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()
```

In [3]:
```
driver.get("https://www.amazon.in./")
```

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

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

In [7]:
```
designation=driver.find_element(By.ID,"twotabsearchtextbox" )
designation.send_keys(user_input)
```

In [10]:
```
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]:
```
driver=webdriver.Chrome()
driver.get("https://www.amazon.in./")
```

In [39]:
```
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 [40]:
```
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 [14]:
```
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]:
```
Brand=[]
```

```
Price=[]
Exchange=[]
Delivary=[]
```

In [17]:
```
product_click=driver.find_element(By.XPATH,'//a[@class="a-link-normal s-no-outline"]/div/img')
product_click.click()
```

In [ ]:
```
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]:
```
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 [10]:
```
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 [11]:
```
driver=webdriver.Chrome()
```

In [13]:
```
driver.get("https://www.google.com/")
```

In [14]:
```
search=driver.find_element(By.CLASS_NAME,"gLFyf")
search.send_keys("image.google.com")
```

In [21]:
```
search_button=driver.find_element(By.XPATH,'//div[@class="FPdoLc lJ9FBc"]/center/input ')
search_button.click()
```

In [22]:
```
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 [ ]: