1. Write a python program which searches all the product under a particular product from www.amazon.in. The product to be searched will be taken as input from user. For e.g. If user input is ‘guitar’. Then search for guitars.

In [1]:
```
import selenium
import time
import pandas as pd
from selenium import webdriver
from selenium.common.exceptions import StaleElementReferenceException,NoSuchElementException
from bs4 import BeautifulSoup
import requests
from selenium.webdriver.common.by import By
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.support.ui import WebDriverWait
import re
```

In [2]:
```
driver=webdriver.Chrome()  #connecting the web
```

In [3]:
```
driver.get("https://www.amazon.in./") # getting mentioned url and opening amazon web page.
```

In [4]:
```
user_input=input('enter the input by user  ') #mention user input code.
```

```
enter the input by user  guitar
```

In [7]:
```
# getting search_bar xpath and send_keys user input.
designation=driver.find_element(By.ID,"twotabsearchtextbox" )
designation.send_keys(user_input)
```

In [10]:
```
#getting xpath of search_button and click them.
search_button=driver.find_element(By.XPATH,'//div[@class="nav-search-submit nav-sprite"]/span/input ')
search_button.click()
```

2.-In the above question, now scrape the following details of each product listed in first 3 pages of your search results and save it in a data frame and csv. In case if any product has less than 3 pages in search results then scrape all the products available under that product name. Details to be scraped are: "Brand Name", "Name of the Product", "Price", "Return/Exchange", "Expected Delivery", "Availability" and “Product URL”. In case, if any of the details are missing for any of the product then replace it by “-“.

In [11]:
```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import requests
import time
from selenium.common.exceptions import NoSuchElementException
```

In [38]:
```
#connect the web page and mention the url and opening the amazon.
driver=webdriver.Chrome()
driver.get("https://www.amazon.in./")
```

In [39]:
```
#getting search_bar xpath and find guitar and click search_button.
speaker_tag=driver.find_element(By.XPATH,"//div[@class='nav-search-field ']//input")
speaker_tag.send_keys('guitar')
search=driver.find_element(By.XPATH,"//div[@class='nav-search-submit nav-sprite']//span")
search.click()
time.sleep(3)
```

In [14]:
```
# written code for getting urls for pages.
product_urls=[]
start=0
end=3

for page in range(start,end):
    url=driver.find_elements(By.XPATH,'//a[@class="a-link-normal s-no-outline"] ')
    for i in url:
        product_urls.append(i.get_attribute("href"))
    nxt_button=driver.find_element(By.XPATH,'//a[@class="s-pagination-item s-pagination-next s-pagination-butto
    nxt_button.click()
    time.sleep(2)
```

In [15]:
```
len(product_urls)
```

Out[15]:
```
184
```

In [16]:
```
#creating empty list.
Brand=[]
Price=[]
Exchange=[]
Delivary=[]
```

In [17]:
```
#written a code to click the product for getting more information.
```

```
product_click=driver.find_element(By.XPATH,'//a[@class="a-link-normal s-no-outline"]/div/img')
product_click.click()
```

In [ ]:
```
#written code for getting all infomation in the urls.
for URL in product_urls:
    driver.get(URL)
    time.sleep(2)

    try:
        brand=driver.find_element(By.XPATH,'//div[@id="productOverview_feature_div"]/div/table/tbody/tr[1]/td[2
        Brand.append(brand.text)
    except NoSuchElementException:
        Brand.append('-')

    try:
        price=driver.find_element(By.XPATH,'//div[@class="a-box-group"]//div[3]/div/div/div/div/span/span/span[
        Price.append(price.text)
    except NoSuchElementException:
        Price.append('-')

    try:
        exchange=driver.find_element(By.XPATH,'//div[@id="anonCarousel4"]/ol/li/div/span/div[2]/span')
        Exchange.append(exchange.text)
    except NoSuchElementException:
        Exchange.append('-')

    try:
        delivary=driver.find_element(By.XPATH,'//div[@id="deliveryBlockContainer"]/div/div/div/div/span/span')
        Delivary.append(delivary.text)
    except NOSuchElementException:
        Delivary.append('-')
```

In [19]:
```
len(Brand)
```

Out[19]:
```
51
```

In [20]:
```
len(Price)
```

Out[20]:
```
51
```

In [21]:
```
len(Exchange)
```

Out[21]:
```
51
```

In [22]:
```
len(Delivary)
```

Out[22]:
```
50
```

In [27]:
```
# written code for making dataframe.
df=pd.DataFrame({"BRAND":Brand[0:50],"PRICE":Price[0:50],"RETURN_EXCHANGE":Exchange[0:50],"DELIVARY":Delivary})
df
```

Out[27]:

	BRAND	PRICE	RETURN_EXCHANGE	DELIVARY
0	Kadence	5,399	-	Saturday, 8 June
1	Kadence	4,899	-	Saturday, 8 June
2	VAULT	4,899	-	Saturday, 8 June
3	Kadence	10,499	-	Saturday, 8 June
4	Intern	2,199	-	Thursday, 6 June
5	JUAREZ	1,999	-	Thursday, 13 June
6	Intern	2,199	-	Saturday, 8 June
7	musoo	3,699	-	Thursday, 6 June
8	musoo	3,699	-	Thursday, 6 June
9	JUAREZ	2,706	-	Sunday, 9 June
10	Intern	2,199	-	Sunday, 9 June
11	JUAREZ	2,723	-	Sunday, 9 June
12	Intern	2,199	-	Thursday, 13 June
13	YAMAHA	7,399	-	Saturday, 8 June
14	REVEL	1,999	-	Saturday, 8 June
15	Henrix	3,299	-	Saturday, 8 June
16	REVEL	1,949	-	7 - 15 June
17	Medellin	2,199	-	Sunday, 9 June
18	Kadence	5,399	-	Saturday, 8 June
19	JUAREZ	2,413	-	Sunday, 9 June
20	Medellin	2,599	-	Sunday, 9 June
21	Kadence	4,899	-	Saturday, 8 June
22	DEVICE OF URBAN INFOTECH	5,989	-	Friday, 7 June
23	blueberry	2,495	-	Friday, 7 June
24	blueberry	2,999	-	Friday, 7 June
25	Intern	3,599	-	Saturday, 8 June
26	JUAREZ	2,365	-	Sunday, 9 June
27	Yamaha	8,480	-	Friday, 7 June
28	blueberry	2,895	-	Friday, 7 June
29	Belear	2,699	-	Sunday, 9 June
30	Intern	2,160	-	Saturday, 8 June
31	JUAREZ	3,296	-	Sunday, 9 June
32	Yamaha	7,900	-	Saturday, 8 June
33	Medellin	2,199	-	Sunday, 9 June
34	Yamaha	7,799	-	Saturday, 8 June
35	YAMAHA	7,870	-	Tuesday, 11 June
36	blueberry	2,495	-	Friday, 7 June
37	YAMAHA	10,490	-	Saturday, 8 June
38	YAMAHA	12,399	-	Tuesday, 11 June
39	Kadence	15,749	-	Sunday, 9 June
40	JUAREZ	3,177	-	Wednesday, 12 June
41	Kadence	6,799	-	Sunday, 9 June
42	Kadence	9,999	-	Sunday, 9 June
43	Yamaha	9,441	-	Saturday, 8 June
44	Yamaha	9,299	-	Saturday, 8 June
45	Kadence	6,999	-	Sunday, 9 June
46	ENYA	17,000	-	Sunday, 9 June
47	Kadence	6,299	-	Saturday, 8 June
48	blueberry	2,495	-	Friday, 7 June
49	VAULT	4,899	-	Wednesday, 5 June

In [ ]:
```
df.to_csv("text.txt")
```

3. Write a python program to access the search bar and search button on images.google.com and scrape 10 images each for keywords ‘fruits’, ‘cars’ and ‘Machine Learning’, ‘Guitar’, ‘Cakes’.

In [58]:
```
#calling library
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import requests
import time
from selenium.common.exceptions import NoSuchElementException
```

In [59]:
```
driver=webdriver.Chrome()
```

In [60]:
```
driver.get("https://www.google.com/")
```

In [61]:
```
search=driver.find_element(By.CLASS_NAME,"gLFyf")
search.send_keys("image.google.com")
```

In [62]:
```
search_button=driver.find_element(By.XPATH,'//div[@class="FPdoLc lJ9FBc"]/center/input ')
search_button.click()
```

In [63]:
```
option_click=driver.find_element(By.XPATH,'//div[@class="yuRUbf"]/div/span/a/h3')
option_click.click()
```

In [23]:
```
url=["fruits","cars","machine learning","guitar","cakes"]
```

In [28]:
```
search_mention=driver.find_element(By.ID,"APjFqb")
search_mention.send_keys("fruits")
```

In [30]:
```
search_btn=driver.find_element(By.XPATH,'/html/body/div[4]/div/div[7]/div/div/div/div/div[1]/div/div/div/div[2]
search_btn.click()
```

In [31]:
```
Fruit_image=[]

fruit_image=driver.find_elements(By.XPATH,'//div[@class="H8Rx8c"]/g-img/img')
for i in fruit_image:
    fruit=i.text
    Fruit_image.append(fruit)
```

In [33]:
```
len(Fruit_image)
```

Out[33]:
```
100
```

In [48]:
```

img_urls=[]

images=driver.find_elements(By.XPATH,'//img[@class="YQ4gaf"] ')

for image in images:
    source=image.get_attribute('src')
    if source is not None:
        if(source[0:4]=='http'):
            img_urls.append(source)


for i in range(len(img_urls)):
    if i>10:
        breakBy.XPATH,
    print("Downloading {0} of {1} images".format(i,10))
    response=requests.get(img_urls[1])
    file=open(r"C:\jan20\fruitsimage"+str(i)+".jpg","wb")
    file.write(response.content)
```

```
---------------------------------------------------------------------------
WebDriverException                        Traceback (most recent call last)
Cell In[48], line 3
      1 img_urls=[]
----> 3 images=driver.find_elements(By.XPATH,'//img [@class="YQ4gaf"] ')
      5 for image in images:
      6     source=image.get_attribute('src')

File ~\anaconda3\Lib\site-packages\selenium\webdriver\remote\webdriver.py:771, in WebDriver.find_elements(self, by, value)
    767     value = f'[name="{value}"]'
    769 # Return empty list if driver returns null
    770 # See https://github.com/SeleniumHQ/selenium/issues/4555
--> 771 return self.execute(Command.FIND_ELEMENTS, {"using": by, "value": value})["value"] or []

File ~\anaconda3\Lib\site-packages\selenium\webdriver\remote\webdriver.py:347, in WebDriver.execute(self, driver_command, params)
    345 response = self.command_executor.execute(driver_command, params)
    346 if response:
--> 347     self.error_handler.check_response(response)
    348     response["value"] = self._unwrap_value(response.get("value", None))
    349     return response

File ~\anaconda3\Lib\site-packages\selenium\webdriver\remote\errorhandler.py:229, in ErrorHandler.check_response(self, response)
    227         alert_text = value["alert"].get("text")
    228     raise exception_class(message, screen, stacktrace, alert_text)  # type: ignore[call-arg]  # mypy is not smart enough here
--> 229 raise exception_class(message, screen, stacktrace)

WebDriverException: Message: disconnected: not connected to DevTools
  (failed to check if window was closed: disconnected: not connected to DevTools)
  (Session info: chrome=125.0.6422.113)
Stacktrace:
        GetHandleVerifier [0x00007FF7B3CC1F52+60322]
        (No symbol) [0x00007FF7B3C3CEC9]
        (No symbol) [0x00007FF7B3AF7EBA]
        (No symbol) [0x00007FF7B3ADF1CC]
        (No symbol) [0x00007FF7B3ADF090]
        (No symbol) [0x00007FF7B3AFA4E1]
        (No symbol) [0x00007FF7B3B8B359]
        (No symbol) [0x00007FF7B3B6BFC3]
        (No symbol) [0x00007FF7B3B39617]
        (No symbol) [0x00007FF7B3B3A211]
        GetHandleVerifier [0x00007FF7B3FD94AD+3301629]
        GetHandleVerifier [0x00007FF7B40236D3+3605283]
        GetHandleVerifier [0x00007FF7B4019450+3563680]
        GetHandleVerifier [0x00007FF7B3D74326+790390]
        (No symbol) [0x00007FF7B3C4750F]
        (No symbol) [0x00007FF7B3C43404]
        (No symbol) [0x00007FF7B3C43592]
        (No symbol) [0x00007FF7B3C32F9F]
        BaseThreadInitThunk [0x00007FFAE431257D+29]
        RtlUserThreadStart [0x00007FFAE584AA48+40]
```

In [64]:
```
search_car=driver.find_element(By.ID,"APjFqb")
search_car.send_keys("cars")
```

In [67]:
```
search_button=driver.find_element(By.XPATH,'/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/button')
search_button.click()
```

In [73]:
```
img_urls=[]
images=driver.find_elements(By.XPATH,'//img[@class="YQ4gaf"]')
for image in images:
    source=image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)

for i in range(len(img_urls)):
    if i>10:
        breakBy.XPATH,
    print("Downloading {0} of {1} images" .format(i,10))
    response=requests.get(img_urls[i])
    file=open(r"C:\jan20\fruitsimage"+str(i)+".jpg","wb")
    file.write(response.content)
```

```
Downloading 0 of 10 images
Downloading 1 of 10 images
Downloading 2 of 10 images
Downloading 3 of 10 images
```

In [75]:
```
search=driver.find_element(By.ID,"APjFqb")
search.send_keys("Machine Learning")
```

In [76]:
```
search_button=driver.find_element(By.XPATH,'/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/button')
```

```
search_button.click()
```

In [91]:
```
img_urls=[]
images=driver.find_elements(By.XPATH,'//img[@class="YQ4gaf"]')

for image in images:
    source=image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)

for i in range(len(img_urls)):
    if i>10:
        breakBy.XPATH,
    print("Downloading {0} of {1} images".format(i,10))
    response=requests.get(img_urls[i])
    file=open(r"C:\jan20\fruitsimage\Machine_Learning"+str(i)+".jpg","wb")
    file.write(response.content)
```

```
Downloading 0 of 10 images
Downloading 1 of 10 images
Downloading 2 of 10 images
Downloading 3 of 10 images
Downloading 4 of 10 images
Downloading 5 of 10 images
Downloading 6 of 10 images
Downloading 7 of 10 images
Downloading 8 of 10 images
Downloading 9 of 10 images
Downloading 10 of 10 images
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[91], line 12
     10 for i in range(len(img_urls)):
     11     if i>10:
---> 12         breakBy.XPATH,
     13     print("Downloading {0} of {1} images".format(i,10))
     14     response=requests.get(img_urls[i])

NameError: name 'breakBy' is not defined
```

In [96]:
```
search=driver.find_element(By.ID,"APjFqb")
search.send_keys("guitar")
```

In [97]:
```
search_button=driver.find_element(By.XPATH,'/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/button')
search_button.click()
```

In [99]:
```
img_urls=[]
images=driver.find_elements(By.XPATH,'//img[@class="YQ4gaf"]')

for image in images:
    source=image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)

for i in range(len(img_urls)):
    if i > 10:
        breakBy.XPATH,
    print("Downloading {0} of {1} images".format(i,10))
    response=requests.get(img_urls[1])
    file=open(r"C:\jan20\fruitsimage\Machine_Learning"+str(i)+".jpg","wb")
    file.write(response.content)
```

```
Downloading 0 of 10 images
Downloading 1 of 10 images
Downloading 2 of 10 images
Downloading 3 of 10 images
Downloading 4 of 10 images
Downloading 5 of 10 images
Downloading 6 of 10 images
Downloading 7 of 10 images
Downloading 8 of 10 images
Downloading 9 of 10 images
Downloading 10 of 10 images
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[99], line 12
     10 for i in range(len(img_urls)):
     11     if i > 10:
---> 12         breakBy.XPATH,
     13     print("Downloading {0} of {1} images".format(i,10))
     14     response=requests.get(img_urls[1])


NameError: name 'breakBy' is not defined
```

In [100…
```
search=driver.find_element(By.ID,'APjFqb')
search.send_keys("cakes")
```

In [101…
```
search_button=driver.find_element(By.XPATH,'/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/button')
search_button.click()
```

In [102…
```
img_images=[]
images=driver.find_elements(By.XPATH,'//img[@class="YQ4gaf"]')

for image in images:
    source=image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)

for i in range(len(img_urls)):
    if i > 10:
        breakBy.XPATH,
    print("Downloading {0} of {1} images" .format(i,10))
    response=requests.get(img_urls[i])
    file=open(r"C:\jan20\fruitsimage\Machine_Learning"+str(i)+".jpg","wb")
    file.write(response.content)
```

```
Downloading 0 of 10 images
Downloading 1 of 10 images
Downloading 2 of 10 images
Downloading 3 of 10 images
Downloading 4 of 10 images
Downloading 5 of 10 images
Downloading 6 of 10 images
Downloading 7 of 10 images
Downloading 8 of 10 images
Downloading 9 of 10 images
Downloading 10 of 10 images
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[102], line 12
     10 for i in range(len(img_urls)):
     11     if i > 10:
---> 12         breakBy.XPATH,
     13     print("Downloading {0} of {1} images" .format(i,10))
     14     response=requests.get(img_urls[i])

NameError: name 'breakBy' is not defined
```

4. Write a python program to search for a smartphone(e.g.: Oneplus Nord, pixel 4A, etc.) on www.flipkart.com and scrape following details for all the search results displayed on 1st page. Details to be scraped: “Brand Name”, “Smartphone name”, “Colour”, “RAM”, “Storage(ROM)”, “Primary Camera”, “Secondary Camera”, “Display Size”, “Battery Capacity”, “Price”, “Product URL”. Incase if any of the details is missing then replace it by “- “. Save your results in a dataframe and CSV.

In [23]:
```
import selenium
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.common.exceptions import NoSuchElementException
import requests
```

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

In [25]:
```
driver.get("https:/www.flipkart.com/")
```

In [26]:
```
search=driver.find_element(By.CLASS_NAME,'Pke_EE')
search.send_keys("smartphone")
```

In [27]:
```
search_button=driver.find_element(By.XPATH,'/html/body/div[1]/div/div[1]/div/div/div/div/div[1]/div/div[1]/div/
search_button.click()
```

In [28]:
```
brand_search=driver.find_element(By.XPATH,'//input[@class="XPD6hh"]')
brand_search.send_keys("OnePlus")
```

In [29]:
```
cheak_option=driver.find_element(By.XPATH,'//div[@class="XqNaEv"]')
```

```
cheak_option.click()
```

In [30]:
```
click_image=driver.find_element(By.XPATH,'//img[@class="DByuf4"]')
click_image.click()
```

In [31]:
```
Product_urls=[]
start=0
end=1
for page in range(start,end):
    url=driver.find_elements(By.XPATH,'//a[@class="CGtC98"]')
    for i in url:
        Product_urls.append(i.get_attribute("href"))
    next_button=driver.find_element(By.XPATH,'//a[@class="_9QVEpD"]')
    next_button.click()
    time.sleep(2)
```

In [32]:
```
len(Product_urls)
```

Out[32]:
```
24
```

In [33]:
```
Mobile_image=driver.find_element(By.XPATH,'//img[@class="DByuf4"]')
Mobile_image.click()
```

In [36]:
```
Brand=[]
Model=[]
Colore=[]
Ram=[]
Rom=[]
Camera=[]
Display_size=[]
Battery_capcity=[]
Price=[]
Product_url=[]

for url in Product_urls:
    driver.get(url)
    time.sleep(2)

    try:
        brand=driver.find_elements(By.XPATH,'//h1[@class="_6EBuvT"]/span')
        Brand.append(brand.text)
    except NoSuchElementException:
        Brand.append('-')

    try:
        model=driver.find_elements(By.XPATH,'//table[@class="_0ZhAN9"]/tbody/tr[3]/td[2]/ul/li')
        Model.append(brand.text)
    except NoSuchElementException:
        Model.append(brand.text)

    try:
        colore=driver.find_elements(By.XPATH,'//table[@class="_0ZhAN9"]/tbody/tr[4]/td[2]/ul/li')
        Colore.append(colore.text)
    except NoSuchElementException:
        Colore.append(colore.text)

    try:
        ram=driver.find_elements(By.XPATH,'//div[@class="xFVion"]/ul/li')
        Ram.append(ram.text)
    except NoSuchElemenetException:
        Ram.append(ram.text)

    try:
        display=driver.find_elements(By.XPATH,'//div[@class="xFVion"]/ul/li[2]')
        Display_size.append(display.text)
    except NoSuchElementException:
        Display_size.append(display.text)

    try:
        camera=driver.find_elements(By.XPATH,'//div[@class="xFVion"]/ul/li[3]')
        Camera.append(camera.text)
    except NoSuchElementException:
        Camera.append(camera.text)

    try:
        battery=driver.find_elements(By.XPATH,'//div[@class="xFVion"]/ul/li[4]')
        Battery_capcity.append(battery.text)
    except NoSuchElementException:
        Battery_capcity.append(battery.text)

    try:
        price=driver.find_elements(By.XPATH,'//div[@class="hl05eU"]/div')
        Price.append(price.text)
```

```
    except NoSuchElementException:
        Price.append(price.text)
    try:
        product=driver.find_elements(By.XPATH,'//a[@class="CGtC98"]')
        Product_url.append(product.text)
    except NoSuchElementException:
        Product_url.append(product.text)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[36], line 18
     16 try:
     17     brand=driver.find_elements(By.XPATH,'//h1[@class="_6EBuvT"]/span')
---> 18     Brand.append(brand.text)
     19 except NoSuchElementException:
     20     Brand.append('-')

AttributeError: 'list' object has no attribute 'text'
```

5. Write a program to scrap geospatial coordinates (latitude, longitude) of a city searched on google maps.

In [1]:
```
import selenium
import pandas as pd
import requests
import re
from bs4 import BeautifulSoup
from selenium.common.exceptions import NoSuchElementException, StaleElementReferenceException
from selenium.webdriver.support.ui import WebDriverWait
import time
import warnings
warnings.filterwarnings("ignore")
from selenium.webdriver.common.by import By
from selenium.common.exceptions import NoSuchElementException
from selenium import webdriver
```

In [2]:
```
driver=webdriver.Chrome() # connecting to the webdriver
```

In [6]:
```
# getting mentioned url and opening google maps web page
url="https://www.google.co.in/maps"
driver.get(url)
time.sleep(2)
```

In [19]:
```
# entering the city name in search bar

City = input('Enter City name that has to be searched : ')
search_bar = driver.find_element(By.CLASS_NAME,"searchboxinput")
search_bar.click()
time.sleep(2)

#sending keys to find cities
search_bar.send_keys(City)

#checking for webelement and clicking on search button
search_btn = driver.find_element(By.ID,"searchbox-searchbutton")
search_btn.click()
time.sleep(2)

try:
    url_str = driver.current_url
    print("URL Extracted: ", url_str)
    latitude_longitude = re.findall(r'@(.*)data',url_str)
    if len(latitude_longitude):
        lat_lng_list = latitude_longitude[0].split(",")
        if len(lat_lng_list)>=2:
            latitude = lat_lng_list[0]
            longitude = lat_lng_list[1]
        print("Latitude = {}, Longitude = {}".format(latitude, longitude))
except Exception as e:
        print("Error: ", str(e))
```

```
Enter City name that has to be searched : jaipur
URL Extracted:  https://www.google.co.in/maps/search/Jaipurjaipurjaipur/@27.1504598,75.6009089,8z/data=!3m1!4b1
?entry=ttu
Latitude = 27.1504598, Longitude = 75.6009089
```

In [ ]:
```
driver.close()
```

6. Write a program to scrap all the available details of best gaming laptops from digit.in.

In [3]:
```
import selenium
import pandas as pd
import requests
import re
from bs4 import BeautifulSoup
from selenium.common.exceptions import NoSuchElementException, StaleElementReferenceException
from selenium.webdriver.support.ui import WebDriverWait
import time
```

```
import warnings
warnings.filterwarnings("ignore")
from selenium.webdriver.common.by import By
from selenium.common.exceptions import NoSuchElementException
from selenium import webdriver
```

In [7]:
```
#connect to the web.
driver=webdriver.Chrome()
url="https:/digit.in./"
driver.get(url)
```

In [5]:
```
click_product=driver.find_element(By.XPATH,'/html/body/div[1]/div[3]/div/div/div/article/div/div/div/div[2]/div
click_product.click()
```

In [6]:
```
 #searching for best Laptop
best_gam_laptops = driver.find_element(By.XPATH,"/html/body/div[1]/div[3]/div/div/div/article/div/div/div/div[2
time.sleep(3)
```

In [12]:
```
product_urls=[]
url=driver.find_elements(By.XPATH,'//div[@class="medianews-body floatright width-100-calc"]/h5/a')
for i in url:
    product_urls.append(i.get_attribute("href"))
```

In [13]:
```
len(product_url)
```

Out[13]:
```
15
```

In [14]:
```
# creating empty list
Laptop_Name = []
Operating_sys = []
Display = []
Processor = []
Memory = []
Weight = []
Dimensions = []
Graph_proc = []
Price = []
```

In [16]:
```
#scraping the data of laptop names
laptop_name = driver.find_elements(By.XPATH,'//span[@id="productTitle"]')
for name in laptop_name:
    Laptop_Name.append(name.text)

#scraping the data of operating system
try:
    op_sys = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[37]/td')
    for os in op_sys:
        Operating_sys.append(os.text)
except NoSuchElementException:
    pass


#scraping data of display of the Laptop
try:
    display = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[6]/td')
    for disp in display:
        Display.append(disp.text)
except NoSuchElementException:
    pass


# scraping data of processor
try:
    processor = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[13]/td
    for pro in processor:
        Processor.append(pro.text)
except NoSuchElementException:
    pass


# scraping the data of memory
try:
    memory = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[16]/td')
    for memo in memory:
        Memory.append(memo.text)
except NoSuchElementException:
    pass


# scraping data of weight
try:
    weight = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[42]/td')
    for wgt in weight:
        Weight.append(wgt.text)
except NoSuchElementException:
    pass
```

```
# scraping data of dimensions
try:
    dimension = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[9]/td'
    for dim in dimension:
        Dimensions.append(dim.text)
except NoSuchElementException:
    pass


# scraping data of graph processor
try:
    graph = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[24]/td')
    for gra in graph:
        Graph_proc.append(gra.text)
except NoSuchElementException:
    pass


# scraping the data of price
try:
    price = driver.find_elements(By.XPATH,'//span[@class="a-price aok-align-center reinventPricePriceToPayMargi
    for pri in price:
        Price.append(pri.text.replace('₹ ','Rs'))
except NoSuchElementException:
    pass
```

In [17]:
```
print(len(Laptop_Name),
len(Operating_sys),
len(Display),
len(Processor),
len(Memory),
len(Weight),
len(Dimensions),
len(Graph_proc),
len(Price))
```

```
0 0 0 0 0 0 0 0 0
```

In [18]:
```
#creating DataFrame for scraped data
Gaming_Laptop=pd.DataFrame({})
Gaming_Laptop['Laptop Name'] = Laptop_Name
Gaming_Laptop['Operating System'] =Operating_sys
Gaming_Laptop['Display'] = Display
Gaming_Laptop['Processor'] = Processor
Gaming_Laptop['Memory'] = Memory
Gaming_Laptop['Weight'] = Weight
Gaming_Laptop['Dimensions'] = Dimensions
Gaming_Laptop['Graphical Processor'] = Graph_proc
Gaming_Laptop['Price'] = Price
Gaming_Laptop
```

Out[18]:

Laptop Name	Operating System	Display	Processor	Memory	Weight	Dimensions	Graphical Processor	Price

In [ ]:
```

```

In [23]:
```
for url in product_urls:
    driver.get(url)
    time.sleep(2)
    #scraping the data of laptop names
    try:
        laptop_name = driver.find_elements(By.XPATH,'//span[@id="productTitle"]')
        Laptop_Name.append(laptop_name.text)
    except NoSuchElementException:
        pass
    #scraping the data of operating system
    try:
        op_sys = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[37]/t
        Operating_sys.append(os_sys.text)
    except NoSuchElementException:
        pass


#scraping data of display of the Laptop
    try:
        display = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[6]/t
        Display.append(display.text)
    except NoSuchElementException:
        pass


# scraping data of processor
    try:
        processor = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[13
        Processor.append(processor.text)
    except NoSuchElementException:
```

```
        pass


# scraping the data of memory
    try:
        memory = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[16]/t
        Memory.append(memory.text)
    except NoSuchElementException:
        pass


# scraping data of weight
    try:
        weight = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[42]/t
        Weight.append(weight.text)
    except NoSuchElementException:
        pass


# scraping data of dimensions
    try:
        dimension = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[9]
        Dimensions.append(dimension.text)
    except NoSuchElementException:
        pass


# scraping data of graph processor
    try:
        graph = driver.find_elements(By.XPATH,'//table[@id="productDetails_techSpec_section_1"]/tbody/tr[24]/td
        Graph_proc.append(graph.text)
    except NoSuchElementException:
        pass


# scraping the data of price
    try:
        price = driver.find_elements(By.XPATH,'//span[@class="a-price aok-align-center reinventPricePriceToPayM
        Price.append(price.text.replace('₹ ','Rs'))
    except NoSuchElementException:
        pass
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[23], line 7
      5 try:
      6     laptop_name = driver.find_elements(By.XPATH,'//span[@id="productTitle"]')
----> 7     Laptop_Name.append(laptop_name.text)
      8 except NoSuchElementException:
      9     pass

AttributeError: 'list' object has no attribute 'text'
```

In [ ]:

7. Write a python program to scrape the details for all billionaires from www.forbes.com. Details to be scrapped: “Rank”, “Name”, “Net worth”, “Age”, “Citizenship”, “Source”, “Industry”.

In [1]:
```
import selenium
import pandas as pd
import requests
import re
from bs4 import BeautifulSoup
from selenium.common.exceptions import NoSuchElementException, StaleElementReferenceException
from selenium.webdriver.support.ui import WebDriverWait
import time
import warnings
warnings.filterwarnings("ignore")
from selenium.webdriver.common.by import By
from selenium import webdriver
```

In [2]:
```
driver=webdriver.Chrome()
```

In [3]:
```
url="https://www.forbes.com/?sh=41bd46d2254c"
driver.get(url)
```

In [4]:
```
option1=driver.find_element(By.XPATH,'//div[@class="dN9h-rXs"]/div[1]')
option1.click()
```

In [5]:
```
option2=driver.find_element(By.XPATH,'//div[@class="R0tvspq-"]')
option2.click()
```

In [6]:
```
option3=driver.find_element(By.XPATH,'//div[@class="LYbP3oHd"]/ul/li/a')
option3.click()
```

In [16]:
```
Product_urls=[]
```

```
start=0
end=4
for page in range(start,end):
    url=driver.find_elements(By.XPATH,'//div[@class="qXW-dtvf"]/button[2]')
    for i in url:
        Product_urls.append(i.get_attribute("src"))
    next_button=driver.find_element(By.XPATH,'//div[@class="qXW-dtvf"]/button[7]')
    next_button.click()
    time.sleep(2)
```

In [17]:
```
len(Product_urls)
```

Out[17]:
```
4
```

In [18]:
```
Product_urls
```

Out[18]:
```
[None, None, None, None]
```

In [ ]:
```
#let's get option button from the page
opt_btn = driver.find_element_by_xpath("//div[@class='header__left']//button")
opt_btn.click()
time.sleep(3)

#select billionaires from options
blns = driver.find_element_by_xpath("/html/body/div[1]/header/nav/div[3]/ul/li[1]")
blns.click()
time.sleep(3)
#select world billionaire
bln_list = driver.find_element_by_xpath("/html/body/div[1]/header/nav/div[3]/ul/li[1]/div[2]/ul/li[2]/a")
bln_list.click()
time.sleep(4
```

In [ ]:
```
# scraping required data from the web page
# creating empty lists
Rank = []
Person_Name = []
Net_worth = []
Age = []
Citizenship = []
Source = []
Industry = []


while(True):

    # scraping the data of rank of the billionaires
    rank_tag = driver.find_elements_by_xpath("//div[@class='rank']")
    for rank in rank_tag:
        Rank.append(rank.text)
    time.sleep(1)



    # scraping the data  of names of the billionaires
    name_tag = driver.find_elements_by_xpath("//div[@class='personName']/div")
    for name in name_tag:
        Person_Name.append(name.text)
    time.sleep(1)


    # scraping the data of age of the billionaires
    age_tag = driver.find_elements_by_xpath("//div[@class='age']/div")
    for age in age_tag:
        Age.append(age.text)
    time.sleep(1)


    # scraping the data of citizenship of the billionaires
    cit_tag = driver.find_elements_by_xpath("//div[@class='countryOfCitizenship']")
    for cit in cit_tag:
        Citizenship.append(cit.text)
    time.sleep(1)


    # scraping the data of source of income of the billionaires
    sour_tag = driver.find_elements_by_xpath("//div[@class='source']")
    for sour in sour_tag:
        Source.append(sour.text)
    time.sleep(1)


    # scraping data of industry of the billionaires
    ind_tag = driver.find_elements_by_xpath("//div[@class='category']//div")
    for ind in ind_tag:
        Industry.append(ind.text)
    time.sleep(1)
```

```
    # scraping data of net_worth of billionaires
    net_tag = driver.find_elements_by_xpath("//div[@class='netWorth']/div")
    for net in net_tag:
        Net_worth.append(net.text)
    time.sleep(1)


    # clicking on next button
    try:
        next_button = driver.find_element_by_xpath("//button[@class='pagination-btn pagination-btn--next ']")
        next_button.click()
    except:
        break
```

In [ ]:
```
print(len(Rank),
len(Person_Name),
len(Net_worth),
len(Age),
len(Citizenship),
len(Source),
len(Industry))
```

In [ ]:
```
# framing Data
Billionaires = pd.DataFrame({})
Billionaires['Rank'] = Rank
Billionaires['Name'] = Person_Name
Billionaires['Net Worth'] = Net_worth
Billionaires['Age'] = Age
Billionaires['Citizenship'] = Citizenship
Billionaires['Source'] = Source
Billionaires['Industry'] = Industry
Billionaires
```

In [ ]:

8-Write a program to extract at least 500 Comments, Comment upvote and time when comment was posted from any YouTube Video.

In [19]:
```
driver=webdriver.Chrome()
```

In [20]:
```
# opening the youtube.com
url = "https://www.youtube.com/"
driver.get(url)
time.sleep(2)
```

In [23]:
```
# finding element for search bar
search_bar = driver.find_element(By.XPATH,'//div[@class="ytd-searchbox-spt"]/input')
search_bar.send_keys("GOT")      # entering video name
time.sleep(2)
```

In [24]:
```
#clicking on search button
search_btn = driver.find_element(By.ID,"search-icon-legacy")
search_btn.click()
time.sleep(2)
```

In [25]:
```
# clicking on first video
video = driver.find_element(By.XPATH,'//yt-formatted-string[@class="style-scope ytd-video-renderer"]')
video.click()
```

In [32]:
```
# 1000 times we scroll down by 10000 in order to generate more comments
for _ in range(1000):
    driver.execute_script("window.scrollBy(0,10000)")
```

In [34]:
```
# creating empty lists
comments = []
comment_time = []
Time = []
Likes = []
No_of_Likes = []

# scrape comments
cm = driver.find_elements(By.ID,"content-text")
for i in cm:
    if i.text is None:
        comments.append("--")
    else:
        comments.append(i.text)
time.sleep(4)


# scrape time when comment was posted
tm = driver.find_elements(By.XPATH,"//a[contains(text(),'ago')]")
for i in tm:
    Time.append(i.text)
```

```
for i in range(0,len(Time),2):
    comment_time.append(Time[i])
time.sleep(4)


# scrape the comment likes
like = driver.find_elements(By.XPATH,"//span[@class='style-scope ytd-comment-action-buttons-renderer']")
for i in like:
    Likes.append(i.text)

for i in range(1,len(Likes),2):
    No_of_Likes.append(Likes[i])
```

In [35]:
```
print(len(comments),len(comment_time),len(No_of_Likes))
```

```
120 60 0
```

In [30]:
```
# creating empty lists
comments = []
comment_time = []
Time = []
Likes = []
No_of_Likes = []

# scrape comments
cm = driver.find_elements(By.ID,"content-text")
for i in cm:
    if i.text is None:
        comments.append("--")
    else:
        comments.append(i.text)
time.sleep(4)


# scrape time when comment was posted
tm = driver.find_elements(By.XPATH,"//a[contains(text(),'ago')]")
for i in tm:
    Time.append(i.text)

for i in range(0,len(Time),2):
    comment_time.append(Time[i])
time.sleep(4)


# scrape the comment likes
like = driver.find_elements(By.XPATH,'//span[@class="style-scope ytd-comment-engagement-bar"]')
for i in like:
    Likes.append(i.text)

for i in range(1,len(Likes),2):
    No_of_Likes.append(Likes[i])
```

In [31]:
```
print(len(comments),len(comment_time),len(No_of_Likes))
```

```
60 30 30
```

In [ ]:
```
# creating dataframe for scraped data

Youtube = pd.DataFrame({})
Youtube['Comment'] = comments[:500]
Youtube['Comment Time'] = comment_time[:500]
Youtube['Comment Upvotes'] = No_of_Likes[:500]
Youtube//span[@class="style-scope ytd-comment-engagement-bar"]
```

In [ ]:
```
```

9. Write a python program to scrape a data for all available Hostels from https://www.hostelworld.com/ in “London” location. You have to scrape hostel name, distance from city centre, ratings, total reviews, overall reviews, privates from price, dorms from price, facilities and property description.

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

In [ ]:
```
# getting the web page of mentioned url
url = "https://www.hostelworld.com/"
driver.get(url)
time.sleep(3)
```

In [ ]:
```
# locating the location search bar
search_bar = driver.find_element_by_id("search-input-field")

# entering London in search bar
search_bar.send_keys("London")
```

In [ ]:
```
# select London
London = driver.find_element_by_xpath("//ul[@id='predicted-search-results']//li[2]")
#clicking on button
London.click()
```

```
# do click on Let's Go button
search_btn = driver.find_element_by_id('search-button')
search_btn.click()
```

In [ ]:
```
# creating empty list & find required data
hostel_name = []
distance = []
pvt_prices = []
dorms_price = []
rating = []
reviews = []
over_all = []
facilities = []
description = []
url = []
```

In [ ]:
```
# scraping the required informations
for i in driver.find_elements_by_xpath("//div[@class='pagination-item pagination-current' or @class='pagination
    i.click()
    time.sleep(3)


    # scraping  hostel name
    try:
        name = driver.find_elements_by_xpath("//h2[@class='title title-6']")
        for i in name:
            hostel_name.append(i.text)
    except NoSuchElementException:
        hostel_name.append('-')


    # scraping distance from city centre
    try:
        dist = driver.find_elements_by_xpath("//div[@class='subtitle body-3']//a//span[1]")
        for i in name:
            distance.append(i.text.replace('Hostel - ',''))
    except NoSuchElementException:
        distance.append('-')


    for i in driver.find_elements_by_xpath("//div[@class='prices-col']"):
    # scraping privates from price
        try:
            pvt_price = driver.find_element_by_xpath("//a[@class='prices']//div[1]//div")
            pvt_prices.append(pvt_price.text)
        except NoSuchElementException:
            pvt_prices.append('-')


    for i in driver.find_elements_by_xpath("//div[@class='prices-col']"):
    # scraping dorms from price
        try:
            dorms = driver.find_element_by_xpath("//a[@class='prices']//div[2]/div")
            dorms_price.append(dorms.text)
        except NoSuchElementException:
            dorms_price.append('-')


    # scraping facilities
    try:
        fac1 = driver.find_elements_by_xpath("//div[@class='has-wifi']")
        fac2 = driver.find_elements_by_xpath("//div[@class='has-sanitation']")
        for i in fac1:
            for j in fac2:
                facilities.append(i.text +', '+ j.text)
    except NoSuchElementException:
        facilities.append('-')


    #fetching url of each hostel
    p_url = driver.find_elements_by_xpath("//div[@class='prices-col']//a[2]")
    for i in p_url:
        url.append(i.get_attribute("href"))

for i in url:
    driver.get(i)
    time.sleep(3)


    # scraping ratings
    try:
        rat = driver.find_element_by_xpath("//div[@class='score orange big' or @class='score gray big']")
        rating.append(rat.text)
    except NoSuchElementException:
        rating.append('-')
```

```
    # scraping total review
    try:
        rws = driver.find_element_by_xpath("//div[@class='reviews']")
        reviews.append(rws.text.replace('Total Reviews',''))
    except NoSuchElementException:
        reviews.append('-')


    # fetching over all review
    try:
        overall = driver.find_element_by_xpath("//div[@class='keyword']//span")
        over_all.append(overall.text)
    except NoSuchElementException:
        over_all.append('-')


    # fetching property description
    try:
        disc = driver.find_element_by_xpath("//div[@class='content']")
        description.append(disc.text)
    except NoSuchElementException:
        over_all.append('-')

    # do click on show more button for description
    try:
        driver.find_element_by_xpath("//a[@class='toggle-content']").click()
        time.sleep(4)
    except NoSuchElementException:
        pass
```

In [ ]:
```
print(len(hostel_name),
len(distance),
len(pvt_prices),
len(dorms_price),
len(rating),
len(reviews),
len(over_all),
len(facilities),
len(description),
len(url))
```

In [ ]:
```
# creating DataFrame
Hostel = pd.DataFrame({})
Hostel['Hostel Name'] = hostel_name
Hostel['Distance from City Centre'] = distance
Hostel['Ratings'] = rating
Hostel['Total Reviews'] = reviews
Hostel['Overall Reviews'] = over_all
Hostel['Privates from Price'] = pvt_prices
Hostel['Dorms from Price'] = dorms_price
Hostel['Facilities'] = facilities[:74]
Hostel['Description'] = description
Hostel
```

In [ ]:

In [ ]:

In [ ]: