

# UIUC CS410 Fall 2023 Project Documentation

Group Name: Developer Team

- Team members: Allan (shengqi) Huang (Shengqi5), Zengjie Tang (zengjie3), Danmeng Zheng(danmeng2), Chuching Ho (cch11)
- Allan Huang is the captain

Tutorial video link: [https://mediaspace.illinois.edu/media/t/1\\_aqy9j3a4](https://mediaspace.illinois.edu/media/t/1_aqy9j3a4)

**1) An overview of the function of the code (i.e., what it does and what it can be used for). What topic have you chosen? Why is it a problem? How does it relate to the theme and to the class?**

**Our team has chosen theme 1:** Intelligent Browsing. We have improved a browser extension for searching Campuswire posts from the project of last year's students. The link to the fall 2022 original project is here: <https://github.com/tenkinoko/CourseProject>. Our browser extension allows users to search Campuswire posts based on the keyword and see the top 5 relevant posts and the number of likes of that post. Clicking on the title of the post will take the users to the corresponding page.

The search function in Campuswire only returns posts that match the keyword. However, the results are not ranked by their relevance to the query. The project of previous students improved the retrieval results to be the most relevant posts. However, the search results also don't have other information like several likes or the exact date of the posts, which makes it difficult for the users to judge the usability of the posts. Furthermore, the codes of extension and crawler from the previous project did not work like what they showed in the demo. So we solved the crawler and extension malfunctions and we integrated additional information related to the posts with BM25 for more relevant retrieval results. We also improved the extension visibility and showed more information related to the posts for a better user experience.

**2) Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement. Briefly describe any datasets, algorithms, or techniques you plan to use. Which programming language do you plan to use?**

The browser extension CamperaExt is an application that facilitates the retrieval of related Campuswire posts based on user search queries. It consists of two main parts:

1. Crawler

The crawler includes the three subparts:

- Text preprocessing and ranking:
  - Programming language: Python

- Libraries: NLTK, re, NumPy
- Description: These modules handle the preprocessing of the Campuswire posts and then implement the ranking algorithm using BM25 and pivoted length normalization and IDF weighting. The sigmoid-normalized likes\_count for each post is also integrated using weighting.
- Web Crawler:
  - Programming language: Python
  - Libraries: Selenium
  - Description: The module is used for scraping posts from the CS410 class channel. It logs in to the Campuswire and navigates to the channel and retrieves the information from a fixed number of posts, now 20. The number can be changed if the user would like to scrape more pages. The scraped information including PostID, Category, Title, Content, and Likes is stored in cw.txt.
- Flask:
  - Programming language: Python
  - Framework: Flask
  - Libraries: Selenium, NumPy, Flask, Flask-CORS
  - Description: The module creates a Flask web application and initializes the web scraping before the first request. It defines an API endpoint for retrieving the Campuswire posts.

## 2. Chrome extension:

- Programming language: HTML, CSS, Javascript
- Description: The extension “CamperaExt” provides a user-friendly interface to interact with the search system. Users can input a keyword and click the search button, then the top five related posts are retrieved and displayed with clickable links and the number of likes.

Note that please see the source code for a more detailed and step-by-step documentation.

## **3) Documentation of the usage of the software including either documentation of usages of APIs or detailed instructions on how to install and run a software, whichever is applicable—link to presentation video.**

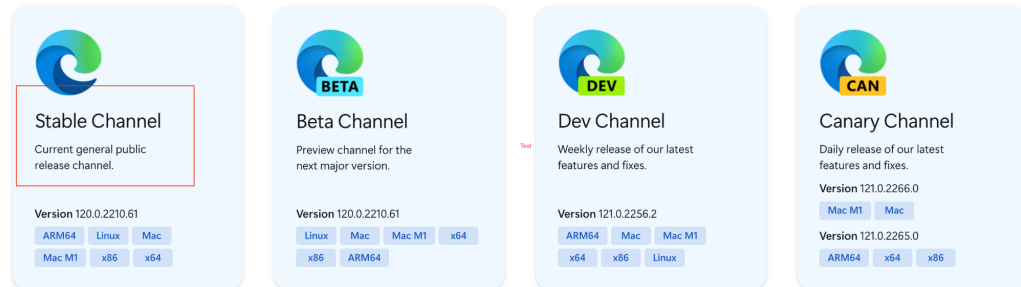
Set up:

- I. Download our source code. <https://github.com/CassyHo/CS410-GroupProject>  
You should see two subfolders: Crawler and Extension
- II. Download WebDriver, pick one of the below methods based on your browser
  1. Edge: The driver for both Windows and macOS is in the Crawler/Drivers folder so you don't need to download or modify anything in the code.  
In case the driver isn't working, here's the link to download

<https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/#download>

Download the driver and put it in the Crawler folder.

For your reference, we used Stable Channel to test on both Windows and macOS and it's working.



## 2. Chrome

Windows User: The driver is in the Crawler/Drivers folder so you don't need to download or modify anything in the code.

Mac user: download the driver from this link:

<https://chromedriver.chromium.org/downloads>

Download the driver and put it in this folder on your PC if you are using macOS:  
`/usr/local/bin`

For your reference, we tested on a macOS M1 and the chrome version is  
Version 120.0.6099.71 (Official Build) (arm64), and this selected driver works


### Stable

Version: 120.0.6099.71 (r1217362)


Binary	Platform	URL	HTTP status
chrome	linux64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/linux64/chrome-linux64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/linux64/chrome-linux64.zip</a>	200
chrome	mac-arm64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-arm64/chrome-mac-arm64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-arm64/chrome-mac-arm64.zip</a>	200
chrome	mac-x64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-x64/chrome-mac-x64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-x64/chrome-mac-x64.zip</a>	200
chrome	win32	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win32/chrome-win32.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win32/chrome-win32.zip</a>	200
chrome	win64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win64/chrome-win64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win64/chrome-win64.zip</a>	200
chromedriver	linux64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/linux64/chromedriver-linux64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/linux64/chromedriver-linux64.zip</a>	200
chromedriver	mac-arm64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-arm64/chromedriver-mac-arm64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-arm64/chromedriver-mac-arm64.zip</a>	200
chromedriver	mac-x64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-x64/chromedriver-mac-x64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-x64/chromedriver-mac-x64.zip</a>	200
chromedriver	win32	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win32/chromedriver-win32.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win32/chromedriver-win32.zip</a>	200
chromedriver	win64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win64/chromedriver-win64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win64/chromedriver-win64.zip</a>	200
chrome-headless-shell	linux64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/linux64/chrome-headless-shell-linux64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/linux64/chrome-headless-shell-linux64.zip</a>	200
chrome-headless-shell	mac-arm64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-arm64/chrome-headless-shell-mac-arm64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-arm64/chrome-headless-shell-mac-arm64.zip</a>	200
chrome-headless-shell	mac-x64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-x64/chrome-headless-shell-mac-x64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/mac-x64/chrome-headless-shell-mac-x64.zip</a>	200
chrome-headless-shell	win32	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win32/chrome-headless-shell-win32.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win32/chrome-headless-shell-win32.zip</a>	200
chrome-headless-shell	win64	<a href="https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win64/chrome-headless-shell-win64.zip">https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/120.0.6099.71/win64/chrome-headless-shell-win64.zip</a>	200

### III. Install the extension on the browser

1. In your browser, go to Extensions -> Manage Extensions
2. Turn on developer mode

Developer mode 

3. Hit Load unpacked -> Navigate to the Extension folder and hit "Select" Edge:

 Load Unpacked  Pack Extension  Update

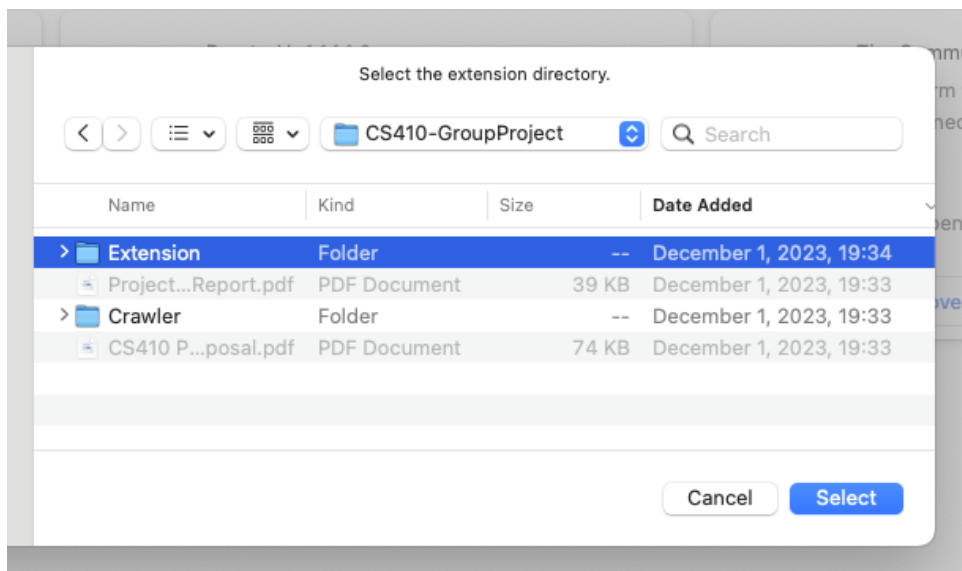
Chrome:

 **Extensions**

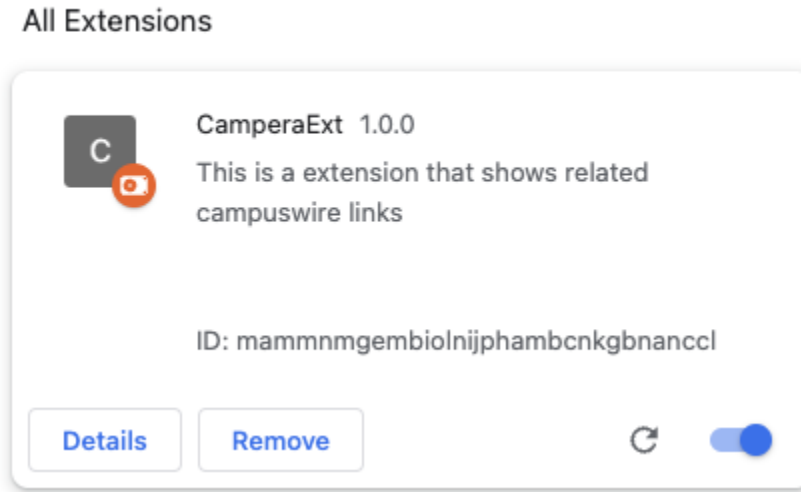
Load unpacked

Pack extension

Update



4. Then you should see it installed



- IV. Install python packages
  - a. Python version: 3.11.4
  - b. selenium: 4.9.0
  - c. nltk: 3.8.1
  - d. flask: 2.2.2
  - e. If you encountered any error that ask you to install any missing packages, please follow the prompt to install
- V. Run crawler
  1. Modify password and username to be your account in the config.py file in Crawler folder

```
1 # login credentials stored in config file to enhance security
2 # enter your Campuswire login
3 username = "xxx"
4 password = "xxx"
```

2. Modify driver option in crawler.py file in Crawler folder.

```

23  def __init__(self):
24      desired_cap = {}
25      # determine which operating system
26      if sys.platform == "win32":
27          # Use this line if you are using Edge
28          self.browser = webdriver.Edge('Drivers/msedgedriver.exe', capabilities=desired_cap)
29
30          # Use this line if you are using Chrome
31          # self.browser = webdriver.Chrome('Drivers/chromedriver.exe')
32      elif sys.platform == "darwin":
33          # Use this line if you are using Edge
34          self.browser = webdriver.Edge('Drivers/msedgedriver', capabilities=desired_cap)
35
36          # Use this line if you are using Chrome
37          # self.browser = webdriver.Chrome()
38

```

3. (Optional) In the crawler.py file in Crawler folder, you can also change the number of most recent posts you'd like to scrape. The default is 100.

```

64      # loop through a fixed number of posts
65      for i in range(0, 100):

