Harshit Mishra (19BCE0799)

CSE-3024 WEB MINING

LAB ASSIGNMENT 4

Aim: Use BeautifulSoup or Scrapy to crawl any one of the E-commerce websites of your choice and perform the same. The following information needs to be extracted from the page:

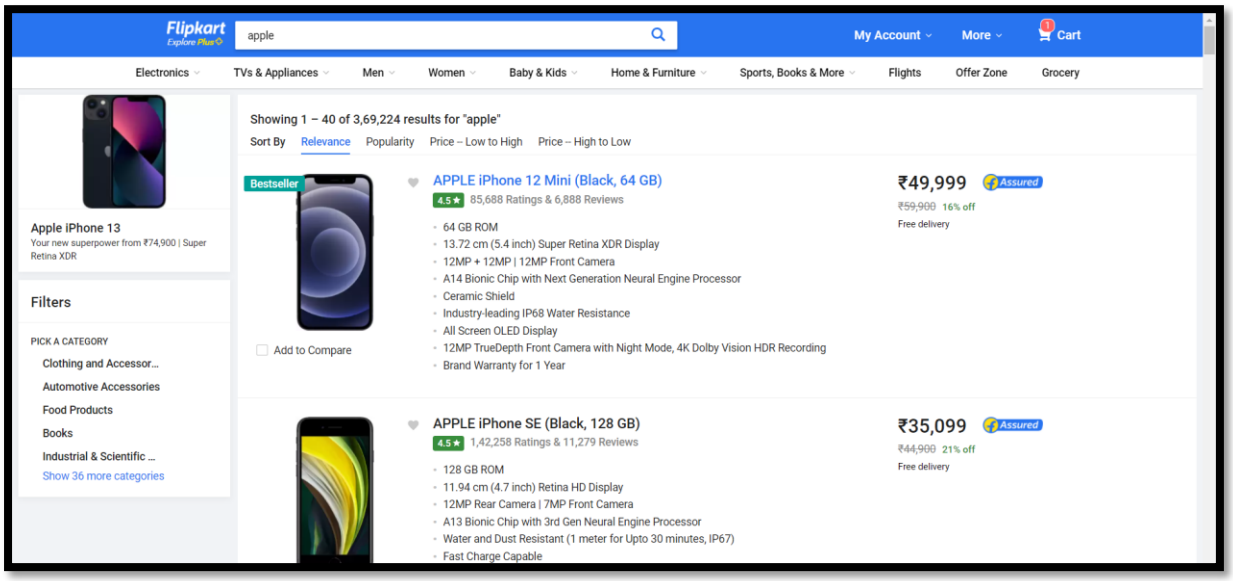
a) Product Name
b) Product Price
c) Product Discount
d) Product Image

Procedure:

- Firstly, we import our scraping libraries of BeautifulSoup and requests.
- We then assign a url variable that stores the URL of web page to be scrapped. In our case it is going to be the URL of a flipkart page featuring apple products.
- Using requests library's get method we take in the page as a text input and store it in variable named text.
- We next create an HTML parser using BeautifulSoup library and store it in soup variable.
- Then we create in a tags variable that store our div block of information that we are seeking. The class name depends on the site we are looking and its DOM structure.
- Then we import our csv related libraries to store our results in csv format.

- Then we open a writeonly file and write in the headers named as Name, Price, Discount and Image.
- Next, we use the getattr and tag methods to successfully parse in the DOM structure of our tags variable.
- This will include details of Name of the product, Price of the product, Discount on the product and Image tag of the product.
- We are getting the image tag to be stored in csv file.

Site we are scrapping:



URL of the above website:

https://www.flipkart.com/search?q=apple&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=on&as=off

Harshit Mishra (19BCE0799)

Code:

```
#Importing scrapper librariess
from bs4 import BeautifulSoup
import requests

#Scrapping the website
url =
"https://www.flipkart.com/search?q=apple&otracker=search&otracker1=search&marketplace=FLIP
KART&as-show=on&as=off"
page = requests.get(url).text
soup = BeautifulSoup(page, 'html.parser')
tags = soup.find_all('div', class_="_1AtVbE col-12-12")


#Creating csv
from csv import writer
import colorama
from colorama import Fore
print(Fore.WHITE+"Scraping data "+Fore.GREEN+"done...")
with open('result.csv','w', encoding='utf8',newline='') as f:
    thewriter = writer(f)
    header = ['Name','Price','Discount','Image']
    thewriter.writerow(header)
    for tag in tags:
        name = getattr(tag.find('div',class_="_4rR01T"),'text', None)
        price = getattr(tag.find('div', class_="_30jeq3 _1_WHN1"),'text', None)
        discount = getattr(tag.find('div',class_="_3Ay6Sb"),'text', None)
        image = tag.find('img', class_="_396cs4 _3exPp9")
        info = [name, price, discount, image]
        thewriter.writerow(info)


print(Fore.WHITE+"Successfully scrapped data!! Results in " + Fore.GREEN+"result.csv")


#Reading the CSV File
import pandas as pd
data = pd.read_csv("result.csv")

#Printing the data
data
```

Code Snippet and Outputs:

```
In [1]: #Importing scrapper librariess
        from bs4 import BeautifulSoup
        import requests
```

Here we are importing our BeautifulSoup and requests libraries to be used for scrapping the web page.

```
In [2]: #Scrapping the website
        url = "https://www.flipkart.com/search?q=apple&otracker=search&otracker1=searc
        page = requests.get(url).text
        soup = BeautifulSoup(page, 'html.parser')
        tags = soup.find_all('div', class_="_1AtVbE col-12-12")
```

Here we are passing in the URL of our webpage to the variable and then fetching its resources in a text format to store it in a variable named page.

Then we pass it as an html source for Document Object Manipulation to another variable named soup.

Finally, to get its div tags we use find_all method of our bs4 library.

```
In [3]: #Creating csv
from csv import writer
import colorama
from colorama import Fore
print(Fore.YELLOW+"Scraping data "+Fore.GREEN+"done...")
with open('result2.csv','w', encoding='utf8',newline='') as f:
    thewriter = writer(f)
    header = ['Name','Price','Discount','Image']
    thewriter.writerow(header)
    for tag in tags:
        name = getattr(tag.find('div',class_="_4rR01T"),'text', None)
        price = getattr(tag.find('div', class_="_30jeq3 _1_WHN1"),'text', None
        discount = getattr(tag.find('div',class_="_3Ay6Sb"),'text', None)
        image = tag.find('img', class_="_396cs4 _3exPp9")
        info = [name, price, discount, image]
        thewriter.writerow(info)

Scraping data done...
```

Here we are storing our scrapped data in a csv file named result. This file is headed with titles of Name, Price, Discount and Image. We are using getattr method to find the respective name, price and etc of a product and assign them to the variables. Finally, we make a list of these values and write this info in results.csv file. This will update the contents of our result file. We repeat the above process until we run out of tags which we obtained using soup.find_all method in cell 2.

```
In [4]: print(Fore.YELLOW+"Successfully scrapped data!! Results in " + Fore.GREEN+"res

Successfully scrapped data!! Results in result.csv
```

Finally, we print that the scrapped data is successfully stored in our working directory.

```
In [5]: #Reading the CSV File
        import pandas as pd
        data = pd.read_csv("result.csv")
```

Here we are importing the result.csv file into our workspace.

```
In [6]: #Printing the data
        data
```

Out[6]:

	Name	Price	Discount	Image
0	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN
2	APPLE iPhone SE (Black, 128 GB)	₹35,099	21% off	<img alt="APPLE iPhone SE (Black, 128 GB)" cla...
3	APPLE iPhone SE (White, 128 GB)	₹35,099	21% off	<img alt="APPLE iPhone SE (White, 128 GB)" cla...
4	APPLE iPhone 12 Mini (Black, 64 GB)	₹49,999	16% off	<img alt="APPLE iPhone 12 Mini (Black, 64 GB)"...
5	APPLE iPhone SE (Red, 64 GB)	₹30,099	24% off	<img alt="APPLE iPhone SE (Red, 64 GB)" class=...
6	APPLE iPhone SE (White, 64 GB)	₹30,099	24% off	<img alt="APPLE iPhone SE (White, 64 GB)" clas...
7	APPLE iPhone SE (Black, 64 GB)	₹30,099	24% off	<img alt="APPLE iPhone SE (Black, 64 GB)" clas...

Here we are printing the data. We can see that first column contains Name of the product, second attribute contains price of the product in Indian Rupee, third attribute contains the discount offered in that product and the last attribute contains the HTML element of image formatted in the csv file.

Harshit Mishra (19BCE0799)

Results:

CSV file as printed in python script:

Out[6]:

	Name	Price	Discount	Image
0	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN
2	APPLE iPhone SE (Black, 128 GB)	₹35,099	21% off	<img alt="APPLE iPhone SE (Black, 128 GB)" cla...
3	APPLE iPhone SE (White, 128 GB)	₹35,099	21% off	<img alt="APPLE iPhone SE (White, 128 GB)" cla...
4	APPLE iPhone 12 Mini (Black, 64 GB)	₹49,999	16% off	<img alt="APPLE iPhone 12 Mini (Black, 64 GB)"...
5	APPLE iPhone SE (Red, 64 GB)	₹30,099	24% off	<img alt="APPLE iPhone SE (Red, 64 GB)" class=...
6	APPLE iPhone SE (White, 64 GB)	₹30,099	24% off	<img alt="APPLE iPhone SE (White, 64 GB)" clas...
7	APPLE iPhone SE (Black, 64 GB)	₹30,099	24% off	<img alt="APPLE iPhone SE (Black, 64 GB)" clas...

result.csv file: