Web Crawler

Ankit Nimje

Surbhi Kanthed

PROJECT MANUAL

This project is designed with data processing in python and front end using HTML, Bootstrap. We tried our best to make it as much as interactive as possible.

SETUP REQUIREMENTS:

- Python 3 : Download from https://www.python.org/download/releases/3.0/
- Spyder : It should have environment for flask, python3

LIBRARIES REQUIRED:

- **Bs4 -- pip3 install bs4**
- **Urilib.request -- pip3 install urllib.request**
- **Request -- pip3 install request**
- **Time – pip3 install time**
- **Flask – pip3 install flask**

After importing the libraries in your local machine. You are done with the installation part .

EXCECUTION:

1. Open the environment where you can execute the python files.
2. In that environment open app.py from Final/crawl_data/app.py folder.
3. If you want to see that how crawler works. You can see files in Final/crawl_data/crawlers/ folder.
4. Run app.py.
5. You will be able to see the local server on which that service is running in command prompt.
6. Again go in the Final folder, run index.html.
7. Then give the input you want from search box. You will get the desired output in diaglox boxes.
8. If the input from search box doesn't work. Then please give input from app.py

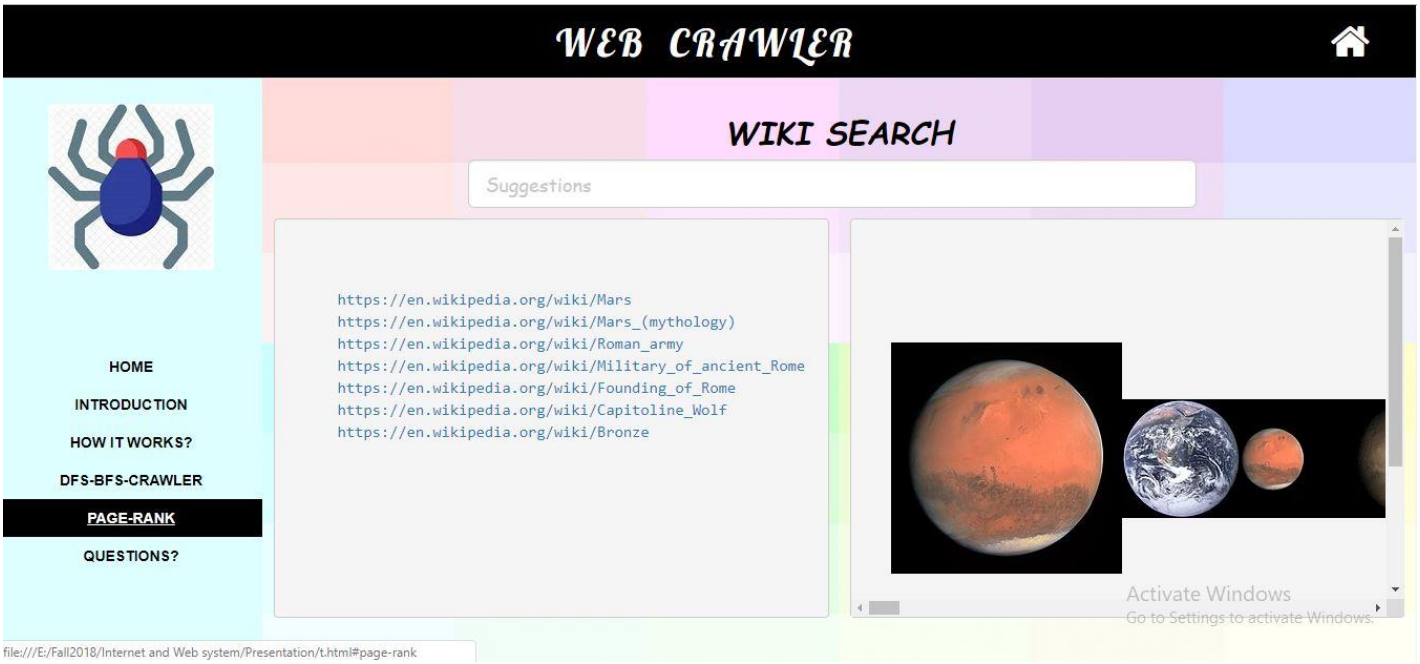
If you need any other assistance, have some queries or something is not working. Please let us know.

Here are the few screenshots

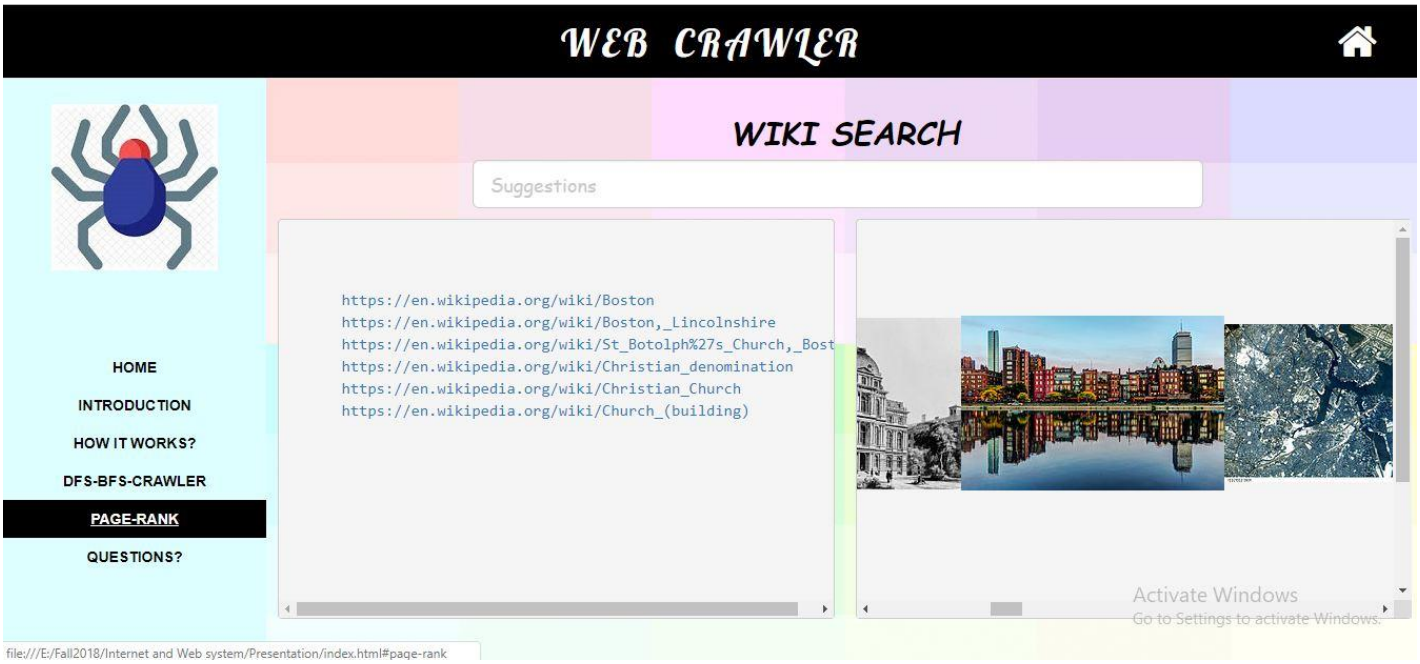
First glace of webpage:

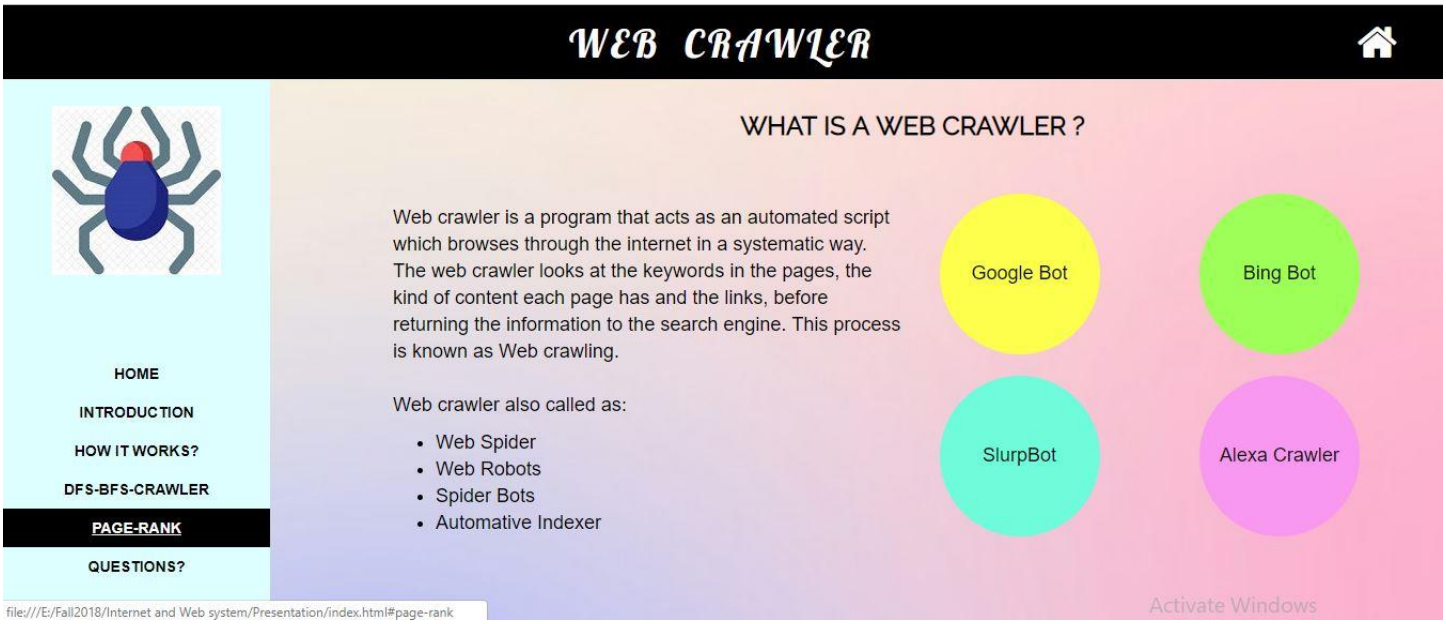
After you search anything in one dialog box it will give the set of links and In other box it will give the set of images related to search.

After search input as MARS



After search input as BOSTON:

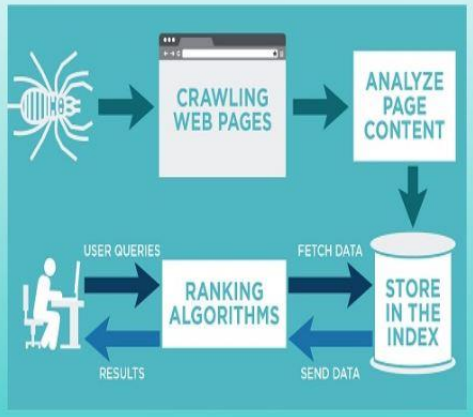




WEB CRAWLER

HOME
INTRODUCTION
HOW IT WORKS?
DFS-BFS-CRAWLER
PAGE-RANK
QUESTIONS?

How does Web Crawler Works ?



A Web-Crawler is an automated script which means all of its actions are predefined.

1. A Crawler first begins with an initial list of URLs to visit, these URLs are called seeds.
2. Then it identifies all the hyperlinks to other pages that are listed on the initial seed page and adds them to frontier.
3. The web crawler then saves these web pages in form of HTML documents which are later worked upon by the search engine and an index is created.

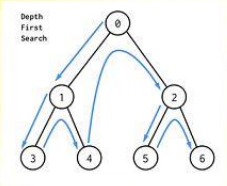
WEB CRAWLER

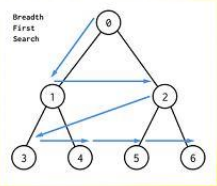
HOME
INTRODUCTION
HOW IT WORKS?
DFS-BFS-CRAWLER
PAGE-RANK
QUESTIONS?

DFS AND BFS CRAWLER

- Implemented with STACK (LIFO)
- Wander away ("lost in cyberspace")
- DFS is considered as a good algorithm if the graph is dense, but not sure when to stop.
- Use MAX LINK DEPTH,Record urls that you have crawled and omit a new request, if the url has been crawled.

- Implemented with QUEUE (FIFO)
- Finds pages along shortest paths
- If we start with "good" pages, this keeps us close; maybe other good stuff…
- BFS is good for graphs but it's too slow on dense graph and consumes hell lot of memory.





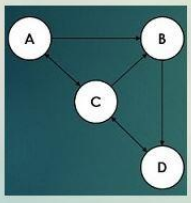
WEB CRAWLER

HOME
INTRODUCTION
HOW IT WORKS?
DFS-BFS-CRAWLER
PAGE-RANK
QUESTIONS?

PAGE RANK ALGORITHM

The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

HOW PAGE RANK IS CALCULATED



	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/4	1/12	1.5/12	1
B	1/4	2.5/12	2/12	2
C	1/4	4.5/12	4.5/12	4
D	1/4	4/12	4/12	3