CS104 Project Report : Web Crawler

DHVANIL GHEEWALA
22B0923

14 June 2023

Introduction

I am Dhvanil gheewala this is my project for CS104 :Web Crawler.
A web crawler, crawler or web spider, is a computer program that's used to search and automatically index website content and other information over the internet. These programs, or bots, are most commonly used to create entries for a search engine index.

Libraries

The libraries and dependencies used in this project are:

- **BeautifulSoup** : Beautiful Soup is a Python package for parsing HTML and XML documents
- **requests** : Used to get send '*http*' requests from the site
- **argparse** : To receive the arguments passed by user while running the code
- **numpy and matplotlib** : For creativity and data representation[1]
- **warnings**: To remove unnecessary warnings and errors
- **os**: To remove existing specific png

Usage

To use this program, head to the terminal and type ”*python3 web_crawler.py -u* $< site - name >$ *-t* $< threshold >$ *-o* $< output - file - name >$”

- **-u**: Enter a valid URL
  for the URL, If not given then it will print an error on the command line.

- **-t**: Enter a positive integer
  for the threshold of recursiveness, must be greater than 0, give an error for an invalid threshold

- **-o**: Enter the name of the file where the links will be added (not append). It will be created if it doesn't exist

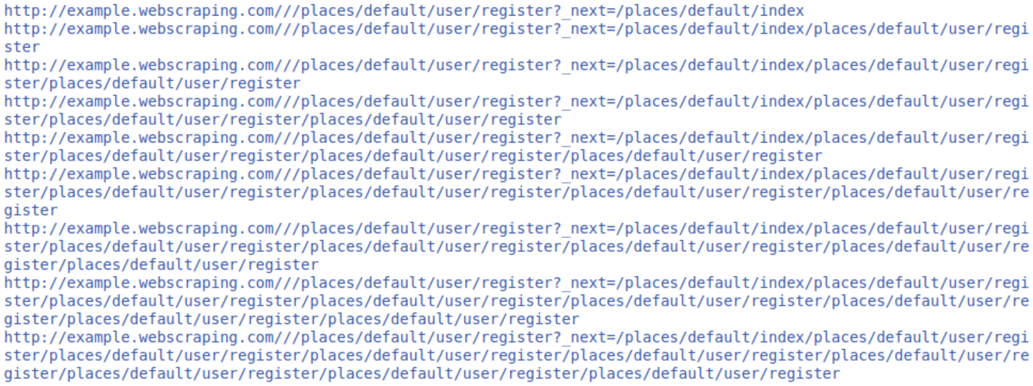
For an output file, If not provided then by default print on the command line.

- **-f**:type *Yes* or *y* (first letter is case insensitive) if you want size of links

Code Structure

Function of base code[2]

Using the base code from *https://www.geeksforgeeks.org/*, we can scrape all urls from the given site and it prints them all without any segregation.Such a data is a very back-breaking and tedious job. Figure 1 was taken from [1]



```
http://example.webscraping.com///places/default/user/register?_next=/places/default/index
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register/places/default/user/re
gister
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register/places/default/user/re
gister/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register/places/default/user/re
gister/places/default/user/register/places/default/user/register
http://example.webscraping.com///places/default/user/register?_next=/places/default/index/places/default/user/regi
ster/places/default/user/register/places/default/user/register/places/default/user/register/places/default/user/re
gister/places/default/user/register/places/default/user/register/places/default/user/register
```

Figure 1: Output of a basic Web crawler

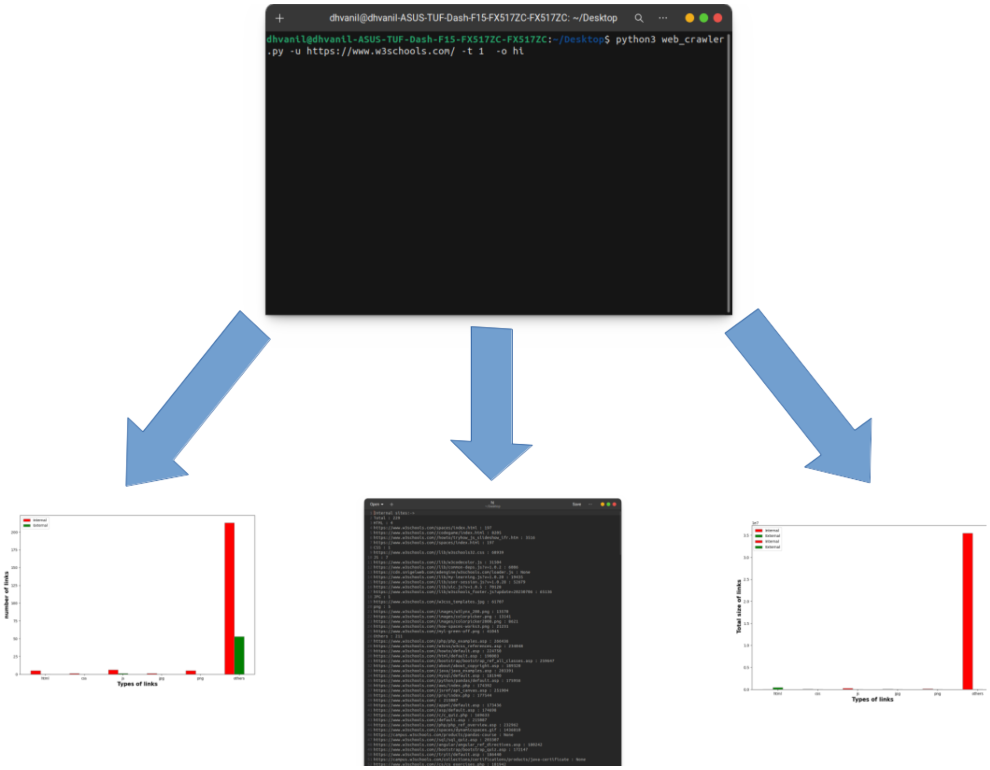
Functions of modified code and Extra features

Using *web_crawler.py* will provide a detailed segregated output according to whether it is Internal or External link and further divided into file types along with size(if required) of each link

It can save the links crawled into a file of you choice and also saves a detailed plot of :

1. Number of files vs type of file
2. Total file size vs type of file **(only if -*f* tag is given value *yes*)**

Note : It deletes the earlier png files to avoid confusion in some scenarios.

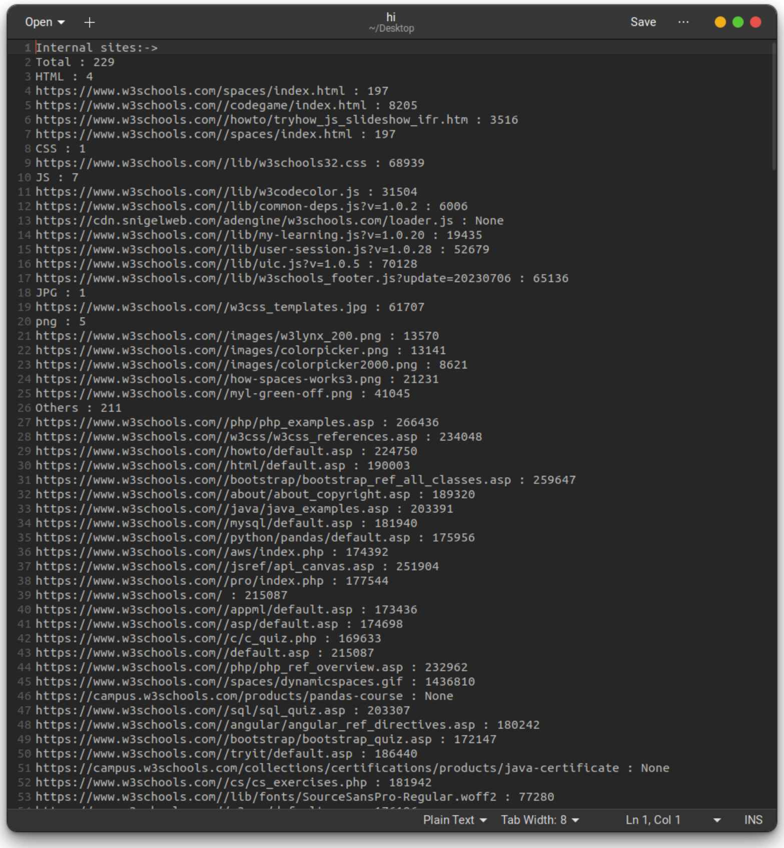


```
dhvanil@dhvanil-ASUS-TUF-Dash-F15-FX517ZC-FX517ZC:~/Desktop$ python3 web_crawler
.py -u https://www.w3schools.com/ -t 1  -o hi
```

Figure 2: Functioning of web_crawler.py

Working

- With the help of BeautifulSoup, the code extracts the links in the *href* and *src* tags and then changing them to make valid urls(some are relative ,so the code prefixes the main site etc.).
- The code is based on recursive crawling through the url given by the user.
- The code finally segregates all the files into various categories like *HTML, CSS, JS, JPG, PNG and Others* using a nested dictionary.



```
Internal sites:->
Total : 229
HTML : 4
https://www.w3schools.com/spaces/index.html : 197
https://www.w3schools.com//codegame/index.html : 8205
https://www.w3schools.com//howto/tryhow_js_slideshow_ifr.htm : 3516
https://www.w3schools.com//spaces/index.html : 197
CSS : 1
https://www.w3schools.com//lib/w3schools32.css : 68939
JS : 7
https://www.w3schools.com//lib/w3codecolor.js : 31504
https://www.w3schools.com//lib/common-deps.js?v=1.0.2 : 6006
https://cdn.snigelweb.com/adengine/w3schools.com/loader.js : None
https://www.w3schools.com//lib/my-learning.js?v=1.0.20 : 19435
https://www.w3schools.com//lib/user-session.js?v=1.0.28 : 52679
https://www.w3schools.com//lib/uic.js?v=1.0.5 : 70128
https://www.w3schools.com//lib/w3schools_footer.js?update=20230706 : 65136
JPG : 1
https://www.w3schools.com//w3css_templates.jpg : 61707
png : 5
https://www.w3schools.com//images/w3lynx_200.png : 13570
https://www.w3schools.com//images/colorpicker.png : 13141
https://www.w3schools.com//images/colorpicker2000.png : 8621
https://www.w3schools.com//how-spaces-works3.png : 21231
https://www.w3schools.com//myl-green-off.png : 41045
Others : 211
https://www.w3schools.com//php/php_examples.asp : 266436
https://www.w3schools.com//w3css/w3css_references.asp : 234048
https://www.w3schools.com//howto/default.asp : 224750
https://www.w3schools.com//html/default.asp : 190003
https://www.w3schools.com//bootstrap/bootstrap_ref_all_classes.asp : 259647
https://www.w3schools.com//about/about_copyright.asp : 189320
https://www.w3schools.com//java/java_examples.asp : 203391
https://www.w3schools.com//mysql/default.asp : 181940
https://www.w3schools.com//python/pandas/default.asp : 175956
https://www.w3schools.com//aws/index.php : 174392
https://www.w3schools.com//jsref/api_canvas.asp : 251904
https://www.w3schools.com//pro/index.php : 177544
https://www.w3schools.com/ : 215087
https://www.w3schools.com//appml/default.asp : 173436
https://www.w3schools.com//asp/default.asp : 174698
https://www.w3schools.com//c/c_quiz.php : 169633
https://www.w3schools.com//default.asp : 215087
https://www.w3schools.com//php/php_ref_overview.asp : 232962
https://www.w3schools.com//spaces/dynamicspaces.gif : 1436810
https://campus.w3schools.com/products/pandas-course : None
https://www.w3schools.com//sql/sql_quiz.asp : 203307
https://www.w3schools.com//angular/angular_ref_directives.asp : 180242
https://www.w3schools.com//bootstrap/bootstrap_quiz.asp : 172147
https://www.w3schools.com//tryit/default.asp : 186440
https://campus.w3schools.com/collections/certifications/products/java-certificate : None
https://www.w3schools.com//cs/cs_exercises.php : 181942
https://www.w3schools.com//lib/fonts/SourceSansPro-Regular.woff2 : 77280
```

Figure 3: Output of web_crawler.py

References

[1] URL: https://www.geeksforgeeks.org/bar-plot-in-matplotlib/.

[2] URL: https://www.geeksforgeeks.org/python-program-to-recursively-scrape-all-the-urls-of-the-website/.