WEB MINING
LAB
CSE 3024

Faculty: Dr.Sridhar.R

LAB3

DATE : 19TH AUG 2021

VIT CHENNAI

Aim : Do the following operations using request and beautiful soup

Q1.Web page: https://www.vit.ac.in

Print the “title” of the page

Print out all the anchor tags with the class = “nav-link”

Code:

```
import requests
from bs4 import BeautifulSoup

URL = 'https://www.vit.ac.in'
response = requests.get(URL,verify=False)
soup = BeautifulSoup(response.text, 'lxml')

print("Titles:\n")
title = soup.findAll('title')
print(title)


print("\n\nA tags with class nav-link\n")
for link in soup.find_all('a',class_='nav-link'):
    print(link.text)
```

OutPut:

```
Titles:

VIT | No.1 Private Institution for Innovation


A tags with class nav-link

 Menu

Home
About Us
Overview
Vision & Mission
VIT Milestones
Leadership
Governance
Administrative Offices
Infrastructure
Ranking & Accreditation
Sustainability
True Green project
Community Outreach
Community Radio
Archieved News
Events
NIRF
MHRD/UGC
Careers@VIT
Newsletter
Academics
Overview
Academic Regulations
Programmes Offered
AY 2021-22
AY 2020-21
Curriculum
FFCS
Library
Schools
Feedback
Admissions
Overview
Programmes Offered
Undergraduate
Postgraduate
```

Q2.Web page : https://vit.ac.in/school/allfaculty/site/computer-applications

a.Print out all the faculty names using the class id “title2” and their research area by devising appropriate algorithm – you need to use methods of beautifulsoup

Code:

```
import termtables as tt

import requests
from bs4 import BeautifulSoup

URL = "https://vit.ac.in/school/allfaculty/site/computer-applications"
response = requests.get(URL,headers={'User-Agent':"Mozilla/5.0"}).text
soup = BeautifulSoup(response,'lxml')

finalList=[]
print("\n\nFaculties with class title2 with research work: \n")
for link in soup.findAll('div',class_='lightbox_course fancybox-content'):
    facList = link.text.strip().split("\n")
    if(facList[0]!=facList[-1]):
        finalList.append([facList[0],facList[-1]])
    print("\n")

string = tt.to_string(
    finalList,
    header=["Fac Name", "Research Area"],
    style=tt.styles.ascii_thin_double,
    # alignment="ll",
    # padding=(0, 1),
)
print(string)
```

OutPut:

Fac Name	Research Area
Dr.Ramkumar T	Research Area :Data Mining & Big Data Analytics
Dr.Ephzibah E.P	Research Area :Data Mining and Artificial Intelligence
Dr. Karthikeyan P	Research Area :Cloud computing, Web Services
Dr.Manivannan S.S	Research Area :Network and Information Security, IoT and Machine Learning
Dr. Meenatchi S	Research Area :Computer Science Hardware and Architecture
Dr.Shynu P.G	Research Area :Cloud computing, Information Security, Data Science
Dr. Uma Maheswari G	Research Area :Computer Science Information Systems
Dr.Deepa N	Research Area :Predictive Analytics
Dr.Jayalakshmi P	Research Area :Networks
Dr. Senthil Kumar N	Research Area :Semantic Web; Information Retrieval
Ms. Manisha R. Patil	Email : manishapatil.r@vit.ac.in
Ms. Manjupriya R	Research Area :

b.Find the facebook link, twitter link, instagram link and linkedin link of VIT from the page content and identify the class names given for each.

```
import requests
from bs4 import BeautifulSoup


URL = 'https://vit.ac.in/school/allfaculty/site/computer-applications'
response = requests.get(URL,verify=False)
soup = BeautifulSoup(response.text, 'html.parser').find('span',class_='soclia_links')

print("\nSocial links: ")

for link in soup.children:
    print("Link: " + link['href'])
    print("Class: " + str(link['class'])+"\n")
```

OutPut:

```
Social links:
Link: https://www.facebook.com/VITuniversity/
Class: ['face_book_icon', 'f_icon_be']

Link: https://twitter.com/vit_univ
Class: ['twitter_icon', 'f_icon_be']

Link: https://www.linkedin.com/school/vellore-institute-of-technology/
Class: ['linkedin_icon', 'f_icon_be']
```

c.List out the DOM hierarchy of the page [Find all the children and the relationship between the children

```
import requests
from bs4 import BeautifulSoup

URL = "https://vit.ac.in/school/allfaculty/site/computer-applications"
response = requests.get(URL,headers={'User-Agent':"Mozilla/5.0"}).text
soup = BeautifulSoup(response,'lxml')

def printDom(element,maxDepth,space=0):
    if maxDepth>0:
        if space==0:
            print("┌>",element.name)
        else:
            print("| "*space+"├>",element.name)
    for i in element.findChildren(recursive=False):
        printDom(i,maxDepth-1,space+1)
    return
printDom(soup.html,7)
```

OutPut:

```
┌> html
| ├> head
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> link
| | ├> meta
| | ├> link
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> title
| | ├> style
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| | ├> meta
| ├> body
| | ├> div
```


| |> body
| | |> div
| | | |> div
| | | | |> a
| | | |> div
| | | | |> script
| | | | |> script
| | | | |> script
| | | | |> section
| | | | | |> div
| | | | | | |> div
| | | | |> header
| | | | | |> div
| | | | | | |> div
| | | | | | |> div
| | | | | | |> div
| | | | |> div
| | | | |> div
| | | | | |> div
| | | | | | |> div
| | | | | |> script
| | | | | |> footer
| | | | | | |> div
| | | | | |> a
| | | | | | |> i
| | | | | |> div
| | | | |> script
| | | | |> script
| | | | |> script
| | | | |> script
| | | | |> script
| | | | |> script
| | | | |> script
| | | | |> script
| | | | |> script
| | | | |> link
| | | | |> noscript
| | | | | |> link
| | | | | |> div
| | | | |> script
| | | | |> script
| | | | |> script
| | | | |> script

Q3.Web page : https://sermitsiaq.ag/english

1.Find all the items of class=”menu” and print out the items of the menu with class names With class names “first leaf”, “leaf” and “last leaf”.

Code:

```
URL = "https://sermitsiaq.ag/english"
response = requests.get(URL,headers={'User-Agent':"Mozilla/5.0"}).text
soup = BeautifulSoup(response,'lxml')

print("\033[92mEle with class Leaf")
for links in soup.findAll(class_='leaf'):
    print("     \033[96m"+links.text)

print("\033[92mEle with class firstLeaf")
for links in soup.findAll(class_='first leaf'):
    print("     \033[96m"+links.text)

print("\033[92mEle with class lastLeaf")
for links in soup.findAll(class_='last leaf'):
    print("     \033[96m"+links.text)
```

OutPut:

```
Ele with class Leaf
     Bestil foto
     Abonnement
     Annoncer
     Kontakt
     E-aviser
     JOB
     Pilivik
     Forsiden
     Indland
     Nuuk
     Politik
     Erhverv
     Politi
     Udland
     Kultur
     Sport
     Nyhedsoversigt
     Job
Ele with class firstLeaf
     Bestil foto
     Forsiden
Ele with class lastLeaf
     Pilivik
```

2.Find all items with ids containing string “menu” in them

Code:

```
URL = "https://sermitsiaq.ag/english"
response = requests.get(URL,headers={'User-Agent':"Mozilla/5.0"}).text
soup = BeautifulSoup(response,'lxml').find(id='mainmenu')

for link in soup:
    print(link.text)
```

OutPut:

```
<nav class="mainmenu"><div class="panel-pane pane-block pane-system-main-menu">
<ul class="menu"><li class="first leaf"><a href="/">Forsiden</a></li>
<li class="leaf"><a class="menu-indland" href="/indland">Indland</a></li>
<li class="leaf"><a href="/nuuk">Nuuk</a></li>
<li class="leaf"><a class="menu-politik" href="/politik">Politik</a></li>
<li class="leaf"><a href="/taxonomy/term/6/">Erhverv</a></li>
<li class="leaf"><a href="/politi">Politi</a></li>
<li class="leaf"><a class="menu-udland" href="/udland">Udland</a></li>
<li class="leaf"><a href="/kultur">Kultur</a></li>
<li class="leaf"><a href="/sport">Sport</a></li>
<li class="leaf"><a href="/nyhedsoversigt">Nyhedsoversigt</a></li>
```

3.Find all items with tag “article”.

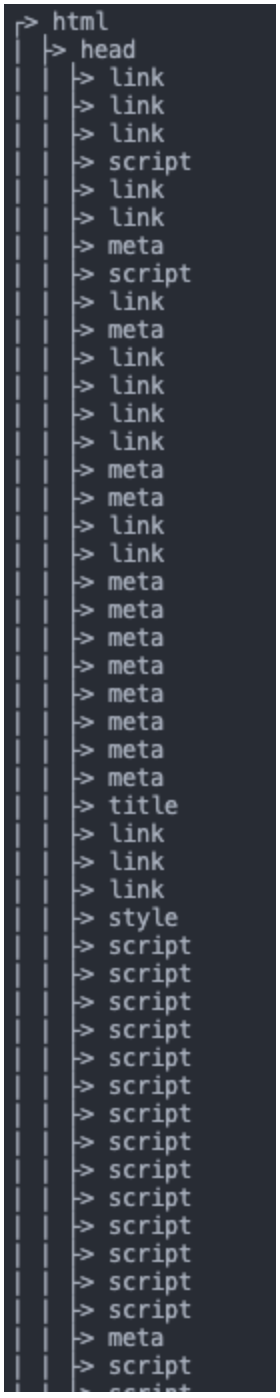
```
import requests
from bs4 import BeautifulSoup

URL = "https://sermitsiaq.ag/english"
response = requests.get(URL,headers={'User-Agent':"Mozilla/5.0"}).text
soup = BeautifulSoup(response,'lxml')

print("\033[91mEle with article tags")
for article in soup.findAll('article'):
    print("\033[96m"+article.text.strip().replace(' ', '')+"\n")
```

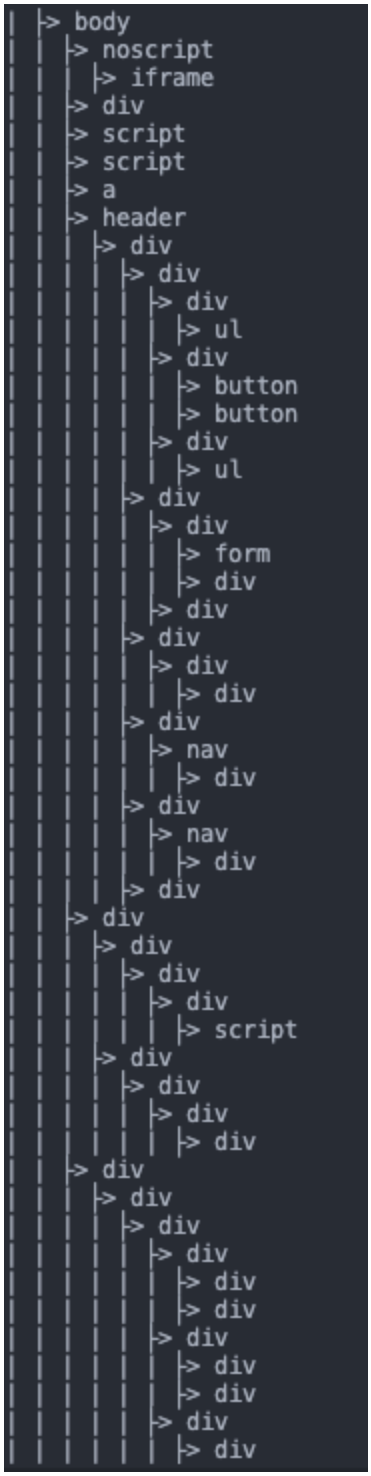
OutPut:

```
Ele with article tags
Arcticpeopleconcernedforthefuture

Pressrelease:CommonArcticsearchandrescueserviceagreed

AddressKuupikKleistArcticCouncil7Meeting2011

OpenlettersenttotheForeignMinistersofCanada,U.S.,Norway,Denmark,GreenlandandRussia

ResourceDevelopmentPrinciplesinInuitNunaat

ArcticCouncilNuukMinisterialAgenda

Editorial:OurfriendsinNorway

Climatechange–it'saboutthepeople

LivingconditionsandeconomicdevelopmentinthefaceofclimatechangearethechallengestheA
nd'spremiersays

TightsecurityduringArcticsummit

Terrorismthreatlevelsremain"serious"asArcticforeignministersgatherinNuuk

NewArcticstrategytobepresentedinNuuk

Asclimatescientistspaintanincreasinglydirepictureofglobalwarming,ithasbeguntosinki
mentrequiresimmediateaction

ArcticCouncilNuukMinisterialAgenda

ParticipantsinArcticCouncilNuukMinisterial

Thesmallestdelegationshaveasfewassixmembers,thelargestupto25
```

4.List out the DOM hierarchy of the page.

```python
import requests
from bs4 import BeautifulSoup

URL = "https://vit.ac.in/school/allfaculty/site/computer-applications"
response = requests.get(URL,headers={'User-Agent':"Mozilla/5.0"}).text
soup = BeautifulSoup(response,'lxml')

def printDom(element,maxDepth,space=0):
    if maxDepth>0:
        if space==0:
            print("┌>",element.name)
        else:
            print("| "*space+"├>",element.name)
    for i in element.findChildren(recursive=False):
        printDom(i,maxDepth-1,space+1)
    return
printDom(soup.html,7)
```


> html
> head
> link
> link
> link
> script
> link
> link
> meta
> script
> link
> meta
> link
> link
> link
> link
> meta
> meta
> link
> link
> meta
> meta
> meta
> meta
> meta
> meta
> meta
> meta
> meta
> title
> link
> link
> link
> style
> script
> script
> script
> script
> script
> script
> script
> script
> script
> script
> script
> script
> script
> meta
> script


|-> body
|-> noscript
|-> iframe
|-> div
|-> script
|-> script
|-> a
|-> header
|-> div
|-> div
|-> div
|-> ul
|-> div
|-> button
|-> button
|-> div
|-> ul
|-> div
|-> div
|-> form
|-> div
|-> div
|-> div
|-> div
|-> div
|-> nav
|-> div
|-> div
|-> nav
|-> div
|-> div
|-> div
|-> div
|-> script
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div
|-> div

Q4.Take website :https://www.batimes.com.ar

A.List items of class “nav-item text-uppercase px-0”

Code:

```
import requests
from bs4 import BeautifulSoup

URL = "https://www.batimes.com.ar"
response = requests.get(URL,verify=False)
soup = BeautifulSoup(response.text, 'html.parser')

print("\nItems with given class:")
for item in soup.findAll(class_='nav-item text-uppercase px-0'):
    print(item.text.strip())
```

OutPut:

```
Items with given class:
Topics


Olivos party


Messi moves to Paris


Matías Kulfas interview


IPCC report
```

B.Search for string “Matías Lammens” and list out the HTML item of in which the string occurs

Code:

```
import requests
from bs4 import BeautifulSoup as bs

url='https://www.batimes.com.ar/'
html_text=requests.get(url,headers={'User-Agent':'Mozilla/5.0'}).text
soup=bs(html_text,'html.parser')
def WordFind(element):
    string=str(element.find(text=True,recursive=False))
    if "Matías Lammens" in string:
        return True
    return False
for list1 in soup.find_all(WordFind):
    print(list1)
```

OutPut:

```
<h4>Sport &amp;amp; Tourism Minister Matías Lammens says "football with spectators
y September or October.
</h4>
```

C.List all the images in the page

Code:

```
import requests
from bs4 import BeautifulSoup

URL = "https://www.batimes.com.ar"
response = requests.get(URL,verify=False)
soup = BeautifulSoup(response.text, 'html.parser')


print("All the image sources:\n")
for ele in soup.findAll('img'):
    if(ele['src']):
        print("Image Src: " + str(ele['src']))
        print("\n")
```

OutPut:

```
All the image sources:

Image Src: https://www.batimes.com.ar/img/logo_perfil.svg


Image Src: https://www.batimes.com.ar/img/logo.svg


Image Src: https://via.placeholder.com/1140x540?text=BATIMES


Image Src: https://via.placeholder.com/720x355?text=BATIMES


Image Src: https://via.placeholder.com/720x355?text=BATIMES


Image Src: https://via.placeholder.com/720x355?text=BATIMES


Image Src: https://via.placeholder.com/720x355?text=BATIMES
```

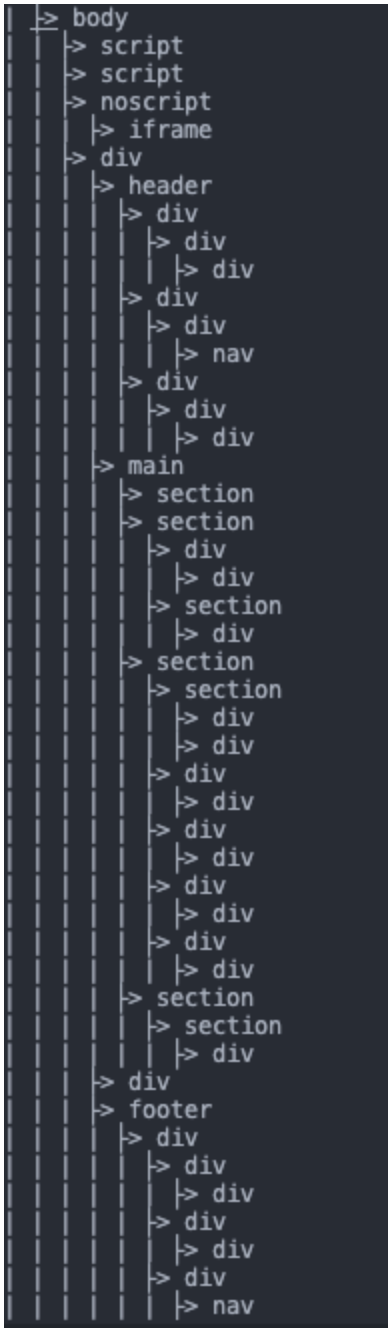
D.List out the DOM hierarchy of the page.

Code:

```
def printDom(element,maxDepth,space=0):
    if maxDepth>0:
        if space==0:
            print("┌>",element.name)
        else:
            print("| "*space+"├>",element.name)
    for i in element.findChildren(recursive=False):
        printDom(i,maxDepth-1,space+1)
    return
printDom(soup.html,7)
```

OutPut:

```
┌> html
|  ├> head
|  |  ├> meta
|  |  ├> meta
|  |  ├> meta
|  |  |  ├> meta
|  |  |  ├> meta
|  |  |  ├> meta
|  |  |  ├> meta
|  |  |  ├> meta
|  |  |  |  ├> meta
|  |  |  |  ├> meta
|  |  |  |  ├> meta
|  |  |  |  ├> meta
|  |  |  |  ├> meta
|  |  |  |  ├> meta
|  |  |  |  ├> meta
|  |  |  |  |  ├> meta
|  |  |  |  |  ├> meta
|  |  |  |  |  ├> meta
|  |  |  |  |  ├> meta
|  |  |  |  |  ├> meta
|  |  |  |  |  ├> meta
|  |  |  |  |  ├> meta
|  |  |  |  |  ├> meta
|  |  |  |  |  |  ├> title
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> link
|  |  |  |  |  |  ├> linl
```


|-> body
| |-> script
| |-> script
| |-> noscript
| | |-> iframe
| |-> div
| | |-> header
| | | |-> div
| | | | |-> div
| | | | | |-> div
| | | |-> div
| | | | |-> div
| | | | | |-> nav
| | | |-> div
| | | | |-> div
| | | | | |-> div
| | |-> main
| | | |-> section
| | | |-> section
| | | | |-> div
| | | | | |-> div
| | | | |-> section
| | | | | |-> div
| | | |-> section
| | | | |-> section
| | | | | |-> div
| | | | | |-> div
| | | | |-> div
| | | | | |-> div
| | | | |-> div
| | | | | |-> div
| | | | |-> div
| | | | | |-> div
| | | | |-> div
| | | | | |-> div
| | | |-> section
| | | | |-> section
| | | | | |-> div
| | |-> div
| | |-> footer
| | | |-> div
| | | | |-> div
| | | | | |-> div
| | | | |-> div
| | | | | |-> div
| | | | |-> div
| | | | | |-> nav