In [ ]:
```
#Name-Nikhil Lalwani
#project_name- web scraping assignment 4
#company_name- FlipRobo
#Batch number- DS2307
```

In [147…
```
import selenium
from selenium import webdriver
import pandas as pd
from selenium.webdriver.common.by import By
import warnings
warnings.filterwarnings('ignore')
import time

from bs4 import BeautifulSoup
import pandas as pd
from tabulate import tabulate
import requests
import urllib.request
```

1. Scrape the details of most viewed videos on YouTube from Wikipedia. Url = https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos You need to find following details: A) Rank B) Name C) Artist D) Upload date E) Views

In [23]:
```
url='https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos'
r=requests.get(url)
htmlcontent=r.content
soup=BeautifulSoup(htmlcontent,'html.parser')

table_content=[]
table=soup.find_all('table',class_="wikitable")
for x in table:
    table_content.append(x.text)

df=pd.read_html(str(table))[0]

df=df.drop(df.index[-1])
df.index = df.index + 1
df.drop(df.columns[-1], axis=1, inplace=True)
df.drop(df.columns[-1], axis=1, inplace=True)
df
```

Out[23]:

	No.	Video name	Uploader	Views (billions)	Publication date
1	1.	"Baby Shark Dance"[6]	Pinkfong Baby Shark - Kids' Songs & Stories	13.36	June 17, 2016
2	2.	"Despacito"[9]	Luis Fonsi	8.26	January 12, 2017
3	3.	"Johny Johny Yes Papa"[16]	LooLoo Kids - Nursery Rhymes and Children's Songs	6.80	October 8, 2016
4	4.	"Bath Song"[17]	Cocomelon - Nursery Rhymes	6.41	May 2, 2018
5	5.	"Shape of You"[18]	Ed Sheeran	6.08	January 30, 2017
6	6.	"See You Again"[21]	Wiz Khalifa	6.03	April 6, 2015
7	7.	"Wheels on the Bus"[26]	Cocomelon - Nursery Rhymes	5.56	May 24, 2018
8	8.	"Phonics Song with Two Words"[27]	ChuChu TV Nursery Rhymes & Kids Songs	5.48	March 6, 2014
9	9.	"Uptown Funk"[28]	Mark Ronson	5.03	November 19, 2014
10	10.	"Learning Colors – Colorful Eggs on a Farm"[29]	Miroshka TV	4.97	February 27, 2018
11	11.	"Gangnam Style"[30]	officialpsy	4.90	July 15, 2012
12	12.	"Masha and the Bear – Recipe for Disaster"[35]	Get Movies	4.56	January 31, 2012
13	13.	"Dame Tu Cosita"[36]	Ultra Records	4.44	April 5, 2018
14	14.	"Axel F"[37]	Crazy Frog	4.06	June 16, 2009
15	15.	"Sugar"[38]	Maroon 5	3.93	January 14, 2015
16	16.	"Counting Stars"[39]	OneRepublic	3.87	May 31, 2013
17	17.	"Roar"[40]	Katy Perry	3.87	September 5, 2013
18	18.	"Baa Baa Black Sheep"[41]	Cocomelon - Nursery Rhymes	3.78	June 25, 2018
19	19.	"Waka Waka (This Time for Africa)"[42]	Shakira	3.73	June 4, 2010
20	20.	"Sorry"[43]	Justin Bieber	3.71	October 22, 2015
21	21.	"Lakdi Ki Kathi"[44]	Jingle Toons	3.69	June 14, 2018
22	22.	"Thinking Out Loud"[45]	Ed Sheeran	3.66	October 7, 2014
23	23.	"Dark Horse"[46]	Katy Perry	3.59	February 20, 2014
24	24.	"Humpty the train on a fruits ride"[47]	Kiddiestv Hindi - Nursery Rhymes & Kids Songs	3.55	January 26, 2018
25	25.	"Perfect"[48]	Ed Sheeran	3.54	November 9,

	No.	Video name	Uploader	Views (billions)	Publication date
					2017
26	26.	"Faded"[49]	Alan Walker	3.51	December 3, 2015
27	27.	"Let Her Go"[50]	Passenger	3.51	July 25, 2012
28	28.	"Girls Like You"[51]	Maroon 5	3.48	May 31, 2018
29	29.	"Lean On"[52]	Major Lazer Official	3.46	March 22, 2015
30	30.	"Bailando"[53]	Enrique Iglesias	3.45	April 11, 2014

2. Scrape the details team India's international fixtures from bcci.tv. Url = https://www.bcci.tv/. You need to find following details: A) Match title (I.e. 1 ODI) B) Series C) Place D) Date E) Time

In [24]:
```
driver=webdriver.Chrome()
```

In [25]:
```
driver.get('https://www.bcci.tv/')
```

In [37]:
```
click_international=driver.find_element(By.XPATH,'/html/body/nav/div[1]/div[2]/ul[1]/l
click_international.click()
```

In [52]:
```
Match_title=[]
title=driver.find_elements(By.XPATH,'//h5[@class="match-tournament-name ng-binding"]')
for x in title:
    Match_title.append(x.text)
```

In [58]:
```
Match_series=[]
series=driver.find_elements(By.XPATH,'//div[@class="match-card-middle__inner d-flex ju
for x in series:
    Match_series.append(x.text)
```

In [54]:
```
Match_place=[]
place=driver.find_elements(By.XPATH,'//span[@class="ng-binding"]')
for x in place:
    Match_place.append(x.text)
```

In [55]:
```
Match_date=[]
date=driver.find_elements(By.XPATH,'//div[@class="match-dates ng-binding"]')
for x in date:
    Match_date.append(x.text)
```

In [56]:
```
Match_time=[]
time=driver.find_elements(By.XPATH,'//div[@class="match-time no-margin ng-binding"]')
for x in time:
    Match_time.append(x.text)
```

In [59]:
```
df=pd.DataFrame({'Match_title':Match_title,'Match_series':Match_series,'Match_place':M
df
```

Out[59]:

	Match_title	Match_series	Match_place	Match_date	Match_time
0	ASIA CUP 2023	India\nvs\nSri Lanka	Colombo	17 SEP 2023	3:00 PM IST
1	19TH ASIAN GAMES 2023	India Women\nvs\nTBD	Hangzhou	21 SEP 2023	6:30 AM IST
2	AUSTRALIA TOUR OF INDIA 2023-24	India\nvs\nAustralia	Mohali	22 SEP 2023	1:30 PM IST
3	AUSTRALIA TOUR OF INDIA 2023-24	India\nvs\nAustralia	Indore	24 SEP 2023	1:30 PM IST
4	AUSTRALIA TOUR OF INDIA 2023-24	India\nvs\nAustralia	Rajkot	27 SEP 2023	1:30 PM IST
5	ICC MENS WORLD CUP 2023 WARM-UP MATCHES	India\nvs\nEngland	Guwahati	30 SEP 2023	2:00 PM IST
6	19TH ASIAN GAMES 2023	India\nvs\nTBD	Hangzhou	3 OCT 2023	6:30 AM IST
7	ICC MENS WORLD CUP 2023 WARM-UP MATCHES	India\nvs\nNetherlands	Thiruvananthapuram	3 OCT 2023	2:00 PM IST

3. Scrape the details of State-wise GDP of India from statisticstime.com. Url = http://statisticstimes.com/ You have to find following details: A) Rank B) State C) GSDP(18-19)- at current prices D) GSDP(19-20)- at current prices E) Share(18-19) F) GDP($ billion)

In [74]:
```
url='https://www.statisticstimes.com/economy/india/indian-states-gdp.php'
r=requests.get(url)
htmlcontent=r.content
soup=BeautifulSoup(htmlcontent,'html.parser')

table_content=[]
table=soup.find_all('table',class_='display')
for x in table:
    table_content.append(x.text)

df=pd.read_html(str(table))[0]
df=df.drop(df.index[-1])
df.index = df.index + 1
df
```

Out[74]:

	Rank	State	GSDP (Cr INR at Current prices)		Share	GDP ($billion)	GSDP (Cr INR at 2011-12 prices)	
	Rank	State	19-20	18-19	18-19	2019	19-20	18-19
1	1.0	Maharashtra	-	2632792	13.94%	399.921	-	2039074
2	2.0	Tamil Nadu	1845853	1630208	8.63%	247.629	1312929	1215307
3	3.0	Uttar Pradesh	1687818	1584764	8.39%	240.726	1166817	1123982
4	4.0	Gujarat	-	1502899	7.96%	228.290	-	1186379
5	5.0	Karnataka	1631977	1493127	7.91%	226.806	1156039	1091077
6	6.0	West Bengal	1253832	1089898	5.77%	165.556	793223	739525
7	7.0	Rajasthan	1020989	942586	4.99%	143.179	711627	677428
8	8.0	Andhra Pradesh	972782	862957	4.57%	131.083	672018	621301
9	9.0	Telangana	969604	861031	4.56%	130.791	663258	612828
10	10.0	Madhya Pradesh	906672	809592	4.29%	122.977	561801	522009
11	11.0	Kerala	-	781653	4.14%	118.733	-	559412
12	12.0	Delhi	856112	774870	4.10%	117.703	634408	590569
13	13.0	Haryana	831610	734163	3.89%	111.519	572240	531085
14	14.0	Bihar	611804	530363	2.81%	80.562	414977	375651
15	15.0	Punjab	574760	526376	2.79%	79.957	418868	397669
16	16.0	Odisha	521275	487805	2.58%	74.098	396499	376877
17	17.0	Assam	-	315881	1.67%	47.982	-	234048
18	18.0	Chhattisgarh	329180	304063	1.61%	46.187	243477	231182
19	19.0	Jharkhand	328598	297204	1.57%	45.145	240036	224986
20	20.0	Uttarakhand	-	245895	1.30%	37.351	-	193273
21	21.0	Jammu & Kashmir	-	155956	0.83%	23.690	-	112755
22	22.0	Himachal Pradesh	165472	153845	0.81%	23.369	124403	117851
23	23.0	Goa	80449	73170	0.39%	11.115	63408	57787
24	24.0	Tripura	55984	49845	0.26%	7.571	40583	36963
25	25.0	Chandigarh	-	42114	0.22%	6.397	-	31192
26	26.0	Puducherry	38253	34433	0.18%	5.230	25093	23013
27	27.0	Meghalaya	36572	33481	0.18%	5.086	26695	24682
28	28.0	Sikkim	32496	28723	0.15%	4.363	20017	18722
29	29.0	Manipur	31790	27870	0.15%	4.233	20673	19300
30	30.0	Nagaland	-	27283	0.14%	4.144	-	17647
31	31.0	Arunachal Pradesh	-	24603	0.13%	3.737	-	16676
32	32.0	Mizoram	26503	22287	0.12%	3.385	18797	16478

	Rank	State	GSDP (Cr INR at Current prices)		Share	GDP ($billion)	GSDP (Cr INR at 2011-12 prices)	
	Rank	State	19-20	18-19	18-19	2019	19-20	18-19
33	33.0	Andaman & Nicobar Islands	-	-	-	-	-	-

4. Scrape the details of trending repositories on Github.com. Url = https://github.com/ You have to find the following details: A) Repository title B) Repository description C) Contributors count D) Language used

In [76]:
```
driver=webdriver.Chrome()
```

In [80]:
```
driver.get('https://github.com/')
```

In [81]:
```
click_trending=driver.find_element(By.XPATH,'/html/body/div[1]/div[1]/header/div/div[2
click_trending.click()
```

In [87]:
```
repository_title=[]
title=driver.find_elements(By.XPATH,'//h2[@class="h3 lh-condensed"]')
for x in title:
    repository_title.append(x.text)
```

In [88]:
```
repository_description=[]
description=driver.find_elements(By.XPATH,'//p[@class="col-9 color-fg-muted my-1 pr-4'
for x in description:
    repository_description.append(x.text)
```

In [93]:
```
Contributors_count=[]
count=driver.find_elements(By.XPATH,'//span[@class="d-inline-block float-sm-right"]')
for x in count:
    Contributors_count.append(x.text)
```

In [91]:
```
Language_used=[]
language=driver.find_elements(By.XPATH,'//span[@itemprop="programmingLanguage"]')
for x in language:
    Language_used.append(x.text)
```

In [94]:
```
df=pd.DataFrame({'repository_title':repository_title,'repository_description':reposito
df
```

Out[94]:

	repository_title	repository_description	Contributors_count	Language_used
0	coqui-ai / TTS	🐸💬 - a deep learning toolkit for Text-to-Speec...	164 stars today	Python
1	aiwaves-cn / agents	An Open-source Framework for Autonomous Langua...	80 stars today	Python
2	isocpp / CppCoreGuidelines	The C++ Core Guidelines are a set of tried-and...	35 stars today	Python
3	godotengine / godot	Godot Engine – Multi-platform 2D and 3D game e...	625 stars today	C++
4	godotengine / godot-docs	Godot Engine official documentation	37 stars today	reStructuredText
5	FlaxEngine / FlaxEngine	Flax Engine – multi-platform 3D game engine	123 stars today	C++
6	yoheinakajima / instagraph	Converts text input or URL into knowledge grap...	743 stars today	Python
7	stride3d / stride	Stride Game Engine (formerly Xenko)	43 stars today	C#
8	nuejs / nuejs	Build user interfaces with 10x less code. Alte...	154 stars today	JavaScript
9	raysan5 / raylib	A simple and easy-to-use library to enjoy vide...	40 stars today	C
10	godotengine / godot-cpp	C++ bindings for the Godot script API	9 stars today	C++
11	sxyazi / yazi	⚡ Blazing fast terminal file manager written ...	244 stars today	Rust
12	lodash / lodash	A modern JavaScript utility library delivering...	78 stars today	JavaScript
13	TheCherno / Hazel	Hazel Engine	18 stars today	C++
14	kholia / OSX-KVM	Run macOS on QEMU/KVM. With OpenCore + Big Sur...	10 stars today	Python
15	tldraw / tldraw	a very good whiteboard	873 stars today	TypeScript
16	bevyengine / bevy	A refreshingly simple data-driven game engine ...	39 stars today	Rust
17	DroidKaigi / conference-app-2023	The Official Conference App for DroidKaigi 2023	3 stars today	Kotlin
18	godotengine / godot-demo-projects	Demonstration and Template Projects	93 stars today	GDScript
19	MarlinFirmware / Marlin	Marlin is an optimized firmware for RepRap 3D ...	5 stars today	C++
20	ethereum-lists / chains	provides metadata for chains	6 stars today	Kotlin
21	commaai / openpilot	openpilot is an open source driver assistance ...	7 stars today	Python
22	public-apis / public-apis	A collective list of free APIs	68 stars today	Python

	repository_title	repository_description	Contributors_count	Language_used
23	Kr328 / ClashForAndroid	A rule-based tunnel for Android.	26 stars today	Kotlin
24	JetBrains / compose-multiplatform	Compose Multiplatform, a modern UI framework f...	13 stars today	Kotlin

5. Scrape the details of top 100 songs on billiboard.com. Url = https:/www.billboard.com/ You have to find the following details: A) Song name B) Artist name C) Last week rank D) Peak rank E) Weeks on board

In [95]:
```
driver=webdriver.Chrome()
```

In [96]:
```
driver.get('https://www.billboard.com/')
```

In [97]:
```
click_chart=driver.find_element(By.XPATH,'/html/body/div[3]/div[9]/div/div/div/ul/li[1
click_chart.click()
```

In [98]:
```
click_chart_2=driver.find_element(By.XPATH,'/html/body/div[3]/div[9]/div/div/div/ul/li
click_chart_2.click()
```

In [99]:
```
Song_name=[]
song=driver.find_elements(By.XPATH,'//h3[@class="c-title  a-no-trucate a-font-primary-
for x in song:
    Song_name.append(x.text)
```

In [100…
```
Artist_name=[]
name=driver.find_elements(By.XPATH,'//span[@class="c-label  a-no-trucate a-font-primar
for x in name:
    Artist_name.append(x.text)
```

In [106…
```
df=pd.DataFrame({'Song_name':Song_name,'Artist_name':Artist_name})
df
```

Out[106]:

	Song_name	Artist_name
0	I Remember Everything	Zach Bryan Featuring Kacey Musgraves
1	Fast Car	Luke Combs
2	Cruel Summer	Taylor Swift
3	Last Night	Morgan Wallen
4	Dance The Night	Dua Lipa
...	...	...
94	Stand By Me	Lil Durk Featuring Morgan Wallen
95	Call Your Friends	Rod Wave
96	Your Heart Or Mine	Jon Pardi
97	Primera Cita	Carin Leon
98	S91	Karol G

99 rows × 2 columns

6. Scrape the details of Highest selling novels. compare A) Book name B) Author name C) Volumes sold D) Publisher E) Genre

In [110…

```
url='https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time
r=requests.get(url)
htmlcontent=r.content
soup=BeautifulSoup(htmlcontent,'html.parser')

table_contnt=[]
table=soup.find_all('table',class_="in-article sortable")
for x in table:
    table_contnt.append(x.text)

df=pd.read_html(str(table))[0]
df=df.drop(df.index[-1])
df.index = df.index + 1
df
```

Out[110]:

	Rank	Title	Author	Volume Sales	Publisher	Genre
1	1	Da Vinci Code,The	Brown, Dan	5094805	Transworld	Crime, Thriller & Adventure
2	2	Harry Potter and the Deathly Hallows	Rowling, J.K.	4475152	Bloomsbury	Children's Fiction
3	3	Harry Potter and the Philosopher's Stone	Rowling, J.K.	4200654	Bloomsbury	Children's Fiction
4	4	Harry Potter and the Order of the Phoenix	Rowling, J.K.	4179479	Bloomsbury	Children's Fiction
5	5	Fifty Shades of Grey	James, E. L.	3758936	Random House	Romance & Sagas
...	...	...	...	...	...	...
96	96	Ghost,The	Harris, Robert	807311	Random House	General & Literary Fiction
97	97	Happy Days with the Naked Chef	Oliver, Jamie	794201	Penguin	Food & Drink: General
98	98	Hunger Games,The:Hunger Games Trilogy	Collins, Suzanne	792187	Scholastic Ltd.	Young Adult Fiction
99	99	Lost Boy,The:A Foster Child's Search for the L...	Pelzer, Dave	791507	Orion	Biography: General
100	100	Jamie's Ministry of Food:Anyone Can Learn to C...	Oliver, Jamie	791095	Penguin	Food & Drink: General

100 rows × 6 columns

7. Scrape the details most watched tv series of all time from imdb.com. Url = https://www.imdb.com/list/ls095964455/ You have to find the following details: A) Name B) Year span C) Genre D) Run time E) Ratings F) Votes

In [111…
```
driver=webdriver.Chrome()
```

In [112…
```
driver.get('https://www.imdb.com/list/ls095964455/')
```

In [113…
```
series_name=[]
name=driver.find_elements(By.XPATH,'//h3[@class="lister-item-header"]')
for x in name:
    series_name.append(x.text)
```

In [114…
```
Year_span=[]
year=driver.find_elements(By.XPATH,'//span[@class="lister-item-year text-muted unbold'
for x in year:
    Year_span.append(x.text)
```

In [116…
```
series_Genre=[]
Genre=driver.find_elements(By.XPATH,'//span[@class="genre"]')
for x in Genre:
    series_Genre.append(x.text)
```

In [117…
```
series_Run_time=[]
Run_time=driver.find_elements(By.XPATH,'//span[@class="runtime"]')
for x in Run_time:
    series_Run_time.append(x.text)
```

In [118…
```
series_Ratings=[]
Ratings=driver.find_elements(By.XPATH,'//div[@class="ipl-rating-star small"]')
for x in Ratings:
    series_Ratings.append(x.text)
```

In [119…
```
series_votes=[]
votes=driver.find_elements(By.XPATH,'//span[@name="nv"]')
for x in votes:
    series_votes.append(x.text)
```

In [120…
```
df=pd.DataFrame({'series_name':series_name,'Year_span':Year_span,'series_Genre':series
df
```

Out[120]:

	series_name	Year_span	series_Genre	series_Run_time	series_Ratings	series_votes
0	1. Game of Thrones (2011–2019)	(2011–2019)	Action, Adventure, Drama	57 min	9.2	2,203,961
1	2. Stranger Things (2016–2025)	(2016–2025)	Drama, Fantasy, Horror	51 min	8.7	1,275,570
2	3. The Walking Dead (2010–2022)	(2010–2022)	Drama, Horror, Thriller	44 min	8.1	1,045,781
3	4. 13 Reasons Why (2017–2020)	(2017–2020)	Drama, Mystery, Thriller	60 min	7.5	307,335
4	5. The 100 (2014–2020)	(2014–2020)	Drama, Mystery, Sci-Fi	43 min	7.6	266,435
...	...	...	...	...	...	...
95	96. Reign (2013–2017)	(2013–2017)	Drama	42 min	7.5	52,725
96	97. A Series of Unfortunate Events (2017–2019)	(2017–2019)	Adventure, Comedy, Drama	50 min	7.8	64,776
97	98. Criminal Minds (2005–)	(2005–)	Crime, Drama, Mystery	42 min	8.1	211,022
98	99. Scream: The TV Series (2015–)	(2015–)	Comedy, Crime, Drama	45 min	7	43,897
99	100. The Haunting of Hill House (2018)	(2018)	Drama, Horror, Mystery	572 min	8.6	265,936

100 rows × 6 columns

8. Details of Datasets from UCI machine learning repositories. Url = https://archive.ics.uci.edu/You have to find the following details: A) Dataset name B) Data type C) Task D) Attribute type E) No of instances F) No of attribute G) Year

In [148…
```
driver=webdriver.Chrome()
```

In [150…
```
driver.get('https://archive.ics.uci.edu/datasets')
```

In [156…
```
elements Not scrapable
```

In [ ]:
```
-------------------------------------------------END-----------------------------------
```