PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

Name: NITHIN

RAJULAPATI(Reg.no.

39110831)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGSCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

NOVEMBER - 2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119
www.sathyabamauniversity.ac.in

DEPARTEMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **RAJULAPATI NITHIN (39110831)** who carried out the project entitled "**SCRAPPING THE WEBSITE USING PYTHON"** under my supervision from February 2022 to April 2022

INTERNAL GUIDE

Name: Dr.Jemshia Miriam

HEAD OF DEPARTMENT

Dr. L. LAKSHMANAN M.E., Ph.D.,
Dr.S.VIGNESHWARI M.E., Ph.D.,

Submitted for Viva Voice Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

I RAJULAPATI NITHIN hereby declare that the Project Report entitled **SCRAPPING THE WEBSITE USING PYTHON** done by me under the guidance of **Dr.JEMSHIA MIRIAM** and **Dr. L. LAKSHMANAN M.E., Ph.D., and Dr.S.VIGNESHWARI, M.E.,Ph.D.,**at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Jemshia Miriam** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

ABSTRACT

As the Internet and World Wide Web continue to expand and amass more users, increased rates of crime occur. One such crime is modern slavery, or human trafficking. Social media and web forums are often employed by traffickers to recruit and advertise victims anonymously.

While this issue continues to propagate through both the surface web and the dark web, web scraping tools must be developed to extract and analyse the information on these websites to identify traffickers and victims of human trafficking.

This project aims to extract the data from the selected website and arrange the extracted data into a organised manner like .csv format and make able to understand the data.

The proposed system for this project is a web scraper that is able to access and extract data from websites using a web application as an interface for user interaction. The extracted data isthen stored in a database, as the web application allows the user to search through and query the saved findings. When the system has been fully implemented, a reflection on the completedsystem takes place, judging to see if a web scraper can successfully be implementedto combatthe issue of human trafficking.

TABLE OF CONTENTS:

INTRODUCTION

1. Introduction to the WEB SCRAPING:

Web scraping, **web harvesting**, or **web data extraction** is data scraping used for extracting data from websites. The web scraping software may directly access the world wide web using the Hyper text transfer protocol or a web browser. While web scraping can be done manually bya software user, the term typically refers to automated processes implemented using a bot orweb crawler. It is a form of copying in which specific data is gathered and copied from the web,typically into a central local database or spreadsheet, for later retrieval.

- Web scraping a web page involves fetching it and extracting from it. Fetching is the downloading of a page (which a browser does when a user views a page). Therefore, webcrawling is a main component of web scraping, to fetch pages for later processing. Once fetched, then extraction can take place. The content of a page may be parsed, searched, reformatted, its data copied into a spreadsheet or loaded into a database. Web scrapers typically take something out of a page, to make use of it for another purpose somewhere else.An example would be to find and copy names and telephone numbers, or companiesand their URLs, or e-mail addresses to a list (contact scraping).
- Web pages are built using text-based mark-up languages (HTML and XHTML), andfrequently contain a wealth of useful data in text form. However, most web pages aredesigned for human end-users and not for ease of automated use. As a result, specialized tools and software have been developed to facilitate the scraping of webpages.
- Newer forms of web scraping involve monitoring data feeds from web servers.For example, JSON is commonly used as a transport storagemechanism between the client and the web server.
- There are methods that some websites use to prevent web scraping, such asdetecting and disallowing bots from crawling (viewing) their pages. In response,there are web scraping systems that rely on using techniques in DOM

2. Need for WEB SCRAPING:

Yes, people who need to know when content on a webpage changes. Here's some examples of information that I've personally built tools to scrape (on behalf of clients):

- Names/job titles off of social media to track how/when people change roles.
- Names/locations/contact information off of websites to find future clients.
- Mentions in governmental databases to find products that haven't beenpubliclyannounced yet.
- Social media posts that mention certain companies or products (to seewhatothers are saying about the client).
- Social media posts that mention certain companies or products (to attempttopoach customers from competitors).
- Auction/store websites to find products before anyone else can.

Almost any time people need to be notified of a change, scraping is a cheap way of doing it. As always though, don't let a possible solution dictate your problem and usetheright tool for the job.

3. INTRODUCTION TO PYTHON

Python is a high level, dynamic programming language. Python3.4 version was used as it is a mature, versatile and robust programming language. It is an interpreted language which makes the testing and debugging extremely quickly as there is no compilation step. There are extensive open-source libraries available for this version of python and a large community of users. Python is simple yet powerful, interpreted and dynamic programming language, which is well known for its functionality of processing natural language data, i.e., spoken English using NLTK. Other high level programming languages such as —R and —Matlab were considered because they have many benefits such as ease of use but they do not offer the same flexibility and freedom that Python can deliver.

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible

tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation. It had very less contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data—load, prepare, manipulate, model and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas:

- Fast and efficient Data Frame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favourite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

These instructions illustrate all major features of Beautiful Soup 4, with examples.Ishow you what the library is good for, how it works, how to use it, how to make itdowhat you want, and what to do when it violates your expectations.

This document covers Beautiful Soup version 4.11.0. The examples in thisdocumentation were written for Python 3.8.

You might be looking for the documentation for Beautiful Soup 3. If so, you should know that Beautiful Soup 3 is no longer being developed and that all support for it wasdropped on December 31, 2020. If you want to learn about the differences between Beautiful Soup 3 and Beautiful Soup 4.

AIM AND SCOPE OF THE PRESENT INVESTIGATION

AIM: To extract the data from the website and visualise the data in the understandable format.

SCOPE: Nowadays tonnes and tonnes of data generating every day through websites, applications and browsers. But processing and cleansing of these data is not so easy as generating data. So handling these data is difficult. This article will let you know the scraping data in websites, scopes, benefits and also drawbacks. Everything in this world has two sides good and bad. The betterment here is most of each sides matters.

In this project we are going to scrap/extract the data from the well known website(github.com). We are going to extract only the data related to the beautiful soup in GitHub and also therequired repositories and all the links available in the website for our outcome. Then wearegoing to take that bulk data and arrange the data in the Organised manner like the .CSV formatand make it easy to read for all.

EXPERIMENTAL OR REQUIREMENTS AND METHODS

3. SYSTEM

SPECIFICATIONHardwar

eRequirements:

1. Processor – Intel Core processors or any AMD chips or Mac M1.
2. RAM – 4 GB
3. Hard Disk – 40GB
4. Mouse – Standard Mouse
5. Keyboard – Standard Keyboard
6. Processor Speed – 2.4GHZ

Display Mode:

1. Color Quality – Highest[32bit]
2. Screen Resolution – 1024 by 768Pixels

Software requirements:

1. Jupyiter notebook (anaconda 3) or Google collab.
2. Python 3 or latest.

1. PROJECT DESCRIPTION:

We are going to analyse the data in the website and extract the data which is in the HTML, CSS and JSON format. Collect all the links and addresses which are present in the page in abulk form and arrange them in the understandable form such as .CSV form.

Here we are going to use the most known modules in the Python and also very much useful in cyber technologies.

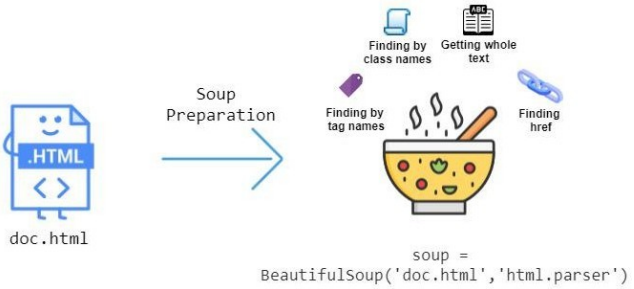
Let us now see the modules

2. MODULES USED:

1. Requests

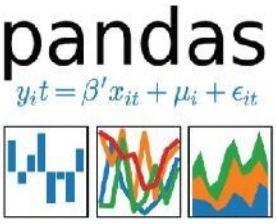
2. Beautiful Soup

3. Pandas



* Beautiful soup

*** Requests**



	BandName	WavelengthMax	WavelengthMin
0	CoastalAerosol	450	430
1	Blue	510	450
2	Green	590	530
3	Red	670	640
4	NearInfrared	880	850
5	ShortWaveInfrared_1	1650	1570
6	ShortWaveInfrared_2	2290	2110
7	Cirrus	1380	1360

APPENDI

XCODING:

```
# In[3]:

import

requests#

In[4]:

topics_url = "https://github.com/topics/beautiful-

soup"# In[5]:

response =

requests.get(topics_url)# In[6]:

type(response

)# In[7]:

response.status_cod

e# In[8]:
```

```
page_contents =

response.text# In[9]:

len(page_contents

)# In[10]:

page_contents[:1000

]# In[11]:

with open('beautiful-soup-topics.html', 'w', encoding="utf-8")
    asfile:file.write(page_contents)

# In[12]:

import

jovian#

In[13]:

jovian.commit(project='python-web-scraping-and-rest-

api')# In[14]:

jovian.commit()

# In[15]:
```

```
from bs4 import

BeautifulSoup# In[16]:

get_ipython().run_line_magic('pinfo',

'BeautifulSoup')# In[17]:

with open('beautiful-soup-topics.html', 'r')
    asf:html_source = f.read()

# In[18]:

html_source[:1000

]# In[19]:

doc = BeautifulSoup(html_source,

'html.parser')# In[20]:

type(doc

)# In[21]:

title_tag = doc.title
```

```
# In[22]:

title_ta

g#

In[23]:

type(title_tag

)# In[24]:

title_tag.nam

e# In[25]:

type(title_tag

)# In[26]:

title_tag.nam

e# In[27]:

title_tag.tex

t# In[28]:

first_link = doc.a
```

```
# In[29]:

first_lin

k#

In[30]:

first_link.tex

t# In[31]:

all_link_tags =

doc.find_all('a')# In[32]:

len(all_link_tags

)# In[33]:

all_link_tags[:3

]# In[34]:

first_lin

k#
```



```
In[35]:
```

first_link['href'

]# In[36]:

first_link['class'

]# In[37]:

first_link.attr

s# In[38]:

doc.find_all('img', { 'alt':

'trivialfis'})# In[39]:

matching_tags = doc.find_all(class_='HeaderMenu-

link')# In[40]:

matching_tag

s# In[41]:

header_link_tags = doc.find_all('a', class_='HeaderMenu-link')

In[42]:

header_link_tag

s# In[43]:

header_link_tags[0]

['href']# In[44]:

```
header_links = []
base_url = 'https://github.com'

for tag in header_link_tags:
    header_links.append({ 'title': tag.text.strip(), 'url': base_url + tag['href']})

header_links
```

In[45]:

```
sample_html = """
<html>
    <body>
        <ul class="top-list">
            <li>Item 1</li>
            <li>Item 2</li>
            <li>
                <ul>
                    <li>Item 3.1</li>
                    <li>Item 3.2</li>
                    <li>Item 3.3</li>
                </ul>
            </li>
        </ul>
    </body>
</html>"""
```

```
# In[46]:

sample_doc =

BeautifulSoup(sample_html)# In[47]:

list_tag = sample_doc.find('ul', class_='top-

list')# In[48]:

list_item_tags = list_tag.find_all('li',

recursive=False)# In[49]:

list_item_tag

s# In[50]:

list_tag.find_all('li'

)# In[51]:

jovian.commit(

)# In[52]:
```

```
def
    get_topic_page(topic):#
    Construct the URL
    topic_repos_url = 'https://github.com/topics/' + topic

    # Get the HTML page content using requests
    response = requests.get(topic_repos_url)

    # Ensure that the reponse is
    validif response.status_code !
    =200:
        print('Status code:', response.status_code)
        raise Exception('Failed to fetch web page ' + topic_repos_url)

    # Construct a beautiful soup
    documentdoc =
    BeautifulSoup(response.text)

    return


doc# In[53]:


doc = get_topic_page('beautiful-


soup')# In[54]:


doc.title.tex


t# In[55]:


doc2 = get_topic_page('data-analysis')


# In[56]:


doc2.title.text
```

In[57]:

article_tags = doc.find_all('article', class_='border rounded color-shadow-smallcolor-bg-subtle my-4')

In[58]:

len(article_tags

)# In[59]:

article_tag =

article_tags[4]# In[60]:

h3_tag =
article_tag.find('h3')h3_tag

In[61]:

a_tags = h3_tag.find_all('a',

recursive=False)# In[62]:

username =
a_tags[0].textusername

In[63]:

```
username =
a_tags[0].text.strip()username


# In[64]:


repo_name =
a_tags[1].text.strip()repo_name


# In[65]:


repo_path = a_tags[1]
['href'].strip()repo_path


# In[66]:


base_url = 'https://
github.com'repo_url =
base_url + repo_pathrepo_url


# In[67]:


article_tags[4


]# In[68]:


a_star_tag = article_tags[4].find('span', class_='Counter js-social-count')


# In[69]:
```

```
a_star_ta


g# In[70]:


a_star_tag.text.strip(


)# In[71]:


def
    parse_star_count(stars_str):
    stars_str =
    stars_str.strip()if
    stars_str[-1] == 'k':
        return int(float(stars_str[:-1])
    *1000)else:
        return


int(stars_str)# In[72]:


parse_star_count('29'


)# In[73]:


parse_star_count('999'


)# In[74]:


star_count = parse_star_count(a_star_tag.text.strip())


# In[75]:
```

```
star_coun

t# In[76]:

print('Repository name:',
repo_name)print("Owner's
username:", username)print('Stars:',
star_count) print('Repository URL:',
repo_url)

# In[77]:

def parse_repository(article_tag):

    # <a> tags containing username, repository name and URL

    a_tags = article_tag.h3.find_all('a')

    # Owner's username

    username =

    a_tags[0].text.strip()#

    Repository name

    repo_name =

    a_tags[1].text.strip()# Repository
```

URL

```
    repo_url = base_url + a_tags[1]['href'].strip()


    # Star count


    stars_tag = article_tag.find('span', class_='Counter js-
    social-count')star_count =
    parse_star_count(stars_tag.text.strip())

    # Return a

    dictionaryreturn {

        'repository_name':
        repo_name,'owner_username'
        :username, 'stars': star_count,
        'repository_url': repo_url
    }


# In[78]:


parse_repository(article_tags[0]


)# In[79]:


parse_repository(article_tags[10]


)# In[80]:


top_repositories = [parse_repository(tag) for tag in


article_tags]# In[81]:
```

```
len(top_repositories

)# In[82]:


top_repositories[:5

]# In[83]:


def get_top_repositories(doc):
    article_tags = doc.find_all('article', class_='border rounded color-
shadow-smallcolor-bg-subtle my-4')
    topic_repos = [parse_repository(tag) for tag
    inarticle_tags]return topic_repos


# In[85]:


topic_page_BS = get_topic_page('beautiful-
soup')top_repos_BS =
get_top_repositories(topic_page_BS)
top_repos_BS[:5]


# In[86]:


topic_page_da = get_topic_page('data-
analysis')top_repos_da =
get_top_repositories(topic_page_da)
top_repos_da[:5]


# In[87]:


get_top_repositories(get_topic_page('python'))[:5]
```

In[88]:

jovian.commit(

)# In[ ]:

In[89]:

WRITING INFORAMATION INTO A CSV FORMAT#

In[ ]:

In[90]:

```
def write_csv(items,
    path):# Open the file in
    write
    modewith open(path, 'w')
    asf:
        # Return if there's nothing
        towriteif len(items) == 0:
            return

        # Write the headers in the first
        lineheaders =
        list(items[0].keys())
        f.write(','.join(headers) + '\n')

        # Write one item per
        linefor item in items:
```

```
            values = []
            for header in headers:
                values.append(str(item.get(header, "")))
            f.write(','.join(values) +


"\n")# In[91]:


len(top_repos_BS


)# In[92]:


top_repos_BS[:3


]# In[93]:


write_csv(top_repositories, 'beautiful-


soup.csv')# In[94]:


with open('beautiful-soup.csv', 'r')
    asf:print(f.read())


# In[95]:


import
requestsfrom
bs4 import
BeautifulSoupbase_url
='https://gitub.com'

def scrape_topic_repositories(topic, path=None):
    """Get the top repositories for a topic and write them to a CSV
    file. If path is None.
    linefor item in items:
```

```
        path = topic + '.csv'
    topic_page_doc =
    get_topic_page(topic)topic_repositories
    = get_top_repositories(topic_page_doc)
    write_csv(topic_repositories, path)
    print('Top repositories for topic "{}" written to file
    "{}"'.format(topic,path))return path

def get_top_repositories(doc):
    """Parse the top repositories for a topic given a Beautiful Soup document"""
    article_tags = doc.find_all('article', class_='border rounded color-shadow-small
color-bg-subtle my-4')
    topic_repos = [parse_repository(tag) for tag
    inarticle_tags]return topic_repos

def get_topic_page(topic):
    """Get the web page containing the top repositories for a topic as
aBeautifulSoup document"""
    topic_repos_url = 'https://github.com/topics/'
    + topicresponse =
    requests.get(topic_repos_url) if
    response.status_code != 200:
        print('Status code:', response.status_code)
        raise Exception('Failed to fetch web page ' +
    topic_repos_url)return BeautifulSoup(response.text)

def parse_repository(article_tag):
    """Parse information about a repository from an
    <article>tag"""a_tags = article_tag.h3.find_all('a')
    username = a_tags[0].text.strip()
    repo_name = a_tags[1].text.strip()
    repo_url = base_url + a_tags[1]['href'].strip()
    stars_tag = article_tag.find('span', class_='Counter js-
    social-count')star_count =
    parse_star_count(stars_tag.text.strip())
    return {'repository_name': repo_name, 'owner_username':
username,'stars':star_count, 'repository_url': repo_url}

def parse_star_count(stars_str):
    """Parse strings like 40.3k and get the no. of stars as
    anumber"""stars_str = stars_str.strip()
    return int(float(stars_str[:-1]) * 1000) if stars_str[-1] == 'k' else int(stars_str)
        linefor item in items:
```



```
def write_csv(items, path):
```

linefor item in items:

```
    """Write a list of dictionaries to a CSV
    file"""with open(path, 'w') as f:
        if len(items)
            ==0:return
        headers =
        list(items[0].keys())
        f.write(','.join(headers) +
        '\n')for item in items:
            values = []
            for header in headers:
                values.append(str(item.get(header, "")))
            f.write(','.join(values) +


"\n")# In[96]:


scrape_topic_repositories('beautiful-


soup')# In[97]:


import pandas as


pd# In[98]:


pd.read_csv('beautiful-


soup.csv')# In[99]:


scrape_topic_repositories('data-


analysis')# In[100]:


pd.read_csv('data-analysis.csv')
        linefor item in items:
```

```
# In[101]:

scrape_topic_repositories('python'

)# In[102]:

pd.read_csv('python.csv'

)# In[103]:

jovian.commit(files=['beautiful-soup.csv', 'python.csv', 'data-

analysis.csv'])# In[ ]:



# In[104]:

response = requests.get('https://api.github.com/repos/octocat/hello-

world')# In[105]:

import json

data_dict = json.loads(response.text)


# In[110]:
```

```
data_dic


t#


In[107]:


def get_repo_details(username, repo_name):
    print('Fetching information for {}/{}'.format(username, repo_name))
    repo_details_url = "https://api.github.com/repos/" + username + "/" +
repo_name
    response =
    requests.get(repo_details_url)if
    notresponse.ok:
        print("Failed
        tofetch!")return
        {}
    repo_data =
    json.loads(response.text)return {
        'description': repo_data['description'],
        'watchers': repo_data['watchers_count'],
        'open_issues':
        repo_data['open_issues_count'],'created_at'
        :repo_data['created_at'], 'updated_at':
        repo_data['updated_at']
    }


# In[108]:


get_repo_details('octocat', 'hello-


world')# In[109]:


# In[110]:
get_repo_details('tensorflow', 'tensorflow')
```

```
def
    add_repo_details(repos):
    return
[dict(**get_repo_details(repo['owner_username'],
repo['repository_name']), **repo) for repo in repos]


# In[111]:


add_repo_details(top_repositories[:5]


)# In[112]:


from getpass import

getpasstoken = getpass()


# In[113]:


jovian.commit(


)# In[114]:


first_div=doc.find_all('div'

)first_div[0


]# In[115]:


first_img=doc.find('img'

)first_img
```

```
# In[116]:


first_span=doc.sp

anfirst_span


# In[117]:


first_p=doc.find_all('p'
)first_p[0


]#


In[118]:


all_images=doc.find_all('img'
)len(all_images


)# In[119]:


fifth_image=all_images[5
]fifth_imag


e# In[120]:


fifth_image['src'


]# In[121]:


fifth_image['alt']
```

```
# In[123]:


topics=['data-analysis','python','deep-


learning']# In[124]:


def
    scrape_topics(topics):
    for topic in topics:
        scrape_topic_repositories(topic)


# In[125]:


scrape_topics(topics


)# In[126]:


pd.read_csv('data-


analysis.csv')# In[127]:


pd.read_csv('python.csv'


)# In[128]:


pd.read_csv('deep-learning.csv')
```

OUTPUT IN BULK FORM


```
<li class="d-inline-flex" data-view-component="true">
<a aria-current="true" class="tabnav-tab f6 px-2 py-1" data-ga-click="Explore, go to repository, location:explore feed" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"REPOSITORY","click_visual_representation":"CODE_TAB","actor_id":null,"record_id":22544574,"originating_url"
:"https://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="176af4525587b219facc73988f290e5a66954930c41cf4316a95b2cdecc00693" data-turbo="false
" data-view-component="true" href="/SylvainDe/ComicBookMaker" id="code-tab-22544574">
<svg aria-hidden="true" class="octicon octicon-code color-fg-default" data-view-component="true" height="16" version="1.1" viewbox="0 0 16 16" width="16">
<path d="M4.72 3.22a.75.75 0 011.06 1.06L2.06 8l3.72 3.72a.75.75 0 11-1.06 1.06L.47 8.53a.75.75 0 010-1.06l4.25-4.25zm6.56 0a.75.75 0 10-1.06 1.06L13.94 8l-3.72 3.72a.7
5.75 0 101.06 1.06l4.25-4.25a.75.75 0 000-1.06l-4.25-4.25z" fill-rule="evenodd"></path>
</svg> Code


</a></li>
<li class="d-inline-flex" data-view-component="true">
<a class="tabnav-tab f6 px-2 py-1" data-ga-click="Explore, go to repository issues, location:explore feed" data-hydro-click='{"event_type":"explore.click","payload":{"c
lick_context":"REPOSITORY_CARD","click_target":"REPOSITORY_ISSUES","click_visual_representation":"ISSUES_TAB","actor_id":null,"record_id":22544574,"originating_url":"ht
tps://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="3842961125e3f9df9937cba092c8981bd00f9647be9f21bd4d7d4a34ba48a4a6" data-turbo="false" da
ta-view-component="true" href="/SylvainDe/ComicBookMaker/issues" id="issues-tab-22544574">
<svg aria-hidden="true" class="octicon octicon-issue-opened color-gray-light" height="16" version="1.1" viewbox="0 0 16 16" width="16"><path d="M8 9.5a1.5 1.5 0 100-3 1
.5 1.5 0 000 3z"></path><path d="M8 0a8 8 0 100 16A8 8 0 008 0zM1.5 8a6.5 6.5 0 1113 0 6.5 6.5 0 01-13 0z" fill-rule="evenodd"></path></svg> Issues


</a></li>
<li class="d-inline-flex" data-view-component="true">
<a class="tabnav-tab f6 px-2 py-1" data-ga-click="Explore, go to repository pulls, location:explore feed" data-hydro-click='{"event_type":"explore.click","payload":{"cl
ick_context":"REPOSITORY_CARD","click_target":"REPOSITORY_PULL_REQUESTS","click_visual_representation":"PULL_REQUESTS_TAB","actor_id":null,"record_id":22544574,"origina
ting_url":"https://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="e6ff71f42bfe1b6ba610c301301c729559a1de7ff3a04975faa37774cdb16f07" data-tur
bo="false" data-view-component="true" href="/SylvainDe/ComicBookMaker/pulls" id="pull-requests-tab-22544574">
<svg aria-hidden="true" class="octicon octicon-git-pull-request color-fg-muted" data-view-component="true" height="16" version="1.1" viewbox="0 0 16 16" width="16">
<path d="M7.177 3.073L9.573.677A.25.25 0 0110 .854v4.792a.25.25 0 01-.427.177L7.177 3.427a.25.25 0 010-.354zM3.75 2.5a.75.75 0 100 1.5.75.75 0 000-1.5zm-2.25.75a2.25 2.
25 0 113 2.122v5.256a2.251 2.251 0 11-1.5 0V5.372A2.25 2.25 0 011.5 3.25zM11 2.5h-1V4h1a1 1 0 011 1v5.628a2.251 2.251 0 101.5 0V5A2.5 2.5 0 0011 2.5zm1 10.25a.75.75 0 1
11.5 0 .75.75 0 01-1.5 0zM3.75 12a.75.75 0 100 1.5.75.75 0 000-1.5z" fill-rule="evenodd"></path>
</svg> Pull requests


</a></li>
</ul>
</nav>
<div class="color-bg-default rounded-bottom-2">
<div class="px-3 pt-3">
<p class="color-fg-muted mb-0"></p><div>Script to fetch webcomics and use them to create ebooks.</div>
```

```
</a></li>
</ul>
</nav>
<div class="color-bg-default rounded-bottom-2">
<div class="px-3 pt-3">
<p class="color-fg-muted mb-0"></p><div>Script to fetch webcomics and use them to create ebooks.</div>
</div>
<div class="d-flex flex-wrap border-bottom color-border-muted px-3 pt-2 pb-2">
<a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click","payl
oad":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":84,"originating_url":"https://githu
b.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="1dba9e20772c5571a51af4e25ab2accc078a2e280e5fc152bd9f5f29f34a2fe6" data-view-component="true" href=
"/topics/python" title="Topic: python">

              python


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":13190,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="a916e07f6fe4194b0ealde19dbad0240570e17112302764ac93ad5675b5230d9" data-view-component="tru
e" href="/topics/tumblr" title="Topic: tumblr">

              tumblr


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":14109,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="f870893b36e669d743e04242f026ead4bfff8e8c9927e189ff241a45a109c8ba" data-view-component="tru
e" href="/topics/web-crawler" title="Topic: web-crawler">

              web-crawler


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":17249,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="fe9c932859a5333aca81081764f253dd6fc4f1aca2f31de4c7fd4e5ff483a906" data-view-component="tru
e" href="/topics/comic" title="Topic: comic">

              comic


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":17260,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="60e0cb6ed39ea699a81f89b6e71f9f6176ef15f5a8d9b49cd7a58a2e99f1b1d0" data-view-component="tru
e" href="/topics/webcomic" title="Topic: webcomic">

              webcomic


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":20655,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="bc7473159405151432f4eec13d62cdd9ff2437ad86089d08ebd89073b9208267" data-view-component="tru
e" href="/topics/ebook" title="Topic: ebook">

              ebook


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":22377,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="c73caa9879667f3aaa999f80f5435075d123802c5d8e7caed1564a566f453e34" data-view-component="tru
e" href="/topics/beautiful-soup" title="Topic: beautiful-soup">

              beautiful-soup


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":23016,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="b4d29bd686783e969e56c97b95273003e19db79f62590cd7f30dd30e17207f77" data-view-component="tru
e" href="/topics/comics" title="Topic: comics">

              comics


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":23680,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="02c43c2349b0944ad8769c18227e79d27ea00ede8beb2e58d74d85a567cba13e" data-view-component="tru
e" href="/topics/mobi" title="Topic: mobi">

              mobi


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":45206,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="41a5e0343ba045aa864548e3e7d9d55e4da514fa827ebb04cdb9853409caf827" data-view-component="tru
e" href="/topics/comic-downloader" title="Topic: comic-downloader">

              comic-downloader


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":45540,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="cbf740b6cdfe8fb0eccc138b119d5dbbaa1f21fa18b0f994100d0d89dcec671b" data-view-component="tru
e" href="/topics/download-comic" title="Topic: download-comic">

              download-comic


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":59963,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="dae1a0159e4f5ca461252214b2ac37dadff3b7c54d59170230908 4d4af15bcc" data-view-component="tru
e" href="/topics/kindle" title="Topic: kindle">

              kindle


</a> <a class="topic-tag topic-tag-link f6 mb-2" data-ga-click="Explore, go to topic, location:explore feed repository" data-hydro-click='{"event_type":"explore.click",
"payload":{"click_context":"REPOSITORY_CARD","click_target":"TOPIC","click_visual_representation":"TOPIC_TAG","actor_id":null,"record_id":74654,"originating_url":"https
://github.com/topics/beautiful-soup","user_id":null}}' data-hydro-click-hmac="820511a0706fde59bb17e468193384532 61a3e1bdfc8519c3de5cb6513beafd9" data-view-component="tru
```

In [97]:
```
import pandas as pd
```

In [98]:
```
pd.read_csv('beautiful-soup.csv')
```

Out[98]:

