

**Name: Gerosh George**

**19BCE1403**

**22/9/2021**

## **BASIC CRAWLER PROGRAM**

### **CODE:**

```
import requests
from bs4 import BeautifulSoup
from queue import Queue
import re

seeds = []
frontier = Queue(maxsize=300)
unique_urls = set()
extracted_urls = 0
extracted_links = []

def stop_criterion(limit=10):
    if extracted_urls >= limit:
        return True
    pass

def save_page(url, page):
    global extracted_urls

    folder = 'pages/'
    name = re.findall(r'https?://(.+)', url)[0]
    if name[-1] == '/':
        name = name[:-1]
    name = name.replace('/', '_')

    fname = folder + name + '.txt'
    with open(fname, "w", encoding="utf-8") as fp:
        fp.write(page)

    extracted_urls += 1
```

```

extracted_links.append(url)

msg = f"[{extracted_urls}] {url} (file name: {name}.txt)"

print(msg)

def initialise_frontier():
    for url in seeds:
        url = url.replace('www.', '')
        if url not in unique_urls:
            frontier.put(url)
            unique_urls.add(url)

def url_filter(url):
    if 'vit.ac.in' in url:

        for word in ['pdf', 'jpg', 'html', '/files']:
            if word in url:
                return False

        return True
    else:
        return False

def fetch_page(url):

    try:

        resp = requests.get(url)

        if resp.ok:

            save_page(url, resp.text)

            soup = BeautifulSoup(resp.text, 'html.parser')

            a_links = [a.get('href') for a in soup.findAll('a')]

            for link in a_links:
                if link and link.startswith("https"): #or link.startswith('http
s')):

                    link = link.replace('www', '')

```

```

        if link[-1]!=" ":
            link = link[:-1]

        if link not in unique_urls and url_filter(link):
            frontier.put(link)
            unique_urls.add(link)

        if frontier.full():
            #print("\n***** FRONTIER IS FULL *****\n")
            break

    else:
        print(f"\n[INFO] Error code for {url} : {resp.status_code}", "\n")

except Exception as e:
    print(f"\n[ERROR] {url} with error: {e.__doc__}", "\n")

def read_seeds():

    with open('seeds.txt', 'r') as fp:
        for url in fp.readlines():
            if url not in unique_urls:
                seeds.append(url)

def start_crawler(limit):

    read_seeds()
    initialise_frontier()

    while(not frontier.empty()):
        url = frontier.get()
        fetch_page(url)

        if stop_criterion(limit):
            break

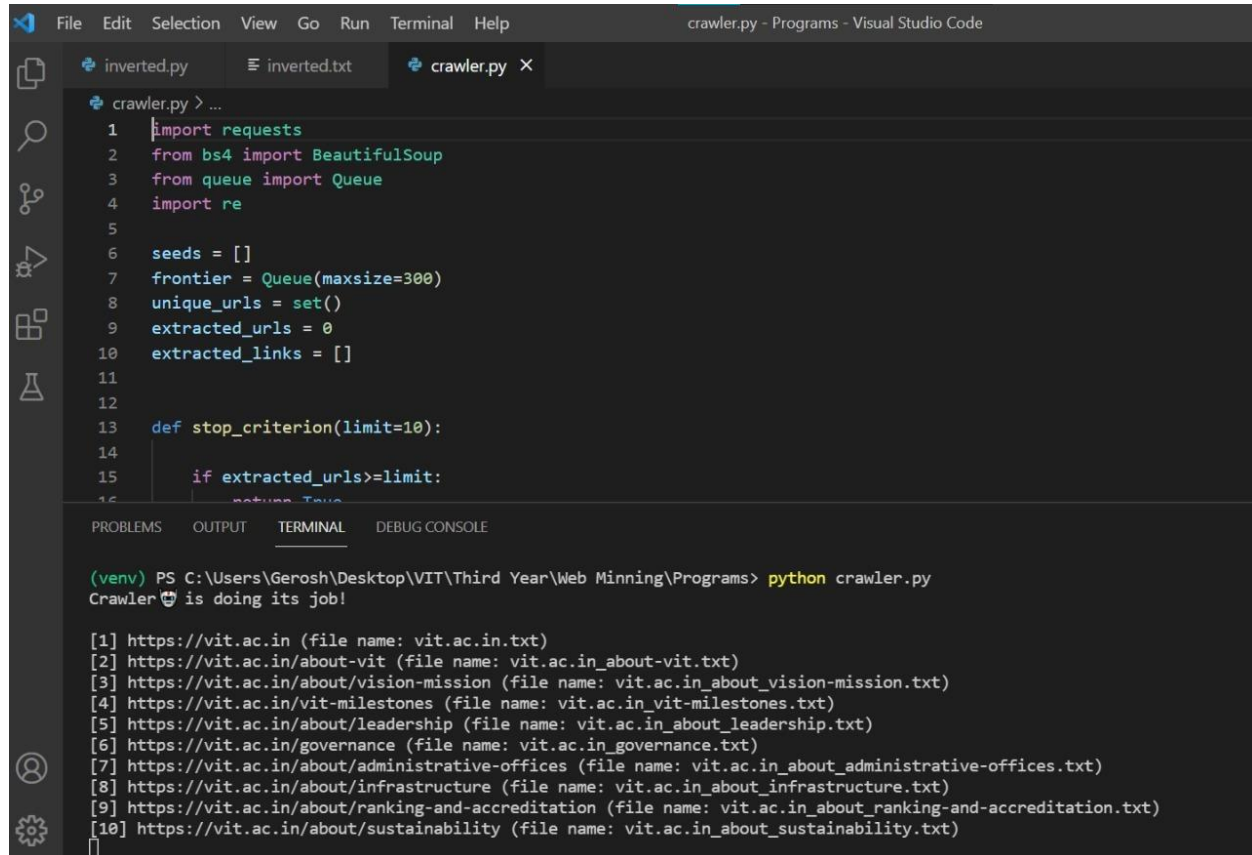
    print("\nNumber of links visited: ", len(extracted_links))

if __name__ == '__main__':

```

```
print('Crawler👾 is doing its job!\n')
start_crawler(300)
print('\nCrawler👾 has finished its job')
```

## OUTPUT:



The screenshot displays the Visual Studio Code interface with a Python script named `crawler.py` open in the editor. The script is a web crawler that uses `requests`, `BeautifulSoup`, and `Queue` to fetch and save web pages. The terminal window at the bottom shows the output of running the script, indicating that the crawler is active and has successfully fetched 10 pages from `vit.ac.in`, saving them with descriptive file names.

```
File Edit Selection View Go Run Terminal Help
crawler.py - Programs - Visual Studio Code

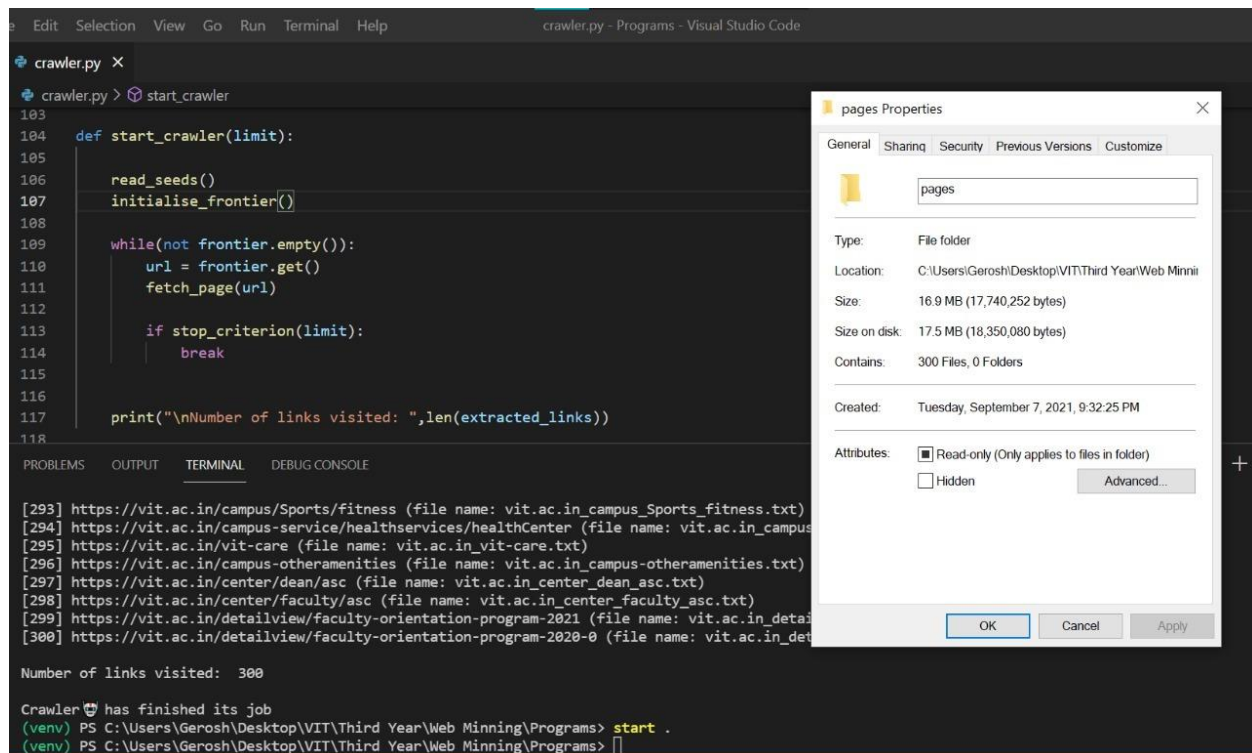
inverted.py inverted.txt crawler.py X

crawler.py > ...
1 import requests
2 from bs4 import BeautifulSoup
3 from queue import Queue
4 import re
5
6 seeds = []
7 frontier = Queue(maxsize=300)
8 unique_urls = set()
9 extracted_urls = 0
10 extracted_links = []
11
12
13 def stop_criterion(limit=10):
14
15     if extracted_urls>=limit:
16         return True

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

(venv) PS C:\Users\Gerosh\Desktop\VIT\Third Year\Web Mining\Programs> python crawler.py
Crawler👾 is doing its job!

[1] https://vit.ac.in (file name: vit.ac.in.txt)
[2] https://vit.ac.in/about-vit (file name: vit.ac.in_about-vit.txt)
[3] https://vit.ac.in/about/vision-mission (file name: vit.ac.in_about_vision-mission.txt)
[4] https://vit.ac.in/vit-milestones (file name: vit.ac.in_vit-milestones.txt)
[5] https://vit.ac.in/about/leadership (file name: vit.ac.in_about_leadership.txt)
[6] https://vit.ac.in/governance (file name: vit.ac.in_governance.txt)
[7] https://vit.ac.in/about/administrative-offices (file name: vit.ac.in_about_administrative-offices.txt)
[8] https://vit.ac.in/about/infrastructure (file name: vit.ac.in_about_infrastructure.txt)
[9] https://vit.ac.in/about/ranking-and-accreditation (file name: vit.ac.in_about_ranking-and-accreditation.txt)
[10] https://vit.ac.in/about/sustainability (file name: vit.ac.in_about_sustainability.txt)
```



Screenshot of one of pages that was extracted

