Project Name:-Amazon Web Scrapper

Purpose:-Scrap Description,Price(Rs),Rating, Review Count,Url of the Product for Product Searched from, the Amazon Website.

Importing Necessary Libraries

In [1]:
```
import csv
import pandas as pd
pd.options.display.max_colwidth = 4000
from bs4 import BeautifulSoup
#!pip install selenium
from selenium import webdriver
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

Creating a Chrome Driver for Scrapping

In [2]:
```
#if someone is using this code for reference,then change this chrome driver with your own
driver=webdriver.Chrome("C:\\Mehul\Data Scientist\\Amazon Scrapper\\amazon_scraper\\chrome
```

```
C:\Users\mehul\AppData\Local\Temp/ipykernel_23216/3550596582.py:2: DeprecationWarning: exe
cutable_path has been deprecated, please pass in a Service object
  driver=webdriver.Chrome("C:\\Mehul\Data Scientist\\Amazon Scrapper\\amazon_scraper\\chro
me_driver\\chromedriver_win32\\chromedriver.exe")
```

Passing the amazon.in url

In [3]:
```
url="https://www.amazon.in/"
```

Passing the url in the chrome driver to fetch the amazon website

In [4]:
```
driver.get(url)
```

In [5]:
```
#template="https://www.amazon.in/s?k={}&crid=3JKZD4EXHPZC&sprefix=%2Caps%2C432&ref=nb_sb_s
```

In [6]:
```
#search_term='mi phone'
```

In [7]:
```
#search_term=search_term.replace(' ','+')
```

In [8]:
```
#template.format(search_term)
```

Creating a template url with blank search term & passing the search term in the url for scrapping the Data

In [9]:
```
def get_url(search_term):
    """ Generate a URL for the search term"""
    template="https://www.amazon.in/s?k={}&crid=3JKZD4EXHPZC&sprefix=%2Caps%2C432&ref=nb_s
```

```
    search_term=search_term.replace(' ','+')
    return template.format(search_term)
```

Passing the search term in the above function

In [10]:
```
url=get_url('laptops')
print(url)
```

```
https://www.amazon.in/s?k=laptops&crid=3JKZD4EXHPZC&sprefix=%2Caps%2C432&ref=nb_sb_ss_recent_1_0_recent
```

Passing the whole url generated above into the driver

In [11]:
```
driver.get(url)
```

Extracting the Collection by using Beautiful Soup

In [12]:
```
soup=BeautifulSoup(driver.page_source,'html.parser')
```

Extracting the Information from the search page after entering the our search criteria

In [13]:
```
results=soup.find_all('div',{'data-component-type':'s-search-result'})
```

Checking the Total No of Products on the Search Page

In [14]:
```
len(results)
```

Out[14]:
```
22
```

Scrapping the Description,Price,Rating,Review_Count,Url of the Product for the first record on the search page

In [15]:
```
item=results[0]
```

Scrapping the Description

In [16]:
```
item.h2.a.text.strip()
```

Out[16]:
```
'HP 15s, 11th Gen Intel Core i3 8GB RAM/1TB HDD+256 GB SSD 15.6-inches/39.6 cm FHD Laptop/ Windows 11/Intel UHD Graphics/Dual Speakers/Alexa/MSO/Fast Charge/1.75 Kg, 15s-du3614TU'
```

Scrapping the Price of the Product

In [17]:
```
price_parent=item.find('span','a-price')
```

In [18]:
```
price=price_parent.find('span','a-price-whole').text.strip()
price
```

Out[18]:
```
'42,390'
```

Scrapping the Rating of the Product

In [19]:
```
rating=item.i.text.strip()
```

Scarpping the Review Counts

In [20]:
```
item.find('span',{'class':'a-size-base'}).text
```

Out[20]:
```
'8'
```

Scrapping the Url of the Product

In [21]:
```
atag=item.h2.a
```

In [22]:
```
url="https://www.amazon.in"+atag.get('href')
```

In [ ]:
```

```

Creating a function for moving our scrapper to next pages for the product searched

In [23]:
```
def get_url(search_term):
    """ Generate a URL for the search term"""
    template="https://www.amazon.in/s?k={}&crid=3JKZD4EXHPZC&sprefix=%2Caps%2C432&ref=nb_s
    search_term=search_term.replace(' ','+')

    #add term query to url
    url=template.format(search_term)

    #add page query place holder
    url += '&page{}'

    return url
```

Putting all the above codes together and a creating a final function for scrapping Description,Price,Rating,Review_Count,Url of the Product for all the records and all pages for the product searched

In [24]:
```
def get_url(search_term):
    """ Generate a URL for the search term"""
   # template="https://www.amazon.ae/s?k=salama+care&crid=2WHGND4NAFUT3&sprefix=salama+ca
    template="https://www.amazon.in/s?k={}&crid=3JKZD4EXHPZC&sprefix=%2Caps%2C432&ref=nb_s
    search_term=search_term.replace(' ','+')

    #add term query to url
    url=template.format(search_term)

    #add page query place holder
    url += '&page{}'

    return url
```

We will save our scrapped data into the csv fie having the columns as Description,Price(Rs),Rating,Review Count,Url of the Product

In [25]:

```
def extract_record(item):
    #description&url
    atag=item.h2.a
    description=atag.text.strip()
    url="https://www.amazon.in"+atag.get('href')

    try:

        #price
        price_parent=item.find('span','a-price')
        price=price_parent.find('span','a-price-whole').text.strip()
    except AttributeError:
        return
    try:
        #rank and rating
        rating=item.i.text
        review_count=item.find('span',{'class':'a-size-base'}).text
    except AttributeError:
        rating=''
        review_count=''

    result=(description,price,rating,review_count,url)

    return result



def search_query(search_term):
    """Run main program routine"""
    #startup the webdriver
    #if someone is using this code for reference,then change this chrome driver with your
    driver=webdriver.Chrome("C:\\Mehul\Data Scientist\\Amazon Scrapper\\amazon_scraper\\ch

    records=[]
    url=get_url(search_term)

    for page in range(1,21):
        driver.get(url.format(page))
        soup=BeautifulSoup(driver.page_source,'html.parser')
        results=soup.find_all('div',{'data-component-type':'s-search-result'})

        for item in results:
            record=extract_record(item)
            if record:
                records.append(extract_record(item))

    driver.close()

    filename = search_term + ".csv"
    #save the data to csv file
    with open(filename,"w",newline='',encoding='utf-8') as f:
        writer=csv.writer(f)
        writer.writerow(['Description','Price(Rs)','Rating','Review Count','Url of the Pro
        writer.writerows(records)
```

By using search query function now will be able to scrap the Description,Price(Rs),Rating,Review Count,Url of the Product for the search results

Now we search for Laptops

In [26]:
```
search_query('laptops')
```

```
C:\Users\mehul\AppData\Local\Temp/ipykernel_23216/1439708076.py:33: DeprecationWarning: ex
ecutable_path has been deprecated, please pass in a Service object
  driver=webdriver.Chrome("C:\\Mehul\Data Scientist\\Amazon Scrapper\\amazon_scraper\\chro
me_driver\\chromedriver_win32\\chromedriver.exe")
```

Our scrapped Data will be saved in the csv file in the format given below:-

In [27]:
```
#importinng the csv file in which our scrapped data is saved
pd.read_csv('laptops.csv')
```

Out[27]:

	Description	Price(Rs)	Rating	Review Count	
0	HP 15s, 11th Gen Intel Core i3 8GB RAM/1TB HDD+256 GB SSD 15.6-inches/39.6 cm FHD Laptop/Windows 11/Intel UHD Graphics/Dual Speakers/Alexa/MSO/Fast Charge/1.75 Kg, 15s-du3614TU	42,390	3.8 out of 5 stars	8	https://www.amazon.in/HP-15s-du3614TU%2Fdp%2FB0B6F6PM6C%2Fref%3Dsr_1_1_sspa%3
1	Honor MagicBook 15, AMD Ryzen 5 5500U 15.6-inch (39.62 cm) FHD IPS Anti-Glare Thin and Light Laptop (8GB/256GB PCIe SSD/Windows 11/ Metal Body/Fingerprint Login/1.54Kg), Gray, BohrM-WDQ9CHNE	36,990	4.2 out of 5 stars	13	https://www.amazon.in/sspa/click?WDQ9CHNE%2Fdp%2FB0BC9S5DLW%2Fref%3Dsr_1_2_sspa%3
2	Lenovo IdeaPad Slim 1 AMD Ryzen 5 3500U 15.6" (39.62cm) FHD Thin & Light Laptop (8GB/512GB SSD/Windows 11/Office 2021/3months Game Pass/Cloud Grey/1.6Kg), 82R1004AIN	35,990	3.1 out of 5 stars	15	https://www.amazon.in/Lenovo-IdeaPad-39-62cm-Windows
3	HP Omen 12th Gen Intel Core i7-12700H 16.1 inch(40.9cm)FHD Gaming Laptop(16GB RAM/1TB SSD/GeForce RTX 3070Ti 8GB Graphics/144Hz/7ms Response Time/Win 11/MSO 2021/Backlit/B&O/Alexa/Xbox Pass)16-k0370TX	2,05,900	NaN	NaN	
4	HP NB 255 G8 Laptop 15.6 inch (39.6 cm) AMD Ryzen 3-3250U/8GB DDR4 Ram/1TB HDD/HD/Windows 11/AMD Radeon Vega 8 Graphics/Dark Ash Silver/1.74Kg) 64Q84PA	30,100	3.2 out of 5 stars	105	https://www.amazon.in/HP-3-3250U-Graphics-6

	Description	Price(Rs)	Rating	Review Count	
...	...	...	...	...	
435	HP 14s, 11th Gen Intel Core i3-1115G4, 8GB RAM/256GB SSD 14-inch(35.6 cm) Micro-Edge, Anti-Glare, FHD Laptop/Alexa Built-in/Win 11/Intel UHD Graphics/Dual Speakers/ MSO 2021/1.41 Kg, 14s-dy2507TU	36,990	4.2 out of 5 stars	993	https://www.amazon.in/HP-i3-1115G4-Micro-Edge-Anti-Glare-
436	Acer One 14 Business Laptop AMD Ryzen 3 3250U Processor (8GB RAM/256GB SSD/AMD Radeon Graphics/Windows 11 Home) Z2-493 with 35.56 cm (14.0") HD Display (Rose Gold)	29,990	3.8 out of 5 stars	31	https://www.amazon.in/Acer-Business-Processor-Graph
437	ASUS Vivobook 14, 14.0-inch (35.56 cms) HD, Intel Core i3-1005G1 10th Gen, Thin and Light Laptop (8GB/512GB SSD/Integrated Graphics/Windows 11/Office 2021/Silver/1.60 kg), X415JA-BV322WS	29,990	NaN	NaN	
438	Lenovo IdeaPad Slim 3 Intel Core i3 11th Gen 14" (35.56cm) FHD IPS Thin & Light Laptop (8GB/512GB SSD/Windows 11/Office 2021/Backlit/FPR/2Yr Warranty/3months Game Pass/Arctic Grey/1.41Kg), 82H701DMIN	37,990	4.2 out of 5 stars	253	https://www.amazon 82H701DMIN%2Fdp%2FB09Y5VD8N7%2Fref%3Dsr_1_21_sspa%
439	Dell Vostro 3420 Laptop, Intel i3-1115G4, 8GB DDR4 & 256GB SSD, Win 11 + MSO'21, 14.0", 35.56Cms FHD WVA AG 250 nits, Carbon Black (D552299WIN9BE, 1.48Kgs) Windows 11	36,990	2.6 out of 5 stars	3	https://www.amazon.in/sspa/ 48Kgs%2Fdp%2FB0B469F1PN%2Fref%3Dsr_1_22_sspa%3F

440 rows × 5 columns

In [ ]: