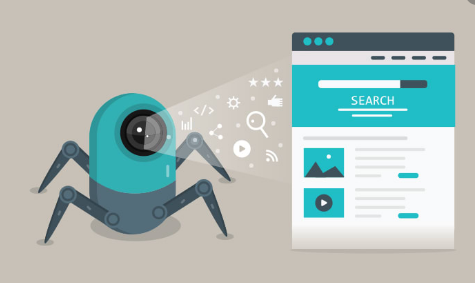
 **Air University Multan campus**

**Department of Computer Science and Engineening**



Web Crawler

**Project Report**

**Group 9 Members**

183158 Aleena Imran

183200 Minal Ejaz

183209 Laiba Naveed

**Contents**

[1.1 Introduction 3](#_Toc73131285)

[1.2 Basic principle of web crawler 3](#_Toc73131286)

[1.3 Several Characteristics: 3](#_Toc73131287)

[2. Problem 4](#_Toc73131288)

[3. Problem Statement 4](#_Toc73131289)

[4. Working 4](#_Toc73131290)

[4.1 Software Tools 4](#_Toc73131291)

[4.2 Libraries 4](#_Toc73131292)

[4.3 Description 5](#_Toc73131293)

[4.4 How to Run 5](#_Toc73131294)

[4.4.1 Requirements 5](#_Toc73131295)

[4.4.2 App Run Commands 5](#_Toc73131296)

[5. Output 6](#_Toc73131297)

[6. Acknowledgement 8](#_Toc73131298)

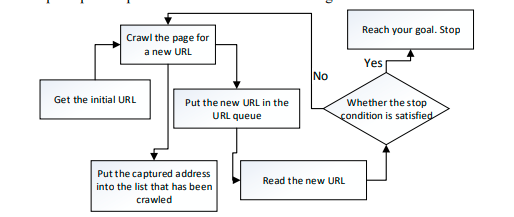
# **1.1 Introduction**

With the emergence of network technology, network data has reached a considerable amount. There are all kinds of big data on the Internet, and the Internet is a collection of these huge data. However, this data is not easily stored in a local database for access and processing. Web crawlers play an important role in collecting network data.

A web crawler is a computer program that traverses hyperlinks and indexes them. As the core part of the vertical search engine, how to make crawlers more accurate and faster to grab information has become an important research direction in the field of crawlers, which has attracted extensive attention from many researchers at home and abroad.

# **1.2 Basic principle of web crawler**

The most important role of the web crawler is to crawl in the big data of the Internet, find effective information, and store the needed information data into the local database.



Crawlers mainly include downloaders, information extractors, schedulers and crawl queues. Scheduler will seed URL provided to download, then downloader get page information from the Internet to find and send information extractor, extractor strategy according to the instruction from information extraction to obtain information and the next level in the URL, then the next level URL to a waiting queue, waiting queue to go to submit the URL of the heavy, filtering and sorting operation into the list, after waiting for the scheduler calls.

# **1.3 Several Characteristics:**

**1-Performance and efficiency: -** A web crawler that traverses a site for the first time can download all available files, enabling system resource utilization efficiency

**2-Quality: -** Web crawlers should give priority to obtaining high-quality pages that users need, and improve the accuracy of obtaining pages

**3-Freshness: -** Keep search engines fresh, crawling their data independently based on the change frequency of each page and database, and crawling new URLs that are randomly created or updated.

# **2. Problem**

Most prominent challenge with current web crawlers, selection of important pages for downloading. It is important for the crawler “To select the pages and to visit “important” pages first by prioritizing the URLs in the queue properly. It minimizing the load on the websites crawled with parallelization of the crawling process.

# **3. Problem Statement**

A web crawler is an automated program script that browses the world-wide-web and it can look and store the contents of the webpage. This process is called web crawling. The search engines like Google use this technique to find up to date information. Make sure to use the multithreading concept.

# **4. Working**

# **4.1 Software Tools**

**Software:**

* VS Code

**Languages:**

* Django
* Python
* HTML
* CSS
* JAVASCRIPT

# **4.2 Libraries**

1- from django.contrib import admin

2- from django.apps import AppConfig

3- from bs4 import BeautifulSoup, SoupStrainer

4- from urllib.request import urlopen

5- from urllib.error import HTTPError, URLError

6- from ordered\_set import OrderedSet

7- from .helpers import is\_url\_valid, get\_clean\_url, is\_link\_internal

8- from concurrent.futures import ThreadPoolExecutor

9- from urllib.parse import urljoin, urlparse

10- from django.urls import path

11- from .views import urlInput

12- from django.views.decorators.csrf import csrf\_exempt

13-from django.http import JsonResponse

14-from. crawler import Crawler

15- from django.contrib import admin

# **4.3 Description**

The crawler will go through many web pages to check the words on the page and where the words are used in other places. Crawlers will build a big index to include all the findings. To put it simply, the index is a list of words as well as the web pages related to those words. When you search "big data" on a certain search engine, the search will check its index and return findings for you.

The major steps executed by a [web crawler](https://www.promptcloud.com/blog/all-you-need-to-know-about-web-crawling/):

1. Select a URL from a group of candidates
2. Download other related and associated web pages
3. Extract the URLs among the associated web pages
4. Add those new URLs to the related candidate

A web crawler starts its work by visiting a list of websites that it has visited before. During the visits, it will also look for other related websites that worth a visit. By constant visiting, web crawlers can discover new pages or URLs, update changes to existing pages, as well as mark those dead links. When web crawlers visit a certain page, it goes through all the content of the page and then conveys it to its database. After the data on the page is captured, the words on the page will be placed into the index of a search engine. You can take [the index](https://www.google.com/search/howsearchworks/crawling-indexing/) as a huge database of words and where they appear on different pages.

Crawlers won't stop completely once they have indexed web pages. They will check if there are any changes made to the web pages from time to time. If there is something new, the index created will also be updated.

# **4.4 How to Run**

## **4.4.1 Requirements**

* Install requests (pip install requests)
* Install beautifulsoup (pip install beautifulsoup)
* Install lxml(pip install lxml)
* Install orderedset(pip install orderedset)

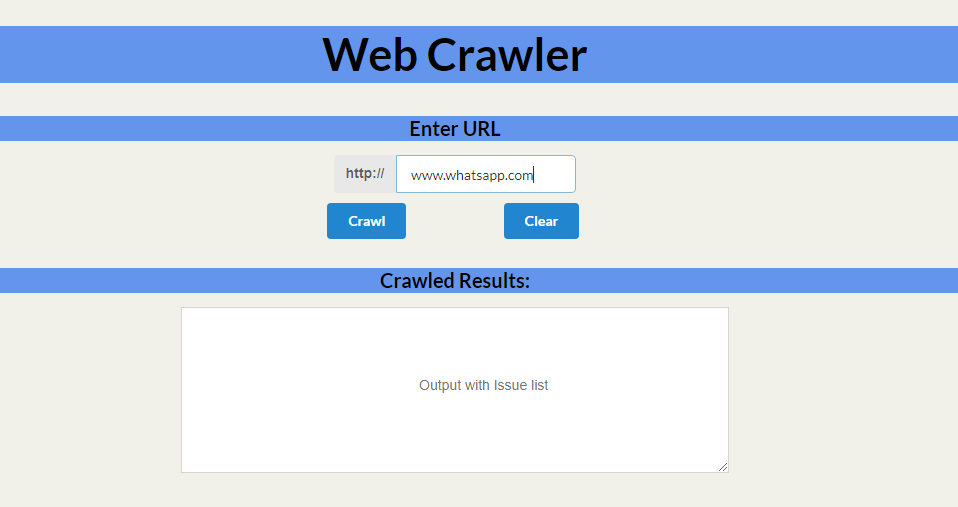
## **4.4.2 App Run Commands**

* python manage.py run server
* Put this URL <http://127.0.0.1:8000/> on webserver and wait for the app to run

# **5. Output**



Enter url then press crawl button



After processing it will show results



By pressing Clear button, everything will be wipe out.



# **6. Acknowledgement**

We would like to thanks Mam Atka Ali for helping us in the completion of this project.