

Web Scraping Project-2 (Selenium):

In [3]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
!pip install selenium
```

```
Requirement already satisfied: selenium in c:\users\user\anaconda3\lib\site-packages (4.5.0)
Requirement already satisfied: urllib3[socks]~=1.26 in c:\users\user\anaconda3\lib\site-packages (from selenium) (1.26.4)
Requirement already satisfied: trio~=0.17 in c:\users\user\anaconda3\lib\site-packages (from selenium) (0.22.0)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\user\anaconda3\lib\site-packages (from selenium) (0.9.2)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\user\anaconda3\lib\site-packages (from selenium) (2022.9.24)
Requirement already satisfied: sortedcontainers in c:\users\user\anaconda3\lib\site-packages (from trio~=0.17->selenium) (2.3.0)
Requirement already satisfied: idna in c:\users\user\anaconda3\lib\site-packages (from trio~=0.17->selenium) (2.10)
Requirement already satisfied: async-generator>=1.9 in c:\users\user\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.10)
Requirement already satisfied: cffi>=1.14 in c:\users\user\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.14.5)
Requirement already satisfied: exceptiongroup>=1.0.0rc9 in c:\users\user\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.0.0rc9)
Requirement already satisfied: sniffio in c:\users\user\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.2.0)
Requirement already satisfied: attrs>=19.2.0 in c:\users\user\anaconda3\lib\site-packages (from trio~=0.17->selenium) (20.3.0)
Requirement already satisfied: outcome in c:\users\user\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.2.0)
Requirement already satisfied: wsproto>=0.14 in c:\users\user\anaconda3\lib\site-packages (from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\users\user\anaconda3\lib\site-packages (from urllib3[socks]~=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\user\anaconda3\lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.20)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\user\anaconda3\lib\site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)
```

In [4]:

```
# Importing Required Libraries

import selenium
import pandas as pd
import numpy as np
from selenium import webdriver
from selenium.common.exceptions import StaleElementReferenceException, NoSuchElementException
from selenium.webdriver.common.by import By
import time
```

Now we will download the webDriver for the Web Browser. Steps for download are- 1.Check the version of your browser. 2.go to the link [\(https://chromedriver.chromium.org/downloads\)](https://chromedriver.chromium.org/downloads) 3.Download the webdriver for your version of your browser.

In [65]:

```
# First connect to the driver

driver=webdriver.Chrome(r"C:\Users\User\Downloads\chromedriver_win32\chromedriver.exe")
```

In []:

```
#Q1: Write a python program to scrape data for “Data Analyst” Job position in “Bangalore” L
#have to scrape the job-title, job-Location, company_name, experience_required. You have to
#jobs data.
#This task will be done in following steps:
#1. First get the webpage https://www.naukri.com/
#2. Enter “Data Analyst” in “Skill, Designations, Companies” field and enter “Bangalore” in
#Location” field.
#3. Then click the search button.
#4. Then scrape the data for the first 10 jobs results you get.
#5. Finally create a dataframe of the scraped data.
```

In [5]:

```
# Opening the naukri page on automated chrome browser

driver.get("https://www.naukri.com/")
```

In [8]:

```
# entering the destination and location as required in the question

designation=driver.find_element(By.CLASS_NAME,"suggestor-input ")
designation.send_keys('Data Analyst')
```

In [9]:

```
location=driver.find_element(By.XPATH,"/html/body/div[1]/div[6]/div/div/div[5]/div/div/div/
designation.send_keys('Bangalore')
```

In [10]:

```
search=driver.find_element(By.CLASS_NAME, "qsbSubmit")
search.click()
```

In [11]:

```
job_title=[]
job_location=[]
company_name=[]
experience_required=[ ]
```

In [12]:

```
# scraping job title
title_tags=driver.find_elements(By.XPATH, '//a[@class="title fw500 ellipsis"]')
for i in title_tags[0:10]:
    title=i.text
    job_title.append(title)

# scraping job location
location_tags=driver.find_elements(By.XPATH, '//li[@class="fleft grey-text br2 placeHolderLi'])
for i in location_tags[0:10]:
    location=i.text
    job_location.append(location)

# scraping company name
company_tags=driver.find_elements(By.XPATH, '//a[@class="subTitle ellipsis fleft"]')
for i in company_tags[0:10]:
    company=i.text
    company_name.append(company)

#scraping experience
experience_tags=driver.find_elements(By.XPATH, '//li[@class="fleft grey-text br2 placeHolderLi'])
for i in experience_tags[0:10]:
    exp=i.text
    experience_required.append(exp)
```

In [16]:

```
job_title
```

Out[16]:

```
['Senior Data Analyst',
 'Data Analyst',
 'Hiring For Data Analyst (DA)/ Team Lead (TL) -Clinical Data management',
 'Call For Clinical Data Analyst - Hyd/Bangalore/Pune/Mumbai/Kolkata',
 'Payroll Transformation Data Analyst',
 'Payroll Transformation Data Analyst',
 'Master Data Management Business Analyst',
 'Data Analytics and Interpretation Business Analyst',
 'HR Data Analyst',
 'Data Analyst']
```

In [13]:

```
job_location
```

Out[13]:

```
['Bangalore/Bengaluru, Chennai',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru, Kolkata, Hyderabad/Secunderabad, Pune, Mumbai (All Areas)',
 'Bangalore/Bengaluru, Kolkata, Hyderabad/Secunderabad, Pune, Mumbai (All Areas)',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru']
```

In [14]:

```
company_name
```

Out[14]:

```
['Latentview',
 'Jar',
 'Cognizant',
 'Cognizant',
 'Arrow Electronics',
 'Arrow Electronics',
 'Accenture',
 'Accenture',
 'Hitachi Ltd.',
 'ANZ']
```

In [15]:

```
experience_required
```

Out[15]:

```
['3-6 Yrs',
 '0-4 Yrs',
 '3-8 Yrs',
 '6-9 Yrs',
 '3-7 Yrs',
 '5-10 Yrs',
 '6-8 Yrs',
 '6-8 Yrs',
 '3-6 Yrs',
 '7-12 Yrs']
```

In [17]:

```
print(len(job_title),len(job_location),len(company_name),len(experience_required))
```

```
10 10 10 10
```

In [18]:

```
df=pd.DataFrame({'Job':job_title,'Location':job_location,'Company':company_name,'Experience':Experience})
```

Out[18]:

	Job	Location	Company	Experience
0	Senior Data Analyst	Bangalore/Bengaluru, Chennai	Latentview	3-6 Yrs
1	Data Analyst	Bangalore/Bengaluru	Jar	0-4 Yrs
2	Hiring For Data Analyst (DA)/ Team Lead (TL) -...	Bangalore/Bengaluru, Kolkata, Hyderabad/Secund...	Cognizant	3-8 Yrs
3	Call For Clinical Data Analyst - Hyd/Bangalore...	Bangalore/Bengaluru, Kolkata, Hyderabad/Secund...	Cognizant	6-9 Yrs
4	Payroll Transformation Data Analyst	Bangalore/Bengaluru	Arrow Electronics	3-7 Yrs
5	Payroll Transformation Data Analyst	Bangalore/Bengaluru	Arrow Electronics	5-10 Yrs
6	Master Data Management Business Analyst	Bangalore/Bengaluru	Accenture	6-8 Yrs
7	Data Analytics and Interpretation Business Ana...	Bangalore/Bengaluru	Accenture	6-8 Yrs
8	HR Data Analyst	Bangalore/Bengaluru	Hitachi Ltd.	3-6 Yrs
9	Data Analyst	Bangalore/Bengaluru	ANZ	7-12 Yrs

In []:

#Q2: Write a python program to scrape data for “Data Scientist” Job position in “Bangalore”
#have to scrape the job-title, job-location, company_name. You have to scrape first 10 jobs
#This task will be done in following steps:
#1. First get the webpage <https://www.naukri.com/>
#2. Enter “Data Scientist” in “Skill, Designations, Companies” field and enter “Bangalore”
#Location” field.
#3. Then click the search button.
#4. Then scrape the data for the first 10 jobs results you get.
#5. Finally create a dataframe of the scraped data.

In [19]:

```
# Opening the naukri page on automated chrome browser
driver.get("https://www.naukri.com/")
```

In [20]:

```
# entering the destination and location as required in the question
designation=driver.find_element(By.CLASS_NAME,"suggestor-input")
designation.send_keys('Data Scientist')
```

In [21]:

```
location=driver.find_element(By.XPATH,"/html/body/div[1]/div[6]/div/div/div[5]/div/div/div/  
designation.send_keys('Bangalore')
```

In [22]:

```
search=driver.find_element(By.CLASS_NAME,"qsbSubmit")  
search.click()
```

In [23]:

```
job_title=[]  
job_location=[]  
company_name=[]
```

In [24]:

```
# scraping job title  
title_tags=driver.find_elements(By.XPATH,'//a[@class="title fw500 ellipsis"]')  
for i in title_tags[0:10]:  
    title=i.text  
    job_title.append(title)  
  
# scraping job location  
location_tags=driver.find_elements(By.XPATH,'//li[@class="fleft grey-text br2 placeHolderLi"]')  
for i in location_tags[0:10]:  
    location=i.text  
    job_location.append(location)  
  
# scraping company name  
company_tags=driver.find_elements(By.XPATH,'//a[@class="subTitle ellipsis fleft"]')  
for i in company_tags[0:10]:  
    company=i.text  
    company_name.append(company)
```

In [25]:

```
job_title
```

Out[25]:

```
['Data Science Specialist',  
'Data Science Manager',  
'Mongodb Database Administrator, Maria DB or Cassandra',  
'Analytics & Modeling Specialist',  
'Assistant Manager - Data Science',  
'Data Scientist',  
'Data Scientist',  
'Senior Data Scientist',  
'Data Scientist: Artificial Intelligence',  
'Industry X - Software Engineering']
```

In [26]:

```
job_location
```

Out[26]:

```
['Bangalore/Bengaluru, Kolkata, Mumbai, Hyderabad/Secunderabad, Pune, Chennai',
 'Bangalore/Bengaluru, Kolkata, Mumbai, Hyderabad/Secunderabad, Pune, Chennai',
 'Bangalore/Bengaluru, Hyderabad/Secunderabad, Pune, Chennai, Delhi / NCR, Mumbai (All Areas)',
 'Bangalore/Bengaluru, Kolkata, Mumbai, Hyderabad/Secunderabad, Pune, Chennai, delhi ncr',
 'Bangalore/Bengaluru, Mumbai, Pune',
 'Bangalore/Bengaluru, New Delhi, Pune, Gurgaon/Gurugram, Delhi / NCR',
 'Bangalore/Bengaluru, Hyderabad/Secunderabad, Pune, Chennai',
 'Bangalore/Bengaluru, Mumbai, New Delhi, Chennai',
 'Bangalore/Bengaluru',
 'Bangalore/Bengaluru']
```

In [27]:

```
company_name
```

Out[27]:

```
['Accenture',
 'Accenture',
 'Mphasis',
 'Accenture',
 'CitiusTech',
 'ZS Associates',
 'Tech Mahindra',
 'Boston Consulting Group',
 'IBM',
 'Accenture']
```

In [28]:

```
print(len(job_title),len(job_location),len(company_name))
```

```
10 10 10
```

In [29]:

```
df2=pd.DataFrame({'Job':job_title,'Location':job_location,'Company':company_name})
df2
```

Out[29]:

	Job	Location	Company
0	Data Science Specialist	Bangalore/Bengaluru, Kolkata, Mumbai, Hyderabad...	Accenture
1	Data Science Manager	Bangalore/Bengaluru, Kolkata, Mumbai, Hyderabad...	Accenture
2	Mongodb Database Administrator, Maria DB or Ca...	Bangalore/Bengaluru, Hyderabad/Secunderabad, P...	Mphasis
3	Analytics & Modeling Specialist	Bangalore/Bengaluru, Kolkata, Mumbai, Hyderabad...	Accenture
4	Assistant Manager - Data Science	Bangalore/Bengaluru, Mumbai, Pune	CitiusTech
5	Data Scientist	Bangalore/Bengaluru, New Delhi, Pune, Gurgaon/...	ZS Associates
6	Data Scientist	Bangalore/Bengaluru, Hyderabad/Secunderabad, P...	Tech Mahindra
7	Senior Data Scientist	Bangalore/Bengaluru, Mumbai, New Delhi, Chennai	Boston Consulting Group
8	Data Scientist: Artificial Intelligence	Bangalore/Bengaluru	IBM
9	Industry X - Software Engineering	Bangalore/Bengaluru	Accenture

In []:

#Q3: You have to use the Location and salary filter.
#You have to scrape data for "Data Scientist" designation for first 10 job results.
#You have to scrape the job-title, job-location, company name, experience required.
#The Location filter to be used is "Delhi/NCR". The salary filter to be used is "3-6" Lakhs

#The task will be done as shown in the below steps:
#1. first get the webpage <https://www.naukri.com/>
#2. Enter "Data Scientist" in "Skill, Designations, and Companies" field.
#3. Then click the search button.
#4. Then apply the location filter and salary filter by checking the respective boxes
#5. Then scrape the data for the first 10 jobs results you get.
#6. Finally create a dataframe of the scraped data.

In [30]:

```
# Opening the naukri page on automated chrome browser
driver.get("https://www.naukri.com/")
```

In [31]:

```
# entering the destination and location as required in the question  
  
designation=driver.find_element(By.CLASS_NAME,"suggestor-input")  
designation.send_keys('Data Scientist')
```

In [32]:

```
location=driver.find_element(By.XPATH,"/html/body/div[1]/div[6]/div/div/div[5]/div/div/div/  
designation.send_keys('Delhi/NCR')
```

In [33]:

```
search=driver.find_element(By.CLASS_NAME,"qsbSubmit")  
search.click()
```

In [34]:

```
job_title=[]  
job_location=[]  
company_name=[]  
experience_required=[]
```

In [35]:

```
# scraping job title  
title_tags=driver.find_elements(By.XPATH,'//a[@class="title fw500 ellipsis"]')  
for i in title_tags[0:10]:  
    title=i.text  
    job_title.append(title)  
  
# scraping job location  
location_tags=driver.find_elements(By.XPATH,'//li[@class="fleft grey-text br2 placeHolderLi"]')  
for i in location_tags[0:10]:  
    location=i.text  
    job_location.append(location)  
  
# scraping company name  
company_tags=driver.find_elements(By.XPATH,'//a[@class="subTitle ellipsis fleft"]')  
for i in company_tags[0:10]:  
    company=i.text  
    company_name.append(company)  
  
#scraping experience  
experience_tags=driver.find_elements(By.XPATH,'//li[@class="fleft grey-text br2 placeHolderLi"]')  
for i in experience_tags[0:10]:  
    exp=i.text  
    experience_required.append(exp)
```

In [36]:

```
job_title
```

Out[36]:

```
['Lead Data Scientist',
 'Data Scientist',
 'DigitalBCG GAMMA Data Scientist',
 'Data Scientist',
 'Data Scientist',
 'Data Scientist / Chat-bot Developer',
 'Data Scientist - Engine Algorithm',
 'Data Activation Specialist - Adobe Target',
 'Data Scientist',
 'Data Scientist']
```

In [37]:

```
job_location
```

Out[37]:

```
['Noida(Sector-59 Noida)\n(WFH during Covid)',
 'Noida, Nagpur, Bangalore/Bengaluru',
 'New Delhi, Bangalore/Bengaluru',
 'Gurgaon/Gurugram',
 'Gurgaon/Gurugram',
 'New Delhi, Bangalore/Bengaluru, Mumbai (All Areas)',
 'Delhi / NCR, Kolkata, Mumbai, Hyderabad/Secunderabad, Lucknow, Chennai, Ahmedabad, Bangalore/Bengaluru',
 'Delhi / NCR, Kolkata, Mumbai, Hyderabad/Secunderabad, Lucknow, Chennai, Ahmedabad, Bangalore/Bengaluru',
 'Noida',
 'Noida']
```

In [38]:

```
company_name
```

Out[38]:

```
['R Systems International',
 'GlobalLogic',
 'Boston Consulting Group',
 'IHS Markit',
 'Optum',
 'Big Seo Buzz',
 'Primo Hiring',
 'Okda Solutions',
 'NGI Ventures',
 'Intrics Solutions']
```

In [39]:

experience_required

Out[39]:

```
['7-10 Yrs',
 '8-10 Yrs',
 '2-5 Yrs',
 '3-6 Yrs',
 '2-7 Yrs',
 '3-7 Yrs',
 '1-3 Yrs',
 '7-10 Yrs',
 '0-5 Yrs',
 '3-8 Yrs']
```

In [40]:

```
print(len(job_title), len(job_location), len(company_name), len(experience_required))
```

10 10 10 10

In [41]:

```
df3=pd.DataFrame({'Job':job_title,'Location':job_location,'Company':company_name,'Experience':experience_required})
df3
```

Out[41]:

	Job	Location	Company	Experience
0	Lead Data Scientist	Noida(Sector-59 Noida)\n(WFH during Covid)	R Systems International	7-10 Yrs
1	Data Scientist	Noida, Nagpur, Bangalore/Bengaluru	GlobalLogic	8-10 Yrs
2	DigitalBCG GAMMA Data Scientist	New Delhi, Bangalore/Bengaluru	Boston Consulting Group	2-5 Yrs
3	Data Scientist	Gurgaon/Gurugram	IHS Markit	3-6 Yrs
4	Data Scientist	Gurgaon/Gurugram	Optum	2-7 Yrs
5	Data Scientist / Chat-bot Developer	New Delhi, Bangalore/Bengaluru, Mumbai (All Ar...	Big Seo Buzz	3-7 Yrs
6	Data Scientist - Engine Algorithm	Delhi / NCR, Kolkata, Mumbai, Hyderabad/Secund...	Primo Hiring	1-3 Yrs
7	Data Activation Specialist - Adobe Target	Delhi / NCR, Kolkata, Mumbai, Hyderabad/Secund...	Okda Solutions	7-10 Yrs
8	Data Scientist	Noida	NGI Ventures	0-5 Yrs
9	Data Scientist	Noida	Intrics Solutions	3-8 Yrs

In []:

```
#Q4: Scrape data of first 100 sunglasses listings on flipkart.com. You have to scrape four
#1. Brand
#2. Product Description
#3. Price

#To scrape the data you have to go through following steps:
#1. Go to Flipkart webpage by url : https://www.flipkart.com/
#2. Enter "sunglasses" in the search field where "search for products, brands and more" is
#click the search icon
#3. After that you will reach to the page having a lot of sunglasses. From this page you can
#required data as usual.
#4. After scraping data from the first page, go to the "Next" Button at the bottom other page
#click on it.
#5. Now scrape data from this page as usual
#6. Repeat this until you get data for 100 sunglasses.
```

In [42]:

```
# Opening the flipkart page on automated chrome browser

driver.get("https://www.flipkart.com/")
```

In [43]:

```
# Entering the Sunglasses in the search box

Sunglass=driver.find_element(By.CLASS_NAME, "_3704LK")
Sunglass.send_keys('Sunglasses')
```

In [44]:

```
search=driver.find_element(By.CLASS_NAME, "L0Z3Pu")
search.click()
```

In [45]:

```
brand=[]
product=[]
price=[]
```

In [54]:

```
start=0
end=3
for page in range(start,end):

    brand_tag=driver.find_elements(By.XPATH,'//div[@class="_2WkVRV"]')
    for i in brand_tag:
        brand.append(i.text)

    product_tag=driver.find_elements(By.XPATH,'//a[@class="IRpwTa"]')
    for j in product_tag:
        product.append(j.text)

    price_tag=driver.find_elements(By.XPATH,'//div[@class="_30jeq3"]')
    for k in price_tag:
        price.append(k.text)

    next_button=driver.find_element(By.XPATH,'//a[@class="_1LKT03"]')
    next_button.click()
    time.sleep(3)
```

In [57]:

```
brand=brand[0:100]
```

In [59]:

```
len(brand)
```

Out[59]:

```
100
```

In [60]:

```
product=product[0:100]
product
```

Out[60]:

```
['Others Rectangular Sunglasses (58)',
 'Others Aviator Sunglasses (57)',
 'UV Protection Wayfarer Sunglasses (Free Size)',
 'UV Protection Round Sunglasses (54)',
 'Others Round Sunglasses (59)',
 'UV Protection Aviator Sunglasses (54)',
 'UV Protection Rectangular Sunglasses (Free Size)',
 'Polarized, UV Protection Round Sunglasses (51)',
 'UV Protection Aviator Sunglasses (57)',
 'UV Protection, Mirrored Wayfarer Sunglasses (54)',
 'UV Protection Wayfarer Sunglasses (Free Size)',
 'Polarized Retro Square Sunglasses (53)',
 'UV Protection, Mirrored Wayfarer Sunglasses (54)',
 'UV Protection Aviator Sunglasses (Free Size)',
 'UV Protection Wayfarer Sunglasses (57)',
 'Others Round Sunglasses (53)',
 'UV Protection Aviator Sunglasses (58)',
 'UV Protection Rectangular Sunglasses (Free Size)']
```

In [61]:

```
price=price[0:100]
price
```

Out[61]:

```
['₹2,999',
 '₹5,049',
 '₹399',
 '₹79',
 '₹79',
 '₹79',
 '₹79',
 '₹899',
 '₹119',
 '₹79',
 '₹79',
 '₹3,079',
 '₹79',
 '₹79',
 '₹117',
 '₹2,199',
 '₹809',
```

In [62]:

```
print(len(brand),len(product),len(price))
```

100 100 100

In [63]:

```
df4=pd.DataFrame({'Brand':brand,'Product Description':product,'Price':price})
df4
```

Out[63]:

	Brand	Product Description	Price
0	CARRERA	Others Rectangular Sunglasses (58)	₹2,999
1	CARRERA	Others Aviator Sunglasses (57)	₹5,049
2	Fastrack	UV Protection Wayfarer Sunglasses (Free Size)	₹399
3	SHAAH COLLECTIONS	UV Protection Round Sunglasses (54)	₹79
4	Bedst	Others Round Sunglasses (59)	₹79
...
95	POLAROID	Polarized Retro Square Sunglasses (53)	₹3,079
96	ROZZETTA CRAFT	UV Protection, Gradient Rectangular Sunglasses...	₹255
97	NuView	UV Protection, Mirrored Aviator Sunglasses (57)	₹118
98	SHAAH COLLECTIONS	UV Protection Round Sunglasses (52)	₹79
99	POLAROID	Polarized Round Sunglasses (51)	₹3,079

100 rows × 3 columns

In []:

```
#Q5: Scrape 100 reviews data from flipkart.com for iphone11 phone.  
#This task will be done in following steps:  
#1. First get the webpage https://www.flipkart.com/  
#2. Enter "iphone 11" in "Search" field .  
#3. Then click the search button.  
  
#As shown in the above page you have to scrape the tick marked attributes.These are:  
#1. Rating  
#2. Review summary  
#3. Full review  
#4. You have to scrape this data for first 100 reviews.
```

In [5]:

```
# Opening the flipkart page on automated chrome browser  
driver.get("https://www.flipkart.com/")
```

In [6]:

```
# Entering the iphone_11 in the search box  
iphone=driver.find_element(By.CLASS_NAME,"_3704LK")  
iphone.send_keys('iphone_11')
```

In [7]:

```
search=driver.find_element(By.CLASS_NAME,"L0Z3Pu")  
search.click()
```

In [8]:

```
rating=[]  
review=[]  
full_review=[]
```

In [23]:

```
start=0
end=11
for page in range(start,end):

    rating_tag=driver.find_elements(By.XPATH,'//div[@class=" _3LWZlK _1BLPMq"]')
    for p in rating_tag:
        rating.append(p.text)

    review_tag=driver.find_elements(By.XPATH,'//p[@class=" _2-N8zT"]')
    for q in review_tag:
        review.append(q.text)

    full_tag=driver.find_elements(By.XPATH,'//div[@class="t-ZTKy"]')
    for r in full_tag:
        full_review.append(r.text)

next_button=driver.find_element(By.XPATH,'//a[@class=" _1LKT03"]')
next_button.click()
time.sleep(3)
```

In [28]:

```
rating=rating[0:100]
rating
```

Out[28]:

```
['5',
 '5',
 '4',
 '5',
 '4',
 '5',
 '5',
 '5',
 '5',
 '5',
 '5',
 '5',
 '5',
 '5',
 '4',
 '5',
 '4',
 '5',
 '5',
 '5',
 '5',
 '5',
 '5',
 '5']
```

In [29]:

```
review=review[0:100]
review
```

Out[29]:

```
['Perfect product!',  
'Terrific',  
'Good quality product',  
'Perfect product!',  
'Value-for-money',  
'Wonderful',  
'Fabulous!',  
'Mind-blowing purchase',  
'Perfect product!',  
'Terrific',  
'Good quality product',  
'Perfect product!',  
'Value-for-money',  
'Wonderful',  
'Fabulous!',  
'Mind-blowing purchase',  
'Perfect product!',  
'Terrific'.
```

In [27]:

```
full_review=full_review[0:100]
full_review
```

Out[27]:

```
["After using 3 years mobile review. Excellent & Awesome Mobile fully I lo  
ve it mobile don't wait take it perfectly suits everyone very stylish and  
operating a...\\nRead full review",  
 'I am using the phone for last 5 years and found no cons as such. The cam  
era quality is bit low as compared to other iphone varient but it can be m  
anagable. B...\\nRead full review',  
 "impressively Nice.....\\nOne of the greatest iPhone i ever used ....\\nAl  
l was like Never before ...\\nit's just Amazing ...\\nBattery Life is too go  
od ...2 Days wi...\\nRead full review",  
 'Nice products thanks flkat',  
 "I'm Really happy with the product\\nDelivery was fast as well\\n..it was a  
gift for my sister and she loved it so much.",  
 'Nice product..just love it',  
 'Fast performance to previous iPhone x\\nGood camera quality but the best  
part of night mode #killing\\n\\nI am already One-plus7\\nHuge difference bet  
ween in night m...\\nRead full review',  
 'Fantastic and prompt delivery.',  
 "After using 3 years mobile review. Excellent & Awesome Mobile fullv I lo
```

In [30]:

```
print(len(rating),len(review),len(full_review))
```

100 100 100

In [31]:

