author's primary thesis advisor, or post-doctoral advisor
5. The PC member is a relative or close personal friend of the author

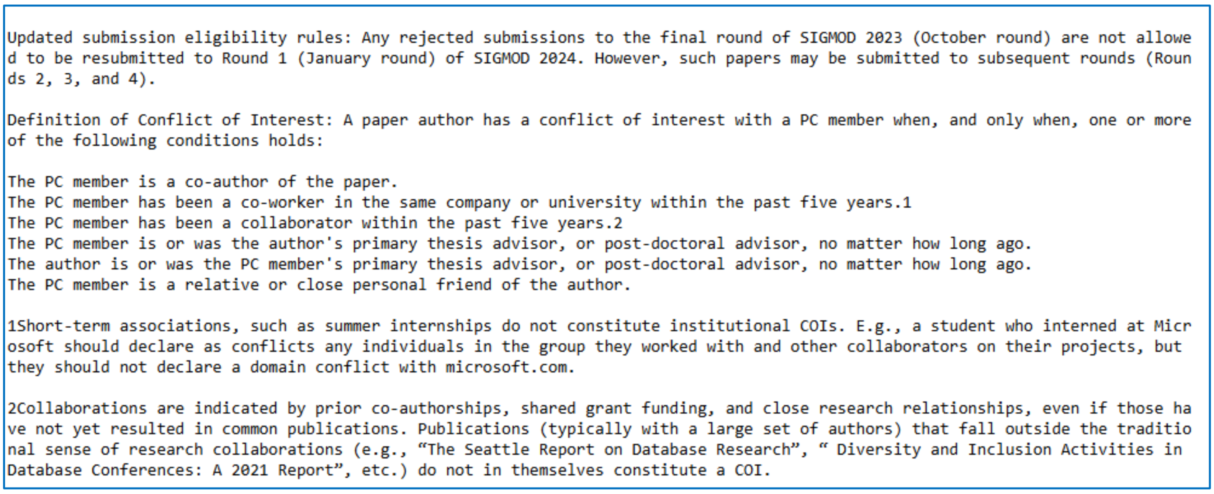
When comparing the common COI obtained via code via manual checks, both of them have mostly similar COI with the exception of COI related to financial aid and funding.

From this, it can be seen there are still many issues with the code and much improvements to be made as the number of COI that could not be scraped and the success obtained from the code are less than expected. This will be further addressed in the Chapter 6 and 7.

6. Issues

While implementing the code, some difficulties were encountered as indicated below.

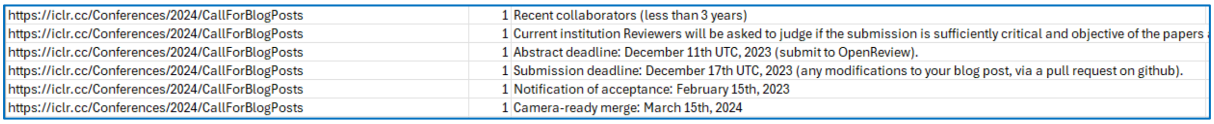
Firstly, as the code can only scrape items that are in ol or ul element, this would result in those webpages that have the COI placed in either p element or other elements from being scraped by the code. However, we cannot allow the code to look for p element as this will lead to the issue where the entire paragraph is scraped. Below shows the result of the scrapping if the limitation of item in ol or ul element is not imposed.



```
Updated submission eligibility rules: Any rejected submissions to the final round of SIGMOD 2023 (October round) are not allowe
d to be resubmitted to Round 1 (January round) of SIGMOD 2024. However, such papers may be submitted to subsequent rounds (Roun
ds 2, 3, and 4).

Definition of Conflict of Interest: A paper author has a conflict of interest with a PC member when, and only when, one or more
of the following conditions holds:

The PC member is a co-author of the paper.
The PC member has been a co-worker in the same company or university within the past five years.1
The PC member has been a collaborator within the past five years.2
The PC member is or was the author's primary thesis advisor, or post-doctoral advisor, no matter how long ago.
The author is or was the PC member's primary thesis advisor, or post-doctoral advisor, no matter how long ago.
The PC member is a relative or close personal friend of the author.

1Short-term associations, such as summer internships do not constitute institutional COIs. E.g., a student who interned at Micr
osoft should declare as conflicts any individuals in the group they worked with and other collaborators on their projects, but
they should not declare a domain conflict with microsoft.com.

2Collaborations are indicated by prior co-authorships, shared grant funding, and close research relationships, even if those ha
ve not yet resulted in common publications. Publications (typically with a large set of authors) that fall outside the traditio
nal sense of research collaborations (e.g., "The Seattle Report on Database Research", " Diversity and Inclusion Activities in
Database Conferences: A 2021 Report", etc.) do not in themselves constitute a COI.
```

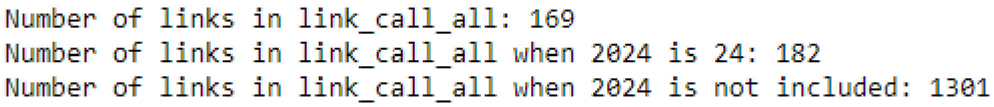
Secondly, since the code also look for all following siblings and descendant, it would result in the subsequent items after the COI to be scraped as well, thus the limitation of ul or ol element is needed to prevent the issue from occurring. This is to reduce excessive information that is not related to COI from being scraped as it would scrape all the information that is stored in paragraph and such.

However, this limitation of ul or ol element would also result in scraping more details for COI or scraping wrong details. This is due to the code where it will find the text 'conflict' in the class, id and from there, search the descendant and descending sibling for any items that is in ul or ol element. This will result in other details that are in another list that are just below it from being scraped as well, resulting in excessive information that is not related to COI being scraped. Examples of this can be seen below

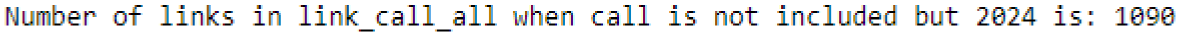


https://iclr.cc/Conferences/2024/CallForBlogPosts	1	Recent collaborators (less than 3 years)
https://iclr.cc/Conferences/2024/CallForBlogPosts	1	Current institution Reviewers will be asked to judge if the submission is sufficiently critical and objective of the papers
https://iclr.cc/Conferences/2024/CallForBlogPosts	1	Abstract deadline: December 11th UTC, 2023 (submit to OpenReview).
https://iclr.cc/Conferences/2024/CallForBlogPosts	1	Submission deadline: December 17th UTC, 2023 (any modifications to your blog post, via a pull request on github).
https://iclr.cc/Conferences/2024/CallForBlogPosts	1	Notification of acceptance: February 15th, 2023
https://iclr.cc/Conferences/2024/CallForBlogPosts	1	Camera-ready merge: March 15th, 2024

Thirdly, another issue faced is the problem in limiting the number of URLS that can be obtained from 'scape_links' so that less links will be obtained. This was done via restricting the links that can be scraped in the second round to those links that contain both '2024' and 'call' in the url. Otherwise, there would be a lot more of the links that will be scraped. For example, if we use '24' instead of '2024', there will be 182 links scraped. On the other hand, if '2024' is excluded and only 'call' is included, 1301 links will be scraped. If 'call' is excluded but '2024' is included, 1090 links will be scraped. This can be seen below.



```
Number of links in link_call_all: 169
Number of links in link_call_all when 2024 is 24: 182
Number of links in link_call_all when 2024 is not included: 1301
```



```
Number of links in link_call_all when call is not included but 2024 is: 1090
```

However, this would result in some of the links containing COI being excluded in the ‘link_call_all’ as they contain only ‘24’ instead of ‘2024’, or ‘call’ is not present in the URL.

Such examples are ‘https://www.asplos-conference.org/asplos2024/cfp/index.html’ and ‘https://www.usenix.org/conference/nsdi24/call-for-papers’.

Lastly, there is also the issue of where the COI is placed in another page such as code of ethics, further details or submission page. As the COI is placed in another webpage, the code cannot find and scrape the COI from the webpage.

7. Improvement

To improve the code, a feature can be added where if conflict is in h1 class, then it will only search until it reaches the next h1 class. This will prevent the issue where details not related to COI will be scraped.

With this, the code can scrape those websites that have the conflict in ‘p’ element as the amount of content of being scaped can be controlled. This would resolve the problem where the COI is not in ul or ol but in p element or another element.

Furthermore, this feature can be utilized to search for hyperlinks that are placed in the conflict which can lead to the page containing COI for further scrape to obtain the COI. This can also resolve the problem of COI being left in another webpage as the link to that webpage can be found via the hyperlink that is left in the COI content.

8. References

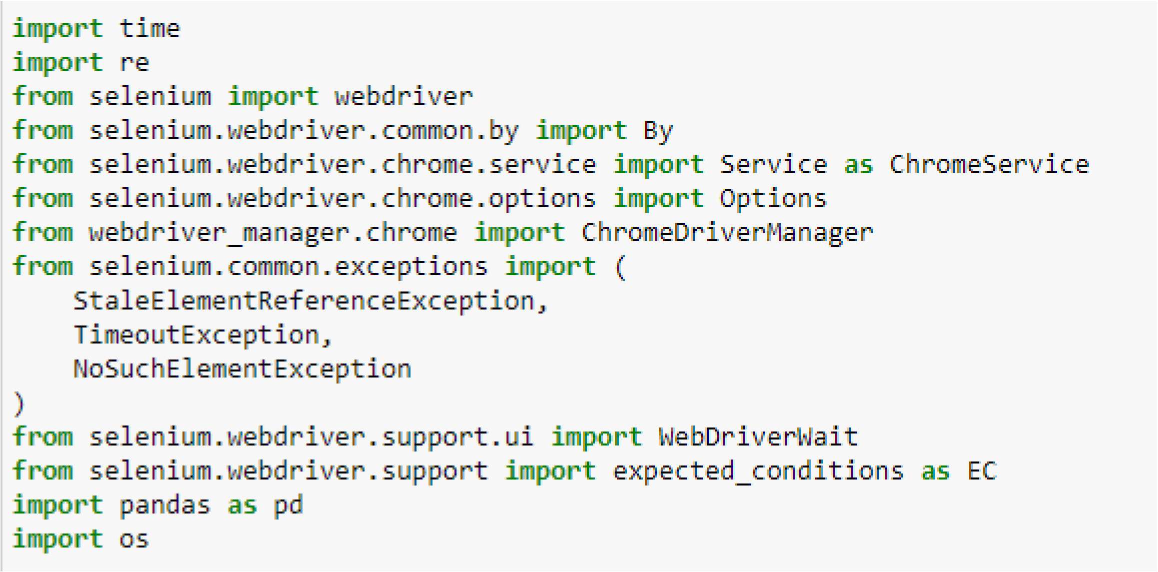
1. Nechvatal, J. , Barker, E. , Bassham, L. , Burr, W. , Dworkin, M. , Foti, J. and Roback, E. (2001), Report on the Development of the Advanced Encryption Standard (AES), Journal of Research (NIST JRES), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=151226 (Accessed November 15, 2024)

2. The 2024 ACM SIGMOD/PODS Conference:Santiago, Chile - Call for Papers: SIGMOD Research. (2024). Sigmod.org. https://2024.sigmod.org/calls_papers_sigmod_research.shtml

3. Selenium. (n.d.). Selenium documentation. Selenium. Retrieved November 12, 2024, from https://www.selenium.dev/documentation/

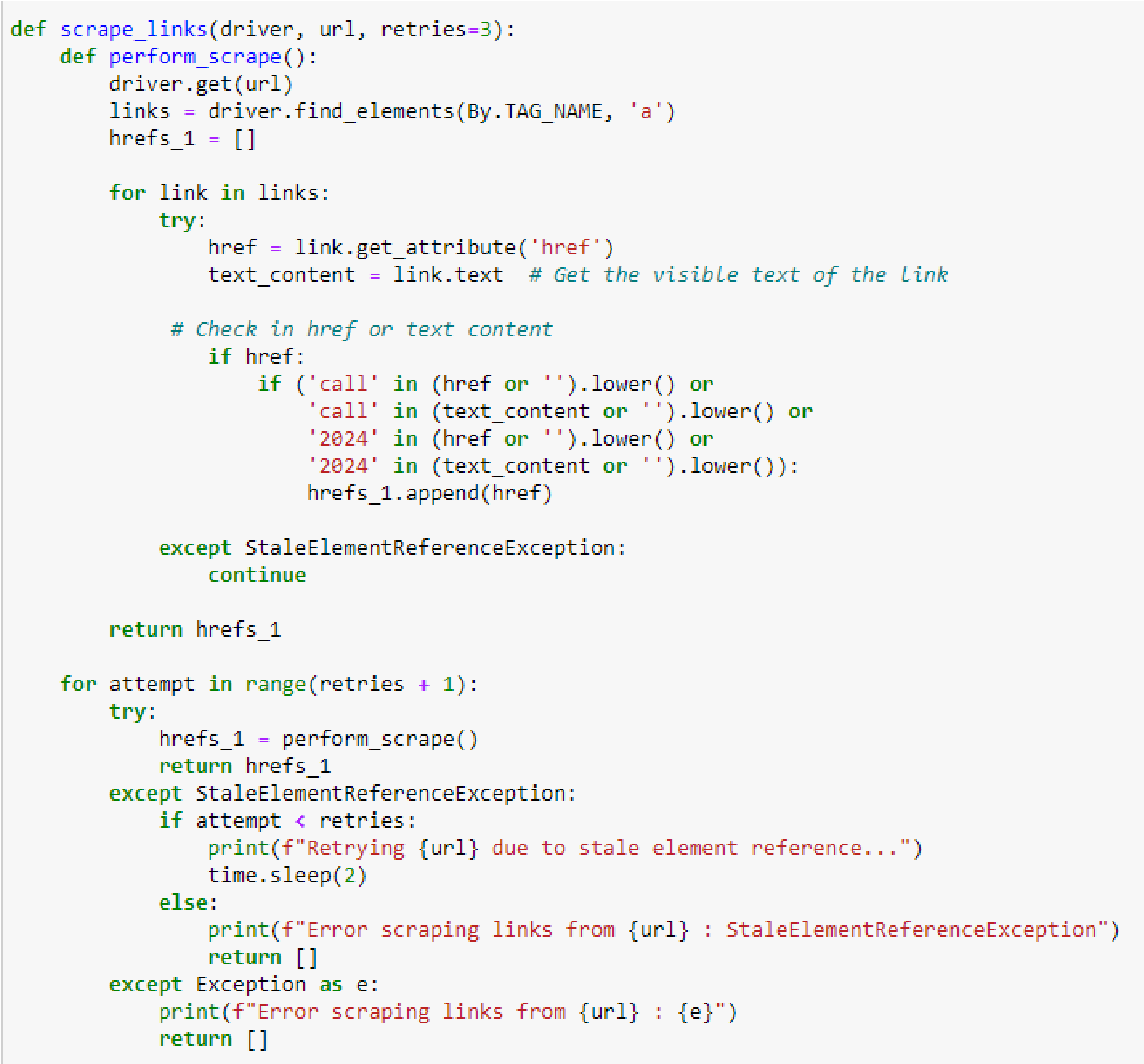
9. Appendix

Packages used:



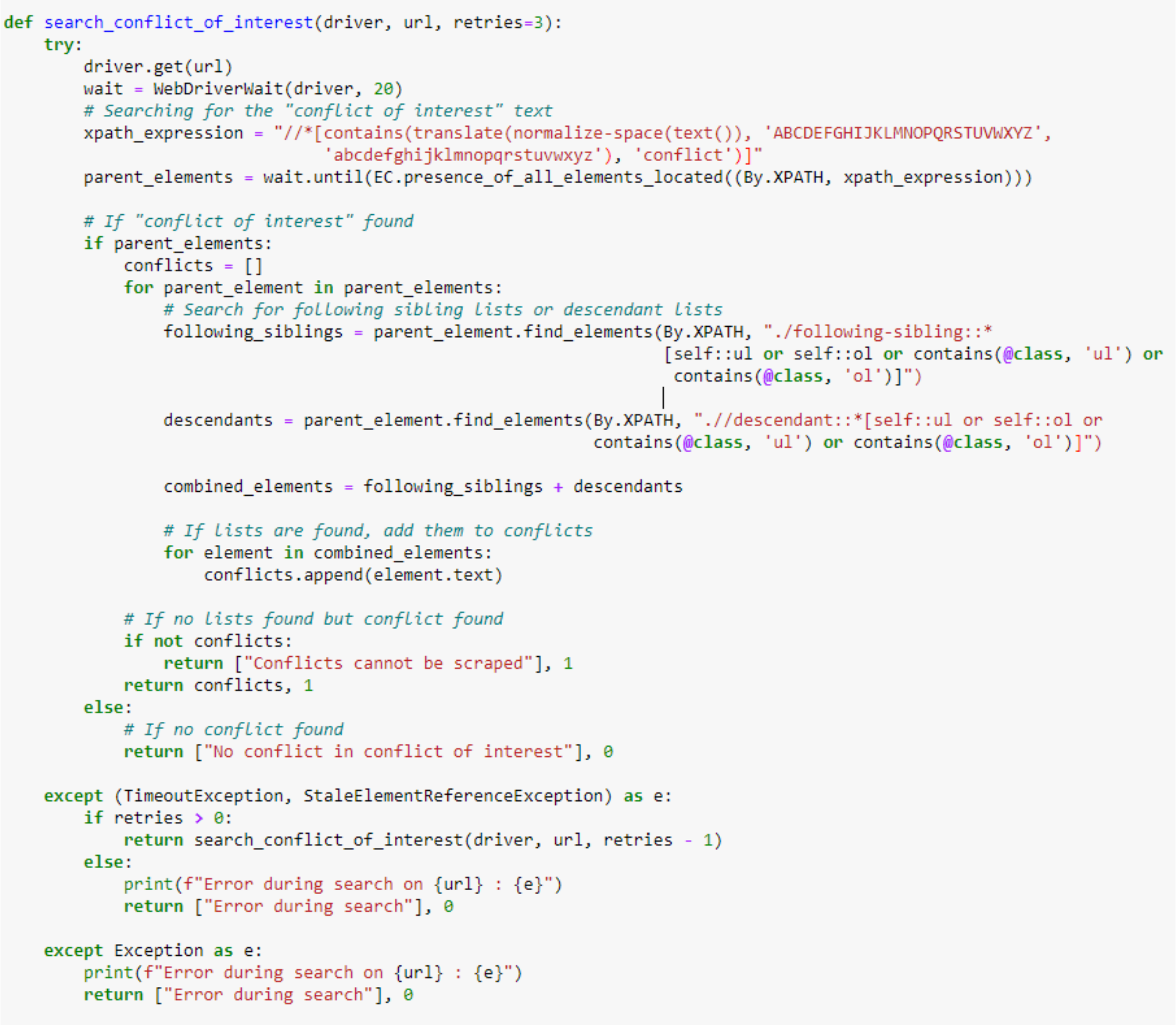
```
import time
import re
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service as ChromeService
from selenium.webdriver.chrome.options import Options
from webdriver_manager.chrome import ChromeDriverManager
from selenium.common.exceptions import (
    StaleElementReferenceException,
    TimeoutException,
    NoSuchElementException
)
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import pandas as pd
import os
```

scrape_links Function:



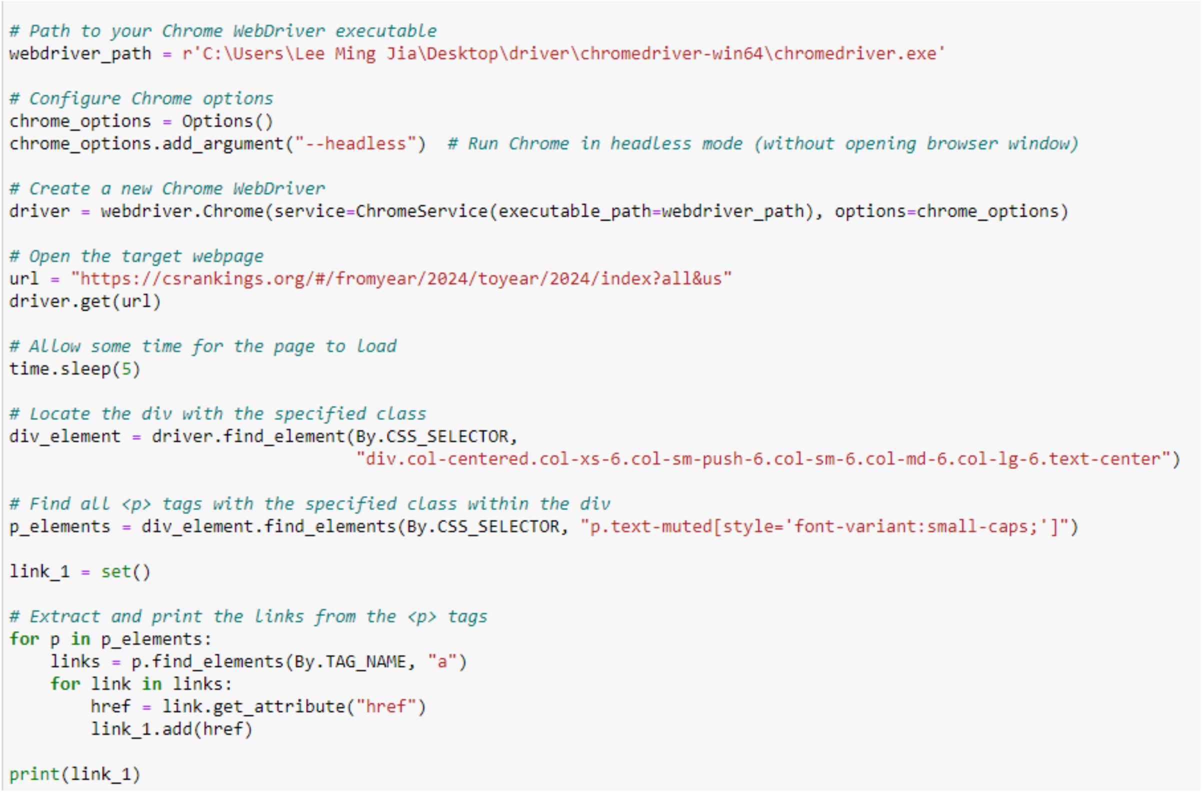
```python
def scrape_links(driver, url, retries=3):
    def perform_scrape():
        driver.get(url)
        links = driver.find_elements(By.TAG_NAME, 'a')
        hrefs_1 = []

        for link in links:
            try:
                href = link.get_attribute('href')
                text_content = link.text  # Get the visible text of the link

             # Check in href or text content
                if href:
                    if ('call' in (href or '').lower() or
                        'call' in (text_content or '').lower() or
                        '2024' in (href or '').lower() or
                        '2024' in (text_content or '').lower()):
                        hrefs_1.append(href)

            except StaleElementReferenceException:
                continue

        return hrefs_1

    for attempt in range(retries + 1):
        try:
            hrefs_1 = perform_scrape()
            return hrefs_1
        except StaleElementReferenceException:
            if attempt < retries:
                print(f"Retrying {url} due to stale element reference...")
                time.sleep(2)
            else:
                print(f"Error scraping links from {url} : StaleElementReferenceException")
                return []
        except Exception as e:
            print(f"Error scraping links from {url} : {e}")
            return []
```

search_conflict_of_interest Function:



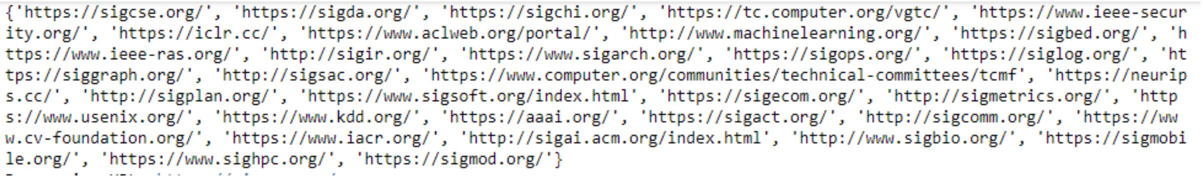
```
def search_conflict_of_interest(driver, url, retries=3):
    try:
        driver.get(url)
        wait = WebDriverWait(driver, 20)
        # Searching for the "conflict of interest" text
        xpath_expression = "//*[contains(translate(normalize-space(text()), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ',
                                'abcdefghijklmnopqrstuvwxyz'), 'conflict')]"
        parent_elements = wait.until(EC.presence_of_all_elements_located((By.XPATH, xpath_expression)))

        # If "conflict of interest" found
        if parent_elements:
            conflicts = []
            for parent_element in parent_elements:
                # Search for following sibling lists or descendant lists
                following_siblings = parent_element.find_elements(By.XPATH, "./following-sibling::*
                                                                  [self::ul or self::ol or contains(@class, 'ul') or
                                                                   contains(@class, 'ol')]")

                descendants = parent_element.find_elements(By.XPATH, ".//descendant::*[self::ul or self::ol or
                                                           contains(@class, 'ul') or contains(@class, 'ol')]")

                combined_elements = following_siblings + descendants

                # If lists are found, add them to conflicts
                for element in combined_elements:
                    conflicts.append(element.text)

            # If no lists found but conflict found
            if not conflicts:
                return ["Conflicts cannot be scraped"], 1
            return conflicts, 1
        else:
            # If no conflict found
            return ["No conflict in conflict of interest"], 0

    except (TimeoutException, StaleElementReferenceException) as e:
        if retries > 0:
            return search_conflict_of_interest(driver, url, retries - 1)
        else:
            print(f"Error during search on {url} : {e}")
            return ["Error during search"], 0

    except Exception as e:
        print(f"Error during search on {url} : {e}")
        return ["Error during search"], 0
```

Code input (Scraping CSRankings):



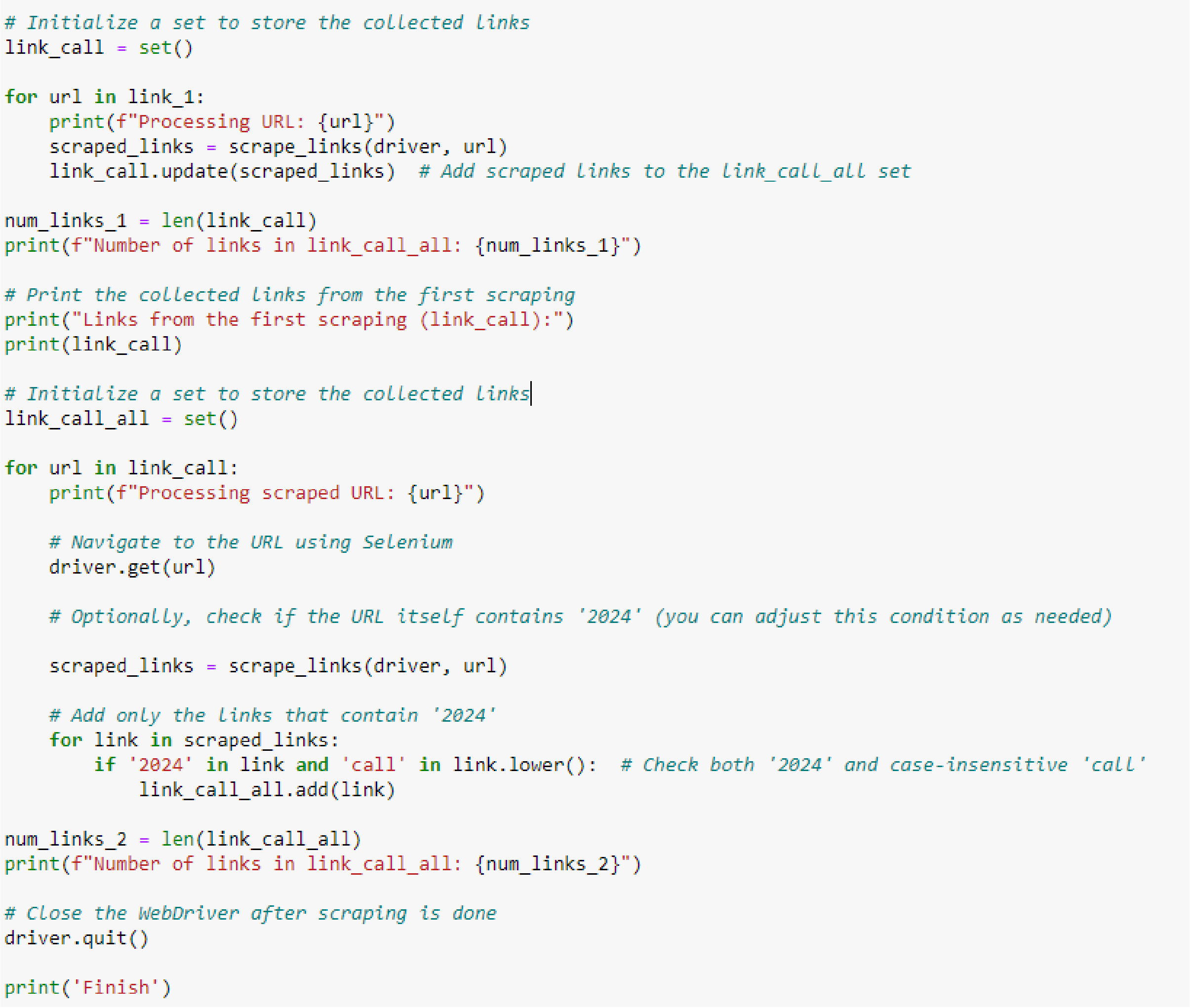
```
# Path to your Chrome WebDriver executable
webdriver_path = r'C:\Users\Lee Ming Jia\Desktop\driver\chromedriver-win64\chromedriver.exe'

# Configure Chrome options
chrome_options = Options()
chrome_options.add_argument("--headless")  # Run Chrome in headless mode (without opening browser window)

# Create a new Chrome WebDriver
driver = webdriver.Chrome(service=ChromeService(executable_path=webdriver_path), options=chrome_options)

# Open the target webpage
url = "https://csrankings.org/#/fromyear/2024/toyear/2024/index?all&us"
driver.get(url)

# Allow some time for the page to load
time.sleep(5)

# Locate the div with the specified class
div_element = driver.find_element(By.CSS_SELECTOR,
                                  "div.col-centered.col-xs-6.col-sm-push-6.col-sm-6.col-md-6.col-lg-6.text-center")

# Find all <p> tags with the specified class within the div
p_elements = div_element.find_elements(By.CSS_SELECTOR, "p.text-muted[style='font-variant:small-caps;']")

link_1 = set()

# Extract and print the links from the <p> tags
for p in p_elements:
    links = p.find_elements(By.TAG_NAME, "a")
    for link in links:
        href = link.get_attribute("href")
        link_1.add(href)

print(link_1)
```

Code output (Scraping CSRankings):



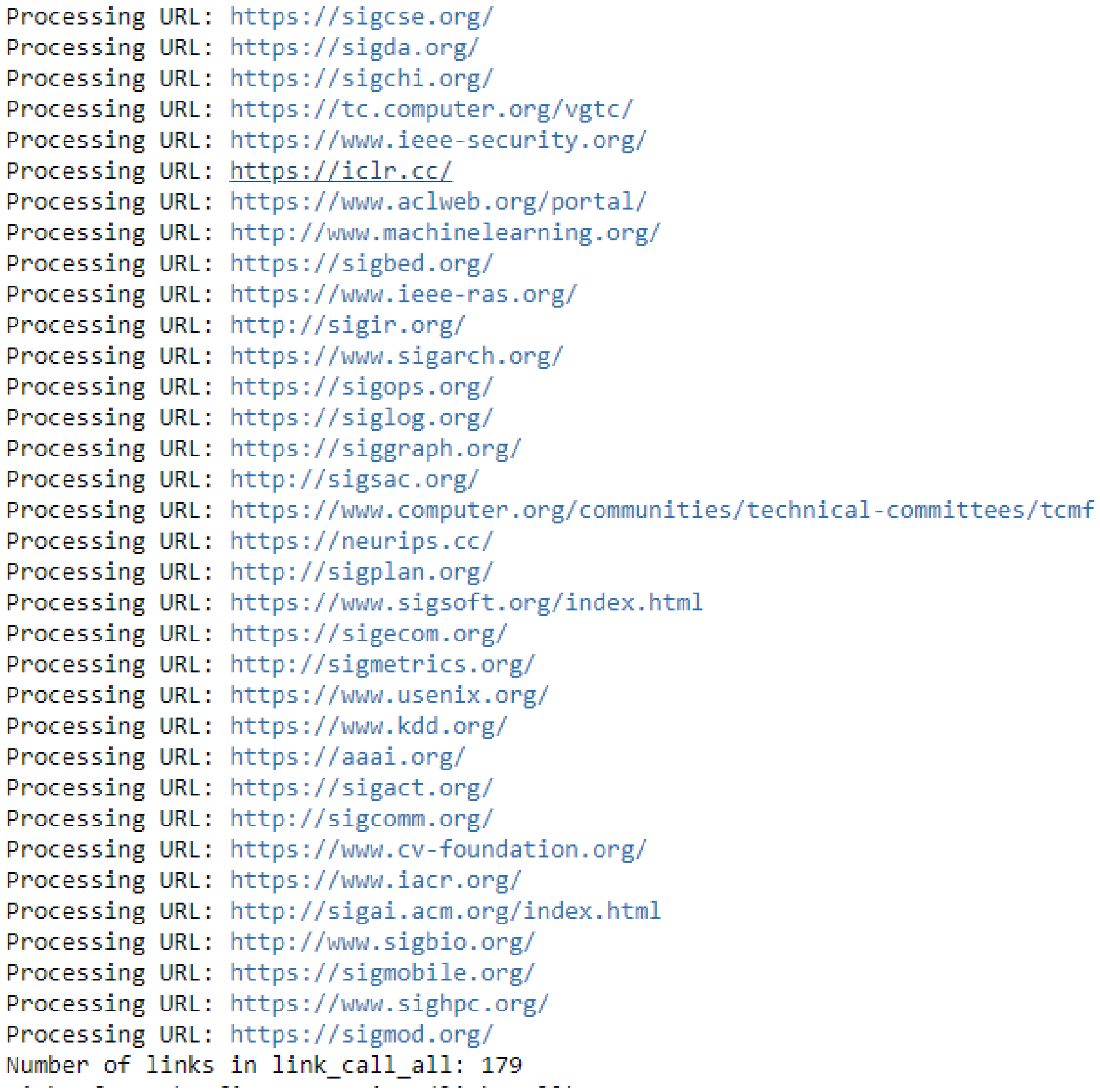
```
{'https://sigcse.org/', 'https://sigda.org/', 'https://sigchi.org/', 'https://tc.computer.org/vgtc/', 'https://www.ieee-secur
ity.org/', 'https://iclr.cc/', 'https://www.aclweb.org/portal/', 'http://www.machinelearning.org/', 'https://sigbed.org/', 'h
ttps://www.ieee-ras.org/', 'http://sigir.org/', 'https://www.sigarch.org/', 'https://sigops.org/', 'https://siglog.org/', 'ht
tps://siggraph.org/', 'http://sigsac.org/', 'https://www.computer.org/communities/technical-committees/tcmf', 'https://neurip
s.cc/', 'http://sigplan.org/', 'https://www.sigsoft.org/index.html', 'https://sigecom.org/', 'http://sigmetrics.org/', 'http
s://www.usenix.org/', 'https://www.kdd.org/', 'https://aaai.org/', 'https://sigact.org/', 'http://sigcomm.org/', 'https://ww
w.cv-foundation.org/', 'https://www.iacr.org/', 'http://sigai.acm.org/index.html', 'http://www.sigbio.org/', 'https://sigmobi
le.org/', 'https://www.sighpc.org/', 'https://sigmod.org/'}
```

Code input (scrape_link):

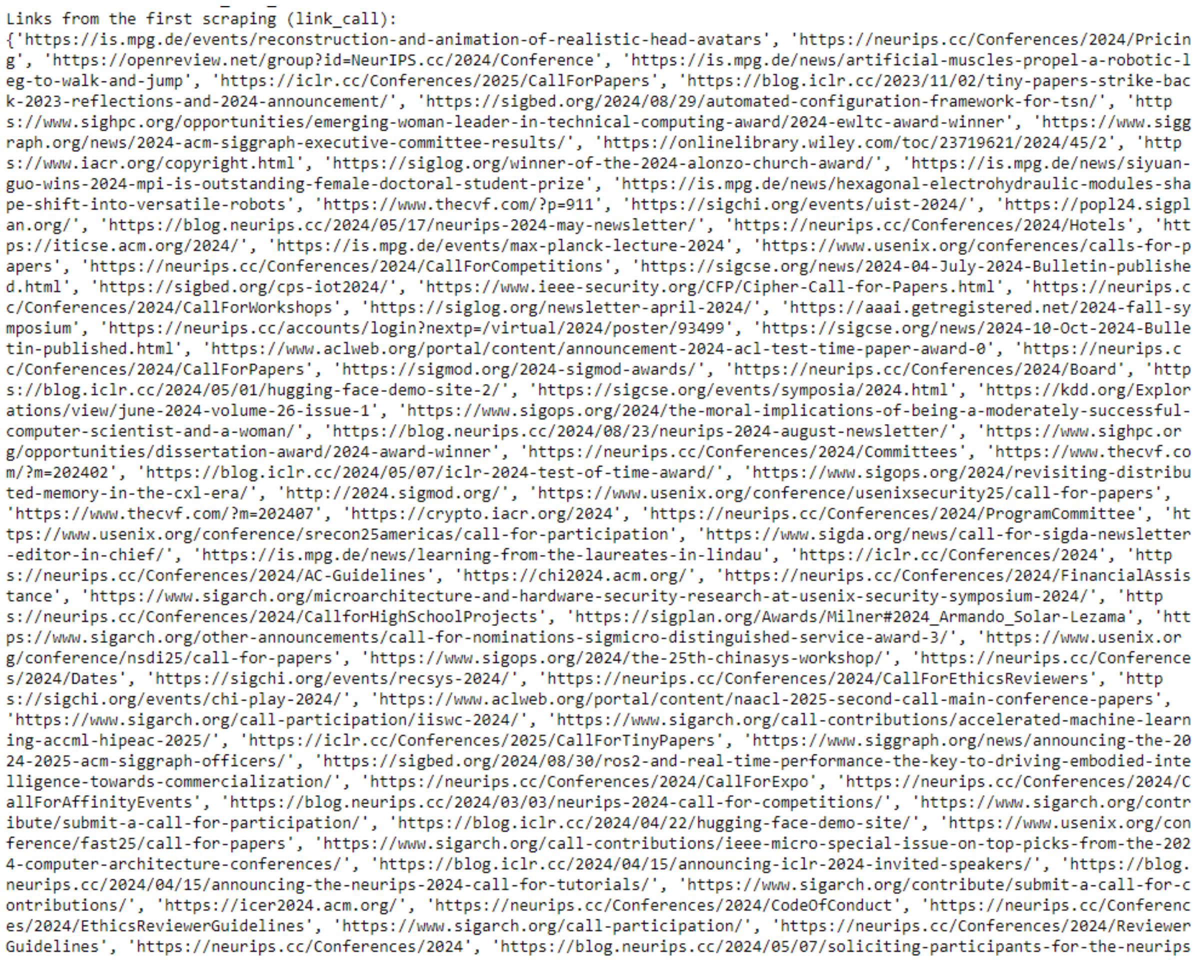


```python
# Initialize a set to store the collected links
link_call = set()

for url in link_1:
    print(f"Processing URL: {url}")
    scraped_links = scrape_links(driver, url)
    link_call.update(scraped_links)  # Add scraped links to the link_call_all set

num_links_1 = len(link_call)
print(f"Number of links in link_call_all: {num_links_1}")

# Print the collected links from the first scraping
print("Links from the first scraping (link_call):")
print(link_call)

# Initialize a set to store the collected links
link_call_all = set()

for url in link_call:
    print(f"Processing scraped URL: {url}")

    # Navigate to the URL using Selenium
    driver.get(url)

    # Optionally, check if the URL itself contains '2024' (you can adjust this condition as needed)

    scraped_links = scrape_links(driver, url)

    # Add only the links that contain '2024'
    for link in scraped_links:
        if '2024' in link and 'call' in link.lower():  # Check both '2024' and case-insensitive 'call'
            link_call_all.add(link)

num_links_2 = len(link_call_all)
print(f"Number of links in link_call_all: {num_links_2}")

# Close the WebDriver after scraping is done
driver.quit()

print('Finish')
```

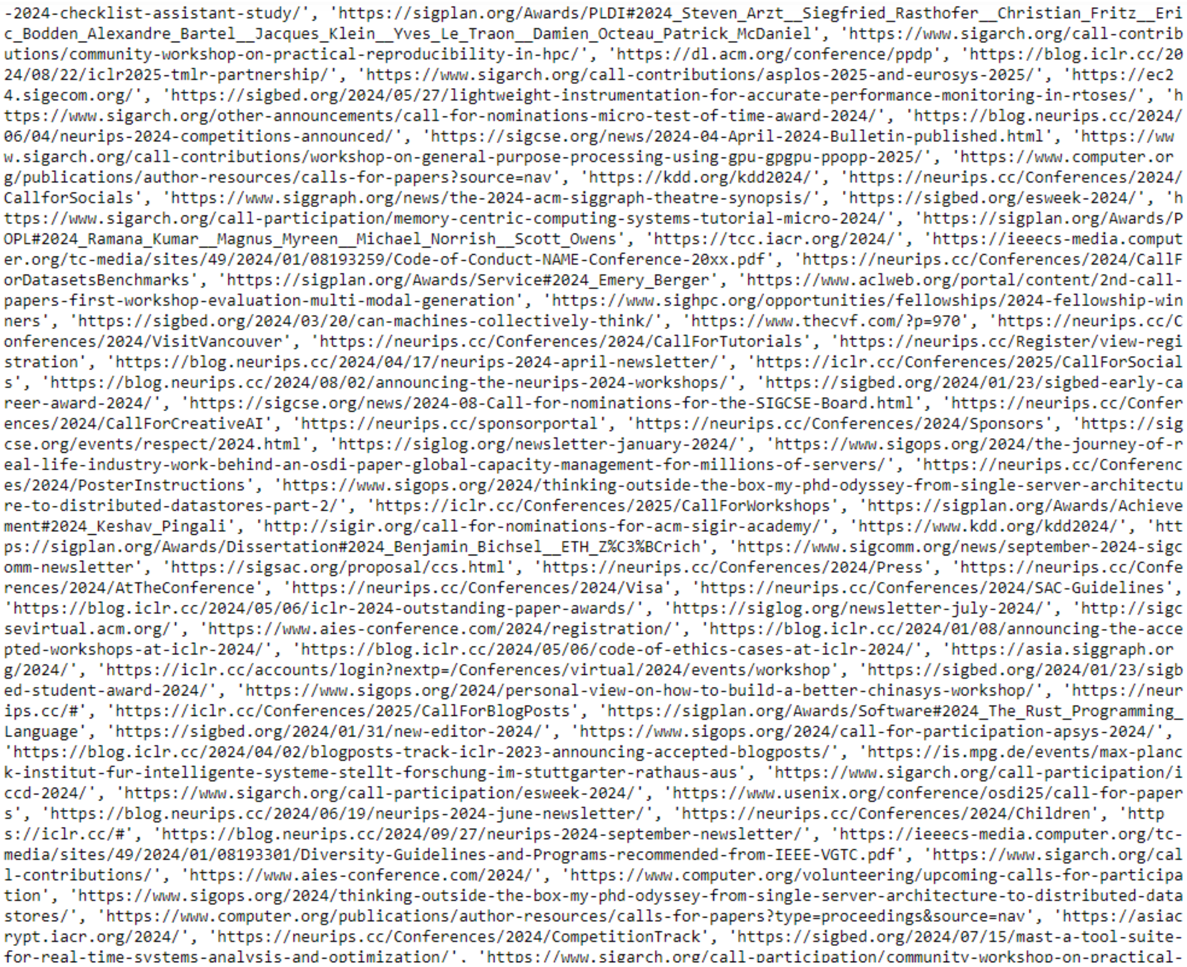
Code output (link_call) :



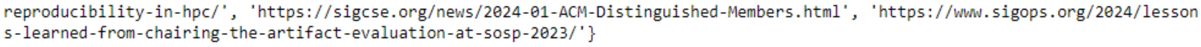
```
Processing URL: https://sigcse.org/
Processing URL: https://sigda.org/
Processing URL: https://sigchi.org/
Processing URL: https://tc.computer.org/vgtc/
Processing URL: https://www.ieee-security.org/
Processing URL: https://iclr.cc/
Processing URL: https://www.aclweb.org/portal/
Processing URL: http://www.machinelearning.org/
Processing URL: https://sigbed.org/
Processing URL: https://www.ieee-ras.org/
Processing URL: http://sigir.org/
Processing URL: https://www.sigarch.org/
Processing URL: https://sigops.org/
Processing URL: https://siglog.org/
Processing URL: https://siggraph.org/
Processing URL: http://sigsac.org/
Processing URL: https://www.computer.org/communities/technical-committees/tcmf
Processing URL: https://neurips.cc/
Processing URL: http://sigplan.org/
Processing URL: https://www.sigsoft.org/index.html
Processing URL: https://sigecom.org/
Processing URL: http://sigmetrics.org/
Processing URL: https://www.usenix.org/
Processing URL: https://www.kdd.org/
Processing URL: https://aaai.org/
Processing URL: https://sigact.org/
Processing URL: http://sigcomm.org/
Processing URL: https://www.cv-foundation.org/
Processing URL: https://www.iacr.org/
Processing URL: http://sigai.acm.org/index.html
Processing URL: http://www.sigbio.org/
Processing URL: https://sigmobile.org/
Processing URL: https://www.sighpc.org/
Processing URL: https://sigmod.org/
Number of links in link_call_all: 179
```



```
Links from the first scraping (link_call):
{'https://is.mpg.de/events/reconstruction-and-animation-of-realistic-head-avatars', 'https://neurips.cc/Conferences/2024/Pricin
g', 'https://openreview.net/group?id=NeurIPS.cc/2024/Conference', 'https://is.mpg.de/news/artificial-muscles-propel-a-robotic-l
eg-to-walk-and-jump', 'https://iclr.cc/Conferences/2025/CallForPapers', 'https://blog.iclr.cc/2023/11/02/tiny-papers-strike-bac
k-2023-reflections-and-2024-announcement/', 'https://sigbed.org/2024/08/29/automated-configuration-framework-for-tsn/', 'http
s://www.sighpc.org/opportunities/emerging-woman-leader-in-technical-computing-award/2024-ewltc-award-winner', 'https://www.sigg
raph.org/news/2024-acm-siggraph-executive-committee-results/', 'https://onlinelibrary.wiley.com/toc/23719621/2024/45/2', 'http
s://www.iacr.org/copyright.html', 'https://siglog.org/winner-of-the-2024-alonzo-church-award/', 'https://is.mpg.de/news/siyuan-
guo-wins-2024-mpi-is-outstanding-female-doctoral-student-prize', 'https://is.mpg.de/news/hexagonal-electrohydraulic-modules-sha
pe-shift-into-versatile-robots', 'https://www.thecvf.com/?p=911', 'https://sigchi.org/events/uist-2024/', 'https://popl24.sigpl
an.org/', 'https://blog.neurips.cc/2024/05/17/neurips-2024-may-newsletter/', 'https://neurips.cc/Conferences/2024/Hotels', 'htt
ps://iticse.acm.org/2024/', 'https://is.mpg.de/events/max-planck-lecture-2024', 'https://www.usenix.org/conferences/calls-for-p
apers', 'https://neurips.cc/Conferences/2024/CallForCompetitions', 'https://sigcse.org/news/2024-04-July-2024-Bulletin-publishe
d.html', 'https://sigbed.org/cps-iot2024/', 'https://www.ieee-security.org/CFP/Cipher-Call-for-Papers.html', 'https://neurips.c
c/Conferences/2024/CallForWorkshops', 'https://siglog.org/newsletter-april-2024/', 'https://aaai.getregistered.net/2024-fall-sy
mposium', 'https://neurips.cc/accounts/login?nextp=/virtual/2024/poster/93499', 'https://sigcse.org/news/2024-10-Oct-2024-Bulle
tin-published.html', 'https://www.aclweb.org/portal/content/announcement-2024-acl-test-time-paper-award-0', 'https://neurips.c
c/Conferences/2024/CallForPapers', 'https://sigmod.org/2024-sigmod-awards/', 'https://neurips.cc/Conferences/2024/Board', 'http
s://blog.iclr.cc/2024/05/01/hugging-face-demo-site-2/', 'https://sigcse.org/events/symposia/2024.html', 'https://kdd.org/Explor
ations/view/june-2024-volume-26-issue-1', 'https://www.sigops.org/2024/the-moral-implications-of-being-a-moderately-successful-
computer-scientist-and-a-woman/', 'https://blog.neurips.cc/2024/08/23/neurips-2024-august-newsletter/', 'https://www.sighpc.or
g/opportunities/dissertation-award/2024-award-winner', 'https://neurips.cc/Conferences/2024/Committees', 'https://www.thecvf.co
m/?m=202402', 'https://blog.iclr.cc/2024/05/07/iclr-2024-test-of-time-award/', 'https://www.sigops.org/2024/revisiting-distribu
ted-memory-in-the-cxl-era/', 'http://2024.sigmod.org/', 'https://www.usenix.org/conference/usenixsecurity25/call-for-papers',
'https://www.thecvf.com/?m=202407', 'https://crypto.iacr.org/2024', 'https://neurips.cc/Conferences/2024/ProgramCommittee', 'ht
tps://www.usenix.org/conference/srecon25americas/call-for-participation', 'https://www.sigda.org/news/call-for-sigda-newsletter
-editor-in-chief/', 'https://is.mpg.de/news/learning-from-the-laureates-in-lindau', 'https://iclr.cc/Conferences/2024', 'http
s://neurips.cc/Conferences/2024/AC-Guidelines', 'https://chi2024.acm.org/', 'https://neurips.cc/Conferences/2024/FinancialAssis
tance', 'https://www.sigarch.org/microarchitecture-and-hardware-security-research-at-usenix-security-symposium-2024/', 'http
s://neurips.cc/Conferences/2024/CallforHighSchoolProjects', 'https://sigplan.org/Awards/Milner#2024_Armando_Solar-Lezama', 'htt
ps://www.sigarch.org/other-announcements/call-for-nominations-sigmicro-distinguished-service-award-3/', 'https://www.usenix.or
g/conference/nsdi25/call-for-papers', 'https://www.sigops.org/2024/the-25th-chinasys-workshop/', 'https://neurips.cc/Conference
s/2024/Dates', 'https://sigchi.org/events/recsys-2024/', 'https://neurips.cc/Conferences/2024/CallForEthicsReviewers', 'http
s://sigchi.org/events/chi-play-2024/', 'https://www.aclweb.org/portal/content/naacl-2025-second-call-main-conference-papers',
'https://www.sigarch.org/call-participation/iiswc-2024/', 'https://www.sigarch.org/call-contributions/accelerated-machine-learn
ing-accml-hipeac-2025/', 'https://iclr.cc/Conferences/2025/CallForTinyPapers', 'https://www.siggraph.org/news/announcing-the-20
24-2025-acm-siggraph-officers/', 'https://sigbed.org/2024/08/30/ros2-and-real-time-performance-the-key-to-driving-embodied-inte
lligence-towards-commercialization/', 'https://neurips.cc/Conferences/2024/CallForExpo', 'https://neurips.cc/Conferences/2024/C
allForAffinityEvents', 'https://blog.neurips.cc/2024/03/03/neurips-2024-call-for-competitions/', 'https://www.sigarch.org/contr
ibute/submit-a-call-for-participation/', 'https://blog.iclr.cc/2024/04/22/hugging-face-demo-site/', 'https://www.usenix.org/con
ference/fast25/call-for-papers', 'https://www.sigarch.org/call-contributions/ieee-micro-special-issue-on-top-picks-from-the-202
4-computer-architecture-conferences/', 'https://blog.iclr.cc/2024/04/15/announcing-iclr-2024-invited-speakers/', 'https://blog.
neurips.cc/2024/04/15/announcing-the-neurips-2024-call-for-tutorials/', 'https://www.sigarch.org/contribute/submit-a-call-for-c
ontributions/', 'https://icer2024.acm.org/', 'https://neurips.cc/Conferences/2024/CodeOfConduct', 'https://neurips.cc/Conferenc
es/2024/EthicsReviewerGuidelines', 'https://www.sigarch.org/call-participation/', 'https://neurips.cc/Conferences/2024/Reviewer
Guidelines', 'https://neurips.cc/Conferences/2024', 'https://blog.neurips.cc/2024/05/07/soliciting-participants-for-the-neurips
```

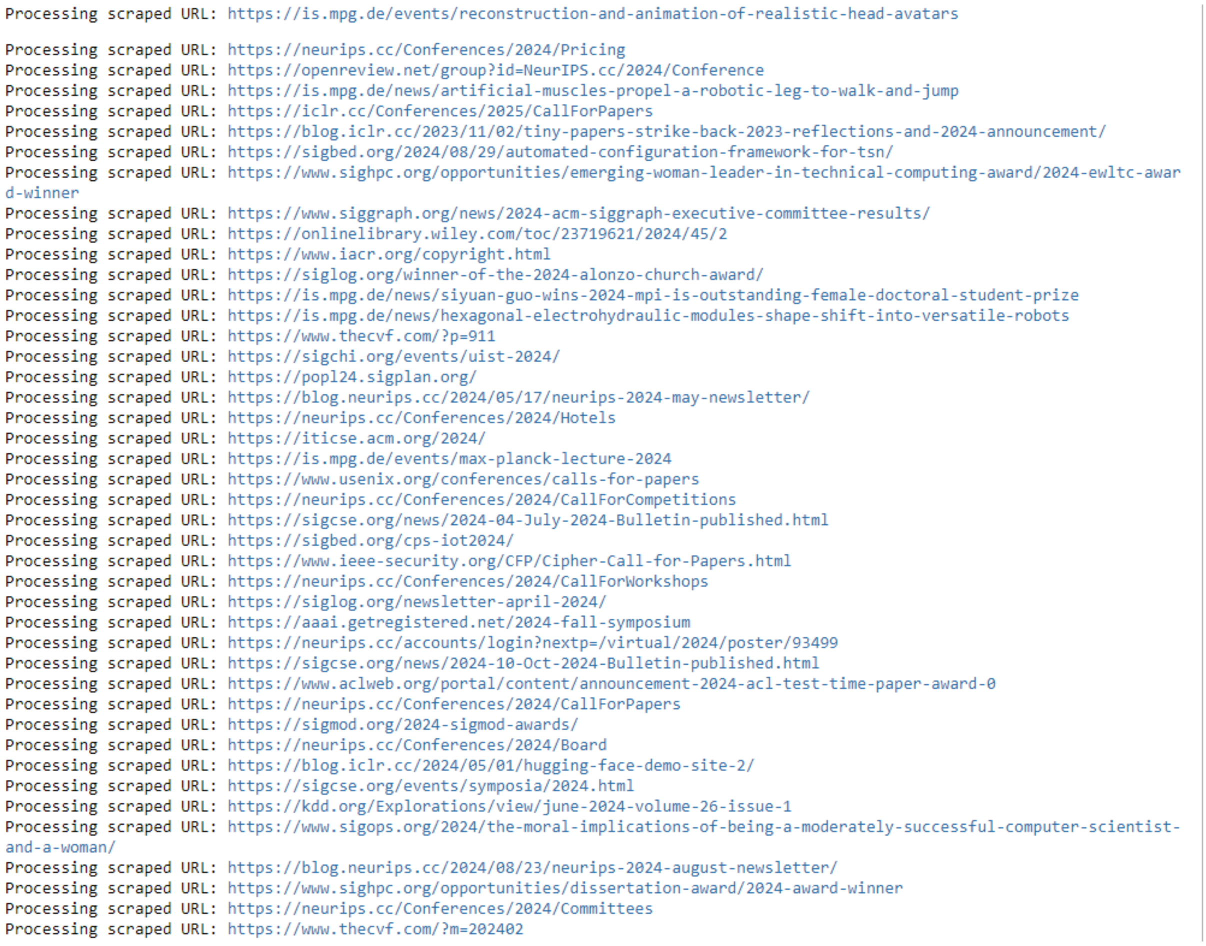


```
-2024-checklist-assistant-study/', 'https://sigplan.org/Awards/PLDI#2024_Steven_Arzt__Siegfried_Rasthofer__Christian_Fritz__Eri
c_Bodden_Alexandre_Bartel__Jacques_Klein__Yves_Le_Traon__Damien_Octeau_Patrick_McDaniel', 'https://www.sigarch.org/call-contrib
utions/community-workshop-on-practical-reproducibility-in-hpc/', 'https://dl.acm.org/conference/ppdp', 'https://blog.iclr.cc/20
24/08/22/iclr2025-tmlr-partnership/', 'https://www.sigarch.org/call-contributions/asplos-2025-and-eurosys-2025/', 'https://ec2
4.sigecom.org/', 'https://sigbed.org/2024/05/27/lightweight-instrumentation-for-accurate-performance-monitoring-in-rtoses/', 'h
ttps://www.sigarch.org/other-announcements/call-for-nominations-micro-test-of-time-award-2024/', 'https://blog.neurips.cc/2024/
06/04/neurips-2024-competitions-announced/', 'https://sigcse.org/news/2024-04-April-2024-Bulletin-published.html', 'https://ww
w.sigarch.org/call-contributions/workshop-on-general-purpose-processing-using-gpu-gpgpu-ppopp-2025/', 'https://www.computer.or
g/publications/author-resources/calls-for-papers?source=nav', 'https://kdd.org/kdd2024/', 'https://neurips.cc/Conferences/2024/
CallforSocials', 'https://www.siggraph.org/news/the-2024-acm-siggraph-theatre-synopsis/', 'https://sigbed.org/esweek-2024/', 'h
ttps://www.sigarch.org/call-participation/memory-centric-computing-systems-tutorial-micro-2024/', 'https://sigplan.org/Awards/P
OPL#2024_Ramana_Kumar__Magnus_Myreen__Michael_Norrish__Scott_Owens', 'https://tcc.iacr.org/2024/', 'https://ieeecs-media.comput
er.org/tc-media/sites/49/2024/01/08193259/Code-of-Conduct-NAME-Conference-20xx.pdf', 'https://neurips.cc/Conferences/2024/CallF
orDatasetsBenchmarks', 'https://sigplan.org/Awards/Service#2024_Emery_Berger', 'https://www.aclweb.org/portal/content/2nd-call-
papers-first-workshop-evaluation-multi-modal-generation', 'https://www.sighpc.org/opportunities/fellowships/2024-fellowship-win
ners', 'https://sigbed.org/2024/03/20/can-machines-collectively-think/', 'https://www.thecvf.com/?p=970', 'https://neurips.cc/C
onferences/2024/VisitVancouver', 'https://neurips.cc/Conferences/2024/CallForTutorials', 'https://neurips.cc/Register/view-regi
stration', 'https://blog.neurips.cc/2024/04/17/neurips-2024-april-newsletter/', 'https://iclr.cc/Conferences/2025/CallForSocial
s', 'https://blog.neurips.cc/2024/08/02/announcing-the-neurips-2024-workshops/', 'https://sigbed.org/2024/01/23/sigbed-early-ca
reer-award-2024/', 'https://sigcse.org/news/2024-08-Call-for-nominations-for-the-SIGCSE-Board.html', 'https://neurips.cc/Confer
ences/2024/CallForCreativeAI', 'https://neurips.cc/sponsorportal', 'https://neurips.cc/Conferences/2024/Sponsors', 'https://sig
cse.org/events/respect/2024.html', 'https://siglog.org/newsletter-january-2024/', 'https://www.sigops.org/2024/the-journey-of-r
eal-life-industry-work-behind-an-osdi-paper-global-capacity-management-for-millions-of-servers/', 'https://neurips.cc/Conferenc
es/2024/PosterInstructions', 'https://www.sigops.org/2024/thinking-outside-the-box-my-phd-odyssey-from-single-server-architectu
re-to-distributed-datastores-part-2/', 'https://iclr.cc/Conferences/2025/CallForWorkshops', 'https://sigplan.org/Awards/Achieve
ment#2024_Keshav_Pingali', 'http://sigir.org/call-for-nominations-for-acm-sigir-academy/', 'https://www.kdd.org/kdd2024/', 'htt
ps://sigplan.org/Awards/Dissertation#2024_Benjamin_Bichsel__ETH_Z%C3%BCrich', 'https://www.sigcomm.org/news/september-2024-sigc
omm-newsletter', 'https://sigsac.org/proposal/ccs.html', 'https://neurips.cc/Conferences/2024/Press', 'https://neurips.cc/Confe
rences/2024/AtTheConference', 'https://neurips.cc/Conferences/2024/Visa', 'https://neurips.cc/Conferences/2024/SAC-Guidelines',
'https://blog.iclr.cc/2024/05/06/iclr-2024-outstanding-paper-awards/', 'https://siglog.org/newsletter-july-2024/', 'http://sigc
sevirtual.acm.org/', 'https://www.aies-conference.com/2024/registration/', 'https://blog.iclr.cc/2024/01/08/announcing-the-acce
pted-workshops-at-iclr-2024/', 'https://blog.iclr.cc/2024/05/06/code-of-ethics-cases-at-iclr-2024/', 'https://asia.siggraph.or
g/2024/', 'https://iclr.cc/accounts/login?nextp=/Conferences/virtual/2024/events/workshop', 'https://sigbed.org/2024/01/23/sigb
ed-student-award-2024/', 'https://www.sigops.org/2024/personal-view-on-how-to-build-a-better-chinasys-workshop/', 'https://neur
ips.cc/#', 'https://iclr.cc/Conferences/2025/CallForBlogPosts', 'https://sigplan.org/Awards/Software#2024_The_Rust_Programming_
Language', 'https://sigbed.org/2024/01/31/new-editor-2024/', 'https://www.sigops.org/2024/call-for-participation-apsys-2024/',
'https://blog.iclr.cc/2024/04/02/blogposts-track-iclr-2023-announcing-accepted-blogposts/', 'https://is.mpg.de/events/max-planc
k-institut-fur-intelligente-systeme-stellt-forschung-im-stuttgarter-rathaus-aus', 'https://www.sigarch.org/call-participation/i
ccd-2024/', 'https://www.sigarch.org/call-participation/esweek-2024/', 'https://www.usenix.org/conference/osdi25/call-for-paper
s', 'https://blog.neurips.cc/2024/06/19/neurips-2024-june-newsletter/', 'https://neurips.cc/Conferences/2024/Children', 'http
s://iclr.cc/#', 'https://blog.neurips.cc/2024/09/27/neurips-2024-september-newsletter/', 'https://ieeecs-media.computer.org/tc-
media/sites/49/2024/01/08193301/Diversity-Guidelines-and-Programs-recommended-from-IEEE-VGTC.pdf', 'https://www.sigarch.org/cal
l-contributions/', 'https://www.aies-conference.com/2024/', 'https://www.computer.org/volunteering/upcoming-calls-for-participa
tion', 'https://www.sigops.org/2024/thinking-outside-the-box-my-phd-odyssey-from-single-server-architecture-to-distributed-data
stores/', 'https://www.computer.org/publications/author-resources/calls-for-papers?type=proceedings&source=nav', 'https://asiac
rypt.iacr.org/2024/', 'https://neurips.cc/Conferences/2024/CompetitionTrack', 'https://sigbed.org/2024/07/15/mast-a-tool-suite-
for-real-time-systems-analysis-and-optimization/', 'https://www.sigarch.org/call-participation/community-workshop-on-practical-
```

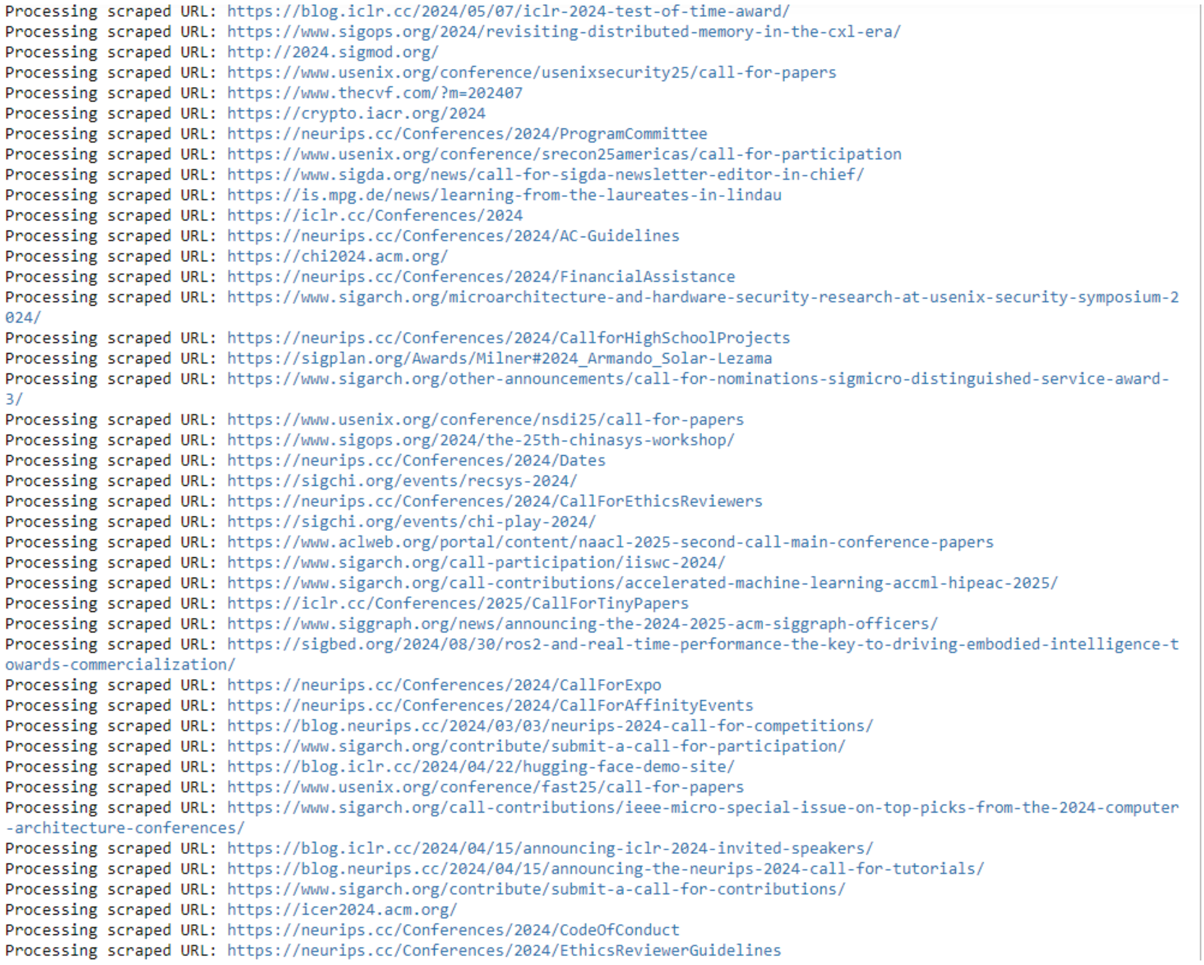


```
reproducibility-in-hpc/', 'https://sigcse.org/news/2024-01-ACM-Distinguished-Members.html', 'https://www.sigops.org/2024/lesson
s-learned-from-chairing-the-artifact-evaluation-at-sosp-2023/'}
```

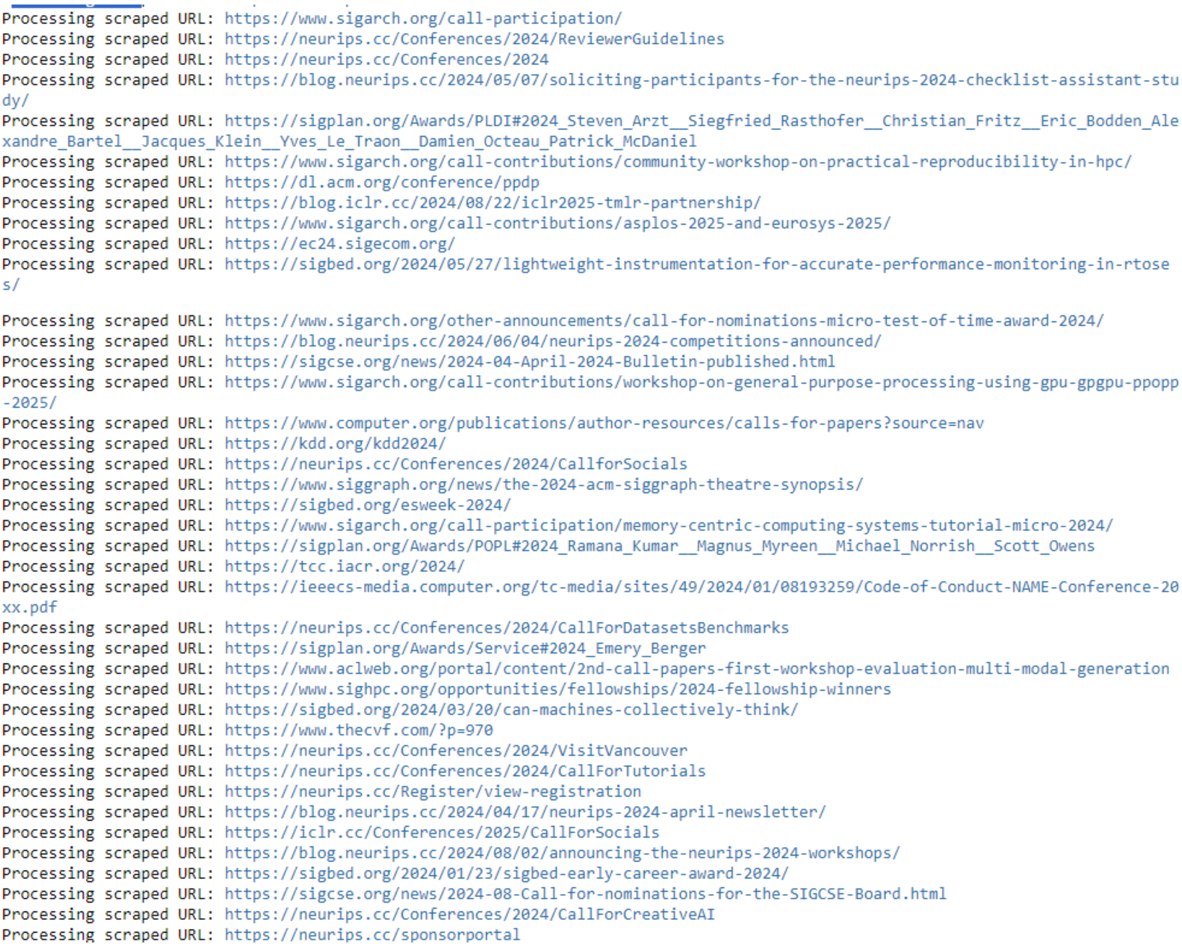
Code output (link_call_all):



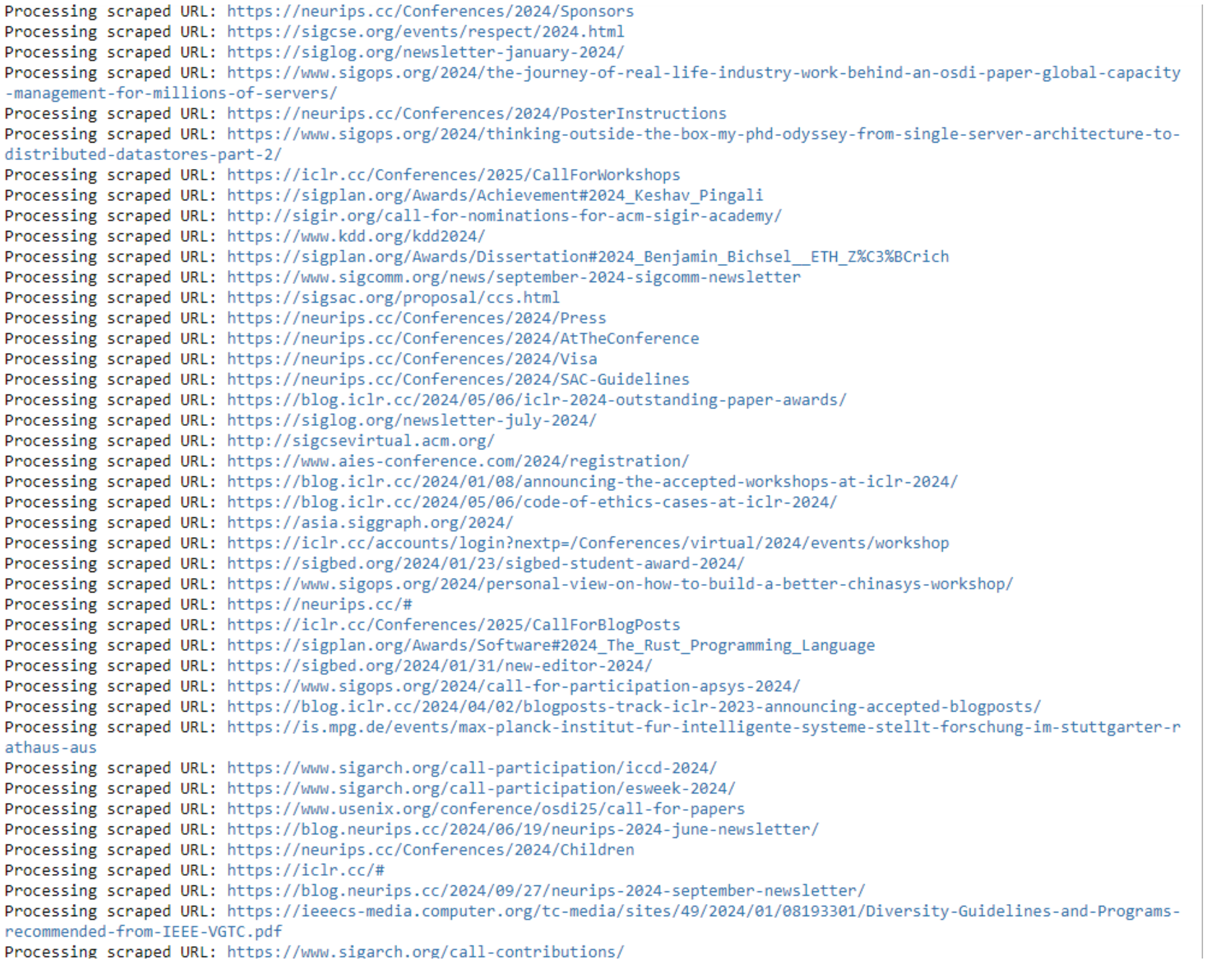
```
Processing scraped URL: https://is.mpg.de/events/reconstruction-and-animation-of-realistic-head-avatars

Processing scraped URL: https://neurips.cc/Conferences/2024/Pricing
Processing scraped URL: https://openreview.net/group?id=NeurIPS.cc/2024/Conference
Processing scraped URL: https://is.mpg.de/news/artificial-muscles-propel-a-robotic-leg-to-walk-and-jump
Processing scraped URL: https://iclr.cc/Conferences/2025/CallForPapers
Processing scraped URL: https://blog.iclr.cc/2023/11/02/tiny-papers-strike-back-2023-reflections-and-2024-announcement/
Processing scraped URL: https://sigbed.org/2024/08/29/automated-configuration-framework-for-tsn/
Processing scraped URL: https://www.sighpc.org/opportunities/emerging-woman-leader-in-technical-computing-award/2024-ewltc-awar
d-winner
Processing scraped URL: https://www.siggraph.org/news/2024-acm-siggraph-executive-committee-results/
Processing scraped URL: https://onlinelibrary.wiley.com/toc/23719621/2024/45/2
Processing scraped URL: https://www.iacr.org/copyright.html
Processing scraped URL: https://siglog.org/winner-of-the-2024-alonzo-church-award/
Processing scraped URL: https://is.mpg.de/news/siyuan-guo-wins-2024-mpi-is-outstanding-female-doctoral-student-prize
Processing scraped URL: https://is.mpg.de/news/hexagonal-electrohydraulic-modules-shape-shift-into-versatile-robots
Processing scraped URL: https://www.thecvf.com/?p=911
Processing scraped URL: https://sigchi.org/events/uist-2024/
Processing scraped URL: https://popl24.sigplan.org/
Processing scraped URL: https://blog.neurips.cc/2024/05/17/neurips-2024-may-newsletter/
Processing scraped URL: https://neurips.cc/Conferences/2024/Hotels
Processing scraped URL: https://iticse.acm.org/2024/
Processing scraped URL: https://is.mpg.de/events/max-planck-lecture-2024
Processing scraped URL: https://www.usenix.org/conferences/calls-for-papers
Processing scraped URL: https://neurips.cc/Conferences/2024/CallForCompetitions
Processing scraped URL: https://sigcse.org/news/2024-04-July-2024-Bulletin-published.html
Processing scraped URL: https://sigbed.org/cps-iot2024/
Processing scraped URL: https://www.ieee-security.org/CFP/Cipher-Call-for-Papers.html
Processing scraped URL: https://neurips.cc/Conferences/2024/CallForWorkshops
Processing scraped URL: https://siglog.org/newsletter-april-2024/
Processing scraped URL: https://aaai.getregistered.net/2024-fall-symposium
Processing scraped URL: https://neurips.cc/accounts/login?nextp=/virtual/2024/poster/93499
Processing scraped URL: https://sigcse.org/news/2024-10-Oct-2024-Bulletin-published.html
Processing scraped URL: https://www.aclweb.org/portal/content/announcement-2024-acl-test-time-paper-award-0
Processing scraped URL: https://neurips.cc/Conferences/2024/CallForPapers
Processing scraped URL: https://sigmod.org/2024-sigmod-awards/
Processing scraped URL: https://neurips.cc/Conferences/2024/Board
Processing scraped URL: https://blog.iclr.cc/2024/05/01/hugging-face-demo-site-2/
Processing scraped URL: https://sigcse.org/events/symposia/2024.html
Processing scraped URL: https://kdd.org/Explorations/view/june-2024-volume-26-issue-1
Processing scraped URL: https://www.sigops.org/2024/the-moral-implications-of-being-a-moderately-successful-computer-scientist-
and-a-woman/
Processing scraped URL: https://blog.neurips.cc/2024/08/23/neurips-2024-august-newsletter/
Processing scraped URL: https://www.sighpc.org/opportunities/dissertation-award/2024-award-winner
Processing scraped URL: https://neurips.cc/Conferences/2024/Committees
Processing scraped URL: https://www.thecvf.com/?m=202402
```



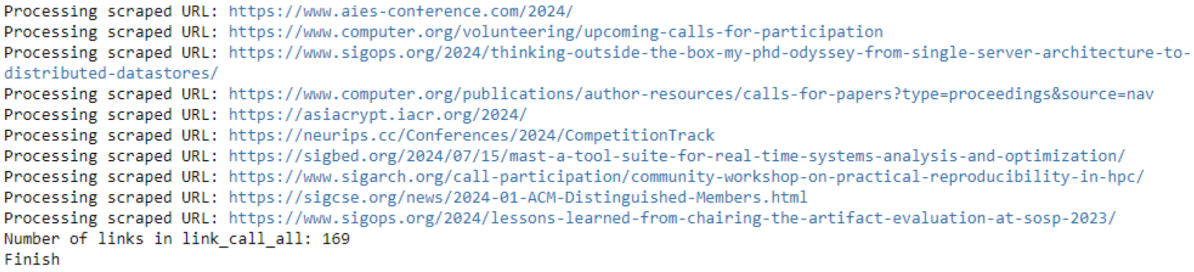
```
Processing scraped URL: https://blog.iclr.cc/2024/05/07/iclr-2024-test-of-time-award/
Processing scraped URL: https://www.sigops.org/2024/revisiting-distributed-memory-in-the-cxl-era/
Processing scraped URL: http://2024.sigmod.org/
Processing scraped URL: https://www.usenix.org/conference/usenixsecurity25/call-for-papers
Processing scraped URL: https://www.thecvf.com/?m=202407
Processing scraped URL: https://crypto.iacr.org/2024
Processing scraped URL: https://neurips.cc/Conferences/2024/ProgramCommittee
Processing scraped URL: https://www.usenix.org/conference/srecon25americas/call-for-participation
Processing scraped URL: https://www.sigda.org/news/call-for-sigda-newsletter-editor-in-chief/
Processing scraped URL: https://is.mpg.de/news/learning-from-the-laureates-in-lindau
Processing scraped URL: https://iclr.cc/Conferences/2024
Processing scraped URL: https://neurips.cc/Conferences/2024/AC-Guidelines
Processing scraped URL: https://chi2024.acm.org/
Processing scraped URL: https://neurips.cc/Conferences/2024/FinancialAssistance
Processing scraped URL: https://www.sigarch.org/microarchitecture-and-hardware-security-research-at-usenix-security-symposium-2
024/
Processing scraped URL: https://neurips.cc/Conferences/2024/CallforHighSchoolProjects
Processing scraped URL: https://sigplan.org/Awards/Milner#2024_Armando_Solar-Lezama
Processing scraped URL: https://www.sigarch.org/other-announcements/call-for-nominations-sigmicro-distinguished-service-award-
3/
Processing scraped URL: https://www.usenix.org/conference/nsdi25/call-for-papers
Processing scraped URL: https://www.sigops.org/2024/the-25th-chinasys-workshop/
Processing scraped URL: https://neurips.cc/Conferences/2024/Dates
Processing scraped URL: https://sigchi.org/events/recsys-2024/
Processing scraped URL: https://neurips.cc/Conferences/2024/CallForEthicsReviewers
Processing scraped URL: https://sigchi.org/events/chi-play-2024/
Processing scraped URL: https://www.aclweb.org/portal/content/naacl-2025-second-call-main-conference-papers
Processing scraped URL: https://www.sigarch.org/call-participation/iiswc-2024/
Processing scraped URL: https://www.sigarch.org/call-contributions/accelerated-machine-learning-accml-hipeac-2025/
Processing scraped URL: https://iclr.cc/Conferences/2025/CallForTinyPapers
Processing scraped URL: https://www.siggraph.org/news/announcing-the-2024-2025-acm-siggraph-officers/
Processing scraped URL: https://sigbed.org/2024/08/30/ros2-and-real-time-performance-the-key-to-driving-embodied-intelligence-t
owards-commercialization/
Processing scraped URL: https://neurips.cc/Conferences/2024/CallForExpo
Processing scraped URL: https://neurips.cc/Conferences/2024/CallForAffinityEvents
Processing scraped URL: https://blog.neurips.cc/2024/03/03/neurips-2024-call-for-competitions/
Processing scraped URL: https://www.sigarch.org/contribute/submit-a-call-for-participation/
Processing scraped URL: https://blog.iclr.cc/2024/04/22/hugging-face-demo-site/
Processing scraped URL: https://www.usenix.org/conference/fast25/call-for-papers
Processing scraped URL: https://www.sigarch.org/call-contributions/ieee-micro-special-issue-on-top-picks-from-the-2024-computer
-architecture-conferences/
Processing scraped URL: https://blog.iclr.cc/2024/04/15/announcing-iclr-2024-invited-speakers/
Processing scraped URL: https://blog.neurips.cc/2024/04/15/announcing-the-neurips-2024-call-for-tutorials/
Processing scraped URL: https://www.sigarch.org/contribute/submit-a-call-for-contributions/
Processing scraped URL: https://icer2024.acm.org/
Processing scraped URL: https://neurips.cc/Conferences/2024/CodeOfConduct
Processing scraped URL: https://neurips.cc/Conferences/2024/EthicsReviewerGuidelines
```



```
Processing scraped URL: https://www.sigarch.org/call-participation/
Processing scraped URL: https://neurips.cc/Conferences/2024/ReviewerGuidelines
Processing scraped URL: https://neurips.cc/Conferences/2024
Processing scraped URL: https://blog.neurips.cc/2024/05/07/soliciting-participants-for-the-neurips-2024-checklist-assistant-stu
dy/
Processing scraped URL: https://sigplan.org/Awards/PLDI#2024_Steven_Arzt__Siegfried_Rasthofer__Christian_Fritz__Eric_Bodden_Ale
xandre_Bartel__Jacques_Klein__Yves_Le_Traon__Damien_Octeau_Patrick_McDaniel
Processing scraped URL: https://www.sigarch.org/call-contributions/community-workshop-on-practical-reproducibility-in-hpc/
Processing scraped URL: https://dl.acm.org/conference/ppdp
Processing scraped URL: https://blog.iclr.cc/2024/08/22/iclr2025-tmlr-partnership/
Processing scraped URL: https://www.sigarch.org/call-contributions/asplos-2025-and-eurosys-2025/
Processing scraped URL: https://ec24.sigecom.org/
Processing scraped URL: https://sigbed.org/2024/05/27/lightweight-instrumentation-for-accurate-performance-monitoring-in-rtose
s/

Processing scraped URL: https://www.sigarch.org/other-announcements/call-for-nominations-micro-test-of-time-award-2024/
Processing scraped URL: https://blog.neurips.cc/2024/06/04/neurips-2024-competitions-announced/
Processing scraped URL: https://sigcse.org/news/2024-04-April-2024-Bulletin-published.html
Processing scraped URL: https://www.sigarch.org/call-contributions/workshop-on-general-purpose-processing-using-gpu-gpgpu-ppopp
-2025/
Processing scraped URL: https://www.computer.org/publications/author-resources/calls-for-papers?source=nav
Processing scraped URL: https://kdd.org/kdd2024/
Processing scraped URL: https://neurips.cc/Conferences/2024/CallforSocials
Processing scraped URL: https://www.siggraph.org/news/the-2024-acm-siggraph-theatre-synopsis/
Processing scraped URL: https://sigbed.org/esweek-2024/
Processing scraped URL: https://www.sigarch.org/call-participation/memory-centric-computing-systems-tutorial-micro-2024/
Processing scraped URL: https://sigplan.org/Awards/POPL#2024_Ramana_Kumar__Magnus_Myreen__Michael_Norrish__Scott_Owens
Processing scraped URL: https://tcc.iacr.org/2024/
Processing scraped URL: https://ieeecs-media.computer.org/tc-media/sites/49/2024/01/08193259/Code-of-Conduct-NAME-Conference-20
xx.pdf
Processing scraped URL: https://neurips.cc/Conferences/2024/CallForDatasetsBenchmarks
Processing scraped URL: https://sigplan.org/Awards/Service#2024_Emery_Berger
Processing scraped URL: https://www.aclweb.org/portal/content/2nd-call-papers-first-workshop-evaluation-multi-modal-generation
Processing scraped URL: https://www.sighpc.org/opportunities/fellowships/2024-fellowship-winners
Processing scraped URL: https://sigbed.org/2024/03/20/can-machines-collectively-think/
Processing scraped URL: https://www.thecvf.com/?p=970
Processing scraped URL: https://neurips.cc/Conferences/2024/VisitVancouver
Processing scraped URL: https://neurips.cc/Conferences/2024/CallForTutorials
Processing scraped URL: https://neurips.cc/Register/view-registration
Processing scraped URL: https://blog.neurips.cc/2024/04/17/neurips-2024-april-newsletter/
Processing scraped URL: https://iclr.cc/Conferences/2025/CallForSocials
Processing scraped URL: https://blog.neurips.cc/2024/08/02/announcing-the-neurips-2024-workshops/
Processing scraped URL: https://sigbed.org/2024/01/23/sigbed-early-career-award-2024/
Processing scraped URL: https://sigcse.org/news/2024-08-Call-for-nominations-for-the-SIGCSE-Board.html
Processing scraped URL: https://neurips.cc/Conferences/2024/CallForCreativeAI
Processing scraped URL: https://neurips.cc/sponsorportal
```

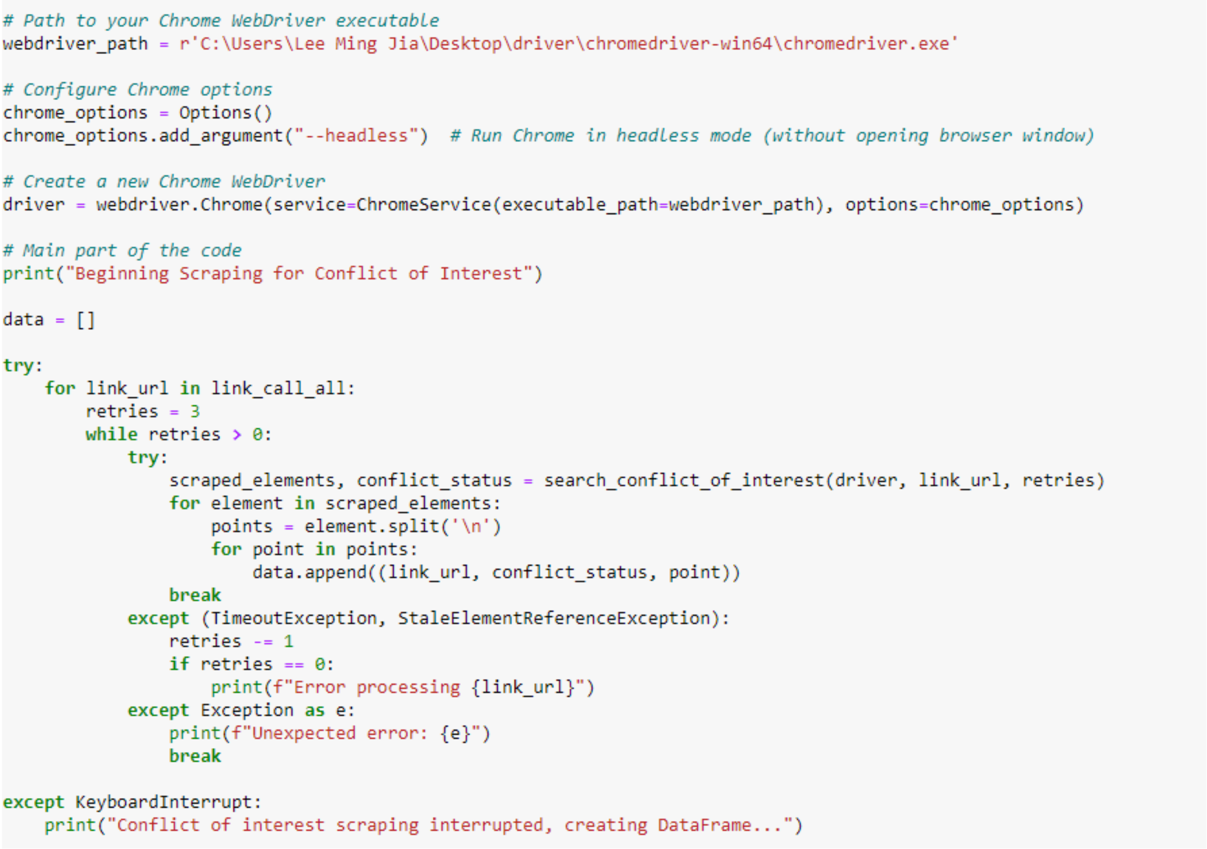


```
Processing scraped URL: https://neurips.cc/Conferences/2024/Sponsors
Processing scraped URL: https://sigcse.org/events/respect/2024.html
Processing scraped URL: https://siglog.org/newsletter-january-2024/
Processing scraped URL: https://www.sigops.org/2024/the-journey-of-real-life-industry-work-behind-an-osdi-paper-global-capacity
-management-for-millions-of-servers/
Processing scraped URL: https://neurips.cc/Conferences/2024/PosterInstructions
Processing scraped URL: https://www.sigops.org/2024/thinking-outside-the-box-my-phd-odyssey-from-single-server-architecture-to-
distributed-datastores-part-2/
Processing scraped URL: https://iclr.cc/Conferences/2025/CallForWorkshops
Processing scraped URL: https://sigplan.org/Awards/Achievement#2024_Keshav_Pingali
Processing scraped URL: http://sigir.org/call-for-nominations-for-acm-sigir-academy/
Processing scraped URL: https://www.kdd.org/kdd2024/
Processing scraped URL: https://sigplan.org/Awards/Dissertation#2024_Benjamin_Bichsel__ETH_Z%C3%BCrich
Processing scraped URL: https://www.sigcomm.org/news/september-2024-sigcomm-newsletter
Processing scraped URL: https://sigsac.org/proposal/ccs.html
Processing scraped URL: https://neurips.cc/Conferences/2024/Press
Processing scraped URL: https://neurips.cc/Conferences/2024/AtTheConference
Processing scraped URL: https://neurips.cc/Conferences/2024/Visa
Processing scraped URL: https://neurips.cc/Conferences/2024/SAC-Guidelines
Processing scraped URL: https://blog.iclr.cc/2024/05/06/iclr-2024-outstanding-paper-awards/
Processing scraped URL: https://siglog.org/newsletter-july-2024/
Processing scraped URL: http://sigcsevirtual.acm.org/
Processing scraped URL: https://www.aies-conference.com/2024/registration/
Processing scraped URL: https://blog.iclr.cc/2024/01/08/announcing-the-accepted-workshops-at-iclr-2024/
Processing scraped URL: https://blog.iclr.cc/2024/05/06/code-of-ethics-cases-at-iclr-2024/
Processing scraped URL: https://asia.siggraph.org/2024/
Processing scraped URL: https://iclr.cc/accounts/login?nextp=/Conferences/virtual/2024/events/workshop
Processing scraped URL: https://sigbed.org/2024/01/23/sigbed-student-award-2024/
Processing scraped URL: https://www.sigops.org/2024/personal-view-on-how-to-build-a-better-chinasys-workshop/
Processing scraped URL: https://neurips.cc/#
Processing scraped URL: https://iclr.cc/Conferences/2025/CallForBlogPosts
Processing scraped URL: https://sigplan.org/Awards/Software#2024_The_Rust_Programming_Language
Processing scraped URL: https://sigbed.org/2024/01/31/new-editor-2024/
Processing scraped URL: https://www.sigops.org/2024/call-for-participation-apsys-2024/
Processing scraped URL: https://blog.iclr.cc/2024/04/02/blogposts-track-iclr-2023-announcing-accepted-blogposts/
Processing scraped URL: https://is.mpg.de/events/max-planck-institut-fur-intelligente-systeme-stellt-forschung-im-stuttgarter-r
athaus-aus
Processing scraped URL: https://www.sigarch.org/call-participation/iccd-2024/
Processing scraped URL: https://www.sigarch.org/call-participation/esweek-2024/
Processing scraped URL: https://www.usenix.org/conference/osdi25/call-for-papers
Processing scraped URL: https://blog.neurips.cc/2024/06/19/neurips-2024-june-newsletter/
Processing scraped URL: https://neurips.cc/Conferences/2024/Children
Processing scraped URL: https://iclr.cc/#
Processing scraped URL: https://blog.neurips.cc/2024/09/27/neurips-2024-september-newsletter/
Processing scraped URL: https://ieeecs-media.computer.org/tc-media/sites/49/2024/01/08193301/Diversity-Guidelines-and-Programs-
recommended-from-IEEE-VGTC.pdf
Processing scraped URL: https://www.sigarch.org/call-contributions/
```



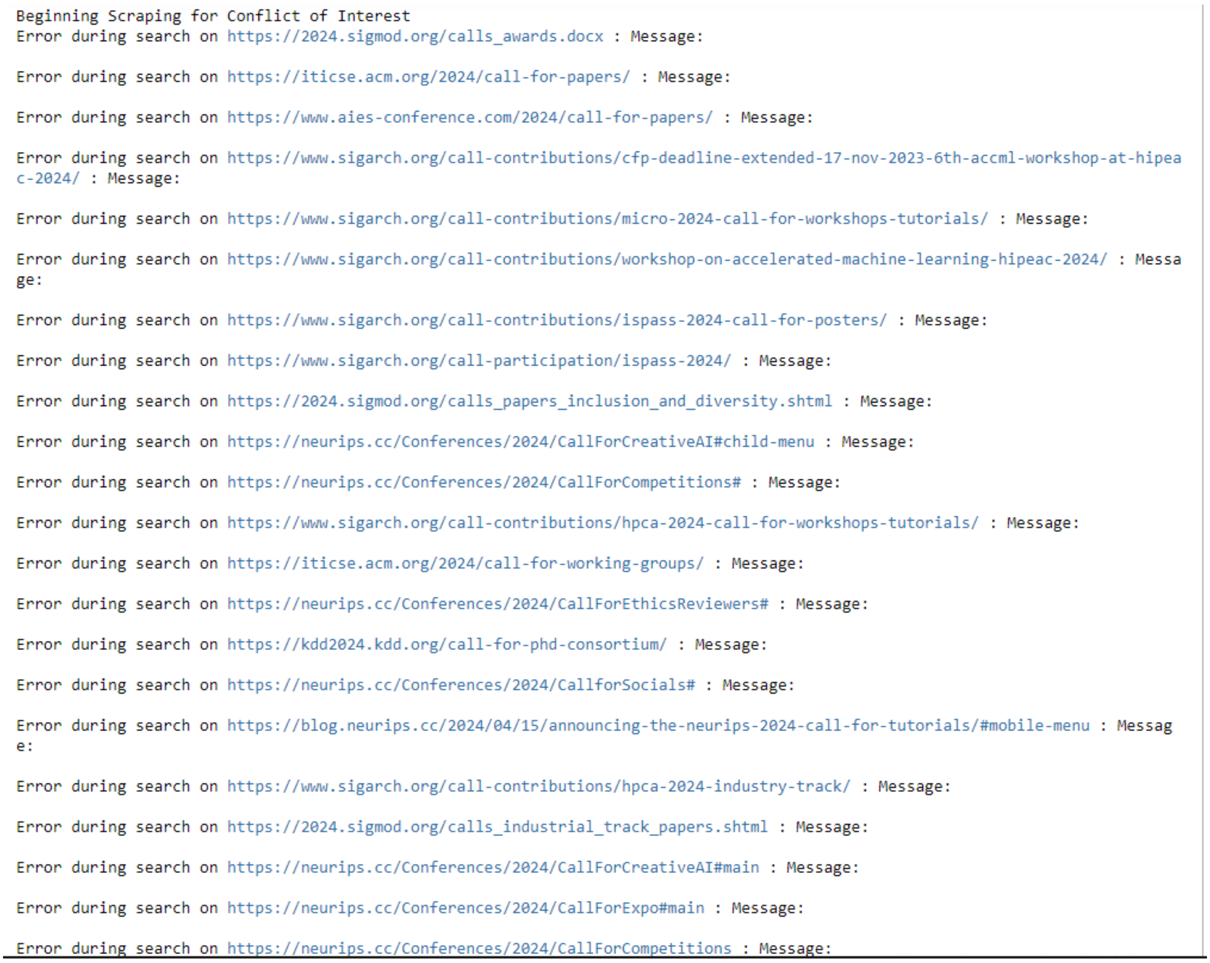
```
Processing scraped URL: https://www.aies-conference.com/2024/
Processing scraped URL: https://www.computer.org/volunteering/upcoming-calls-for-participation
Processing scraped URL: https://www.sigops.org/2024/thinking-outside-the-box-my-phd-odyssey-from-single-server-architecture-to-
distributed-datastores/
Processing scraped URL: https://www.computer.org/publications/author-resources/calls-for-papers?type=proceedings&source=nav
Processing scraped URL: https://asiacrypt.iacr.org/2024/
Processing scraped URL: https://neurips.cc/Conferences/2024/CompetitionTrack
Processing scraped URL: https://sigbed.org/2024/07/15/mast-a-tool-suite-for-real-time-systems-analysis-and-optimization/
Processing scraped URL: https://www.sigarch.org/call-participation/community-workshop-on-practical-reproducibility-in-hpc/
Processing scraped URL: https://sigcse.org/news/2024-01-ACM-Distinguished-Members.html
Processing scraped URL: https://www.sigops.org/2024/lessons-learned-from-chairing-the-artifact-evaluation-at-sosp-2023/
Number of links in link_call_all: 169
Finish
```

Code input (search_conflict_of_interest):

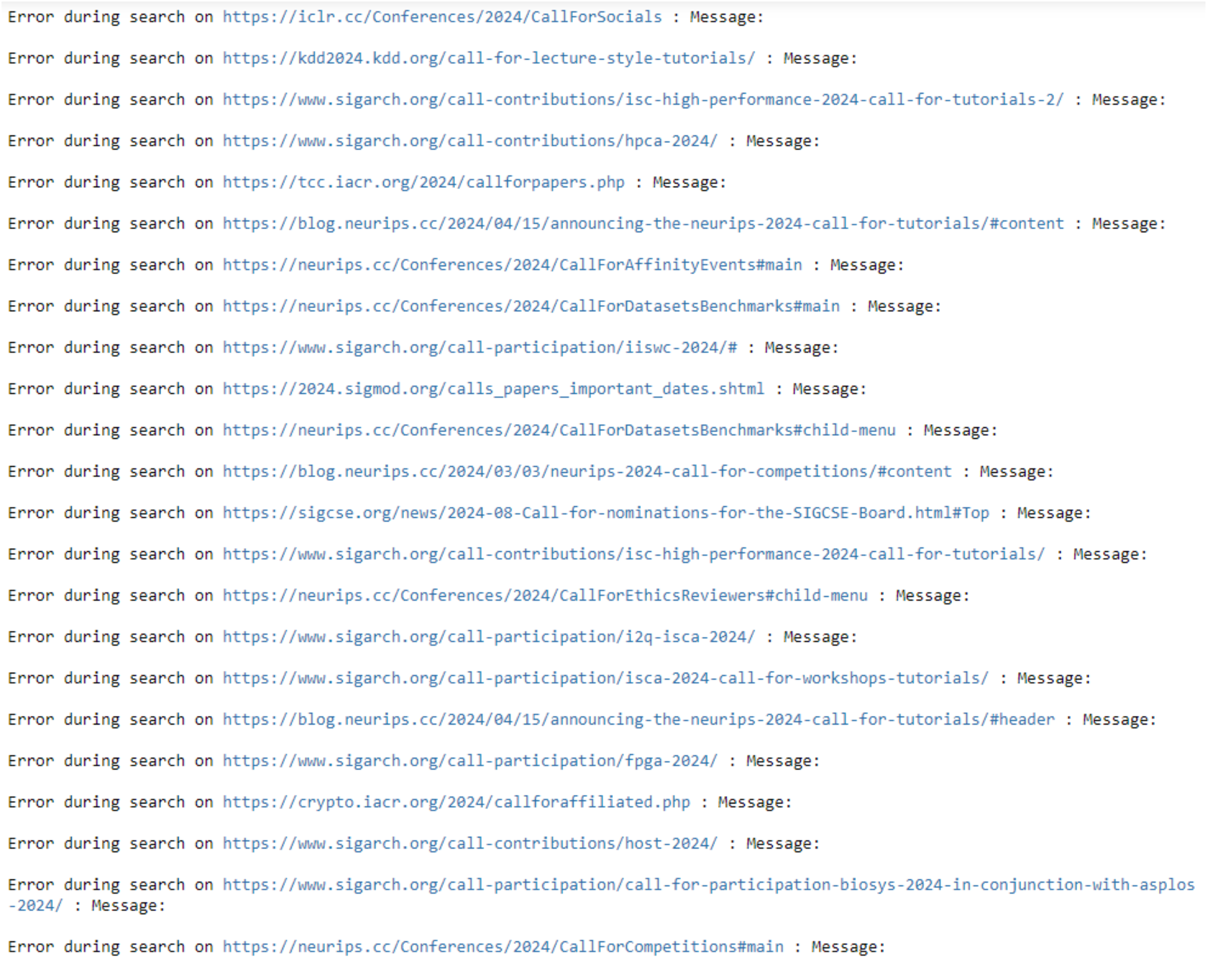


```python
# Path to your Chrome WebDriver executable
webdriver_path = r'C:\Users\Lee Ming Jia\Desktop\driver\chromedriver-win64\chromedriver.exe'

# Configure Chrome options
chrome_options = Options()
chrome_options.add_argument("--headless")  # Run Chrome in headless mode (without opening browser window)

# Create a new Chrome WebDriver
driver = webdriver.Chrome(service=ChromeService(executable_path=webdriver_path), options=chrome_options)

# Main part of the code
print("Beginning Scraping for Conflict of Interest")

data = []

try:
    for link_url in link_call_all:
        retries = 3
        while retries > 0:
            try:
                scraped_elements, conflict_status = search_conflict_of_interest(driver, link_url, retries)
                for element in scraped_elements:
                    points = element.split('\n')
                    for point in points:
                        data.append((link_url, conflict_status, point))
                break
            except (TimeoutException, StaleElementReferenceException):
                retries -= 1
                if retries == 0:
                    print(f"Error processing {link_url}")
            except Exception as e:
                print(f"Unexpected error: {e}")
                break

except KeyboardInterrupt:
    print("Conflict of interest scraping interrupted, creating DataFrame...")
```

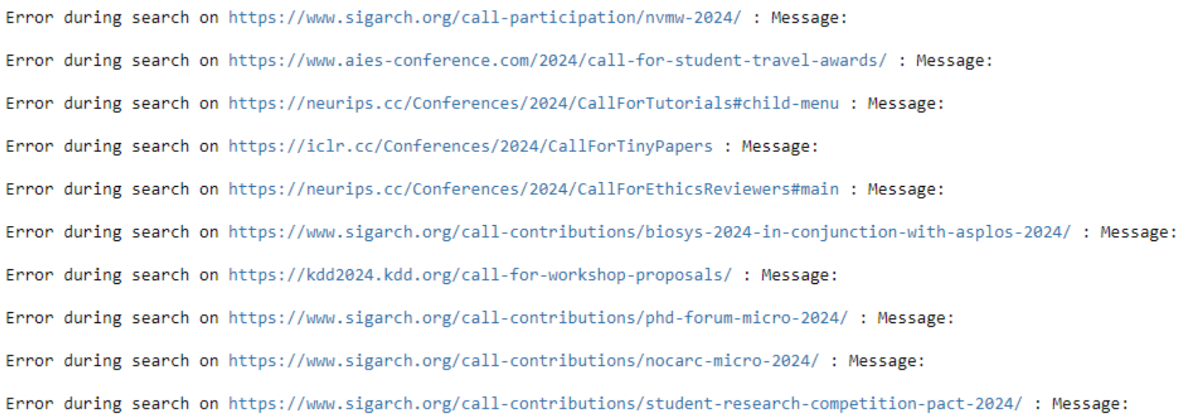
Code output (search_conflict_of_interest):



```
Beginning Scraping for Conflict of Interest
Error during search on https://2024.sigmod.org/calls_awards.docx : Message:

Error during search on https://iticse.acm.org/2024/call-for-papers/ : Message:

Error during search on https://www.aies-conference.com/2024/call-for-papers/ : Message:

Error during search on https://www.sigarch.org/call-contributions/cfp-deadline-extended-17-nov-2023-6th-accml-workshop-at-hipea
c-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/micro-2024-call-for-workshops-tutorials/ : Message:

Error during search on https://www.sigarch.org/call-contributions/workshop-on-accelerated-machine-learning-hipeac-2024/ : Messa
ge:

Error during search on https://www.sigarch.org/call-contributions/ispass-2024-call-for-posters/ : Message:

Error during search on https://www.sigarch.org/call-participation/ispass-2024/ : Message:

Error during search on https://2024.sigmod.org/calls_papers_inclusion_and_diversity.shtml : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForCreativeAI#child-menu : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForCompetitions# : Message:

Error during search on https://www.sigarch.org/call-contributions/hpca-2024-call-for-workshops-tutorials/ : Message:

Error during search on https://iticse.acm.org/2024/call-for-working-groups/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForEthicsReviewers# : Message:

Error during search on https://kdd2024.kdd.org/call-for-phd-consortium/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallforSocials# : Message:

Error during search on https://blog.neurips.cc/2024/04/15/announcing-the-neurips-2024-call-for-tutorials/#mobile-menu : Messag
e:

Error during search on https://www.sigarch.org/call-contributions/hpca-2024-industry-track/ : Message:

Error during search on https://2024.sigmod.org/calls_industrial_track_papers.shtml : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForCreativeAI#main : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForExpo#main : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForCompetitions : Message:
```

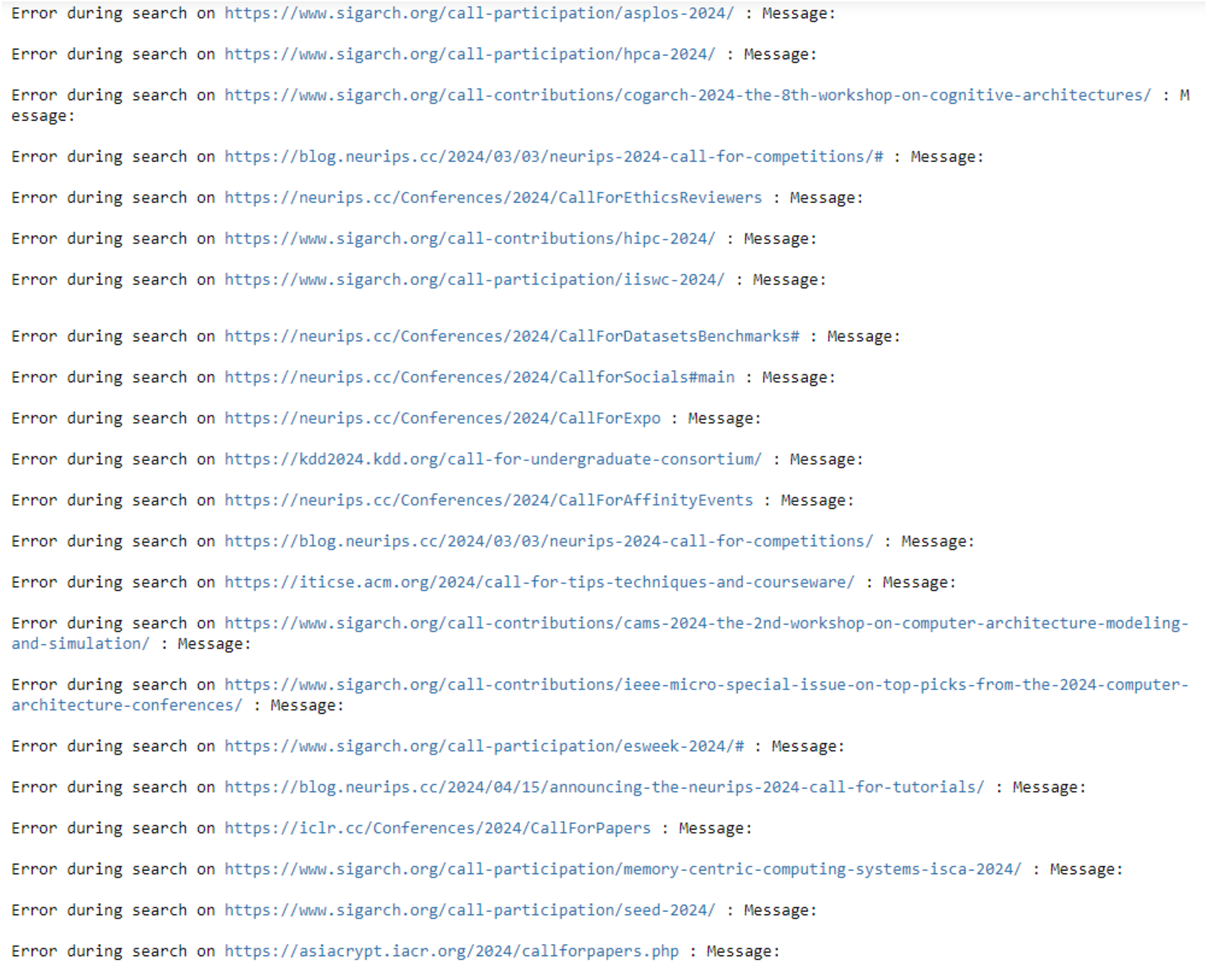


```
Error during search on https://iclr.cc/Conferences/2024/CallForSocials : Message:

Error during search on https://kdd2024.kdd.org/call-for-lecture-style-tutorials/ : Message:

Error during search on https://www.sigarch.org/call-contributions/isc-high-performance-2024-call-for-tutorials-2/ : Message:

Error during search on https://www.sigarch.org/call-contributions/hpca-2024/ : Message:

Error during search on https://tcc.iacr.org/2024/callforpapers.php : Message:

Error during search on https://blog.neurips.cc/2024/04/15/announcing-the-neurips-2024-call-for-tutorials/#content : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForAffinityEvents#main : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForDatasetsBenchmarks#main : Message:

Error during search on https://www.sigarch.org/call-participation/iiswc-2024/# : Message:

Error during search on https://2024.sigmod.org/calls_papers_important_dates.shtml : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForDatasetsBenchmarks#child-menu : Message:

Error during search on https://blog.neurips.cc/2024/03/03/neurips-2024-call-for-competitions/#content : Message:

Error during search on https://sigcse.org/news/2024-08-Call-for-nominations-for-the-SIGCSE-Board.html#Top : Message:

Error during search on https://www.sigarch.org/call-contributions/isc-high-performance-2024-call-for-tutorials/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForEthicsReviewers#child-menu : Message:

Error during search on https://www.sigarch.org/call-participation/i2q-isca-2024/ : Message:

Error during search on https://www.sigarch.org/call-participation/isca-2024-call-for-workshops-tutorials/ : Message:

Error during search on https://blog.neurips.cc/2024/04/15/announcing-the-neurips-2024-call-for-tutorials/#header : Message:

Error during search on https://www.sigarch.org/call-participation/fpga-2024/ : Message:

Error during search on https://crypto.iacr.org/2024/callforaffiliated.php : Message:

Error during search on https://www.sigarch.org/call-contributions/host-2024/ : Message:

Error during search on https://www.sigarch.org/call-participation/call-for-participation-biosys-2024-in-conjunction-with-asplos
-2024/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForCompetitions#main : Message:
```

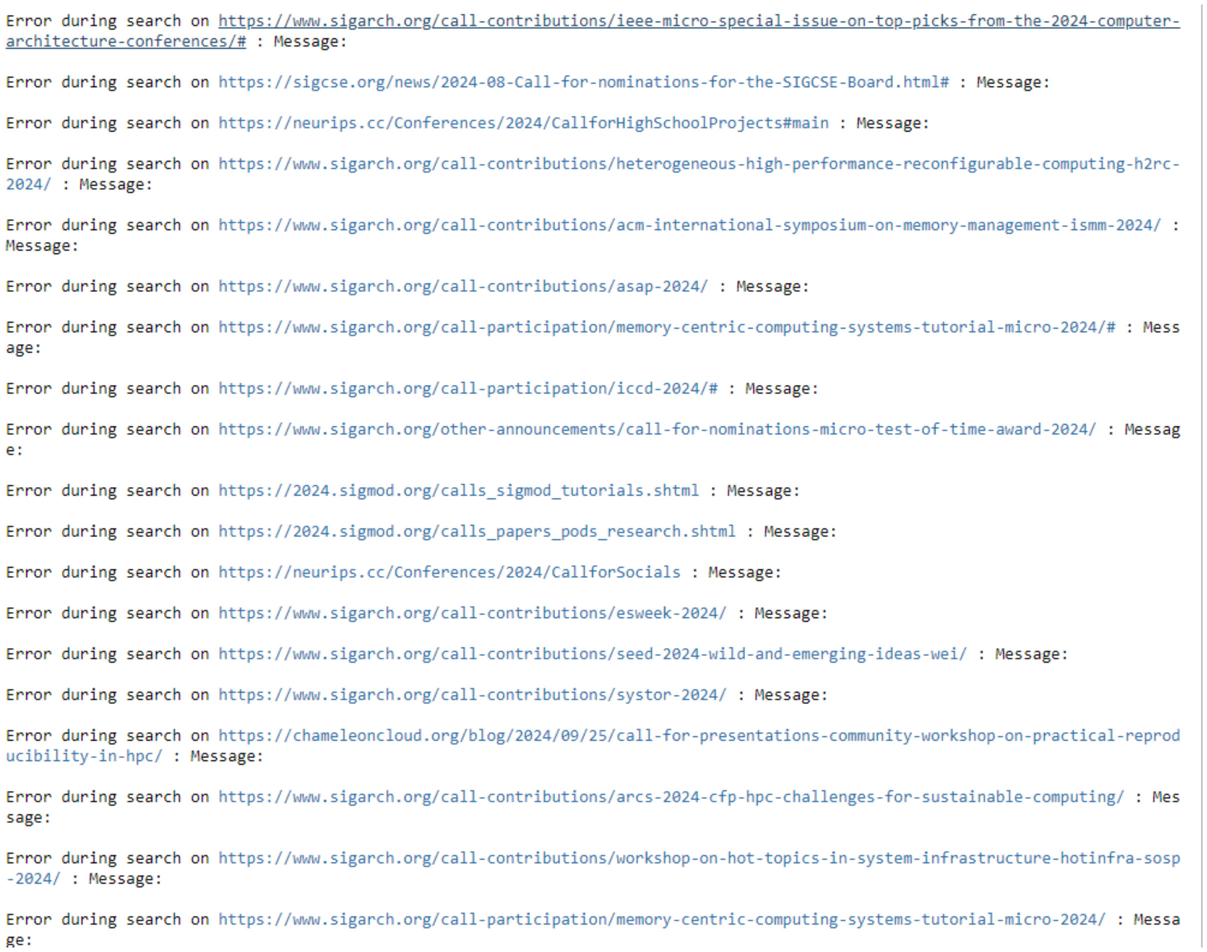


```
Error during search on https://www.sigarch.org/call-participation/nvmw-2024/ : Message:

Error during search on https://www.aies-conference.com/2024/call-for-student-travel-awards/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForTutorials#child-menu : Message:

Error during search on https://iclr.cc/Conferences/2024/CallForTinyPapers : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForEthicsReviewers#main : Message:

Error during search on https://www.sigarch.org/call-contributions/biosys-2024-in-conjunction-with-asplos-2024/ : Message:

Error during search on https://kdd2024.kdd.org/call-for-workshop-proposals/ : Message:

Error during search on https://www.sigarch.org/call-contributions/phd-forum-micro-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/nocarc-micro-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/student-research-competition-pact-2024/ : Message:
```

```
Error during search on https://neurips.cc/Conferences/2024/CallforHighSchoolProjects: Message: unknown error: net::ERR_INTERNET
_DISCONNECTED
  (Session info: chrome=129.0.6668.90)
Stacktrace:
        GetHandleVerifier [0x00007FF7EB06FDA5+29557]
        (No symbol) [0x00007FF7EAFE2240]
        (No symbol) [0x00007FF7EAE9B6EA]
        (No symbol) [0x00007FF7EAE93746]
        (No symbol) [0x00007FF7EAE84439]
        (No symbol) [0x00007FF7EAE861E2]
        (No symbol) [0x00007FF7EAE846FF]
        (No symbol) [0x00007FF7EAE83FAD]
        (No symbol) [0x00007FF7EAE83ECA]
        (No symbol) [0x00007FF7EAE81D52]
        (No symbol) [0x00007FF7EAE823CC]
        (No symbol) [0x00007FF7EAE9E821]
        (No symbol) [0x00007FF7EAF396CE]
        (No symbol) [0x00007FF7EAF174FA]
        (No symbol) [0x00007FF7EAF388A3]
        (No symbol) [0x00007FF7EAF172A3]
        (No symbol) [0x00007FF7EAEE12DF]
        (No symbol) [0x00007FF7EAEE2451]
        GetHandleVerifier [0x00007FF7EB39DCBD+3363469]
        GetHandleVerifier [0x00007FF7EB3E9B47+3674391]
        GetHandleVerifier [0x00007FF7EB3DEAEB+3629243]
        GetHandleVerifier [0x00007FF7EB12FC66+815670]
        (No symbol) [0x00007FF7EAFED6EF]
        (No symbol) [0x00007FF7EAFE92B4]
        (No symbol) [0x00007FF7EAFE9450]
        (No symbol) [0x00007FF7EAFD81FF]
        BaseThreadInitThunk [0x00007FFF32E67374+20]
        RtlUserThreadStart [0x00007FFF3327CC91+33]

Error during search on https://www.sigarch.org/call-contributions/ics-2024/ : Message: 

Error during search on https://neurips.cc/Conferences/2024/CallForCompetitions#child-menu : Message: 

Error during search on https://kdd2024.kdd.org/call-for-kdd-cup-proposals/ : Message: 

Error during search on https://www.sigarch.org/call-participation/gem5-bootcamp-2024/ : Message: 

Error during search on https://www.sigarch.org/call-contributions/hardware-and-architectural-support-for-security-and-privacy-h
asp-2024/ : Message: 

Error during search on https://www.sigarch.org/call-contributions/pact-2024/ : Message: 
```

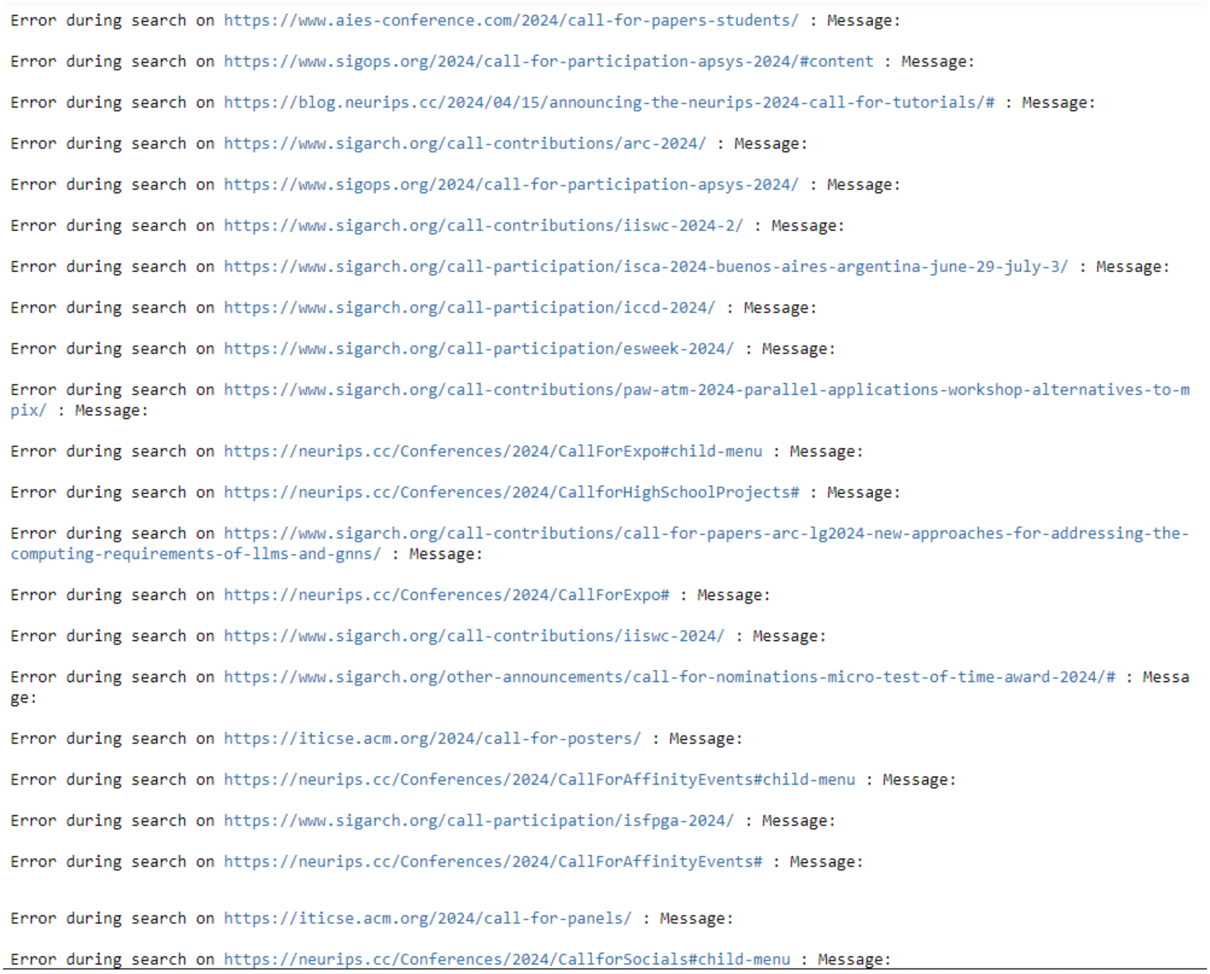


```
Error during search on https://www.sigarch.org/call-participation/asplos-2024/ : Message:

Error during search on https://www.sigarch.org/call-participation/hpca-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/cogarch-2024-the-8th-workshop-on-cognitive-architectures/ : M
essage:

Error during search on https://blog.neurips.cc/2024/03/03/neurips-2024-call-for-competitions/# : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForEthicsReviewers : Message:

Error during search on https://www.sigarch.org/call-contributions/hipc-2024/ : Message:

Error during search on https://www.sigarch.org/call-participation/iiswc-2024/ : Message:


Error during search on https://neurips.cc/Conferences/2024/CallForDatasetsBenchmarks# : Message:

Error during search on https://neurips.cc/Conferences/2024/CallforSocials#main : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForExpo : Message:

Error during search on https://kdd2024.kdd.org/call-for-undergraduate-consortium/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForAffinityEvents : Message:

Error during search on https://blog.neurips.cc/2024/03/03/neurips-2024-call-for-competitions/ : Message:

Error during search on https://iticse.acm.org/2024/call-for-tips-techniques-and-courseware/ : Message:

Error during search on https://www.sigarch.org/call-contributions/cams-2024-the-2nd-workshop-on-computer-architecture-modeling-
and-simulation/ : Message:

Error during search on https://www.sigarch.org/call-contributions/ieee-micro-special-issue-on-top-picks-from-the-2024-computer-
architecture-conferences/ : Message:

Error during search on https://www.sigarch.org/call-participation/esweek-2024/# : Message:

Error during search on https://blog.neurips.cc/2024/04/15/announcing-the-neurips-2024-call-for-tutorials/ : Message:

Error during search on https://iclr.cc/Conferences/2024/CallForPapers : Message:

Error during search on https://www.sigarch.org/call-participation/memory-centric-computing-systems-isca-2024/ : Message:

Error during search on https://www.sigarch.org/call-participation/seed-2024/ : Message:

Error during search on https://asiacrypt.iacr.org/2024/callforpapers.php : Message:
```



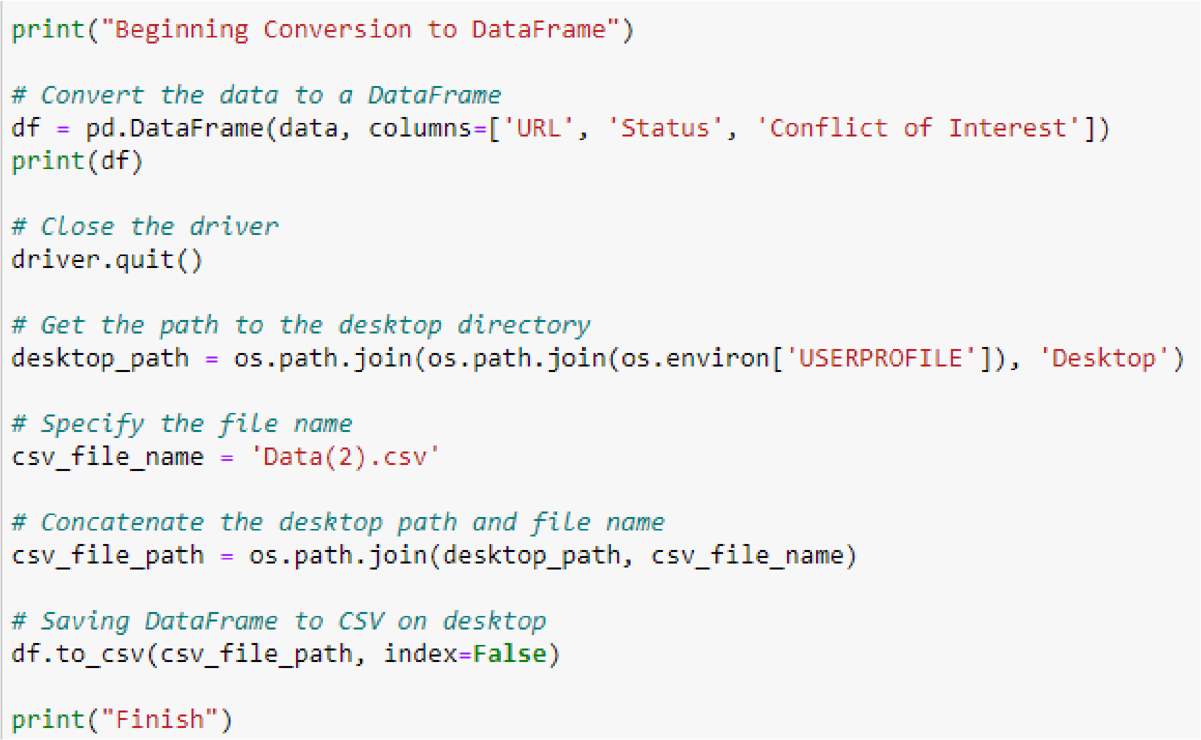
```
Error during search on https://www.sigarch.org/call-contributions/ieee-micro-special-issue-on-top-picks-from-the-2024-computer-
architecture-conferences/# : Message:

Error during search on https://sigcse.org/news/2024-08-Call-for-nominations-for-the-SIGCSE-Board.html# : Message:

Error during search on https://neurips.cc/Conferences/2024/CallforHighSchoolProjects#main : Message:

Error during search on https://www.sigarch.org/call-contributions/heterogeneous-high-performance-reconfigurable-computing-h2rc-
2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/acm-international-symposium-on-memory-management-ismm-2024/ :
Message:

Error during search on https://www.sigarch.org/call-contributions/asap-2024/ : Message:

Error during search on https://www.sigarch.org/call-participation/memory-centric-computing-systems-tutorial-micro-2024/# : Mess
age:

Error during search on https://www.sigarch.org/call-participation/iccd-2024/# : Message:

Error during search on https://www.sigarch.org/other-announcements/call-for-nominations-micro-test-of-time-award-2024/ : Messag
e:

Error during search on https://2024.sigmod.org/calls_sigmod_tutorials.shtml : Message:

Error during search on https://2024.sigmod.org/calls_papers_pods_research.shtml : Message:

Error during search on https://neurips.cc/Conferences/2024/CallforSocials : Message:

Error during search on https://www.sigarch.org/call-contributions/esweek-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/seed-2024-wild-and-emerging-ideas-wei/ : Message:

Error during search on https://www.sigarch.org/call-contributions/systor-2024/ : Message:

Error during search on https://chameleoncloud.org/blog/2024/09/25/call-for-presentations-community-workshop-on-practical-reprod
ucibility-in-hpc/ : Message:

Error during search on https://www.sigarch.org/call-contributions/arcs-2024-cfp-hpc-challenges-for-sustainable-computing/ : Mes
sage:

Error during search on https://www.sigarch.org/call-contributions/workshop-on-hot-topics-in-system-infrastructure-hotinfra-sosp
-2024/ : Message:

Error during search on https://www.sigarch.org/call-participation/memory-centric-computing-systems-tutorial-micro-2024/ : Messa
ge:
```

```
Error during search on https://neurips.cc/Conferences/2024/CallForDatasetsBenchmarks : Message:

Error during search on https://www.sigarch.org/call-contributions/nsf-workshop-on-quantum-operating-systems-and-real-time-control-sosp-2024-and-micro-2024/ : Message:

Error during search on https://blog.neurips.cc/2024/03/03/neurips-2024-call-for-competitions/#header : Message:

Error during search on https://www.sigarch.org/call-participation/arcs-2024/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForTutorials : Message:

Error during search on https://www.sigarch.org/call-contributions/isca-2024/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForCreativeAI : Message:

Error during search on https://kdd2024.kdd.org/call-for-hands-on-tutorials/ : Message:

Error during search on https://blog.neurips.cc/2024/03/03/neurips-2024-call-for-competitions/#mobile-menu : Message:

Error during search on https://kdd2024.kdd.org/call-for-award-nominations/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallforHighSchoolProjects#child-menu : Message:

Error during search on https://www.sigarch.org/call-contributions/pact-2024-call-for-workshops-tutorials/ : Message:

Error during search on https://www.sigarch.org/call-contributions/roboarch-micro-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/heart-2024/ : Message:

Error during search on https://2024.sigmod.org/calls_sigmod_demos.shtml : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForCreativeAI# : Message:

Error during search on https://kdd2024.kdd.org/call-for-student-travel-award-applications/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForTutorials#main : Message:

Error during search on https://www.sigarch.org/call-contributions/wddsa-micro-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/hotcarbon-2024/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForTutorials# : Message:

Error during search on https://www.sigarch.org/call-contributions/ispass-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/iccd-2024-call-for-papers/ : Message:
```



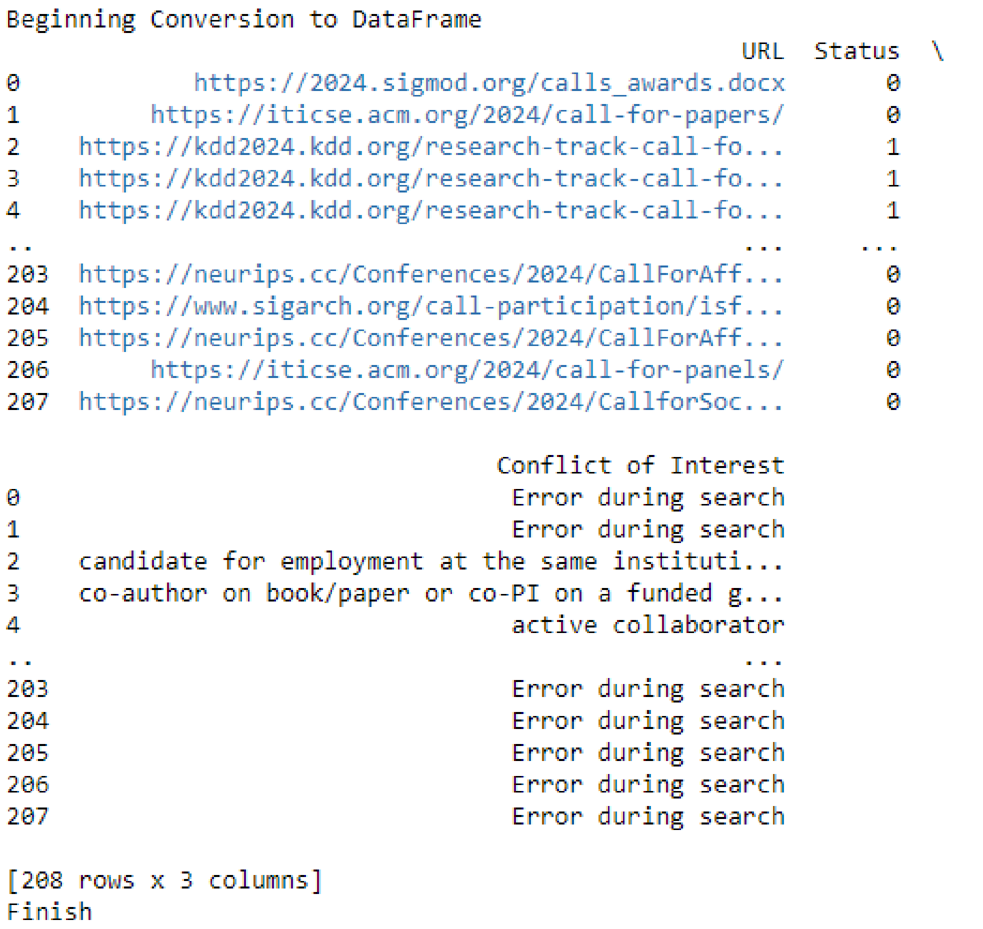
```
Error during search on https://www.aies-conference.com/2024/call-for-papers-students/ : Message:

Error during search on https://www.sigops.org/2024/call-for-participation-apsys-2024/#content : Message:

Error during search on https://blog.neurips.cc/2024/04/15/announcing-the-neurips-2024-call-for-tutorials/# : Message:

Error during search on https://www.sigarch.org/call-contributions/arc-2024/ : Message:

Error during search on https://www.sigops.org/2024/call-for-participation-apsys-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/iiswc-2024-2/ : Message:

Error during search on https://www.sigarch.org/call-participation/isca-2024-buenos-aires-argentina-june-29-july-3/ : Message:

Error during search on https://www.sigarch.org/call-participation/iccd-2024/ : Message:

Error during search on https://www.sigarch.org/call-participation/esweek-2024/ : Message:

Error during search on https://www.sigarch.org/call-contributions/paw-atm-2024-parallel-applications-workshop-alternatives-to-m
pix/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForExpo#child-menu : Message:

Error during search on https://neurips.cc/Conferences/2024/CallforHighSchoolProjects# : Message:

Error during search on https://www.sigarch.org/call-contributions/call-for-papers-arc-lg2024-new-approaches-for-addressing-the-
computing-requirements-of-llms-and-gnns/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForExpo# : Message:

Error during search on https://www.sigarch.org/call-contributions/iiswc-2024/ : Message:

Error during search on https://www.sigarch.org/other-announcements/call-for-nominations-micro-test-of-time-award-2024/# : Messa
ge:

Error during search on https://iticse.acm.org/2024/call-for-posters/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForAffinityEvents#child-menu : Message:

Error during search on https://www.sigarch.org/call-participation/isfpga-2024/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallForAffinityEvents# : Message:


Error during search on https://iticse.acm.org/2024/call-for-panels/ : Message:

Error during search on https://neurips.cc/Conferences/2024/CallforSocials#child-menu : Message:
```

Code input (Conversion to database then to CSV file):



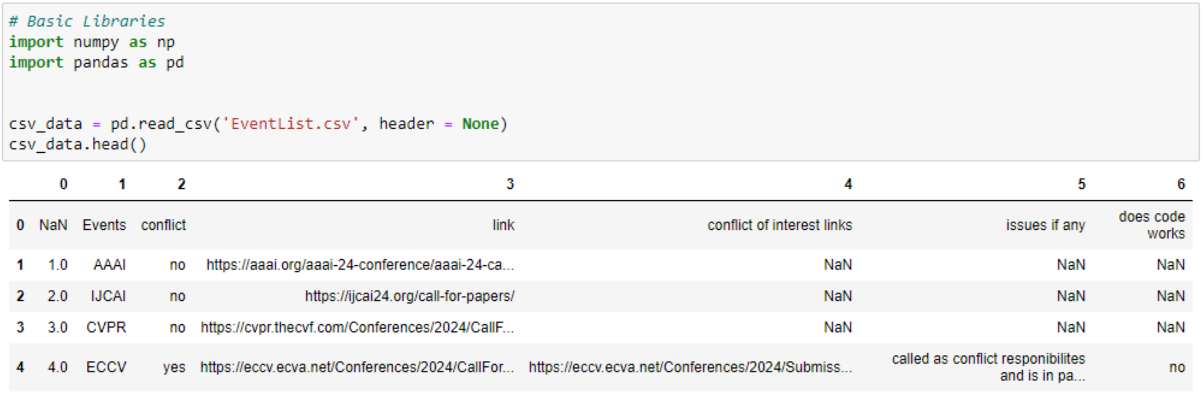
```
print("Beginning Conversion to DataFrame")

# Convert the data to a DataFrame
df = pd.DataFrame(data, columns=['URL', 'Status', 'Conflict of Interest'])
print(df)

# Close the driver
driver.quit()

# Get the path to the desktop directory
desktop_path = os.path.join(os.path.join(os.environ['USERPROFILE']), 'Desktop')

# Specify the file name
csv_file_name = 'Data(2).csv'

# Concatenate the desktop path and file name
csv_file_path = os.path.join(desktop_path, csv_file_name)

# Saving DataFrame to CSV on desktop
df.to_csv(csv_file_path, index=False)

print("Finish")
```

Code output (Conversion to database then to CSV file) :



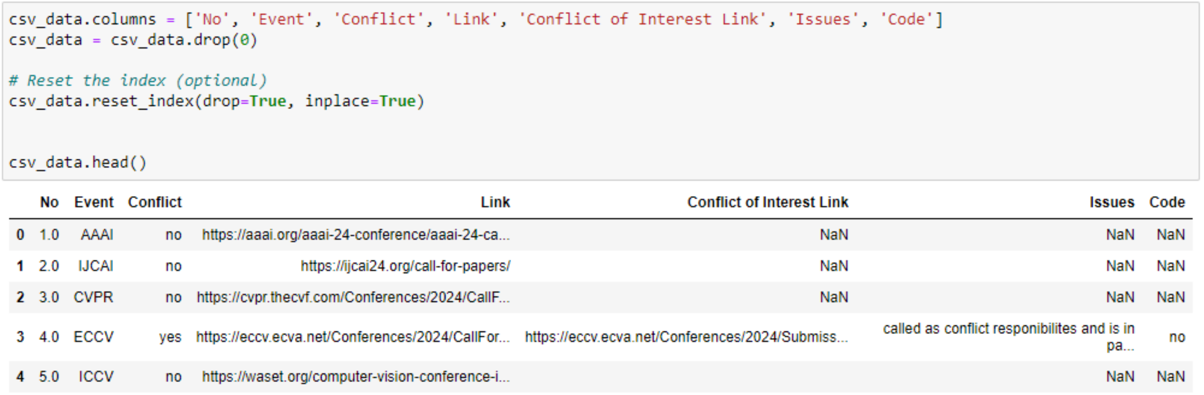
```
Beginning Conversion to DataFrame
                                                   URL  Status  \
0             https://2024.sigmod.org/calls_awards.docx       0   
1          https://iticse.acm.org/2024/call-for-papers/       0   
2     https://kdd2024.kdd.org/research-track-call-fo...       1   
3     https://kdd2024.kdd.org/research-track-call-fo...       1   
4     https://kdd2024.kdd.org/research-track-call-fo...       1   
..                                                  ...     ...   
203   https://neurips.cc/Conferences/2024/CallForAff...       0   
204   https://www.sigarch.org/call-participation/isf...       0   
205   https://neurips.cc/Conferences/2024/CallForAff...       0   
206        https://iticse.acm.org/2024/call-for-panels/       0   
207   https://neurips.cc/Conferences/2024/CallforSoc...       0   

                                  Conflict of Interest  
0                                  Error during search  
1                                  Error during search  
2     candidate for employment at the same instituti...  
3     co-author on book/paper or co-PI on a funded g...  
4                                  active collaborator  
..                                                  ...  
203                                Error during search  
204                                Error during search  
205                                Error during search  
206                                Error during search  
207                                Error during search  

[208 rows x 3 columns]
Finish
```

Data Processing (EventList.csv file is the file I collected all the events that was manually collected and checked):



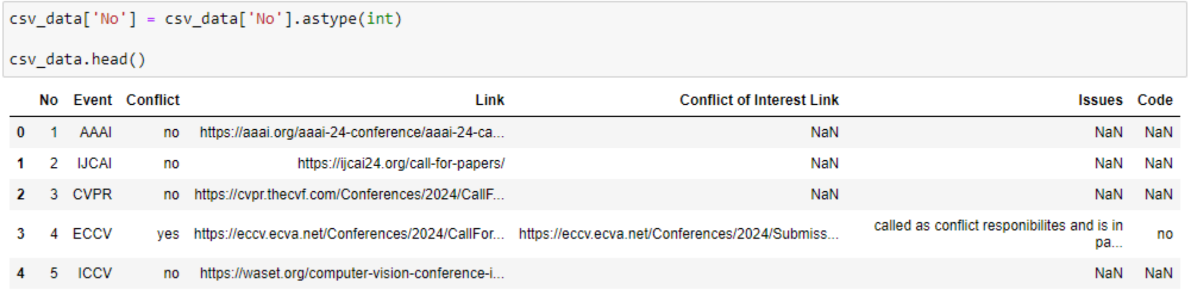
```
# Basic Libraries
import numpy as np
import pandas as pd


csv_data = pd.read_csv('EventList.csv', header = None)
csv_data.head()
```

	0	1	2	3	4	5	6
0	NaN	Events	conflict	link	conflict of interest links	issues if any	does code works
1	1.0	AAAI	no	https://aaai.org/aaai-24-conference/aaai-24-ca...	NaN	NaN	NaN
2	2.0	IJCAI	no	https://ijcai24.org/call-for-papers/	NaN	NaN	NaN
3	3.0	CVPR	no	https://cvpr.thecvf.com/Conferences/2024/CallF...	NaN	NaN	NaN
4	4.0	ECCV	yes	https://eccv.ecva.net/Conferences/2024/CallFor...	https://eccv.ecva.net/Conferences/2024/Submiss...	called as conflict responibilites and is in pa...	no



```
csv_data.columns = ['No', 'Event', 'Conflict', 'Link', 'Conflict of Interest Link', 'Issues', 'Code']
csv_data = csv_data.drop(0)

# Reset the index (optional)
csv_data.reset_index(drop=True, inplace=True)


csv_data.head()
```

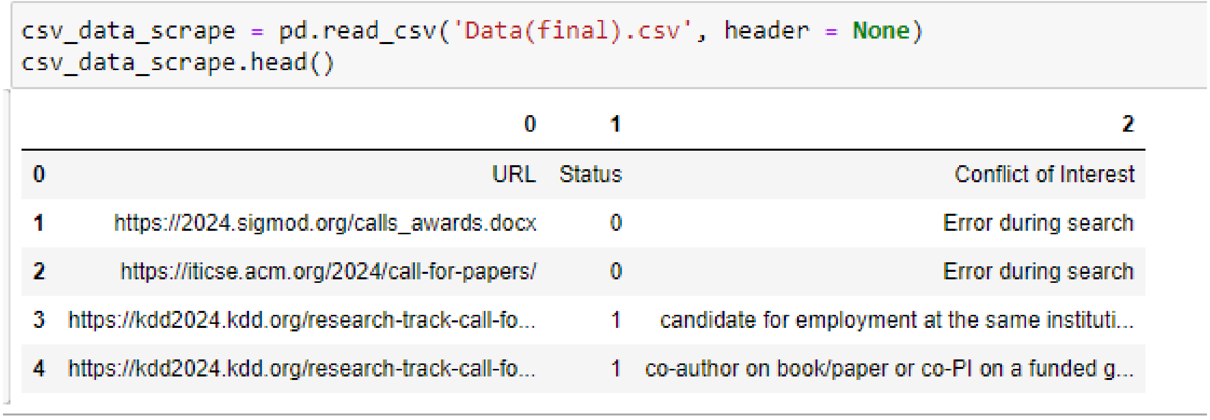
	No	Event	Conflict	Link	Conflict of Interest Link	Issues	Code
0	1.0	AAAI	no	https://aaai.org/aaai-24-conference/aaai-24-ca...	NaN	NaN	NaN
1	2.0	IJCAI	no	https://ijcai24.org/call-for-papers/	NaN	NaN	NaN
2	3.0	CVPR	no	https://cvpr.thecvf.com/Conferences/2024/CallF...	NaN	NaN	NaN
3	4.0	ECCV	yes	https://eccv.ecva.net/Conferences/2024/CallFor...	https://eccv.ecva.net/Conferences/2024/Submiss...	called as conflict responibilites and is in pa...	no
4	5.0	ICCV	no	https://waset.org/computer-vision-conference-i...		NaN	NaN



```
csv_data['No'] = csv_data['No'].astype(int)

csv_data.head()
```

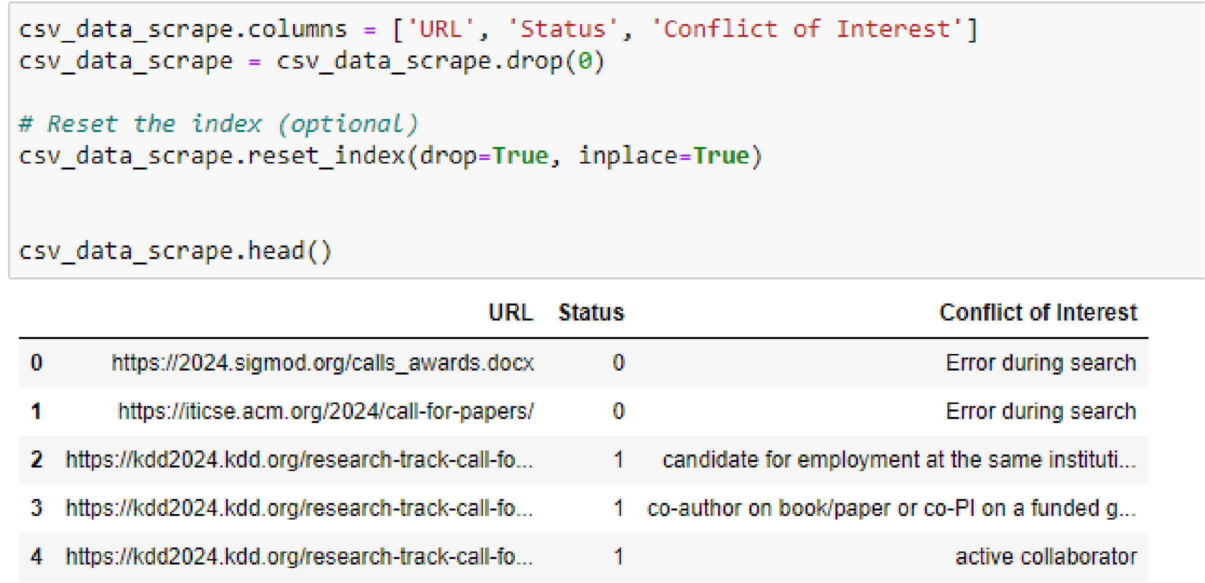
	No	Event	Conflict	Link	Conflict of Interest Link	Issues	Code
0	1	AAAI	no	https://aaai.org/aaai-24-conference/aaai-24-ca...	NaN	NaN	NaN
1	2	IJCAI	no	https://ijcai24.org/call-for-papers/	NaN	NaN	NaN
2	3	CVPR	no	https://cvpr.thecvf.com/Conferences/2024/CallF...	NaN	NaN	NaN
3	4	ECCV	yes	https://eccv.ecva.net/Conferences/2024/CallFor...	https://eccv.ecva.net/Conferences/2024/Submiss...	called as conflict responibilites and is in pa...	no
4	5	ICCV	no	https://waset.org/computer-vision-conference-i...		NaN	NaN

Data Processing (Data(Final).csv file is the file where the data collected by code is stored):



```
csv_data_scrape = pd.read_csv('Data(final).csv', header = None)
csv_data_scrape.head()
```

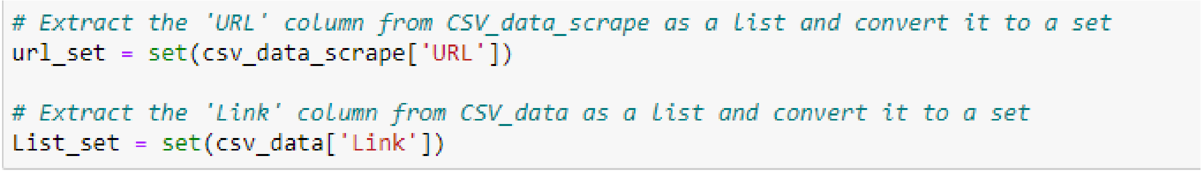
	0	1	2
0	URL	Status	Conflict of Interest
1	https://2024.sigmod.org/calls_awards.docx	0	Error during search
2	https://iticse.acm.org/2024/call-for-papers/	0	Error during search
3	https://kdd2024.kdd.org/research-track-call-fo...	1	candidate for employment at the same instituti...
4	https://kdd2024.kdd.org/research-track-call-fo...	1	co-author on book/paper or co-PI on a funded g...



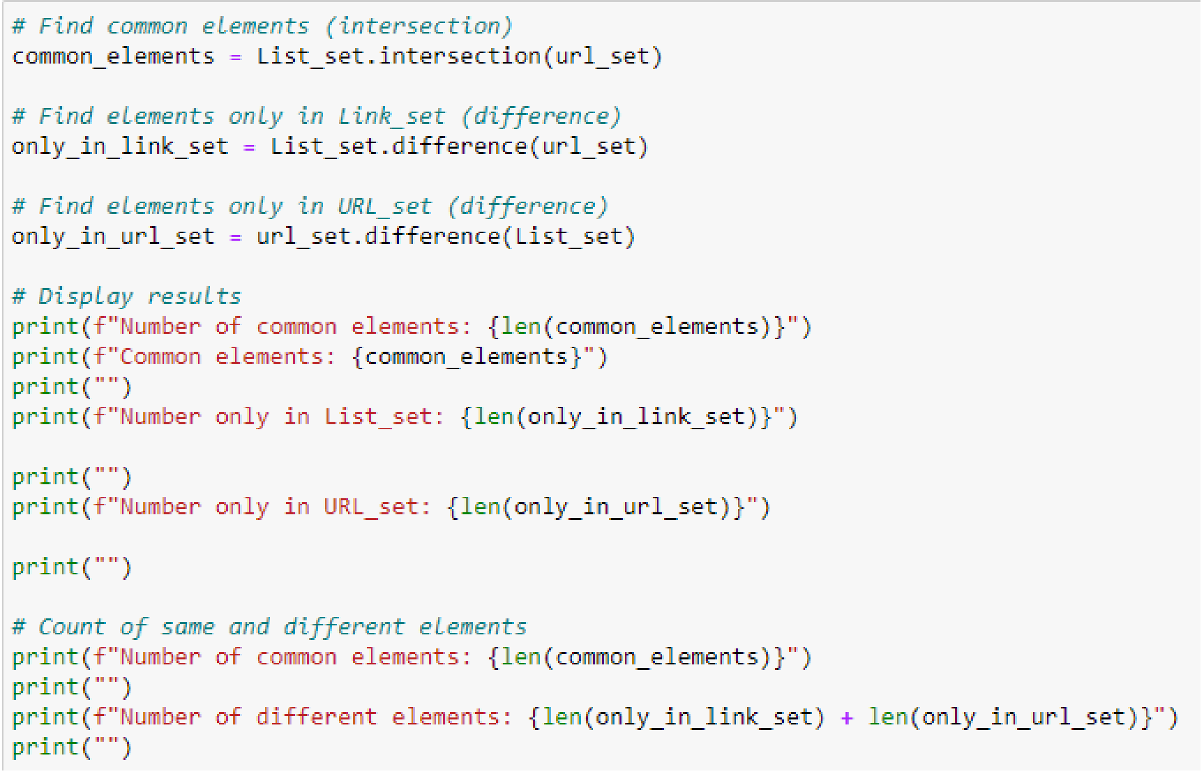
```
csv_data_scrape.columns = ['URL', 'Status', 'Conflict of Interest']
csv_data_scrape = csv_data_scrape.drop(0)

# Reset the index (optional)
csv_data_scrape.reset_index(drop=True, inplace=True)


csv_data_scrape.head()
```

	URL	Status	Conflict of Interest
0	https://2024.sigmod.org/calls_awards.docx	0	Error during search
1	https://iticse.acm.org/2024/call-for-papers/	0	Error during search
2	https://kdd2024.kdd.org/research-track-call-fo...	1	candidate for employment at the same instituti...
3	https://kdd2024.kdd.org/research-track-call-fo...	1	co-author on book/paper or co-PI on a funded g...
4	https://kdd2024.kdd.org/research-track-call-fo...	1	active collaborator

Analysis Comparison of the manually checking and code scraping (input):

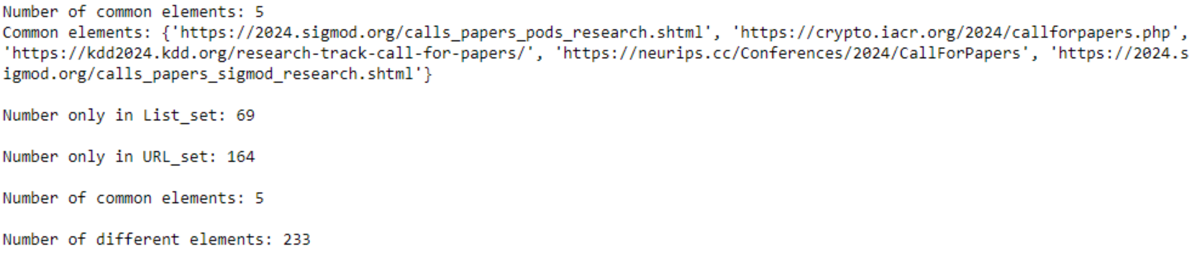


```python
# Extract the 'URL' column from CSV_data_scrape as a list and convert it to a set
url_set = set(csv_data_scrape['URL'])

# Extract the 'Link' column from CSV_data as a list and convert it to a set
List_set = set(csv_data['Link'])
```



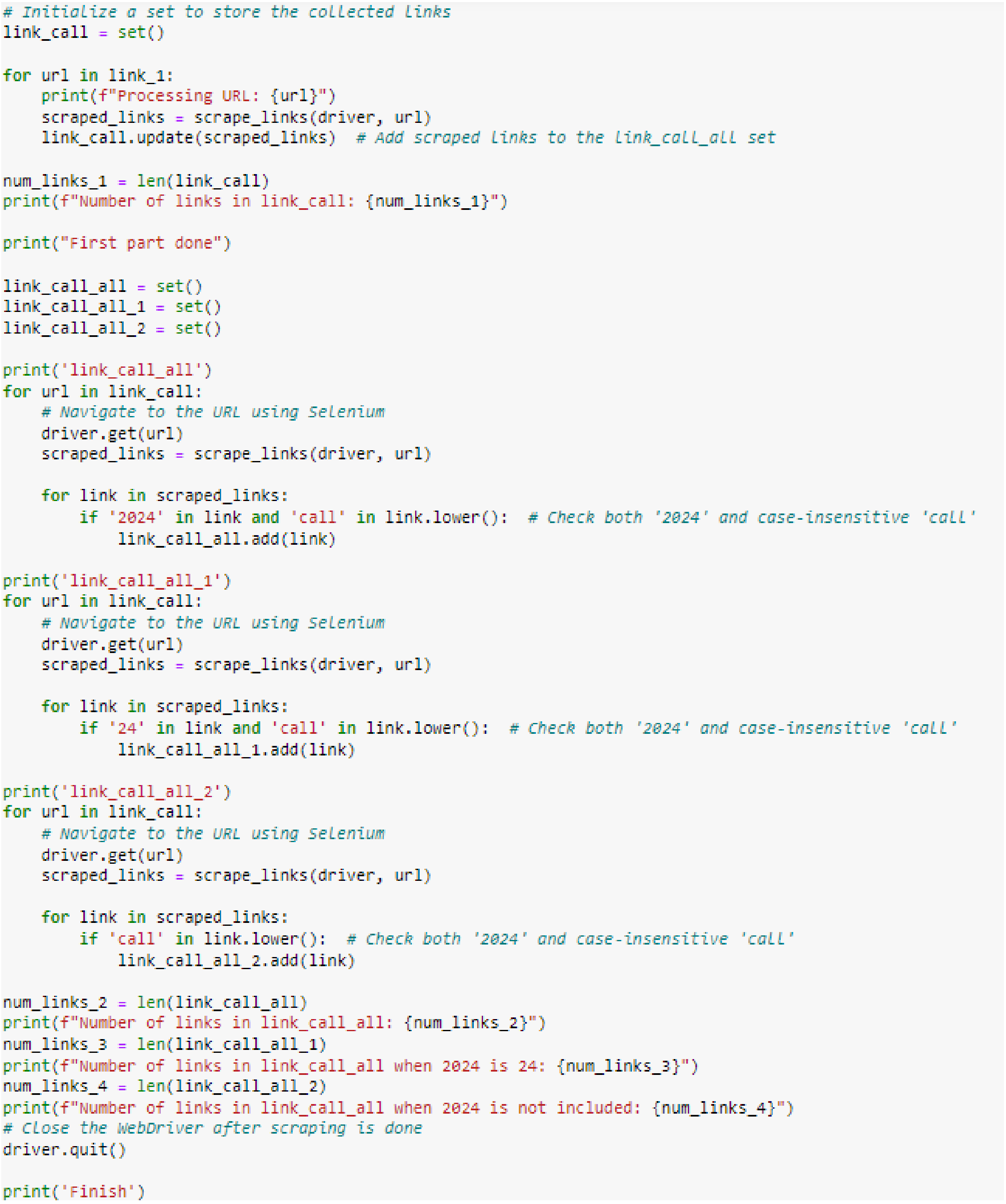
```python
# Find common elements (intersection)
common_elements = List_set.intersection(url_set)

# Find elements only in Link_set (difference)
only_in_link_set = List_set.difference(url_set)

# Find elements only in URL_set (difference)
only_in_url_set = url_set.difference(List_set)

# Display results
print(f"Number of common elements: {len(common_elements)}")
print(f"Common elements: {common_elements}")
print("")
print(f"Number only in List_set: {len(only_in_link_set)}")

print("")
print(f"Number only in URL_set: {len(only_in_url_set)}")

print("")

# Count of same and different elements
print(f"Number of common elements: {len(common_elements)}")
print("")
print(f"Number of different elements: {len(only_in_link_set) + len(only_in_url_set)}")
print("")
```

Analysis Comparison of the manually checking and code scraping (output):



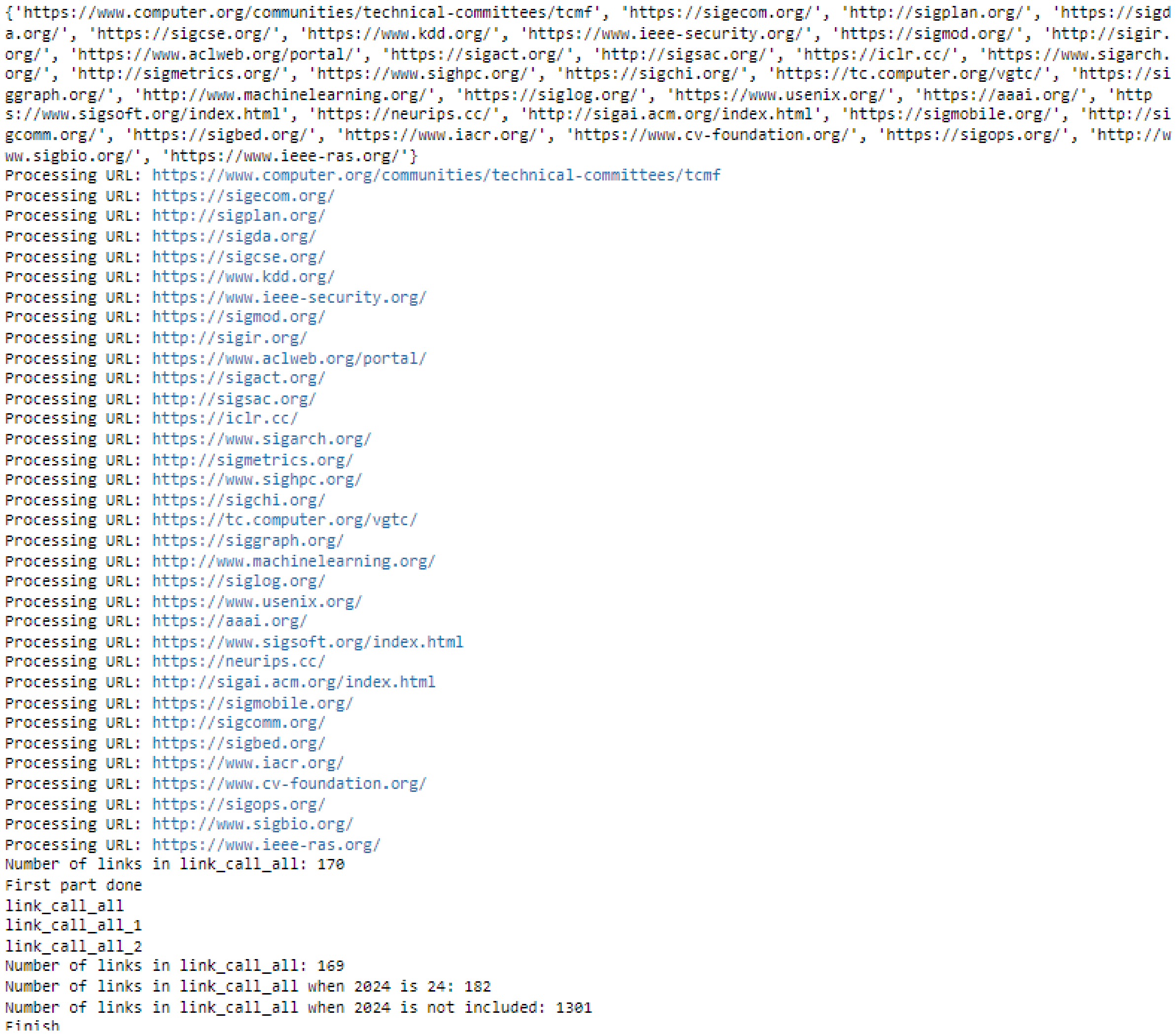
```
Number of common elements: 5
Common elements: {'https://2024.sigmod.org/calls_papers_pods_research.shtml', 'https://crypto.iacr.org/2024/callforpapers.php',
'https://kdd2024.kdd.org/research-track-call-for-papers/', 'https://neurips.cc/Conferences/2024/CallForPapers', 'https://2024.s
igmod.org/calls_papers_sigmod_research.shtml'}

Number only in List_set: 69

Number only in URL_set: 164

Number of common elements: 5

Number of different elements: 233
```

Analysis of the difference in links scraped based on filtering of link_call_all in part 2 of scrape_link (input):



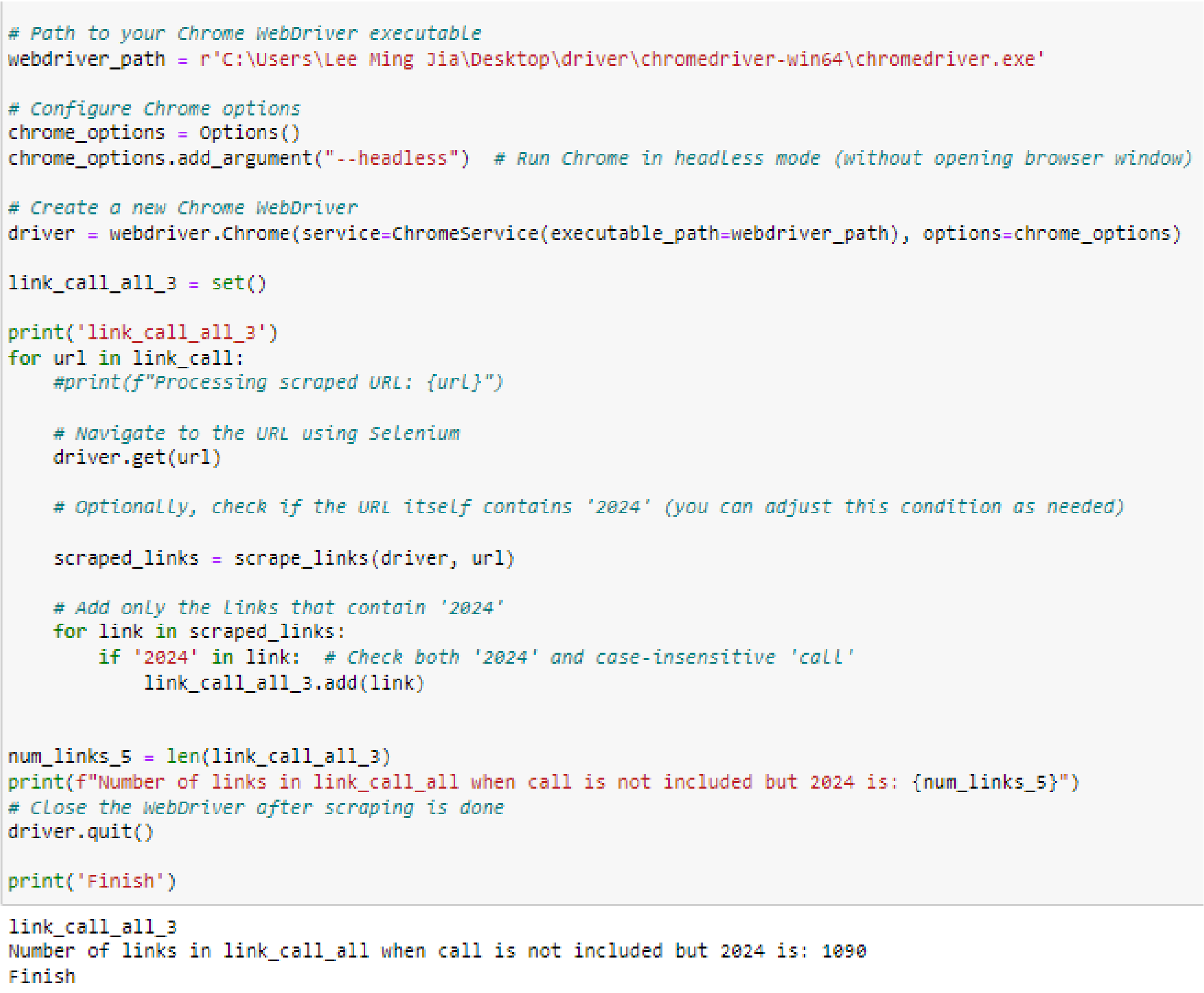
```python
# Initialize a set to store the collected links
link_call = set()

for url in link_1:
    print(f"Processing URL: {url}")
    scraped_links = scrape_links(driver, url)
    link_call.update(scraped_links)  # Add scraped links to the link_call_all set

num_links_1 = len(link_call)
print(f"Number of links in link_call: {num_links_1}")

print("First part done")

link_call_all = set()
link_call_all_1 = set()
link_call_all_2 = set()

print('link_call_all')
for url in link_call:
    # Navigate to the URL using Selenium
    driver.get(url)
    scraped_links = scrape_links(driver, url)

    for link in scraped_links:
        if '2024' in link and 'call' in link.lower():  # Check both '2024' and case-insensitive 'call'
            link_call_all.add(link)

print('link_call_all_1')
for url in link_call:
    # Navigate to the URL using Selenium
    driver.get(url)
    scraped_links = scrape_links(driver, url)

    for link in scraped_links:
        if '24' in link and 'call' in link.lower():  # Check both '2024' and case-insensitive 'call'
            link_call_all_1.add(link)

print('link_call_all_2')
for url in link_call:
    # Navigate to the URL using Selenium
    driver.get(url)
    scraped_links = scrape_links(driver, url)

    for link in scraped_links:
        if 'call' in link.lower():  # Check both '2024' and case-insensitive 'call'
            link_call_all_2.add(link)

num_links_2 = len(link_call_all)
print(f"Number of links in link_call_all: {num_links_2}")
num_links_3 = len(link_call_all_1)
print(f"Number of links in link_call_all when 2024 is 24: {num_links_3}")
num_links_4 = len(link_call_all_2)
print(f"Number of links in link_call_all when 2024 is not included: {num_links_4}")
# Close the WebDriver after scraping is done
driver.quit()

print('Finish')
```

Analysis of the difference in links scraped based on filtering of link_call_all in part 2 of scrape_link (output):



```
{'https://www.computer.org/communities/technical-committees/tcmf', 'https://sigecom.org/', 'http://sigplan.org/', 'https://sigd
a.org/', 'https://sigcse.org/', 'https://www.kdd.org/', 'https://www.ieee-security.org/', 'https://sigmod.org/', 'http://sigir.
org/', 'https://www.aclweb.org/portal/', 'https://sigact.org/', 'http://sigsac.org/', 'https://iclr.cc/', 'https://www.sigarch.
org/', 'http://sigmetrics.org/', 'https://www.sighpc.org/', 'https://sigchi.org/', 'https://tc.computer.org/vgtc/', 'https://si
ggraph.org/', 'http://www.machinelearning.org/', 'https://siglog.org/', 'https://www.usenix.org/', 'https://aaai.org/', 'http
s://www.sigsoft.org/index.html', 'https://neurips.cc/', 'http://sigai.acm.org/index.html', 'https://sigmobile.org/', 'http://si
gcomm.org/', 'https://sigbed.org/', 'https://www.iacr.org/', 'https://www.cv-foundation.org/', 'https://sigops.org/', 'http://w
ww.sigbio.org/', 'https://www.ieee-ras.org/'}
Processing URL: https://www.computer.org/communities/technical-committees/tcmf
Processing URL: https://sigecom.org/
Processing URL: http://sigplan.org/
Processing URL: https://sigda.org/
Processing URL: https://sigcse.org/
Processing URL: https://www.kdd.org/
Processing URL: https://www.ieee-security.org/
Processing URL: https://sigmod.org/
Processing URL: http://sigir.org/
Processing URL: https://www.aclweb.org/portal/
Processing URL: https://sigact.org/
Processing URL: http://sigsac.org/
Processing URL: https://iclr.cc/
Processing URL: https://www.sigarch.org/
Processing URL: http://sigmetrics.org/
Processing URL: https://www.sighpc.org/
Processing URL: https://sigchi.org/
Processing URL: https://tc.computer.org/vgtc/
Processing URL: https://siggraph.org/
Processing URL: http://www.machinelearning.org/
Processing URL: https://siglog.org/
Processing URL: https://www.usenix.org/
Processing URL: https://aaai.org/
Processing URL: https://www.sigsoft.org/index.html
Processing URL: https://neurips.cc/
Processing URL: http://sigai.acm.org/index.html
Processing URL: https://sigmobile.org/
Processing URL: http://sigcomm.org/
Processing URL: https://sigbed.org/
Processing URL: https://www.iacr.org/
Processing URL: https://www.cv-foundation.org/
Processing URL: https://sigops.org/
Processing URL: http://www.sigbio.org/
Processing URL: https://www.ieee-ras.org/
Number of links in link_call_all: 170
First part done
link_call_all
link_call_all_1
link_call_all_2
Number of links in link_call_all: 169
Number of links in link_call_all when 2024 is 24: 182
Number of links in link_call_all when 2024 is not included: 1301
Finish
```

Analysis of the difference in links scraped based on filtering of link_call_all in part 2 of scrape_link (input/ output):



```python
# Path to your Chrome WebDriver executable
webdriver_path = r'C:\Users\Lee Ming Jia\Desktop\driver\chromedriver-win64\chromedriver.exe'

# Configure Chrome options
chrome_options = Options()
chrome_options.add_argument("--headless")  # Run Chrome in headless mode (without opening browser window)

# Create a new Chrome WebDriver
driver = webdriver.Chrome(service=ChromeService(executable_path=webdriver_path), options=chrome_options)

link_call_all_3 = set()

print('link_call_all_3')
for url in link_call:
    #print(f"Processing scraped URL: {url}")

    # Navigate to the URL using Selenium
    driver.get(url)

    # Optionally, check if the URL itself contains '2024' (you can adjust this condition as needed)

    scraped_links = scrape_links(driver, url)

    # Add only the links that contain '2024'
    for link in scraped_links:
        if '2024' in link:  # Check both '2024' and case-insensitive 'call'
            link_call_all_3.add(link)


num_links_5 = len(link_call_all_3)
print(f"Number of links in link_call_all when call is not included but 2024 is: {num_links_5}")
# Close the WebDriver after scraping is done
driver.quit()

print('Finish')
```

```
link_call_all_3
Number of links in link_call_all when call is not included but 2024 is: 1090
Finish
```