In [109]:
```
import selenium
from selenium import webdriver
import time
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup
import requests

import pandas as pd

#importing required exceptions
from selenium.common.exceptions import StaleElementReferenceException , NoSuchElementException

from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
```

In [ ]:
```

```

In [112]:
```
driver = webdriver.Chrome()
```

In [ ]:
```
'''Q1) 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 [9]:
```
driver.get('https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos')
```

In [10]:
```
video_name = []
Name = driver.find_elements(By.XPATH , '//table[@class="sortable wikitable sticky-header static-row-numbers sort-under col3center col4right jquery-tablesorter"]/tbody/tr/td[1]
for name in Name :
    NAME = name.text
    video_name.append(NAME)
len(video_name)
```

Out[10]: 30

In [21]:
```
Artist_Name = []
artist = driver.find_elements(By.XPATH , '//table[@class="sortable wikitable sticky-header static-row-numbers sort-under col3center col4right jquery-tablesorter"]/tbody/tr/td[
for art in artist :
    artst = art.text
    Artist_Name.append(artst)
len(Artist_Name)
```

Out[21]: 30

In [13]:
```
Upload_Date = []
date = driver.find_elements(By.XPATH , '//table[@class="sortable wikitable sticky-header static-row-numbers sort-under col3center col4right jquery-tablesorter"]/tbody/tr/td[4]
for tarikh in date :
    Date = tarikh.text
    Upload_Date.append(Date)
len(Upload_Date)
```

Out[13]: 30

In [14]:
```
Views = []
view = driver.find_elements(By.XPATH , '//table[@class="sortable wikitable sticky-header static-row-numbers sort-under col3center col4right jquery-tablesorter"]/tbody/tr/td[3]
for see in view :
    v = see.text
    Views.append(v)
len(Views)
```

Out[14]: 30

In [24]:
```
details = {
    'VIDEO NAME' : video_name ,
    'ARTIST NAME' : Artist_Name ,
    'UPLOAD DATE' : Upload_Date ,
    'VIEWS' : Views
}

df = pd.DataFrame(details)
df['RANK'] = range(1, len(df) + 1)
df.set_index('RANK', inplace=True)
df
```

Out[24]:

RANK	VIDEO NAME	ARTIST NAME	UPLOAD DATE	VIEWS
1	"Baby Shark Dance"[6]	Pinkfong Baby Shark - Kids' Songs & Stories	June 17, 2016	14.09
2	"Despacito"[9]	Luis Fonsi	January 12, 2017	8.38
3	"Johny Johny Yes Papa"[17]	LooLoo Kids - Nursery Rhymes and Children's Songs	October 8, 2016	6.87
4	"Bath Song"[18]	Cocomelon - Nursery Rhymes	May 2, 2018	6.62
5	"Shape of You"[19]	Ed Sheeran	January 30, 2017	6.20
6	"See You Again"[22]	Wiz Khalifa	April 6, 2015	6.17
7	"Wheels on the Bus"[27]	Cocomelon - Nursery Rhymes	May 24, 2018	5.88
8	"Phonics Song with Two Words"[28]	ChuChu TV Nursery Rhymes & Kids Songs	March 6, 2014	5.70
9	"Uptown Funk"[29]	Mark Ronson	November 19, 2014	5.15
10	"Learning Colors – Colorful Eggs on a Farm"[30]	Miroshka TV	February 27, 2018	5.07
11	"Gangnam Style"[31]	Psy	July 15, 2012	5.05
12	"Masha and the Bear – Recipe for Disaster"[36]	Get Movies	January 31, 2012	4.58
13	"Dame Tu Cosita"[37]	Ultra Records	April 5, 2018	4.55
14	"Axel F"[38]	Crazy Frog	June 16, 2009	4.34
15	"Sugar"[39]	Maroon 5	January 14, 2015	4.00
16	"Counting Stars"[40]	OneRepublic	May 31, 2013	3.97
17	"Baa Baa Black Sheep"[41]	Cocomelon - Nursery Rhymes	June 25, 2018	3.96
18	"Roar"[42]	Katy Perry	September 5, 2013	3.96
19	"Lakdi Ki Kathi"[43]	Jingle Toons	June 14, 2018	3.91
20	"Waka Waka (This Time for Africa)"[44]	Shakira	June 4, 2010	3.85
21	"Sorry"[45]	Justin Bieber	October 22, 2015	3.77
22	"Thinking Out Loud"[46]	Ed Sheeran	October 7, 2014	3.73
23	"Humpty the train on a fruits ride"[47]	Kiddiestv Hindi - Nursery Rhymes & Kids Songs	January 26, 2018	3.73
24	"Shree Hanuman Chalisa"[48]	T-Series Bhakti Sagar	May 10, 2011	3.69
25	"Dark Horse"[49]	Katy Perry	February 20, 2014	3.67
26	"Perfect"[50]	Ed Sheeran	November 9, 2017	3.67
27	"Let Her Go"[51]	Passenger	July 25, 2012	3.61
28	"Faded"[52]	Alan Walker	December 3, 2015	3.59
29	"Girls Like You"[53]	Maroon 5	May 31, 2018	3.56
30	"Lean On"[54]	Major Lazer Official	March 22, 2015	3.55

In [ ]:
```
''' Q2) Scrape the details team India’s international fixtures from bcci.tv.
Url = https://www.bcci.tv/.
You need to find following details:
A) Series
B) Place
C) Date
D) Time
Note: - From bcci.tv home page you have reach to the international fixture page through code.'''
```

In [26]:
```
driver.get('https://www.bcci.tv/')
```

In [27]:
```
search = driver.find_element(By.XPATH , '/html/body/header/div[3]/div[2]/ul/div[1]/a[2]')
search.click()
```

In [29]:
```
Series = []
series = driver.find_elements(By.XPATH , '//div[@class="match-info"]/h5')
for ser in series :
    tour = ser.text
    Series.append(tour)
len(Series)
```

Out[29]: 6

In [30]:
```
Place = []
venue = driver.find_elements(By.XPATH , '//div[@class="match-place ng-scope"]/span[2]')
for i in venue :
    place = i.text
    Place.append(place)
len(Place)
```

Out[30]: 6

In [31]:
```
Stadium = []
stad = driver.find_elements(By.XPATH , '//div[@class="match-place ng-scope"]/span[1]')
for x in stad :
    stadium = x.text
    Stadium.append(stadium)
len(Stadium)
```

Out[31]: 6

In [32]:
```
Date = []
date = driver.find_elements(By.XPATH , '//div[@class="match-date-info"]/div[1]')
for tarikh in date :
    a = tarikh.text
    Date.append(a)
len(Date)
```

Out[32]: 6

In [33]:
```
Time = []
time = driver.find_elements(By.XPATH , '//div[@class="match-date-info"]/div[2]')
for t in time :
    TIME = t.text
    Time.append(TIME)
len(Time)
```

Out[33]: 6

In [34]:
```
details = {
    'SERIES' : Series ,
    'STADIUM' : Stadium ,
    'PLACE' : Place ,
    'DATE' : Date ,
    'TIME' : Time
}
df = pd.DataFrame(details)
df
```

Out[34]:

	SERIES	STADIUM	PLACE	DATE	TIME
0	ENGLAND TOUR OF INDIA 2023-24	Himachal Pradesh Cricket Association Stadium,	Dharamsala	7 MARCH, 2024	9:30 AM IST
1	INDIA TOUR OF ZIMBABWE 2024	Harare Sports Club,	Harare	6 JULY, 2024	8:00 PM IST
2	INDIA TOUR OF ZIMBABWE 2024	Harare Sports Club,	Harare	7 JULY, 2024	8:00 PM IST
3	INDIA TOUR OF ZIMBABWE 2024	Harare Sports Club,	Harare	10 JULY, 2024	8:00 PM IST
4	INDIA TOUR OF ZIMBABWE 2024	Harare Sports Club,	Harare	13 JULY, 2024	8:00 PM IST
5	INDIA TOUR OF ZIMBABWE 2024	Harare Sports Club,	Harare	14 JULY, 2024	8:00 PM IST

In [ ]:
```
''' Q3) 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)
Note: - From statisticstimes home page you have to reach to economy page through code. '''
```

In [36]:
```
driver.get('https://statisticstimes.com/')
```

In [39]:
```
economy_heading = driver.find_element(By.XPATH , '/html/body/div[2]/div[1]/div[2]/div[2]/button')
economy_heading.click()
```

In [40]:
```
search = driver.find_element(By.XPATH , '/html/body/div[2]/div[1]/div[2]/div[2]/div/a[3]')
search.click()
```

In [41]:
```
gdp_state_search = driver.find_element(By.XPATH , '/html/body/div[2]/div[2]/div[2]/ul/li[1]/a')
gdp_state_search.click()
```

In [ ]:
```
gdp_table = driver.find_element(By.CLASS_NAME, 'display.nowrap')

Extracting the data from the table
data = []
for row in gdp_table.find_elements(By.TAG_NAME, 'tr')[1:]:  # Skip the header row
    columns = row.find_elements(By.TAG_NAME, 'td')
    rank, state, gsdp_1819, gsdp_1920, share_1819, gdp_billion = [col.text.strip() for col in columns[0:6]]
    data.append([rank, state, gsdp_1819, gsdp_1920, share_1819, gdp_billion])

Creating a DataFrame
columns = ['Rank', 'State', 'GSDP(18-19) at current prices', 'GSDP(19-20) at current prices', 'Share(18-19)', 'GDP($ billion)']
df = pd.DataFrame(data, columns=columns)

Display the DataFrame
print(df)
```

In [ ]:
```
'''Q4) 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 [43]:
```
driver.get('https://github.com/DravinMishra/MyCompleteProjects')
```

In [55]:
```
repo = driver.find_element(By.XPATH , '/html/body/div[1]/div[4]/div/main/div/div[1]/div[1]/div/strong/a').text
print(repo)
repo_description = driver.find_element(By.XPATH , '/html/body/div[1]/div[4]/div/main/turbo-frame/div/div/div/div[2]/div[2]/div/div[1]/div/div/div[1]').text
print(repo_description)
language_used = driver.find_element(By.XPATH , '/html/body/div[1]/div[4]/div/main/turbo-frame/div/div/div/div[2]/div[2]/div/div[5]/div/ul').text
print(language_used)
```

```
MyCompleteProjects
No description, website, or topics provided.
JavaScript
63.1%
CSS
14.7%
SCSS
10.9%
HTML
9.2%
Python
2.1%
```

In [ ]:
```
'''Q5) 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
 Note: - From the home page you have to click on the charts option then hot 100-page link through code'''
```

In [63]:
```
driver.get('https://www.billboard.com/')
```

In [68]:
```
# top_100 = driver.find_element(By.XPATH , '/html/body/div[3]/main/div[2]/div[1]/div[1]/div[1]/div[2]/div/div[2]/a[3]')
#top_100 = driver.find_element(By.XPATH , '/html/body/div[3]/main/div[2]/div[1]/div[1]/div[1]/div[2]/div/div[2]/a[3]')
top_200 = driver.find_element(By.XPATH , '/html/body/div[3]/main/div[2]/div[1]/div[1]/div[1]/div[2]/div/div[2]/a[1]')
top_200.click()
```

In [73]:
```
Song_Name = []
song = driver.find_elements(By.XPATH , '//div[@class="o-chart-results-list-row-container"]/ul/li[4]/ul/li/h3')
for songs in song :
    Song_Name.append(songs.text)
len(Song_Name)
```

Out[73]:
```
200
```

In [74]:
```
Song_Name
```

Out[74]:
```
['Vultures 1',
 '2093',
 'One Thing At A Time',
 'Stick Season',
 'SOS',
 "1989 (Taylor's Version)",
 'Lover',
 'For All The Dogs',
 'Midnights',
 'American Dream',
 'Zach Bryan',
 'Dangerous: The Double Album',
 'The Highlights',
 'Utopia',
 'Folklore',
 '35 Biggest Hits',
 'Legend: The Best Of Bob Marley And The Wailers',
 'The Diamond Collection',
 'Hazbin Hotel, Season One',
```

In [75]:
```
Artist_Name = []
artist = driver.find_elements(By.XPATH , '//div[@class="o-chart-results-list-row-container"]/ul/li[4]/ul/li[1]/span')
for artists in artist :
    Artist_Name.append(artists.text)
len(Artist_Name)
```

Out[75]:
```
200
```

In [76]:
```
Artist_Name
```

Out[76]:
```
['¥$: Kanye West & Ty Dolla $ign',
 'Yeat',
 'Morgan Wallen',
 'Noah Kahan',
 'SZA',
 'Taylor Swift',
 'Taylor Swift',
 'Drake',
 'Taylor Swift',
 '21 Savage',
 'Zach Bryan',
 'Morgan Wallen',
 'The Weeknd',
 'Travis Scott',
 'Taylor Swift',
 'Toby Keith',
 'Bob Marley And The Wailers',
 'Post Malone',
 'Soundtrack',
```

In [82]:
```
tw_rank = []
# D) Peak rank
# E) Weeks on board
rank_this = driver.find_elements(By.XPATH , '//div[@class="o-chart-results-list-row-container"]/ul/li[1]/span[1]')
for rank in rank_this :
    tw_rank.append(rank.text)
len(tw_rank)
```

Out[82]:
```
200
```

In [81]:
```
last_week_rank = []
lw_rank = driver.find_elements(By.XPATH , '//div[@class="o-chart-results-list-row-container"]/ul/li[4]/ul/li[4]/span')
for rank in lw_rank :
    last_week_rank.append(rank.text)
len(last_week_rank)
```

Out[81]: 200

In [79]:
```
peak_rank = []
p_rank = driver.find_elements(By.XPATH , '//div[@class="o-chart-results-list-row-container"]/ul/li[4]/ul/li[5]/span')
for rank in p_rank :
    peak_rank.append(rank.text)
len(peak_rank)
```

Out[79]: 200

In [80]:
```
week_on_board = []
wob = driver.find_elements(By.XPATH , '//div[@class="o-chart-results-list-row-container"]/ul/li[4]/ul/li[6]/span')
for i in wob :
    week_on_board.append(i.text)
len(week_on_board)
```

Out[80]: 200

In [84]:
```
details = {
    'SONG NAME' : Song_Name ,
    'ARTIST NAME' : Artist_Name ,
    'THIS WEEK RANK' : tw_rank ,
    'LAST WEEK RANK' : last_week_rank ,
    'PEAK RANK' : peak_rank ,
    'WEEK ON BOARD' : week_on_board
}
df = pd.DataFrame(details)
df
```

Out[84]:

	SONG NAME	ARTIST NAME	THIS WEEK RANK	LAST WEEK RANK	PEAK RANK	WEEK ON BOARD
0	Vultures 1	¥: Kanye West & Ty Dolla ign	1	1	1	2
1	2093	Yeat	2	-	2	1
2	One Thing At A Time	Morgan Wallen	3	4	1	51
3	Stick Season	Noah Kahan	4	3	3	65
4	SOS	SZA	5	5	1	63
...	...	...	...	...	...	...
195	Unorthodox Jukebox	Bruno Mars	196	177	1	232
196	Ultimate Sinatra	Frank Sinatra	197	159	32	153
197	Back In Black	AC/DC	198	-	4	594
198	Angel Face	Stephen Sanchez	199	-	90	3
199	Cottonwood 2	NLE Choppa	200	196	21	23

200 rows × 6 columns

In [ ]:
```
'''Q6) Scrape the details of Highest selling novels.
A) Book name
B) Author name
C) Volumes sold
D) Publisher
E) Genre
Url - https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-grey-compare'''
```

In [87]:
```
url = 'https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-grey-compare'
driver.get(url)
```

In [89]:
```
Book_Name = []
book = driver.find_elements(By.XPATH , '//table[@class="in-article sortable"]/tbody/tr/td[2]')
for books in book :
    BOOK = books.text
    Book_Name.append(BOOK)
len(Book_Name)
```

Out[89]: 100

In [90]:
```
Author_Name = []
name = driver.find_elements(By.XPATH , '//table[@class="in-article sortable"]/tbody/tr/td[3]')
for author in name :
    names = author.text
    Author_Name.append(names)
len(Author_Name)
```

Out[90]: 100

In [91]:
```
volume_sold = []
sales = driver.find_elements(By.XPATH , '//table[@class="in-article sortable"]/tbody/tr/td[4]')
for sale in sales :
    volume_sold.append(sale.text)
len(volume_sold)
```

Out[91]: 100

In [92]:
```
publisher = []
publish = driver.find_elements(By.XPATH , '//table[@class="in-article sortable"]/tbody/tr/td[5]')
for i in publish :
    publisher.append(i.text)
len(publisher)
```

Out[92]: 100

In [93]:
```
Genre = []
genre = driver.find_elements(By.XPATH , '//table[@class="in-article sortable"]/tbody/tr/td[6]')
for i in genre :
    Genre.append(i.text)
len(Genre)
```

Out[93]: 100

In [94]:
```
df = pd.DataFrame({
    'BOOK NAME' : Book_Name ,
    'AUTHOR NAME' : Author_Name ,
    'VOLUME SALES' : volume_sold ,
    'PUBLISHER' : publisher ,
    'GENRE' : Genre
})
df
```

Out[94]:

	BOOK NAME	AUTHOR NAME	VOLUME SALES	PUBLISHER	GENRE
0	Da Vinci Code,The	Brown, Dan	5,094,805	Transworld	Crime, Thriller & Adventure
1	Harry Potter and the Deathly Hallows	Rowling, J.K.	4,475,152	Bloomsbury	Children's Fiction
2	Harry Potter and the Philosopher's Stone	Rowling, J.K.	4,200,654	Bloomsbury	Children's Fiction
3	Harry Potter and the Order of the Phoenix	Rowling, J.K.	4,179,479	Bloomsbury	Children's Fiction
4	Fifty Shades of Grey	James, E. L.	3,758,936	Random House	Romance & Sagas
...	...	...	...	...	...
95	Ghost,The	Harris, Robert	807,311	Random House	General & Literary Fiction
96	Happy Days with the Naked Chef	Oliver, Jamie	794,201	Penguin	Food & Drink: General
97	Hunger Games,The:Hunger Games Trilogy	Collins, Suzanne	792,187	Scholastic Ltd.	Young Adult Fiction
98	Lost Boy,The:A Foster Child's Search for the L...	Pelzer, Dave	791,507	Orion	Biography: General
99	Jamie's Ministry of Food:Anyone Can Learn to C...	Oliver, Jamie	791,095	Penguin	Food & Drink: General

100 rows × 5 columns

In [ ]:
```
''' Q7)Scrape the details most watched tv series of all time from imdb.com.
Url = https://www.imdb.com/list/ls512407256/ You have
to find the following details:
A) Name
B) Year span
C) Genre
D) Run time
E) Ratings
F) Votes
'''
```

In [96]:
```
driver.get('https://www.imdb.com/list/ls512407256/')
```

In [97]:
```
Name = []
names = driver.find_elements(By.XPATH , '//div[@class="lister-item-content"]/h3/a')
for name in names :
    Name.append(name.text)
len(Name)
```

Out[97]: 100

In [99]:
```
Year_Span = []
year = driver.find_elements(By.XPATH , '//div[@class="lister-item-content"]/h3/span[2]')
for span in year :
    Year_Span.append(span.text)
len(Year_Span)
```

Out[99]: 100

In [100]:
```
Genre = []
genre = driver.find_elements(By.XPATH , '//div[@class="lister-item-content"]/p/span[5]')
for gen in genre :
    Genre.append(gen.text)
len(Genre)
```

Out[100]: 100

In [101]:
```
Rating = []
ratings = driver.find_elements(By.XPATH , '//div[@class="ipl-rating-star small"]/span[2]')
for rating in ratings :
    Rating.append(rating.text)
len(Rating)
```

Out[101]: 100

In [102]:
```
Run_Times = []
run_time = driver.find_elements(By.XPATH , '//div[@class="lister-item-content"]/p/span[3]')
for time in run_time :
    Run_Times.append(time.text)
len(Run_Times)
```

Out[102]: 100

In [103]:
```
Votes = []
votes = driver.find_elements(By.XPATH , '//div[@class="lister-item-content"]/p[4]/span[2]')
for vote in votes :
    Votes.append(vote.text)
len(Votes)
```

Out[103]: 100

In [110]:
```
df = pd.DataFrame({
    'NAME' : Name ,
    'YEAR SPAN' : Year_Span ,
    'GENRE' : Genre ,
    'RUN TIMES' : Run_Times ,
    'RATING' : Rating ,
    'VOTES' : Votes
})
# df['Serial_Number'] = range(1, len(df) + 1)
# df.set_index['Serial_Number' , implace==True]
df
```

Out[110]:

	NAME	YEAR SPAN	GENRE	RUN TIMES	RATING	VOTES
0	Game of Thrones	(2011–2019)	Action, Adventure, Drama	55 min	9.2	2,262,543
1	Stranger Things	(2016–2025)	Drama, Fantasy, Horror	51 min	8.7	1,320,469
2	The Walking Dead	(2010–2022)	Drama, Horror, Thriller	44 min	8.1	1,072,255
3	13 Reasons Why	(2017–2020)	Drama, Mystery, Thriller	60 min	7.5	313,477
4	The 100	(2014–2020)	Drama, Mystery, Sci-Fi	43 min	7.6	273,410
...	...	...	...	...	...	...
95	True Detective	(2014–)	Crime, Drama, Mystery	55 min	8.9	645,547
96	Teen Wolf	(2011–2017)	Action, Drama, Fantasy	41 min	7.7	162,062
97	The OA	(2016–2019)	Drama, Fantasy, Mystery	60 min	7.8	114,878
98	The Simpsons	(1989–)	Animation, Comedy	22 min	8.7	433,090
99	Desperate Housewives	(2004–2012)	Comedy, Drama, Mystery	45 min	7.6	138,688

100 rows × 6 columns

In [111]:
```
'''Q8) 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
 Note: - from the home page you have to go to the Show All Dataset page through code.'''
```

Out[111]:
```
'Q8) Details of Datasets from UCI machine learning repositories.\nUrl = https://archive.ics.uci.edu/ (https://archive.ics.uci.edu/) You\nhave to find the following details:\n A) Dataset name\nB) Data type\nC) Task\nD) Attribute type\nE) No of instances\nF) No of attribute G) Year\n Note: - from the home page you have to go to the Show All Dataset page through code.'
```

In [113]:
```
driver.get('https://archive.ics.uci.edu/')
```

In [119]:
```
# data = driver.find_elements(By.XPATH , '//div[@class="relative col-span-8 sm:col-span-7"]')
# len(data)
data_set_url = []
data = driver.find_element(By.XPATH , '/html/body/div/div[1]/div[1]/header/nav/ul/li[1]/a')
data_set_url.append(data.get_attribute("href"))
data_set_url
```

Out[119]:
```
['https://archive.ics.uci.edu/datasets']
```

In [120]:
```
for i in data_set_url :
    driver.get(i)
    time.sleep(5)


    try :
```

In [ ]:
```
url = "https://archive.ics.uci.edu/"

# Create a webdriver instance (you may need to specify the path to your webdriver executable)
driver = webdriver.Chrome()

# Navigate to the website
driver.get(url)

# Locate and click on the 'View ALL Data Sets' link
all_datasets_link = driver.find_element(By.LINK_TEXT, 'View ALL Data Sets')
all_datasets_link.click()

# Locate the table containing dataset details
table = driver.find_element(By.XPATH, '//table[@border="1"]')

# Extracting the data from the table
data = []
for row in table.find_elements(By.TAG_NAME, 'tr')[1:]:  # Skip the header row
    columns = row.find_elements(By.TAG_NAME, 'td')
    dataset_name, data_type, task, attribute_type, instances, attributes, year = [col.text.strip() for col in columns[0:7]]
    data.append([dataset_name, data_type, task, attribute_type, instances, attributes, year])

# Creating a DataFrame
columns = ['Dataset Name', 'Data Type', 'Task', 'Attribute Type', 'No of Instances', 'No of Attributes', 'Year']
df = pd.DataFrame(data, columns=columns)

# Display the DataFrame
print(df)
```