CS 172 Final Project Report

1. Collaboration Details

   Team Members: Stanley Zhao (szhao050), David May (dmay004), Lindsey Young (lyoun009)

   Stanley - Worked on the creation of the web crawler in crawler.py, and creation of the flask web app and it's functionality in app.py. Created the output.sh file that is run by the web app to pull data from the documents generated by the indexer. Implemented usage of an output file containing query results that is printed to the web app for the user.

   David - Implemented functionality in crawler.py to read in a .txt file of seed URLs, output html files stored as generic names. Used urllib.robotparser library to check for robots.txt files and avoid crawling prohibited pages. Added multithreading functionality to improve the speed of crawling, and a timer to prevent the crawler from running indefinitely.

   Lindsey - Implemented elasticsearch scripts (indexer.sh), storing of crawled html pages into a folder of output .txt files (in crawler.py). Created the python script (create_json.py) to build a data.json file to bulk load documents into the index.

2. Crawler

   The crawler starts by taking input from a file called seedFile.txt which contains seed URLs, and begins crawling through any links to other pages it can find on each seed page. The algorithm is recursive, so it looks for every possible link it can before the one minute timer runs out, and then returns through all crawled pages to scrape for URLs and html content. The crawler handles duplicates, marking each link inside a set() called visited, and skipping any links it comes across that have already been visited. The crawler also uses multithreading. Three separate threads are created which run concurrently, and each begin crawling at their own respective seed URLs. If more than three seed URLs were provided, the program would handle this by crawling through the first three and waiting until all three threads conclude their crawl before beginning again with the next three seeds located in seedFile.txt. Finally, the crawler avoids prohibited pages with the help of the library urllib.robotparser. This library allows us to quickly read the robots.txt file found in the base directory of any seed URL, and avoid any disallowed URLs found within it as we carry out the crawl.

   The crawler also stores the contents of what it crawls in a folder called "htmls". For each page crawled, a new .txt file is created and the contents are written to that file. We used the Beautiful Soup library to pull only the text content from the pages, as we found that that information would be the most useful to store in the index.

   The crawler is limited in certain ways, as it does not contain itself to only .edu pages, going through any links it can find on any pages indiscriminately. It also stores all

of the found URLs in our html file, but still requires an indexer script in order to organize the raw text from each visited webpage. The crawler is also limited based on the user's connection speed.
To deploy the crawler, the user only needs to use a python command in their terminal, such as "python3 crawler.py".

3. Indexer

   The indexer.sh script allows the user to interact with elasticsearch by typing in a few simple commands such as "b" or "d", rather than having to type out extensive curl commands with many parameters.
   When running the bulk add command (b), indexer.sh runs the python script create_json.py in the background. This script takes the folder of html .txt files that were originally retrieved from the crawler and converts them into the json format. The format is as follows:

   {"index": {"_id": "<id number>"}}
   {"link": "<url>", html: "<html body>"}
   …
   where id_number is an assigned number to keep track of the number of entries, url is the url of the associated document, and html body content of the html.

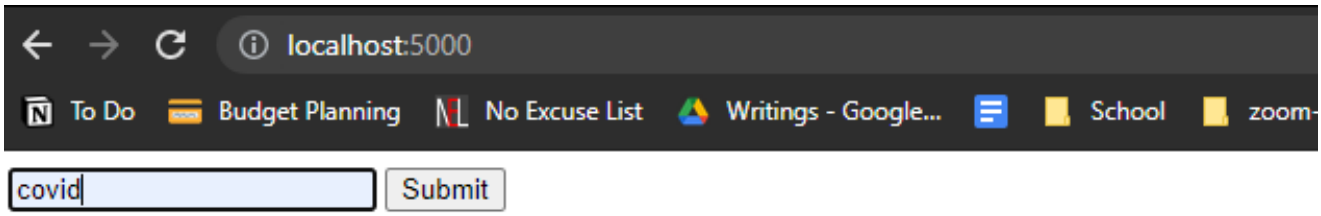
   This is stored in a file called data.json which is used to bulk load the index.

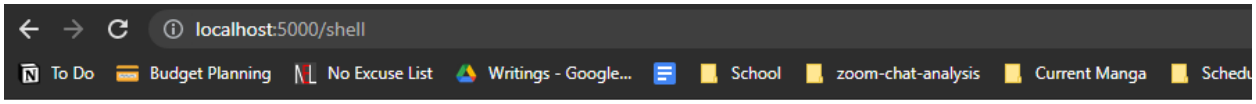
   Instructions to run:

   a. To run, type in the command: ./indexer.sh
   b. Enter in a name for your index (such as "cs172index")
   c. You will be prompted with some options, such as bulk adding files to the index or returning all documents.
      i. enter "b" to bulk add all the data from the crawler
      ii. enter "q" to quit the indexer

4. Extension

   Our extension for the project is a web application that allows the user to input a query into the text box, and then navigate a new page to view the results of the query based on the elasticsearch returned results. The results contain relevant websites, ranked in descending order. The benefit a web app search engine gives to users is allowing them to customize their own search engine, by populating the seed URLs with different websites, users can scrape a specific portion of the internet and then use the web application to search through the results.


localhost:5000
To Do
Budget Planning
No Excuse List
Writings - Google...
School
zoom-
covid
Submit



covid


```
{
  "took" : 153,
  "timed_out" : false,
  "_shards" : {
    "total" : 29,
    "successful" : 29,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 19,
      "relation" : "eq"
    },
    "max_score" : 2.9405324,
    "hits" : [
      {
        "_index" : "cs172index",
        "_type" : "_doc",
        "_id" : "39",
        "_score" : 2.9405324,
        "_source" : {
          "link" : "https://docs.google.com/presentation/?usp=slides_alc",
          "html" : "Learn more about Waze & COVID-19 updates Learn more about Waze & COVID-19 updates"
        }
      },
      {
        "_index" : "cs172index",
        "_type" : "_doc",
        "_id" : "49",
        "_score" : 2.9405324,
        "_source" : {
          "link" : "https://support.google.com/waze/topic/6273402?hl=en&amp;ref_topic=6024567,6024551,",
          "html" : "Learn more about Waze & COVID-19 updates Learn more about Waze & COVID-19 updates"
        }
      },
      {
        "_index" : "cs172index",
        "_type" : "_doc",
        "_id" : "53",
        "_score" : 2.9405324,
        "_source" : {
          "link" : "https://support.google.com/waze/answer/9123774?hl=en&amp;ref_topic=7406710",
          "html" : "Learn more about Waze & COVID-19 updates Learn more about Waze & COVID-19 updates"
        }
      },
      {
        "_index" : "cs172index",
        "_type" : "_doc",
        "_id" : "67",
        "_score" : 2.9390354,
        "_source" : {
          "link" : "https://news.google.com/?tab=un",
          "html" : "Learn more about Google Maps COVID-19 updates.Learn more about Google Maps COVID-19 updates."
        }
      },
      {
```