```

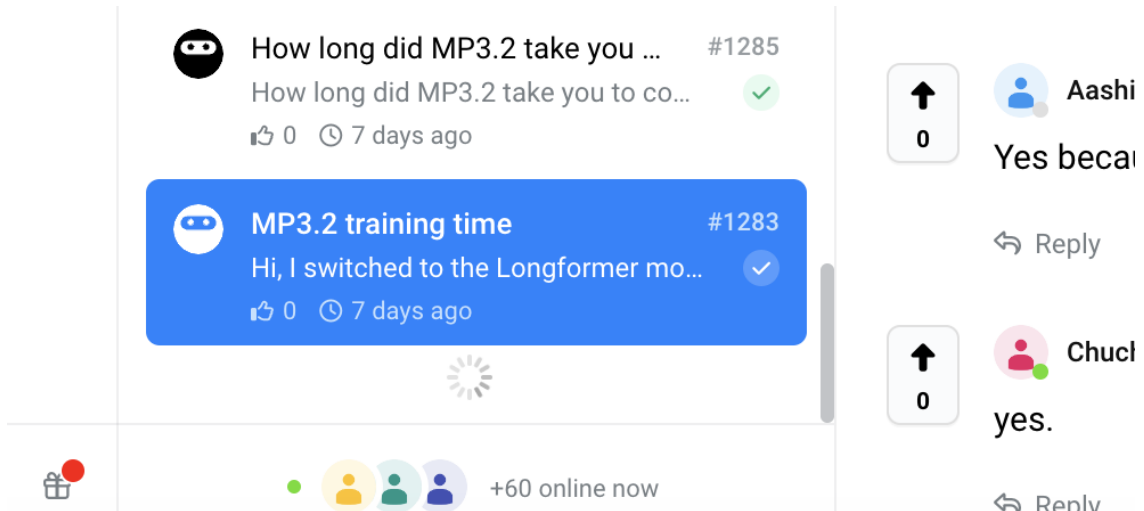
4. In your terminal, navigate to the Crawler folder,  
i.e. `cd /Users/dmz/Desktop/CS410-GroupProject/Crawler`
5. A. For mac, use this command to scrape Campuswire: `FLASK_APP=main.py flask run`

```
(base) dmz@DMZs-MBP crawler % FLASK_APP=main.py flask run
```

- B. For windows, use these two commands `$env:FLASK_APP = "main.py"` and `flask run` to run the application.

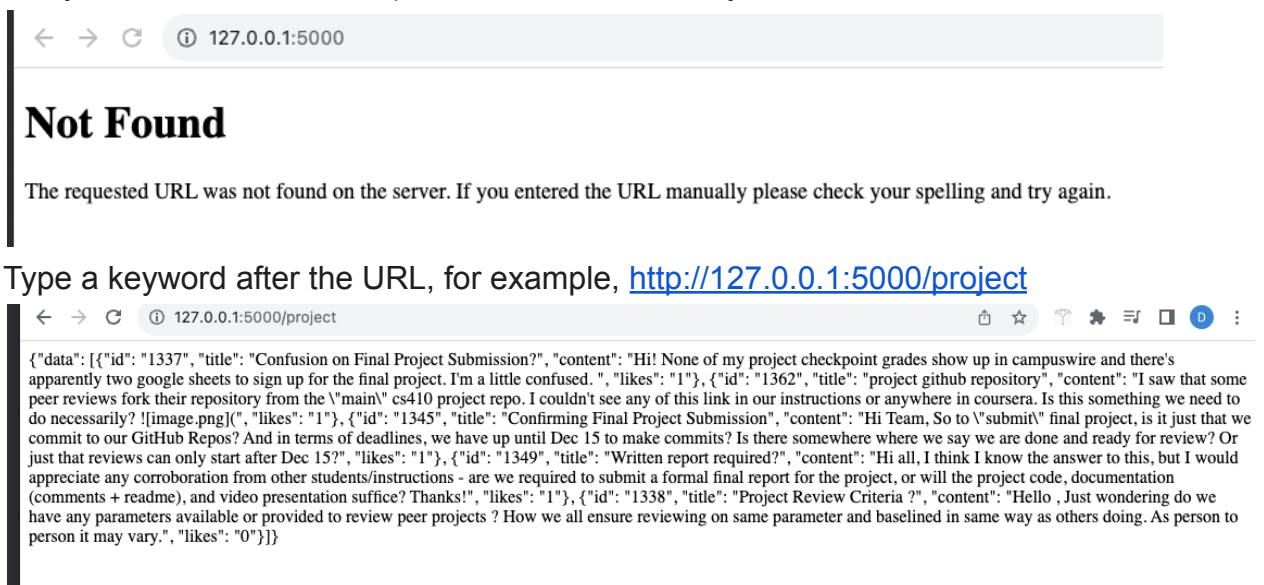
```
$env:FLASK_APP = "main.py"
flask run
```

6. If you encounter any error that asks you to install any missing packages, please follow the prompt to install
7. Go to your browser and type in this URL: <http://127.0.0.1:5000>. You should see a window pop up and start to scrape. If it's stuck at refreshing, i.e. the loading gear keeps spinning for more than 1 minute, hit refresh manually and scroll down to the post where it was stuck to continue scraping.



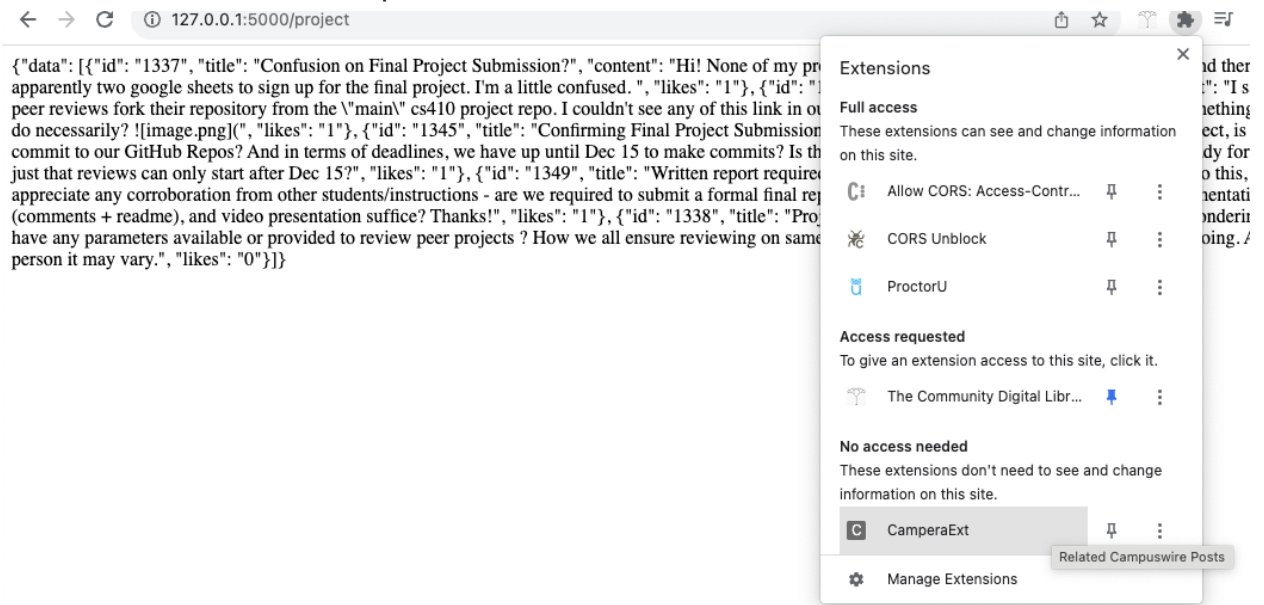
If it's still stuck, go to the terminal and hit ctrl+c to end the crawler and restart.

8. When the process is done, you'll see this. You can either leave this as is or type a keyword to test if the scrape is successful before you test the extension.

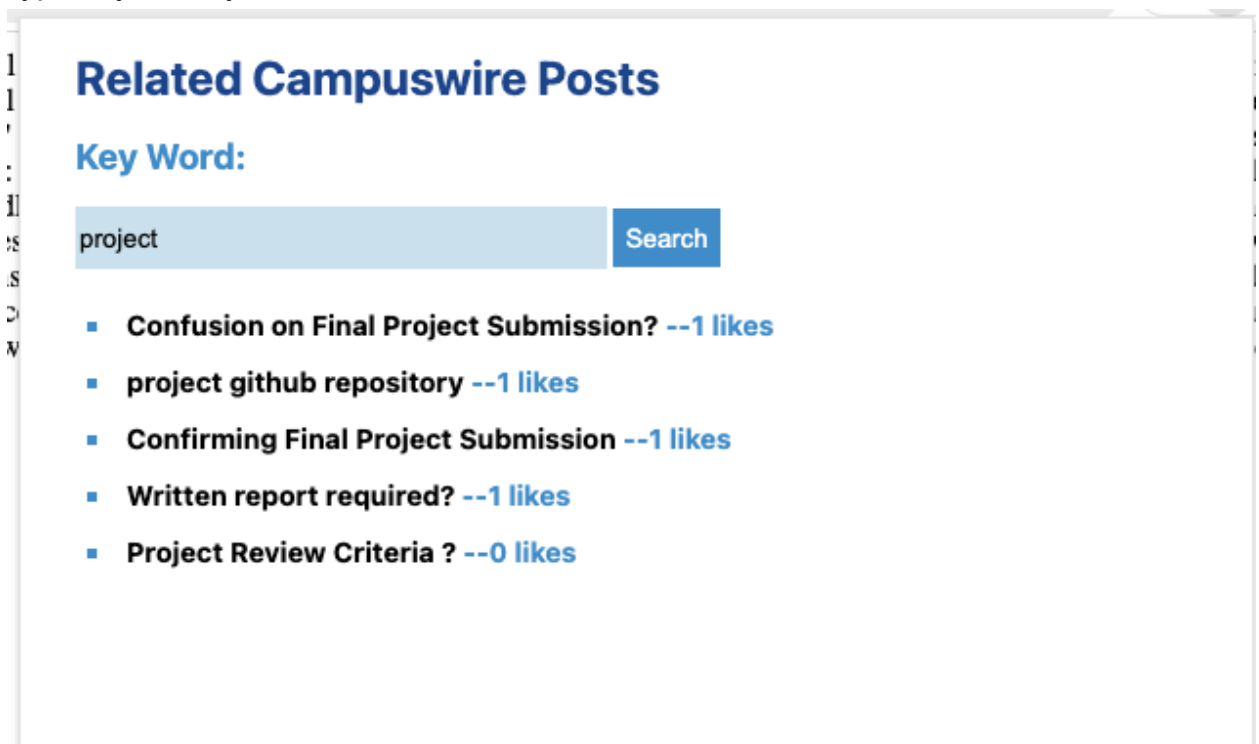


## VI. Test the browser extension

1. Click the extension -> CamperaExt



2. Type keywords you'd like to search and hit the "Search" button



3. Click on the title of the post and the browser will open a tab to take you to the post. Login may be required before you can see the post



**4) How will you demonstrate that your approach will work as expected? If you are improving a function, compare your results to the previously available function. If your implementation works better, show it off. If not, discuss why.**

We entered the same keywords in Campuswire search, the previous project's extension, and our browser extension to compare the results. And it proved that our approach is working as expected and more optimized. The examples shown below are based on 100 most recent posts scraped on Dec 10th, 2023.

- I. First example, we entered "quiz" in all three search platforms. Campuswire results returned any posts that contained "quiz". The screenshot shows the first 5 posts and they are all at least a month ago.

Campuswire results:

The screenshot shows a search interface for 'quiz' on Campuswire. The search bar at the top contains the text 'quiz'. Below the search bar, there are tabs for 'Posts' (49) and 'Chatrooms' (0). A dropdown menu for 'All categories' is visible. The search results are listed below, showing the first 5 posts. Each post includes a profile picture, a title, the content, and the time it was asked. The posts are as follows:

Profile Picture	Title	Content	Time
	<b>==Quiz== Due Dates Clarification</b>	I see that the due date for quizzes in the first half of the course is Oct 8. Is this the "hard deadline" after which you receive 0 credit? Or is this an initial deadline? If it is the "hard deadline", how do we know on what date we start to lose 5 points per day?	Asked 3 months ago
	<b>How can I prepare the Exam1 next week?</b>	How can I prepare the Exam1 next week? any suggestions?	Asked 2 months ago
	<b>Instructor Clarification Required for Below ==Quiz== Due Date Question</b>	quiz due? I saw there was an instructor-endorsed response saying October 8th, but in a previous post	Asked 3 months ago
	<b>Due Date for ==Quizzes==</b>	(i.e. quiz 1 should be done by Sunday on week 1)? Or is it the date noted in Coursera (for quiz 1, it looks to be October 8).	Asked 4 months ago
	<b>Note on Week 12 ==Quiz== Materials</b>	FYI in Week 12, material from both regular and optional lectures is contained in the quiz. It is	Asked a month ago

Our extension and previous students' all retrieved the most recent posts such as quiz 10, 11, 12. And four out of five posts are showing in both extensions. But our extension ranks more liked posts higher. The only post that is different from the previous extension has more likes.

Original extension's results(We added the likes to it for better comparison, the original project does not have likes number):

## Related Campuswire Posts

### Key Word:

- Did anyone make a giant google form with all the previous coursera quiz Q's like last time? --1 likes
- Practice Quiz 11 Explanations Part 2 --2 likes
- Practice Quiz 11 Explanations --1 likes
- Unit 11 Question Practice Quiz --1 likes
- Quiz 12: Which technique to select for a use case? --3 likes

Our extension's results:

## Related Campuswire Posts

### Key Word:

- Quiz 12: Which technique to select for a use case? --3 likes
- Practice Quiz 11 Explanations Part 2 --2 likes
- Did anyone make a giant google form with all the previous coursera quiz Q's like last time? --1 likes
- Unit 10 Practice Quiz Question --2 likes
- Practice Quiz 11 Explanations --1 likes

- II. First example, we entered “peer review” in all three search platforms. The first five of Campuswire relevant posts are related to peer review but the first one is outdated.

## Campuswire results:

peer review


Posts

99+

Chatrooms

0

All categories




Moving due date of ==peer== ==review==

#1087

I just looked at the project overview and it says that "The progress report **peer**-grading should be

Asked 23 days ago




==Peer== ==review==

#1221

Are we assigned **peer review** topics yet?

Asked 10 days ago




==Peer== ==Review== Grade Contribution

#1301

Can the TAs clarify what % of an individual student's grade contributes towards the **peer review**

Asked 6 days ago




==Peer== ==Review== Grading

#1226

Can we get clarification from the course staff on how the **peer review** will factor into our final

Asked 10 days ago



==Peer== ==Review== Clarification

#1268

mark "source code not yet due" on the **peer review** form in CMT, am I expected to change it after the

Asked 8 days ago

The original extension returns most relevant posts but they don't have many likes so they are likely not so useful for the user. Our extension returns two posts that have 6 and 8 likes that are missing in the original extension.

Original extension's results(We added the likes to it for better comparison, the original project does not have likes number):

## Related Campuswire Posts

### Key Word:

- [Peer review --0 likes](#)
- [Peer Review - Rubrics --1 likes](#)
- [How to check for peer review process. --0 likes](#)
- [CMT Project peer review: No projects to review --1 likes](#)
- [Missed Peer Review Accept Deadline --2 likes](#)

Our extension's results:

## Related Campuswire Posts

### Key Word:

- [Peer Review Grade Contribution --8 likes](#)
- [Peer Review - Rubrics --1 likes](#)
- [Missed Peer Review Accept Deadline --2 likes](#)
- [CMT Project peer review: No projects to review --1 likes](#)
- [Peer Review Grading --6 likes](#)

5) Brief description of the contribution of each team member in case of a multi-person team. Everyone is expected to spend at least 20 hours seriously working on their course project as a minimum, not including the time spent preparing the documentation.

#### Allan (Shengqi) Huang:

- Extension popup addition feature: 5 hours
- Crawler debug: 10 hours
- Crawler addition function: 5 hours

#### Danmeng Zheng(danmeng2):

- Crawler debug: 15 hours
- Extension Javascript debug: 10 hours

**Chuching Ho:**

- Crawler and chrome extension debug and implementation: 21 hours;
- Source code documentation: 3 hours;
- Documentation on how software is implemented: 2 hours

**Zengjie Tang:**

- Crawler and chrome extension debug and implementation: 15 hours
- Testing code functionality and style: 5 hours