Marwadi University	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Data Visualization and Dashboard (01CT0410)	**Aim: Web Scrapping using Python**	
Experiment No: 11	**Date: 30-03-2024**	**Enrollment No: 92200133030**

Aim: **Web Scrapping using Python**

IDE: Tableau

Theory:

Web scraping is a valuable technique for extracting data from websites for various purposes such as research, analysis, and automation. In this lab manual, we will explore the fundamentals of web scraping using Python, one of the most popular programming languages for this task. Through hands-on exercises and examples, students will learn how to retrieve data from web pages, parse HTML content, handle different types of data, and store the extracted information for further analysis.

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. Web scraping software may directly access the World Wide Web using the Hypertext Transfer Protocol or a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web 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 or analysis.

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, web crawling is a main component of web scraping, to fetch pages for later processing. Once fetched, extraction can take place. The content of a page may be parsed, searched and reformatted, and 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 finding and copying names and telephone numbers, companies and their URLs, or e-mail addresses to a list (contact scraping).

As well as contact scraping, web scraping is used as a component of applications used for web indexing, web mining and data mining, online price change monitoring and price comparison, product review scraping (to watch the competition), gathering real estate listings, weather data monitoring, website change detection, research, tracking online presence and reputation, web mashup, and web data integration. Web pages are built using text-based mark-up languages (HTML and XHTML), and frequently contain a wealth of useful data in text form. However, most web pages are designed 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 web pages.

Newer forms of web scraping involve monitoring data feeds from web servers. For example, JSON is commonly used as a transport mechanism between the client and the web server. There are methods that some websites use to prevent web scraping, such as detecting and disallowing bots from crawling (viewing) their pages. In response, there are web scraping systems that rely on using techniques in DOM parsing, computer vision and natural language processing to simulate human browsing to enable gathering web page content for offline parsing

Pre Lab Exercise:

a. How do you retrieve HTML content from a web page using Python?

b. What are the steps involved in storing and saving scraped data for further analysis?

c. What are the potential challenges and limitations of web scraping?

Tasks:

Perform the following tasks:

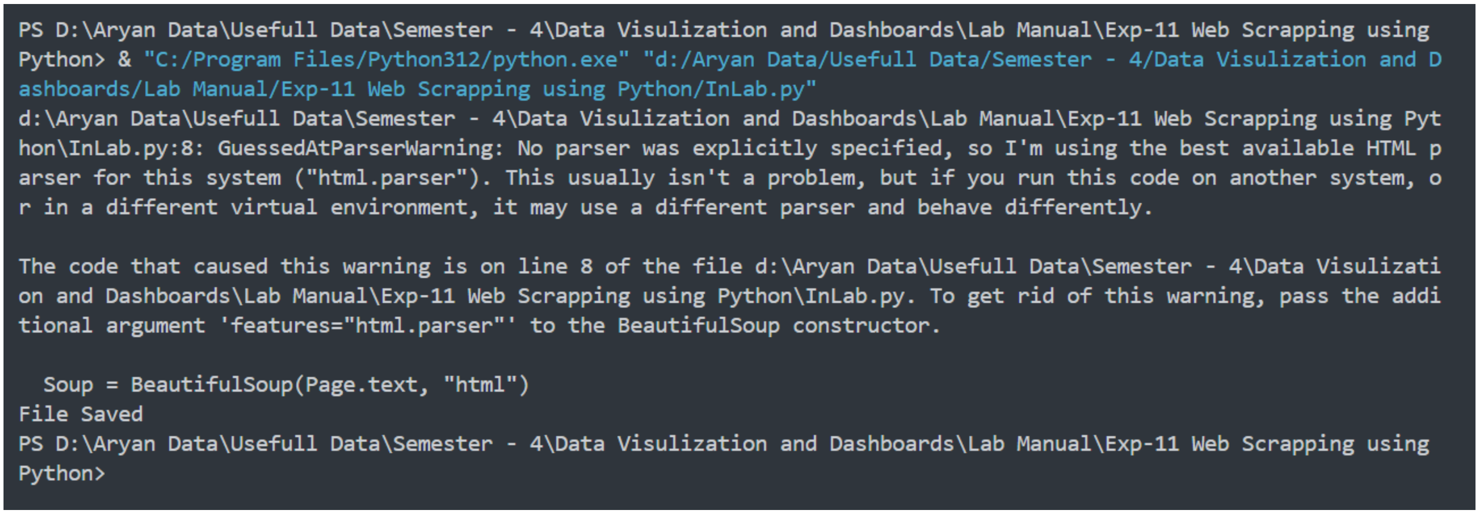
Task 1: Extract Table-1 from the web link https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_United_States_by_revenue.

Code :-

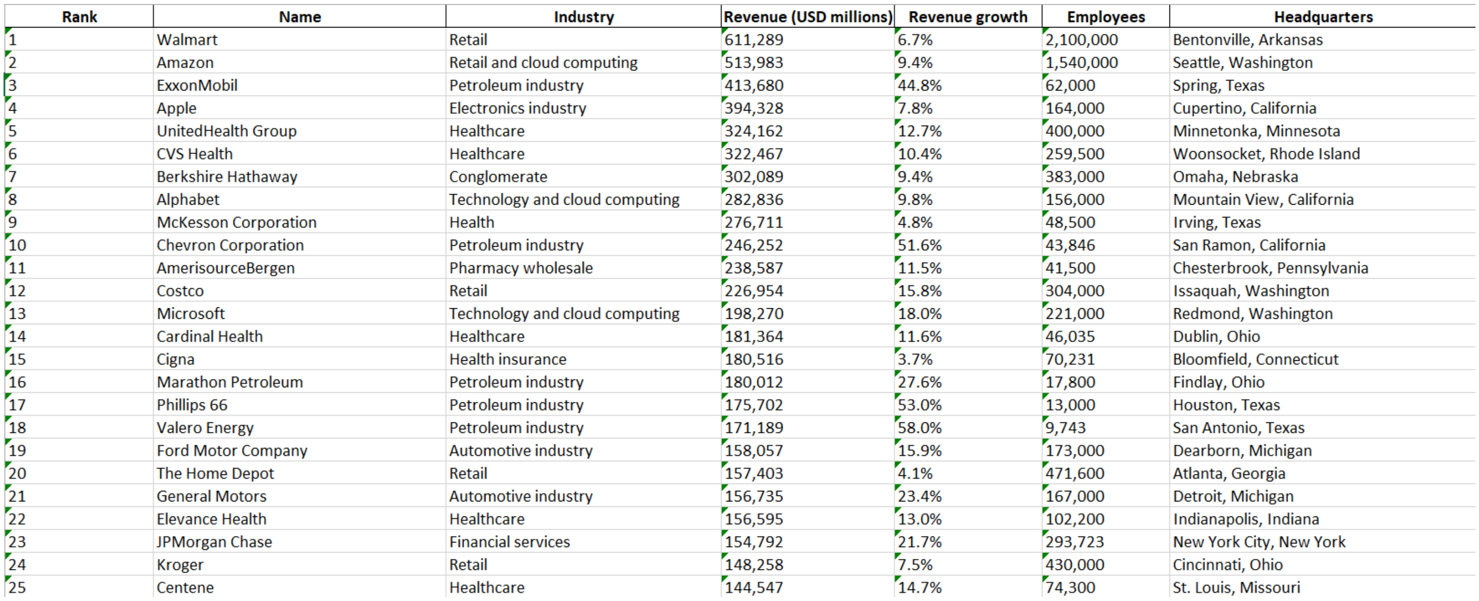
```
import pandas as pd
import requests
from bs4 import BeautifulSoup
url =
"https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_United_States_by_revenue"
Page = requests.get(url)
Soup = BeautifulSoup(Page.text, "html")
Table = Soup.find_all("table")
Table = Table[1]

Headers = Table.find_all("th")
```

```
Titles = [titles.text.strip() for titles in Headers]
Dataset = pd.DataFrame(columns=Titles)
Rows = Table.find_all("tr")
for row in Rows[1:]:
   Data = row.find_all("td")
   Row_Data = [row.text.strip() for row in Data]
   length = len(Dataset)
   Dataset.loc[length] = Row_Data
Dataset.to_excel("D:/Aryan Data/Usefull Data/Semester - 4/Data Visulization and
Dashboards/Lab/Web Scrapping/Largest Companies In USA by Revenue.xlsx",index=False,)
print("File Saved")
```

Results:-



```
PS D:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using
Python> & "C:/Program Files/Python312/python.exe" "d:/Aryan Data/Usefull Data/Semester - 4/Data Visulization and D
ashboards/Lab Manual/Exp-11 Web Scrapping using Python/InLab.py"
d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Pyt
hon\InLab.py:8: GuessedAtParserWarning: No parser was explicitly specified, so I'm using the best available HTML p
arser for this system ("html.parser"). This usually isn't a problem, but if you run this code on another system, o
r in a different virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 8 of the file d:\Aryan Data\Usefull Data\Semester - 4\Data Visulizati
on and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python\InLab.py. To get rid of this warning, pass the addi
tional argument 'features="html.parser"' to the BeautifulSoup constructor.

  Soup = BeautifulSoup(Page.text, "html")
File Saved
PS D:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using
Python>
```



Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters
1	Walmart	Retail	611,289	6.7%	2,100,000	Bentonville, Arkansas
2	Amazon	Retail and cloud computing	513,983	9.4%	1,540,000	Seattle, Washington
3	ExxonMobil	Petroleum industry	413,680	44.8%	62,000	Spring, Texas
4	Apple	Electronics industry	394,328	7.8%	164,000	Cupertino, California
5	UnitedHealth Group	Healthcare	324,162	12.7%	400,000	Minnetonka, Minnesota
6	CVS Health	Healthcare	322,467	10.4%	259,500	Woonsocket, Rhode Island
7	Berkshire Hathaway	Conglomerate	302,089	9.4%	383,000	Omaha, Nebraska
8	Alphabet	Technology and cloud computing	282,836	9.8%	156,000	Mountain View, California
9	McKesson Corporation	Health	276,711	4.8%	48,500	Irving, Texas
10	Chevron Corporation	Petroleum industry	246,252	51.6%	43,846	San Ramon, California
11	AmerisourceBergen	Pharmacy wholesale	238,587	11.5%	41,500	Chesterbrook, Pennsylvania
12	Costco	Retail	226,954	15.8%	304,000	Issaquah, Washington
13	Microsoft	Technology and cloud computing	198,270	18.0%	221,000	Redmond, Washington
14	Cardinal Health	Healthcare	181,364	11.6%	46,035	Dublin, Ohio
15	Cigna	Health insurance	180,516	3.7%	70,231	Bloomfield, Connecticut
16	Marathon Petroleum	Petroleum industry	180,012	27.6%	17,800	Findlay, Ohio
17	Phillips 66	Petroleum industry	175,702	53.0%	13,000	Houston, Texas
18	Valero Energy	Petroleum industry	171,189	58.0%	9,743	San Antonio, Texas
19	Ford Motor Company	Automotive industry	158,057	15.9%	173,000	Dearborn, Michigan
20	The Home Depot	Retail	157,403	4.1%	471,600	Atlanta, Georgia
21	General Motors	Automotive industry	156,735	23.4%	167,000	Detroit, Michigan
22	Elevance Health	Healthcare	156,595	13.0%	102,200	Indianapolis, Indiana
23	JPMorgan Chase	Financial services	154,792	21.7%	293,723	New York City, New York
24	Kroger	Retail	148,258	7.5%	430,000	Cincinnati, Ohio
25	Centene	Healthcare	144,547	14.7%	74,300	St. Louis, Missouri

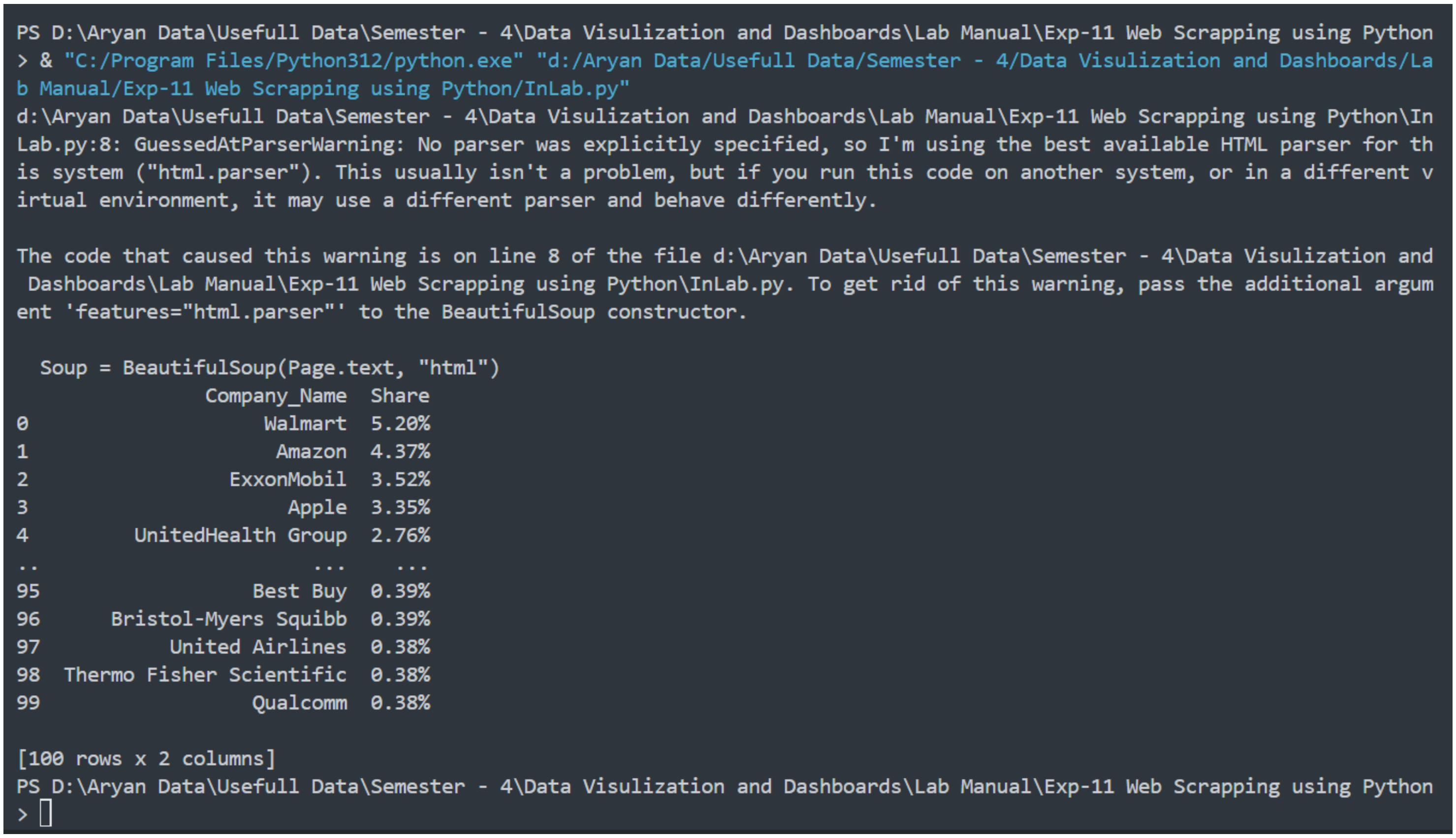
Task 2: Perform data analysis over:

1. Analyze the market share of different companies based on their revenue.

Code:-

```
Dataset["Revenue (USD millions)"] = (
Dataset["Revenue (USD millions)"].str.replace(",", "").astype(float))
total_revenue = Dataset["Revenue (USD millions)"].sum()
Market_Share = {"Company_Name": [], "Share": []}

for index, row in Dataset.iterrows():
   company_name = row["Name"]
   revenue = row["Revenue (USD millions)"]
   share_percentage = (revenue / total_revenue) * 100
   Market_Share["Company_Name"].append(company_name)
   Market_Share["Share"].append(share_percentage)
Market_Share = pd.DataFrame(Market_Share)
Market_Share["Share"] = Market_Share["Share"].map("{:.2f}%".format)
print(Market_Share)
```

Results:-



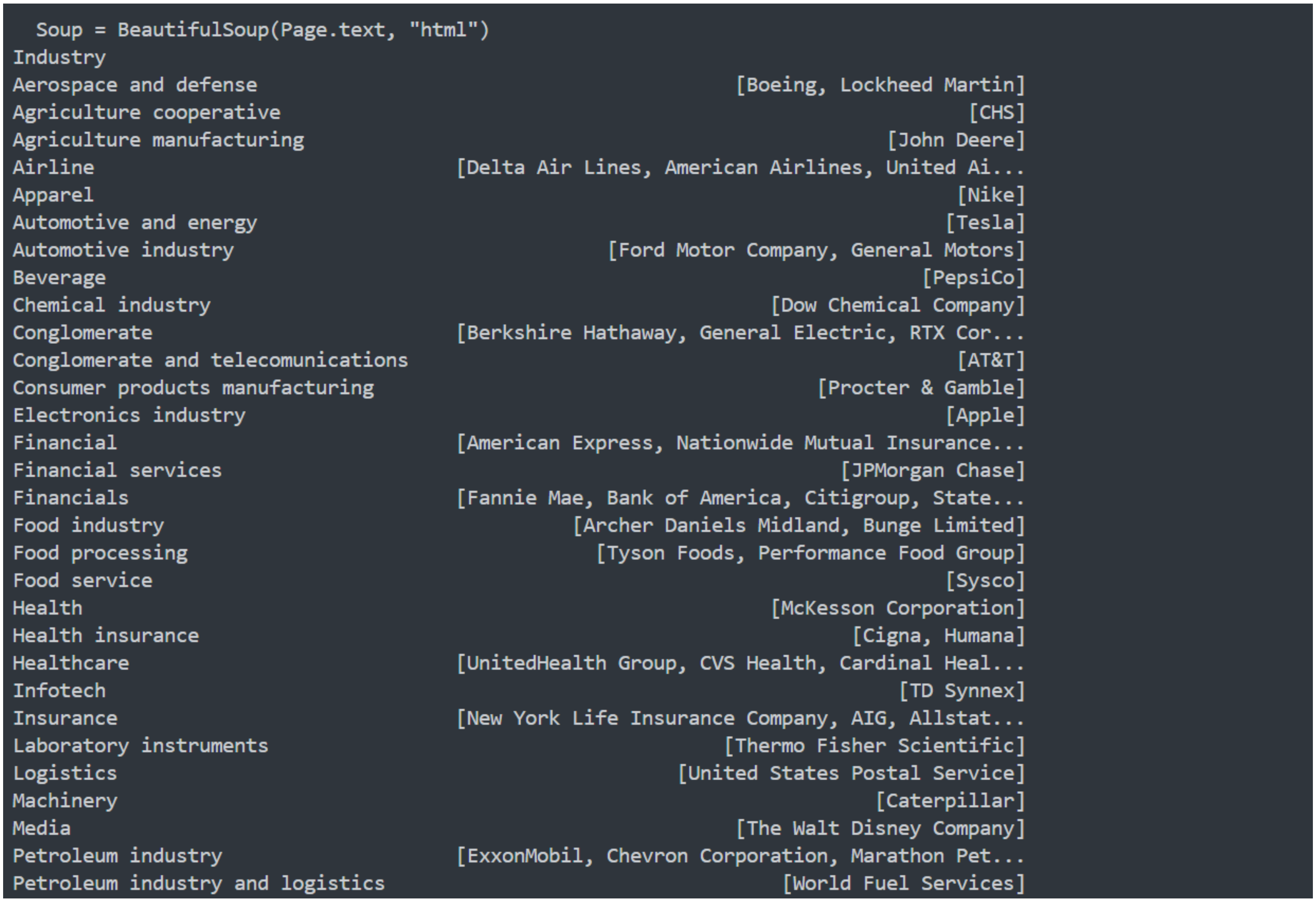
```
PS D:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python
> & "C:/Program Files/Python312/python.exe" "d:/Aryan Data/Usefull Data/Semester - 4/Data Visulization and Dashboards/La
b Manual/Exp-11 Web Scrapping using Python/InLab.py"
d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python\In
Lab.py:8: GuessedAtParserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for th
is system ("html.parser"). This usually isn't a problem, but if you run this code on another system, or in a different v
irtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 8 of the file d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and
 Dashboards\Lab Manual\Exp-11 Web Scrapping using Python\InLab.py. To get rid of this warning, pass the additional argum
ent 'features="html.parser"' to the BeautifulSoup constructor.

  Soup = BeautifulSoup(Page.text, "html")
                Company_Name  Share
0                    Walmart  5.20%
1                     Amazon  4.37%
2                 ExxonMobil  3.52%
3                      Apple  3.35%
4          UnitedHealth Group  2.76%
..                       ...    ...
95                  Best Buy  0.39%
96      Bristol-Myers Squibb  0.39%
97           United Airlines  0.38%
98  Thermo Fisher Scientific  0.38%
99                  Qualcomm  0.38%

[100 rows x 2 columns]
PS D:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python
> 
```

2. Segment companies based on industry to analyze sectors separately.

Code:-

Industry = Dataset.groupby("Industry")["Name"].unique()
print(Industry)

Results:-



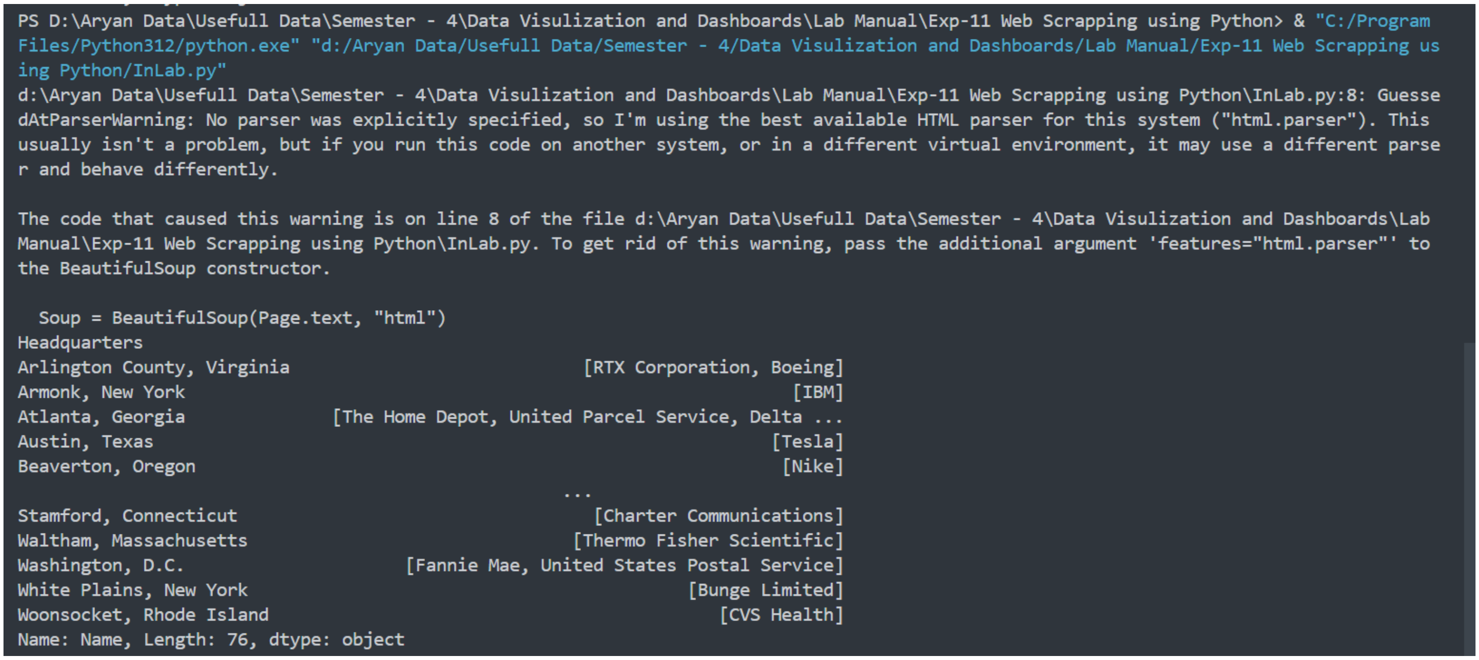
```
  Soup = BeautifulSoup(Page.text, "html")
Industry
Aerospace and defense                                         [Boeing, Lockheed Martin]
Agriculture cooperative                                                           [CHS]
Agriculture manufacturing                                                  [John Deere]
Airline                               [Delta Air Lines, American Airlines, United Ai...
Apparel                                                                          [Nike]
Automotive and energy                                                           [Tesla]
Automotive industry                                [Ford Motor Company, General Motors]
Beverage                                                                      [PepsiCo]
Chemical industry                                                [Dow Chemical Company]
Conglomerate                          [Berkshire Hathaway, General Electric, RTX Cor...
Conglomerate and telecomunications                                               [AT&T]
Consumer products manufacturing                                      [Procter & Gamble]
Electronics industry                                                            [Apple]
Financial                             [American Express, Nationwide Mutual Insurance...
Financial services                                                     [JPMorgan Chase]
Financials                            [Fannie Mae, Bank of America, Citigroup, State...
Food industry                                   [Archer Daniels Midland, Bunge Limited]
Food processing                                   [Tyson Foods, Performance Food Group]
Food service                                                                    [Sysco]
Health                                                           [McKesson Corporation]
Health insurance                                                        [Cigna, Humana]
Healthcare                            [UnitedHealth Group, CVS Health, Cardinal Heal...
Infotech                                                                    [TD Synnex]
Insurance                             [New York Life Insurance Company, AIG, Allstat...
Laboratory instruments                                       [Thermo Fisher Scientific]
Logistics                                                [United States Postal Service]
Machinery                                                                 [Caterpillar]
Media                                                         [The Walt Disney Company]
Petroleum industry                    [ExxonMobil, Chevron Corporation, Marathon Pet...
Petroleum industry and logistics                                  [World Fuel Services]
```

3. Segment companis based on headquarters to analyze sectors separately.

Code:-

Headquarters = Dataset.groupby("Headquarters")["Name"].unique()
print(Headquarters)

Results:-



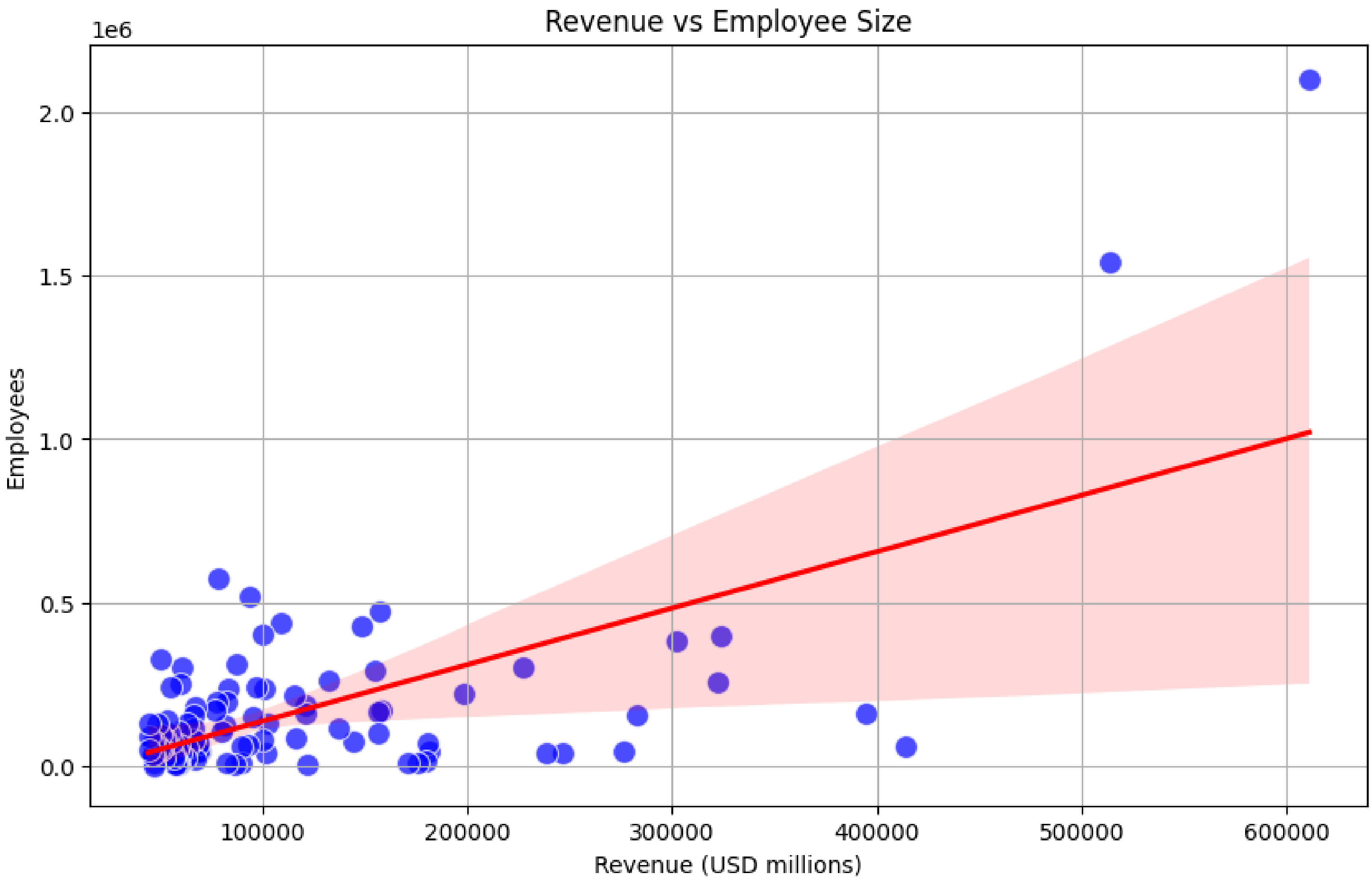
```
PS D:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python> & "C:/Program
Files/Python312/python.exe" "d:/Aryan Data/Usefull Data/Semester - 4/Data Visulization and Dashboards/Lab Manual/Exp-11 Web Scrapping us
ing Python/InLab.py"
d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python\InLab.py:8: Guesse
dAtParserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("html.parser"). This
usually isn't a problem, but if you run this code on another system, or in a different virtual environment, it may use a different parse
r and behave differently.

The code that caused this warning is on line 8 of the file d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab
Manual\Exp-11 Web Scrapping using Python\InLab.py. To get rid of this warning, pass the additional argument 'features="html.parser"' to
the BeautifulSoup constructor.

  Soup = BeautifulSoup(Page.text, "html")
Headquarters
Arlington County, Virginia                            [RTX Corporation, Boeing]
Armonk, New York                                                          [IBM]
Atlanta, Georgia              [The Home Depot, United Parcel Service, Delta ...
Austin, Texas                                                           [Tesla]
Beaverton, Oregon                                                        [Nike]
                                                    ...
Stamford, Connecticut                                  [Charter Communications]
Waltham, Massachusetts                               [Thermo Fisher Scientific]
Washington, D.C.                     [Fannie Mae, United States Postal Service]
White Plains, New York                                          [Bunge Limited]
Woonsocket, Rhode Island                                           [CVS Health]
Name: Name, Length: 76, dtype: object
```

4. Observe the growth of the company based on revenue vs employee size.

Code:-

```
revenue_employee_df = Dataset[['Revenue (USD millions)', 'Employees']]
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Revenue (USD millions)', y='Employees', data=revenue_employee_df, s=100, color='blue',
    alpha=0.7)
plt.title('Revenue vs Employee Size')
plt.xlabel('Revenue (USD millions)')
plt.ylabel('Employees')
plt.grid(True)
sns.regplot(x='Revenue (USD millions)', y='Employees', data=revenue_employee_df, scatter=False, color='red')
 plt.show()
```

Results:-



Observation and Result Analysis:-

Post Lab Exercise:

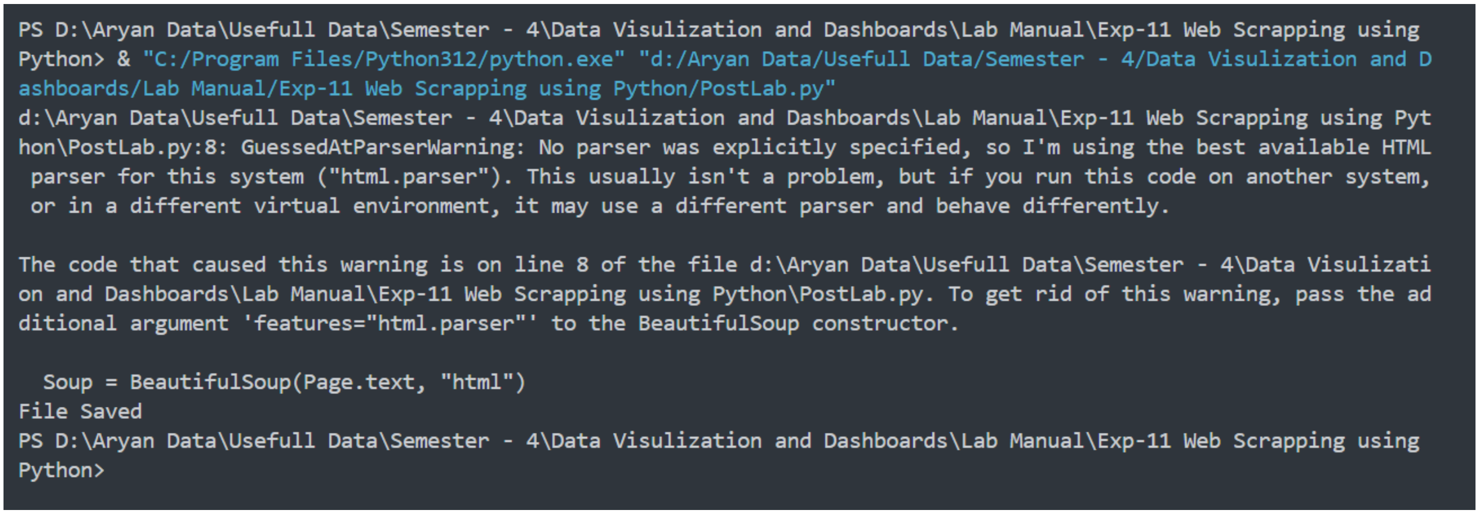
Exercise-1: Extract Table-1 from the weblink https://en.wikipedia.org/wiki/List_of_largest_companies_in_India

Code:-

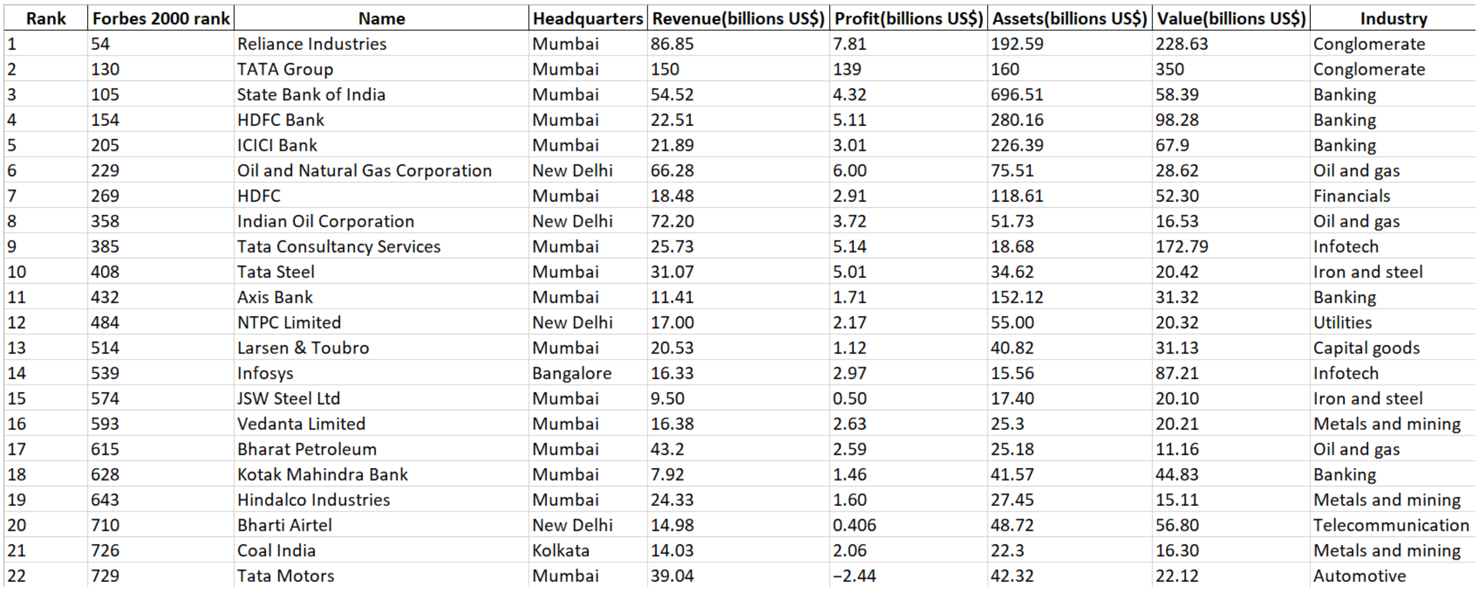
```
import pandas as pd
import requests
from bs4 import BeautifulSoup

url = "https://en.wikipedia.org/wiki/List_of_largest_companies_in_India"
Page = requests.get(url)

Soup = BeautifulSoup(Page.text, "html")
Table = Soup.find_all("table")

Table = Table[0]

Headers = Table.find_all("th")
Titles = [titles.text.strip() for titles in Headers]

Dataset = pd.DataFrame(columns=Titles)
Rows = Table.find_all("tr")

for row in Rows[1:]:
    Data = row.find_all("td")
    Data_filtered = [Data[i] for i in range(len(Data)) if i not in [1, 3]]
    Row_Data = [row.text.strip() for row in Data_filtered]
    Dataset.loc[len(Dataset)] = Row_Data

Dataset.to_excel("D:/Aryan Data/Usefull Data/Semester - 4/Data Visulization and Dashboards/Lab Manual/Exp-11 Web
Scrapping using Python/Largest Companies In India.xlsx",index=False,)
print("File Saved")
```

Marwadi University	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Data Visualization and Dashboard (01CT0410)	**Aim: Web Scrapping using Python**	
Experiment No: 11	**Date: 30-03-2024**	**Enrollment No: 92200133030**

Results:-



```
PS D:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using
Python> & "C:/Program Files/Python312/python.exe" "d:/Aryan Data/Usefull Data/Semester - 4/Data Visulization and D
ashboards/Lab Manual/Exp-11 Web Scrapping using Python/PostLab.py"
d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Pyt
hon\PostLab.py:8: GuessedAtParserWarning: No parser was explicitly specified, so I'm using the best available HTML
 parser for this system ("html.parser"). This usually isn't a problem, but if you run this code on another system,
 or in a different virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 8 of the file d:\Aryan Data\Usefull Data\Semester - 4\Data Visulizati
on and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python\PostLab.py. To get rid of this warning, pass the ad
ditional argument 'features="html.parser"' to the BeautifulSoup constructor.

  Soup = BeautifulSoup(Page.text, "html")
File Saved
PS D:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using
Python>
```



Rank	Forbes 2000 rank	Name	Headquarters	Revenue(billions US$)	Profit(billions US$)	Assets(billions US$)	Value(billions US$)	Industry
1	54	Reliance Industries	Mumbai	86.85	7.81	192.59	228.63	Conglomerate
2	130	TATA Group	Mumbai	150	139	160	350	Conglomerate
3	105	State Bank of India	Mumbai	54.52	4.32	696.51	58.39	Banking
4	154	HDFC Bank	Mumbai	22.51	5.11	280.16	98.28	Banking
5	205	ICICI Bank	Mumbai	21.89	3.01	226.39	67.9	Banking
6	229	Oil and Natural Gas Corporation	New Delhi	66.28	6.00	75.51	28.62	Oil and gas
7	269	HDFC	Mumbai	18.48	2.91	118.61	52.30	Financials
8	358	Indian Oil Corporation	New Delhi	72.20	3.72	51.73	16.53	Oil and gas
9	385	Tata Consultancy Services	Mumbai	25.73	5.14	18.68	172.79	Infotech
10	408	Tata Steel	Mumbai	31.07	5.01	34.62	20.42	Iron and steel
11	432	Axis Bank	Mumbai	11.41	1.71	152.12	31.32	Banking
12	484	NTPC Limited	New Delhi	17.00	2.17	55.00	20.32	Utilities
13	514	Larsen & Toubro	Mumbai	20.53	1.12	40.82	31.13	Capital goods
14	539	Infosys	Bangalore	16.33	2.97	15.56	87.21	Infotech
15	574	JSW Steel Ltd	Mumbai	9.50	0.50	17.40	20.10	Iron and steel
16	593	Vedanta Limited	Mumbai	16.38	2.63	25.3	20.21	Metals and mining
17	615	Bharat Petroleum	Mumbai	43.2	2.59	25.18	11.16	Oil and gas
18	628	Kotak Mahindra Bank	Mumbai	7.92	1.46	41.57	44.83	Banking
19	643	Hindalco Industries	Mumbai	24.33	1.60	27.45	15.11	Metals and mining
20	710	Bharti Airtel	New Delhi	14.98	0.406	48.72	56.80	Telecommunication
21	726	Coal India	Kolkata	14.03	2.06	22.3	16.30	Metals and mining
22	729	Tata Motors	Mumbai	39.04	−2.44	42.32	22.12	Automotive

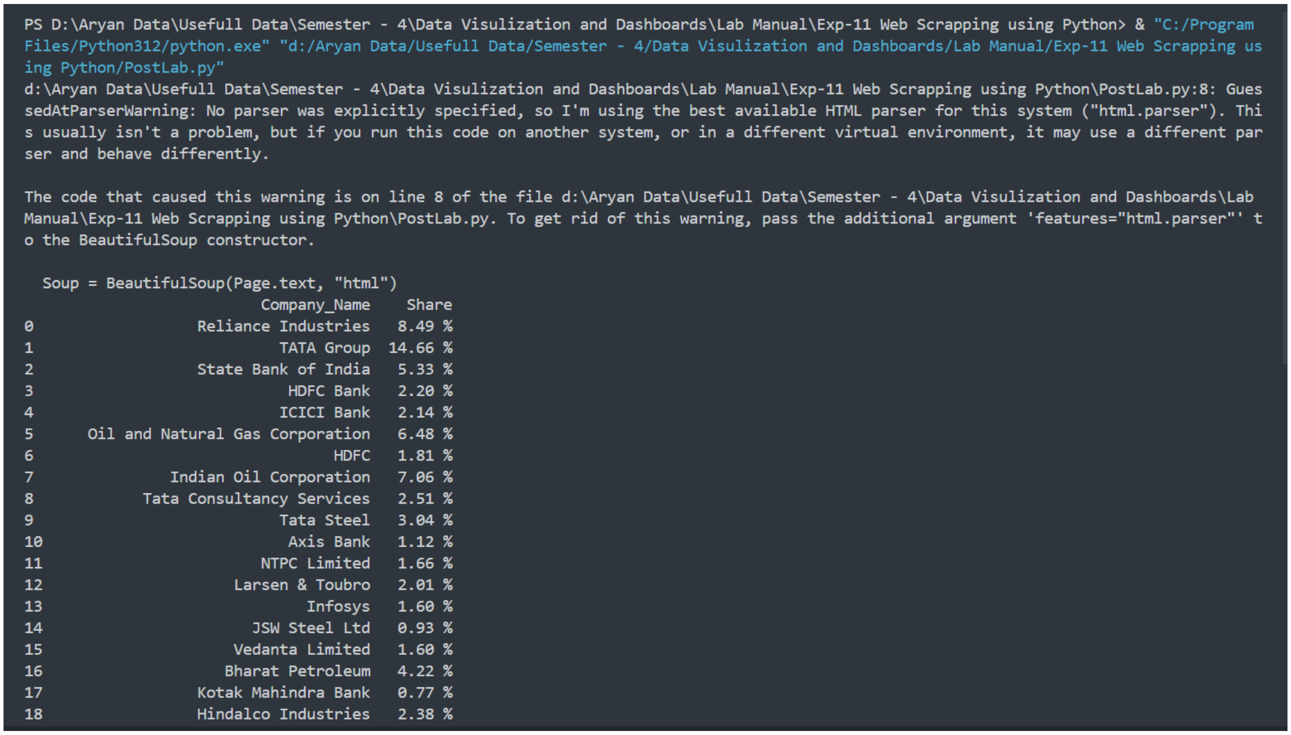
Task 2: Perform data analysis over:

1. Analyze the market share of different companies based on their revenue.

   Code :-

```
Dataset["Revenue(billions US$)"] = Dataset["Revenue(billions US$)"].astype(str)
Dataset["Revenue(billions US$)"] = (Dataset["Revenue(billions US$)"].str.replace(",", "").astype(float))
total_revenue = Dataset["Revenue(billions US$)"].sum()
Market_Share = {"Company_Name": [], "Share": []}
for index, row in Dataset.iterrows():
    company_name = row["Name"]
    revenue = row["Revenue(billions US$)"]
    share_percentage = (revenue / total_revenue) * 100
    Market_Share["Company_Name"].append(company_name)
    Market_Share["Share"].append(f"{share_percentage:.2f} %")
Market_Share = pd.DataFrame(Market_Share)
    print(Market_Share)
```

Result:-



```
PS D:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python> & "C:/Program
Files/Python312/python.exe" "d:/Aryan Data/Usefull Data/Semester - 4/Data Visulization and Dashboards/Lab Manual/Exp-11 Web Scrapping us
ing Python/PostLab.py"
d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python\PostLab.py:8: Gues
sedAtParserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("html.parser"). Thi
s usually isn't a problem, but if you run this code on another system, or in a different virtual environment, it may use a different par
ser and behave differently.

The code that caused this warning is on line 8 of the file d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab
Manual\Exp-11 Web Scrapping using Python\PostLab.py. To get rid of this warning, pass the additional argument 'features="html.parser"' t
o the BeautifulSoup constructor.

  Soup = BeautifulSoup(Page.text, "html")
                        Company_Name    Share
0                Reliance Industries   8.49 %
1                         TATA Group  14.66 %
2                State Bank of India   5.33 %
3                          HDFC Bank   2.20 %
4                         ICICI Bank   2.14 %
5       Oil and Natural Gas Corporation   6.48 %
6                               HDFC   1.81 %
7            Indian Oil Corporation   7.06 %
8         Tata Consultancy Services   2.51 %
9                         Tata Steel   3.04 %
10                         Axis Bank   1.12 %
11                      NTPC Limited   1.66 %
12                    Larsen & Toubro   2.01 %
13                           Infosys   1.60 %
14                     JSW Steel Ltd   0.93 %
15                   Vedanta Limited   1.60 %
16                  Bharat Petroleum   4.22 %
17               Kotak Mahindra Bank   0.77 %
18              Hindalco Industries   2.38 %
```

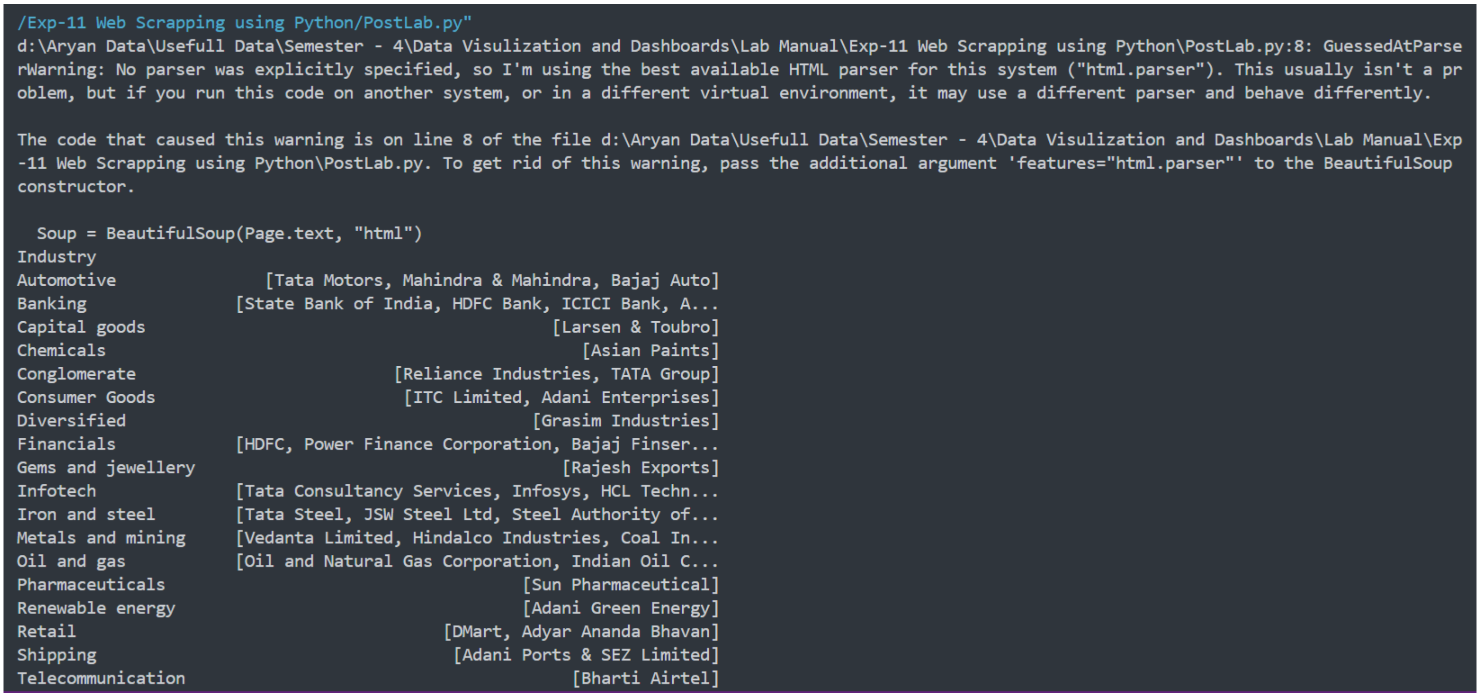
2. Segment companies based on industry to analyze industry sectors separately.

Code :-

Industry = Dataset.groupby("Industry")["Name"].unique()

print(Industry)

Result :-



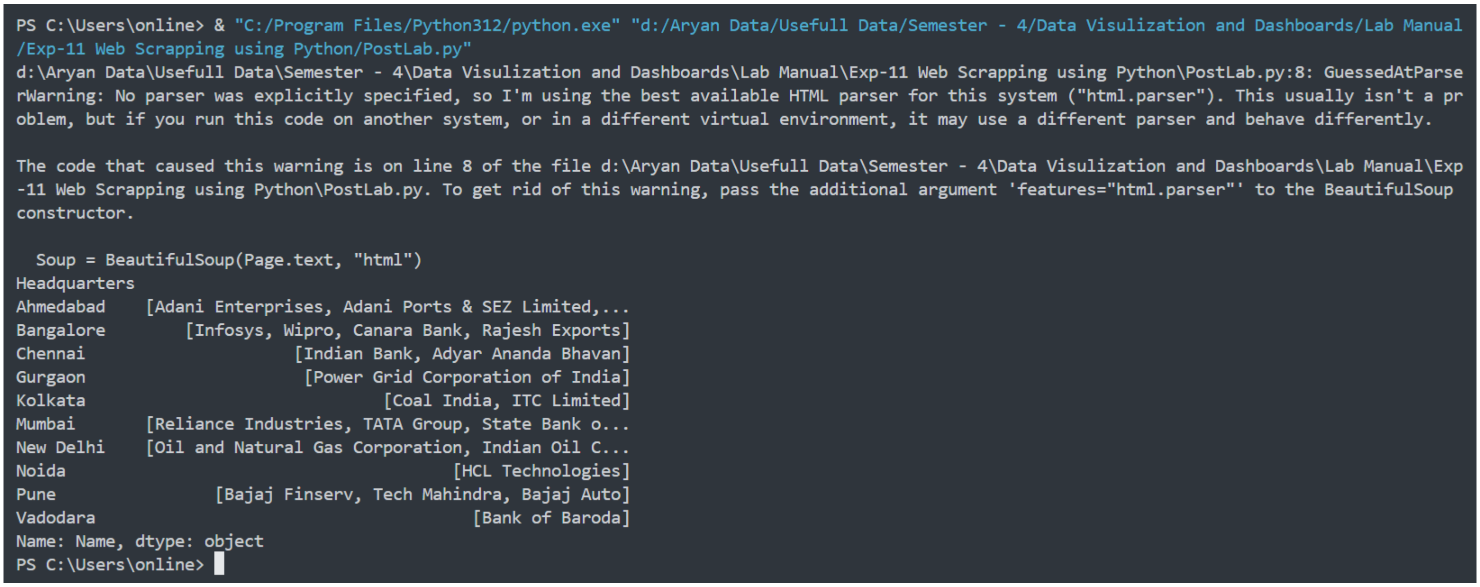
```
/Exp-11 Web Scrapping using Python/PostLab.py"
d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python\PostLab.py:8: GuessedAtParse
rWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("html.parser"). This usually isn't a pr
oblem, but if you run this code on another system, or in a different virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 8 of the file d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp
-11 Web Scrapping using Python\PostLab.py. To get rid of this warning, pass the additional argument 'features="html.parser"' to the BeautifulSoup
constructor.

  Soup = BeautifulSoup(Page.text, "html")
Industry
Automotive               [Tata Motors, Mahindra & Mahindra, Bajaj Auto]
Banking               [State Bank of India, HDFC Bank, ICICI Bank, A...
Capital goods                                         [Larsen & Toubro]
Chemicals                                                [Asian Paints]
Conglomerate                          [Reliance Industries, TATA Group]
Consumer Goods                         [ITC Limited, Adani Enterprises]
Diversified                                         [Grasim Industries]
Financials            [HDFC, Power Finance Corporation, Bajaj Finser...
Gems and jewellery                                     [Rajesh Exports]
Infotech              [Tata Consultancy Services, Infosys, HCL Techn...
Iron and steel        [Tata Steel, JSW Steel Ltd, Steel Authority of...
Metals and mining     [Vedanta Limited, Hindalco Industries, Coal In...
Oil and gas           [Oil and Natural Gas Corporation, Indian Oil C...
Pharmaceuticals                                    [Sun Pharmaceutical]
Renewable energy                                   [Adani Green Energy]
Retail                                     [DMart, Adyar Ananda Bhavan]
Shipping                                    [Adani Ports & SEZ Limited]
Telecommunication                                       [Bharti Airtel]
```

3. Segment companies based on headquarters to analyze sectors separately.

Code :-

Headquarters = Dataset.groupby("Headquarters")["Name"].unique()
print(Headquarters)

Result :-



```
PS C:\Users\online> & "C:/Program Files/Python312/python.exe" "d:/Aryan Data/Usefull Data/Semester - 4/Data Visulization and Dashboards/Lab Manual
/Exp-11 Web Scrapping using Python/PostLab.py"
d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp-11 Web Scrapping using Python\PostLab.py:8: GuessedAtParse
rWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("html.parser"). This usually isn't a pr
oblem, but if you run this code on another system, or in a different virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 8 of the file d:\Aryan Data\Usefull Data\Semester - 4\Data Visulization and Dashboards\Lab Manual\Exp
-11 Web Scrapping using Python\PostLab.py. To get rid of this warning, pass the additional argument 'features="html.parser"' to the BeautifulSoup
constructor.

  Soup = BeautifulSoup(Page.text, "html")
Headquarters
Ahmedabad    [Adani Enterprises, Adani Ports & SEZ Limited,...
Bangalore        [Infosys, Wipro, Canara Bank, Rajesh Exports]
Chennai                     [Indian Bank, Adyar Ananda Bhavan]
Gurgaon                      [Power Grid Corporation of India]
Kolkata                              [Coal India, ITC Limited]
Mumbai       [Reliance Industries, TATA Group, State Bank o...
New Delhi    [Oil and Natural Gas Corporation, Indian Oil C...
Noida                                        [HCL Technologies]
Pune                [Bajaj Finserv, Tech Mahindra, Bajaj Auto]
Vadodara                                       [Bank of Baroda]
Name: Name, dtype: object
PS C:\Users\online> 
```