	repository_name	owner_username	stars	repository_url
0	short-jokes-dataset	amoudgl	229	https://gitub.com/amoudgl/short-jokes-dataset
1	trump-lies	justmarkham	219	https://gitub.com/justmarkham/trump-lies
2	Novel-crawler	ling7334	69	https://gitub.com/ling7334/Novel-crawler
3	RapLyrics-Scraper	fpaupier	29	https://gitub.com/fpaupier/RapLyrics-Scraper
4	ComicBookMaker	SylvainDe	29	https://gitub.com/SylvainDe/ComicBookMaker
5	webscraping	mahmudahsan	23	https://gitub.com/mahmudahsan/webscraping
6	kick-off-web-scraping-python-selenium-beautifu...	devrohaan	18	https://gitub.com/devrohaan/kick-off-web-scrap...
7	tagalog-dictionary-scraper	raymelon	13	https://gitub.com/raymelon/tagalog-dictionary-...
8	100-Days-of-Python	LearnEarn-Fun	8	https://gitub.com/LearnEarn-Fun/100-Days-of-Py...
9	Scraping-Dynamic-JavaScript-Ajax-Websites-With...	oxylabs	8	https://gitub.com/oxylabs/Scraping-Dynamic-Jav...
10	web-scraper	MLArtist	7	https://gitub.com/MLArtist/web-scraper
11	dataset-indian-companies	mratanusarkar	6	https://gitub.com/mratanusarkar/dataset-indian...
12	MoodleTracker	sahilbansal17	5	https://gitub.com/sahilbansal17/MoodleTracker
13	impress-writer	CSEC-NITH	4	https://gitub.com/CSEC-NITH/impress-writer
14	vlr.gg-scraper	aritropaul	4	https://gitub.com/aritropaul/vlr.gg-scraper
15	LinkCheck	aadibajpai	3	https://gitub.com/aadibajpai/LinkCheck
16	Daily-Prayer-Time-API	abdulrcs	3	https://gitub.com/abdulrcs/Daily-Prayer-Time-API
17	lyrics_classification	lenaromanenko	2	https://gitub.com/lenaromanenko/lyrics_classif...
18	GuitarTabs	SylvainDe	2	https://gitub.com/SylvainDe/GuitarTabs
19	Scrapping-drugs-dot-com	m0-k1	2	https://gitub.com/m0-k1/Scrapping-drugs-dot-com
20	WebScraperAllMusic	AntoData	2	https://gitub.com/AntoData/WebScraperAllMusic
21	python-scraping	zhubaiyuan	2	https://gitub.com/zhubaiyuan/python-scraping
22	Web-Scrapping-using-BeautifulSoup2	ankitnayan12	2	https://gitub.com/ankitnayan12/Web-Scrapping-u...
23	AnimalCrossing_PopularityData	ErikaJacobs	2	https://gitub.com/ErikaJacobs/AnimalCrossing_P...
24	xkcd-scraper	omermikhailk	2	https://gitub.com/omermikhailk/xkcd-scraper

OUTPUT IN THE .CSV FORMAT

In [99]:
```
scrape_topic_repositories('data-analysis')
```

```
Top repositories for topic "data-analysis" written to file "data-analysis.csv"
```

Out[99]:
```
'data-analysis.csv'
```

In [100...
```
pd.read_csv('data-analysis.csv')
```

Out[100...

	repository_name	owner_username	stars	repository_url
0	scikit-learn	scikit-learn	49600	https://gitub.com/scikit-learn/scikit-learn
1	superset	apache	45500	https://gitub.com/apache/superset
2	pandas	pandas-dev	33300	https://gitub.com/pandas-dev/pandas
3	metabase	metabase	28000	https://gitub.com/metabase/metabase
4	AI-Expert-Roadmap	AMAI-GmbH	18900	https://gitub.com/AMAI-GmbH/AI-Expert-Roadmap
5	streamlit	streamlit	18500	https://gitub.com/streamlit/streamlit
6	CyberChef	gchq	16000	https://gitub.com/gchq/CyberChef
7	goaccess	allinurl	14600	https://gitub.com/allinurl/goaccess
8	Data-Science-For-Beginners	microsoft	10000	https://gitub.com/microsoft/Data-Science-For-B...
9	pandas-profiling	ydataai	8800	https://gitub.com/ydataai/pandas-profiling
10	OpenRefine	OpenRefine	8700	https://gitub.com/OpenRefine/OpenRefine
11	mlcourse.ai	Yorko	8200	https://gitub.com/Yorko/mlcourse.ai
12	gop	goplus	8000	https://gitub.com/goplus/gop
13	pandas_exercises	guipsamora	7600	https://gitub.com/guipsamora/pandas_exercises
14	statsmodels	statsmodels	7300	https://gitub.com/statsmodels/statsmodels
15	airbyte	airbytehq	6300	https://gitub.com/airbytehq/airbyte
16	imbalanced-learn	scikit-learn-contrib	5800	https://gitub.com/scikit-learn-contrib/imbalan...
17	gonum	gonum	5700	https://gitub.com/gonum/gonum
18	gradio	gradio-app	5700	https://gitub.com/gradio-app/gradio
19	alluxio	Alluxio	5600	https://gitub.com/Alluxio/alluxio
20	dataease	dataease	5500	https://gitub.com/dataease/dataease
21	pyod	yzhao062	5400	https://gitub.com/yzhao062/pyod
22	pachyderm	pachyderm	5400	https://gitub.com/pachyderm/pachyderm

In [101…
```
scrape_topic_repositories('python')
```

```
Top repositories for topic "python" written to file "python.csv"
```

Out[101… 'python.csv'

In [102…
```
pd.read_csv('python.csv')
```

Out[102…

	repository_name	owner_username	stars	repository_url
0	system-design-primer	donnemartin	171000	https://github.com/donnemartin/system-design-pr...
1	tensorflow	tensorflow	164000	https://github.com/tensorflow/tensorflow
2	CS-Notes	CyC2018	149000	https://github.com/CyC2018/CS-Notes
3	Python	TheAlgorithms	134000	https://github.com/TheAlgorithms/Python
4	awesome-python	vinta	123000	https://github.com/vinta/awesome-python
5	free-programming-books-zh_CN	justjavac	90400	https://github.com/justjavac/free-programming-b...
6	thefuck	nvbn	69900	https://github.com/nvbn/thefuck
7	project-based-learning	practical-tutorials	65700	https://github.com/practical-tutorials/project-...
8	django	django	63300	https://github.com/django/django
9	transformers	huggingface	60600	https://github.com/huggingface/transformers
10	flask	pallets	58500	https://github.com/pallets/flask
11	pytorch	pytorch	55200	https://github.com/pytorch/pytorch
12	keras	keras-team	54900	https://github.com/keras-team/keras
13	HelloGitHub	521xueweihan	53800	https://github.com/521xueweihan/HelloGitHub
14	ansible	ansible	52700	https://github.com/ansible/ansible
15	core	home-assistant	51500	https://github.com/home-assistant/core
16	scikit-learn	scikit-learn	49600	https://github.com/scikit-learn/scikit-learn
17	leetcode	azl397985856	47400	https://github.com/azl397985856/leetcode
18	requests	psf	47200	https://github.com/psf/requests
19	superset	apache	45500	https://github.com/apache/superset
20	face_recognition	ageitgey	43800	https://github.com/ageitgey/face_recognition
21	fastapi	tiangolo	43800	https://github.com/tiangolo/fastapi
22	manim	3b1b	43700	https://github.com/3b1b/manim
23	[illegible]	[illegible]	[illegible]	[illegible]

p","user_id":null}}' data-hydro-click-hmac="6450c12cb120d835969d0d20ec4861c007b4c9180467918618345 7b2bdcaa44b" data-view-component="true" href="/login?return_to=%2FSylvainDe%2FComicBookMaker" rel="nofollow"> <svg aria-hidden="true" class="octicon octicon-star v-align-text-bottom d-inline-block mr-2" data-view-component="true" height="16" version="1.1" viewbox="0 0 16 16" width="16">
<path d="M8 .25a.75.75 0 01.673.41811.882 3.815 4.21.612a.75.75 0 01.416 1.2791-3.046 2.97.719 4.192a.75.75 0 01-1.088.791L8 12.3471-3.766 1.98a.75.75 0 01-1.088-.791.72-4.194L.818 6.374a.75.75 0 01.416-1.2814.21-.611L7.327.668A.75.75 0 018 .25zm0 2.445L6.615 5.5a.75.75 0 01-.564.411-3.097.45 2.24 2.184a.75.75 0 01.216.6641-.528 3.0842.769-1.456a.75.75 0 01.698 012.77 1.456-.53-3.084a.75.75 0 01.216-.6612.24-2.183-3.096-.45a.75.75 0 01-.564-.41L8 2.694v.001z" fill-rule="evenodd"></path>
</svg>

In [125…
```
scrape_topics(topics)
```

```
Top repositories for topic "data-analysis" written to file "data-analysis.csv"
Top repositories for topic "python" written to file "python.csv"
Top repositories for topic "deep-learning" written to file "deep-learning.csv"
```

In [126…
```
pd.read_csv('data-analysis.csv')
```

Out[126…

	repository_name	owner_username	stars	repository_url
0	scikit-learn	scikit-learn	49600	https://github.com/scikit-learn/scikit-learn
1	superset	apache	45500	https://github.com/apache/superset
2	pandas	pandas-dev	33300	https://github.com/pandas-dev/pandas
3	metabase	metabase	28000	https://github.com/metabase/metabase
4	AI-Expert-Roadmap	AMAI-GmbH	18900	https://github.com/AMAI-GmbH/AI-Expert-Roadmap
5	streamlit	streamlit	18500	https://github.com/streamlit/streamlit
6	CyberChef	gchq	16000	https://github.com/gchq/CyberChef
7	goaccess	allinurl	14600	https://github.com/allinurl/goaccess
8	Data-Science-For-Beginners	microsoft	10000	https://github.com/microsoft/Data-Science-For-B...
9	pandas-profiling	ydataai	8800	https://github.com/ydataai/pandas-profiling
10	OpenRefine	OpenRefine	8700	https://github.com/OpenRefine/OpenRefine
11	mlcourse.ai	Yorko	8200	https://github.com/Yorko/mlcourse.ai
12	gop	goplus	8000	https://github.com/goplus/gop
13	pandas_exercises	guipsamora	7600	https://github.com/guipsamora/pandas_exercises
14	statsmodels	statsmodels	7300	https://github.com/statsmodels/statsmodels
15	airbyte	airbytehq	6300	https://github.com/airbytehq/airbyte
16	imbalanced-learn	scikit-learn-contrib	5800	https://github.com/scikit-learn-contrib/imbalan...
17	gonum	gonum	5700	https://github.com/gonum/gonum
18	gradio	gradio-app	5700	https://github.com/gradio-app/gradio
19	alluxio	Alluxio	5600	https://github.com/Alluxio/alluxio
20	dataease	dataease	5500	https://github.com/dataease/dataease
21	pyod	yzhao062	5400	https://github.com/yzhao062/pyod
22	pachyderm	pachyderm	5400	https://github.com/pachyderm/pachyderm
23	Data-Analysis-and-Machine-Learning-Projects	rhiever	5200	https://github.com/rhiever/Data-Analysis-and-Ma...

In [127…
```
pd.read_csv('python.csv')
```

Out[127…

	repository_name	owner_username	stars	repository_url
0	system-design-primer	donnemartin	171000	https://github.com/donnemartin/system-design-pr...
1	tensorflow	tensorflow	164000	https://github.com/tensorflow/tensorflow
2	CS-Notes	CyC2018	149000	https://github.com/CyC2018/CS-Notes
3	Python	TheAlgorithms	134000	https://github.com/TheAlgorithms/Python
4	awesome-python	vinta	123000	https://github.com/vinta/awesome-python
5	free-programming-books-zh_CN	justjavac	90400	https://github.com/justjavac/free-programming-b...
6	thefuck	nvbn	69900	https://github.com/nvbn/thefuck
7	project-based-learning	practical-tutorials	65700	https://github.com/practical-tutorials/project-...
8	django	django	63300	https://github.com/django/django
9	transformers	huggingface	60600	https://github.com/huggingface/transformers
10	flask	pallets	58500	https://github.com/pallets/flask
11	pytorch	pytorch	55200	https://github.com/pytorch/pytorch
12	keras	keras-team	54900	https://github.com/keras-team/keras
13	HelloGitHub	521xueweihan	53800	https://github.com/521xueweihan/HelloGitHub
14	ansible	ansible	52700	https://github.com/ansible/ansible
15	core	home-assistant	51500	https://github.com/home-assistant/core
16	scikit-learn	scikit-learn	49600	https://github.com/scikit-learn/scikit-learn
17	leetcode	azl397985856	47400	https://github.com/azl397985856/leetcode
18	requests	psf	47200	https://github.com/psf/requests
19	superset	apache	45500	https://github.com/apache/superset
20	face_recognition	ageitgey	43800	https://github.com/ageitgey/face_recognition
21	fastapi	tiangolo	43800	https://github.com/tiangolo/fastapi
22	manim	3b1b	43700	https://github.com/3b1b/manim
23	scrapy	scrapy	43200	https://github.com/scrapy/scrapy
24	TensorFlow-Examples	aymericdamien	41800	https://github.com/aymericdamien/TensorFlow-Exa...

In [128…
```
pd.read_csv('deep-learning.csv')
```

Out[128…

	repository_name	owner_username	stars	repository_url
0	tensorflow	tensorflow	164000	https://github.com/tensorflow/tensorflow
1	opencv	opencv	60800	https://github.com/opencv/opencv
2	transformers	huggingface	60600	https://github.com/huggingface/transformers
3	pytorch	pytorch	55200	https://github.com/pytorch/pytorch
4	keras	keras-team	54900	https://github.com/keras-team/keras
5	TensorFlow-Examples	aymericdamien	41800	https://github.com/aymericdamien/TensorFlow-Exa...
6	faceswap	deepfakes	40700	https://github.com/deepfakes/faceswap
7	100-Days-Of-ML-Code	Avik-Jain	36600	https://github.com/Avik-Jain/100-Days-Of-ML-Code
8	Real-Time-Voice-Cloning	CorentinJ	34200	https://github.com/CorentinJ/Real-Time-Voice-Cl...
9	caffe	BVLC	32400	https://github.com/BVLC/caffe
10	DeepFaceLab	iperov	32000	https://github.com/iperov/DeepFaceLab
11	Deep-Learning-Papers-Reading-Roadmap	floodsung	32000	https://github.com/floodsung/Deep-Learning-Pape...
12	d2l-zh	d2l-ai	31300	https://github.com/d2l-ai/d2l-zh
13	MadeWithML	GokuMohandas	29900	https://github.com/GokuMohandas/MadeWithML
14	machine-learning-for-software-engineers	ZuzooVn	25800	https://github.com/ZuzooVn/machine-learning-for...
15	tesseract.js	naptha	25800	https://github.com/naptha/tesseract.js
16	yolov5	ultralytics	24500	https://github.com/ultralytics/yolov5
17	handson-ml	ageron	24200	https://github.com/ageron/handson-ml
18	openpose	CMU-Perceptual-Computing-Lab	23800	https://github.com/CMU-Perceptual-Computing-Lab...
19	awesome-deep-learning-papers	terryum	23600	https://github.com/terryum/awesome-deep-learnin...
20	pytorch-tutorial	yunjey	23400	https://github.com/yunjey/pytorch-tutorial
21	spaCy	explosion	23100	https://github.com/explosion/spaCy
22	data-science-ipython-notebooks	donnemartin	22900	https://github.com/donnemartin/data-science-ipy...
23	fastai	fastai	22100	https://github.com/fastai/fastai
24	ML-From-Scratch	eriklindernoren	21000	https://github.com/eriklindernoren/ML-From-Scratch
25	MockingBird	babysor	20500	https://github.com/babysor/MockingBird
26	ray	ray-project	19800	https://github.com/ray-project/ray

RESULTS

RESULT: In this project we have successfully extracted the data from the github.com and also arranged the bulk of data into the readable and understandable way(dataset). Not only the datawe have also extracted all the links and addresses in the website.

CONCLUSION

While this project may not be as sophisticated as web scrapers made by large corporations, there is enough scope in this application to make a significant impact in the world of law enforcement. By utilising a set of buzzwords relating to sex trafficking and a spider targeted towards the right website, many trafficking crimes could be discovered in a matter of seconds, as the spiders in this project crawled through large webpages in under5 seconds.

REFERENCES

- Albert, R. Jeong, H. and Barabasi, A. (1999) Internet: Diameter of the world-wide web,Nature, 401(6749), pp. 130-131. do: 10.1038/43601.
- Balodis, M. (2017) Web Scraper. Available at: http://webscraper.io/ (Accessed: 02/11/17).
- Bin, H.Patel, and Zhen, Z. (2007) Accessing the Deep Web: a survey, Communications of theACM, 50(5), pp. 94-101.
- Boorse, K. (2016) Spotlight Helps Law Enforcement Identify Victims of Sex Trafficking Faster. Available at: https://www.wearethorn.org/blog/spotlight-helps-identify-sex-trafficking-victims- faster/ (Accessed: 29/10/17).
- Broder, A.Z., Najork, M. and Wiener, J.L. (2003) Efficient URL caching for world wide web crawling. , Budapest, Hungary. 20-24 May 2003. New York, NY, USA: ACM, pp. 679.

- Castillo, C. (2004) Effective Web Crawling. Ph.D. in Computer Science. University of Chile. Cordua, J. (2017) Clarity & Focus in 2017. Available at: https://www.wearethorn.org/blog/clarity-and-focus-2017/ (Accessed: 29/10/17).
- Europol (2017) SERIOUS AND ORGANISED CRIME THREAT ASSESSMENT Crime in the age of technologyEuropol. Available at: https://www.europol.europa.eu/sites/default/files/documents/socta2017_0.pdf (Accessed: 02/11/2017).
- Google (2017) Fighting Human Trafficking & Modern Day Slavery. Available at: https://www.blog.google/documents/4/Fighting%20Human%20Trafficking%20and%20Modern%2 0Day%20Slavery.pdf (Accessed: 06/11/17).
- IC3 (2017) 2016 Internet Crime Report Available at: https://pdf.ic3.gov/2016_IC3Report.pdf (Accessed: 19/11/2017).
- ILO (2012) ILO 2012 Global estimate of forced labour Executive summary Available at: http://www.ilo.org/wcmsp5/groups/public/---ed_norm/--- declaration/documents/ publication/wcms_181953.pdf (Accessed: 02/11/2017).
- Import.io (2017) Import.io | Extract data from the web. Available at: https://www.import.io/ (Accessed: 06/11/17).
- ITU (2015) ICT Facts and Figures 2015.

Sincerely Done By:

Nithin Rajulapati

39110831