```
df5=pd.DataFrame({'Rating':rating,'Review summary':review,'Full review':full_review})
df5
```

Out[31]:

	Rating	Review summary	Full review
0	5	Perfect product!	After using 3 years mobile review. Excellent &...
1	5	Terrific	I am using the phone for last 5 years and foun...
2	4	Good quality product	impressively Nice.....\nOne of the greatest i...
3	5	Perfect product!	Nice products thanks flkat
4	4	Value-for-money	I'm Really happy with the product\nDelivery wa...
...
95	5	Mind-blowing purchase	Fantastic and prompt delivery.
96	5	Perfect product!	After using 3 years mobile review. Excellent &...
97	5	Terrific	I am using the phone for last 5 years and foun...
98	4	Good quality product	impressively Nice.....\nOne of the greatest i...
99	5	Perfect product!	Nice products thanks flkat

100 rows × 3 columns

In []:

#Q6: Scrape data for first 100 sneakers you find when you visit flipkart.com and search for #search field.
#You have to scrape 4 attributes of each sneaker:
#1. Brand
#2. Product Description
#3. Price

In [33]:

```
# Opening the flipkart page on automated chrome browser
driver.get("https://www.flipkart.com/")
```

In [34]:

```
# Entering the sneakers in the search box
sneaker=driver.find_element(By.CLASS_NAME,"_3704LK")
sneaker.send_keys('sneakers')
```

In [35]:

```
search=driver.find_element(By.CLASS_NAME,"L0Z3Pu")
search.click()
```

In [36]:

```
brand=[]
product=[]
price=[]
discount_shown=[]
```

In [38]:

```
start=0
end=3
for page in range(start,end):

    brand_tag=driver.find_elements(By.XPATH,'//div[@class="_2WkVRV"]')
    for p in brand_tag:
        brand.append(p.text)

    product_tag=driver.find_elements(By.XPATH,'//a[@class="IRpwTa"]')
    for q in product_tag:
        product.append(q.text)

    price_tag=driver.find_elements(By.XPATH,'//div[@class="_30jeq3"]')
    for r in price_tag:
        price.append(r.text)

    discount_tag=driver.find_elements(By.XPATH,'//div[@class="_3Ay6Sb"]')
    for s in discount_tag:
        discount_shown.append(s.text)

next_button=driver.find_element(By.XPATH,'//a[@class="_1LKTO3"]')
next_button.click()
time.sleep(3)
```

In [40]:

```
brand=brand[0:100]
brand
```

Out[40]:

```
['K- FOOTLANCE',
'BIRDE',
'RED TAPE',
'BRUTON',
'BRUTON',
'World Wear Footwear',
'Labbin',
'BIRDE',
'BRUTON',
'Magnolia',
'World Wear Footwear',
'bacca bucci',
'World Wear Footwear',
'TR',
'Kraasa',
'Layasa',
'Magnolia',
'Deals4vou'.
```

In [41]:

```
product=product[0:100]
product
```

Out[41]:

```
['Sneakers For Men',
 'Premium Casual Shoes White Sneakers For Men',
 'Sneakers For Men',
 'Modern Trendy Sneakers Shoes Sneakers For Men',
 'Lightweight Pack Of 1 Trendy Sneakers Sneakers For Men',
 'Latest Collection-1215 Stylish Casual Sports Sneakers F...',
 'Sneakers For Men',
 'Premium White Casual Shoes Sneakers For Men Sneakers Fo...',
 'Latest Sneakers Shoe Sneakers For Men',
 'Sneakers For Men',
 'Latest Collection Black-349 Trendy & Stylish Casual Sne...',
 '"JUPITER Men's Retro Color Blocked Light Weight Chunky F...',
 'Latest Collection-1216 Stylish Casual Sports Sneakers F...',
 'Casuals, Canvas, Partywear Sneakers For Men',
 'Casual Sneakers White Shoes For Girls And Sneakers For ...',
 'Sneakers For Men',
 'Sneakers For Women',
 'Sneakers For Men'.
```

In [42]:

```
price=price[0:100]
price
```

Out[42]:

```
['₹239',
 '₹359',
 '₹899',
 '₹191',
 '₹143',
 '₹143',
 '₹319',
 '₹319',
 '₹195',
 '₹239',
 '₹179',
 '₹1,320',
 '₹143',
 '₹291',
 '₹342',
 '₹399',
 '₹239',
 '₹260']
```

In [43]:

```
discount_shown=discount_shown[0:100]
discount_shown
```

Out[43]:

```
['76% off',
 '76% off',
 '82% off',
 '85% off',
 '76% off',
 '71% off',
 '68% off',
 '78% off',
 '84% off',
 '76% off',
 '64% off',
 '55% off',
 '71% off',
 '80% off',
 '65% off',
 '60% off',
 '76% off',
 '73% off'].
```

In [44]:

```
print(len(brand),len(product),len(price),len(discount_shown))
```

```
100 100 100 100
```

In [45]:

```
df6=pd.DataFrame({'Brand':brand,'Product Description':product,'Price':price,'Discount':discount_shown})
df6
```

Out[45]:

	Brand	Product Description	Price	Discount
0	K-FOOTLANCE	Sneakers For Men	₹239	76% off
1	BIRDE	Premium Casual Shoes White Sneakers For Men	₹359	76% off
2	RED TAPE	Sneakers For Men	₹899	82% off
3	BRUTON	Modern Trendy Sneakers Shoes Sneakers For Men	₹191	85% off
4	BRUTON	Lightweight Pack Of 1 Trendy Sneakers Sneakers...	₹143	76% off
...
95	aadi	350 smart grey lace-ups casual for men Sneaker...	₹239	68% off
96	aadi	Sneakers For Men	₹216	76% off
97	Airland	Exclusive Rap & Dance Shoes For Boys Sneakers ...	₹135	73% off
98	Shozie	2 Combo Sneaker Shoes Sneakers For Men	₹260	67% off
99	Zixer	Men White Sneakers Sneakers For Men	₹450	84% off

100 rows × 4 columns

In []:

```
#Q7: Go to the Link - https://www.myntra.com/shoes  
#Set second Price filter and Color filter to "Black"  
#And then scrape First 100 shoes data you get.  
#The data should include "Brand" of the shoes , Short Shoe  
#description, price of the shoe
```

In [50]:

```
# Opening the myntra page on automated chrome browser  
  
driver.get("https://www.myntra.com/shoes")
```

In [54]:

```
# Entering the shoes in the search box  
  
shoes=driver.find_element(By.CLASS_NAME,"desktop-searchBar")  
shoes.send_keys('shoes')
```

In [55]:

```
search=driver.find_element(By.CLASS_NAME,"desktop-submit")  
search.click()
```

In [101]:

```
brand=[]  
shoe=[]  
price_cm=[]
```

In [110]:

```
start=0  
end=4  
for page in range(start,end):  
  
    brand_tag=driver.find_elements(By.XPATH,'//h3[@class="product-brand"]')  
    for p in brand_tag:  
        brand.append(p.text)  
  
    shoe_tag=driver.find_elements(By.XPATH,'//h4[@class="product-product"]')  
    for q in shoe_tag:  
        shoe.append(q.text)  
  
    price_tag=driver.find_elements(By.XPATH,'//span[@class="product-discountedPrice"]')  
    for r in price_tag:  
        r=r.text  
        price_cm.append(r)  
  
    next_button=driver.find_element(By.XPATH,'//li[@class="pagination-next"]')  
    next_button.click()  
    time.sleep(3)
```

In [112]:

```
brand=brand[0:100]
brand
```

Out[112]:

```
['Skechers',
 'Skechers',
 'Skechers',
 'Puma',
 'Skechers',
 'ADIDAS',
 'Nike',
 'UNDER ARMOUR',
 'ADIDAS',
 'UNDER ARMOUR',
 'Skechers',
 'Nike',
 'Puma',
 'Puma',
 'UNDER ARMOUR',
 'UNDER ARMOUR',
 'Skechers',
 'ADIDAS Originals'].
```

In [113]:

```
shoe=shoe[0:100]
shoe
```

Out[113]:

```
['Men Go Walk Walking Shoes',
 'Men GO WALK - TERRA Shoes',
 'Men ENIGMA Running Shoes',
 'Men Liberate Running Shoes',
 'GO WALK STRETCH FIT Shoes',
 'Men Response Super 3.0 Run',
 'Men Air Max Sneakers',
 'Men HOVR SonicSE Running Shoes',
 'Men Ultraboost 5.0 DNA Running',
 'Men ChargedEscape 3 BL Running',
 'Men Walking Shoes',
 'Men REACT MILER 3 Running Shoe',
 'Men Fast-Trac Nitro Running',
 'Men Magnify Nitro Running',
 'Unisex 3ZERO IV Basketball',
 'Men HOVR Sonic SE Running Shoe',
 'Men Go Run Hyper Burst Running',
 'Women Superstar Sneakers'].
```

In [114]:

```
price_cm=price_cm[0:100]
price_cm
```

Out[114]:

```
['Rs. 6374',
 'Rs. 8499',
 'Rs. 6649',
 'Rs. 9349',
 'Rs. 6374',
 'Rs. 8499',
 'Rs. 10199',
 'Rs. 7199',
 'Rs. 5949',
 'Rs. 7871',
 'Rs. 8799',
 'Rs. 8449',
 'Rs. 7649',
 'Rs. 6999',
 'Rs. 7224',
 'Rs. 6499',
 'Rs. 5999',
 'Rs. 7999'].
```

In [115]:

```
print(len(brand),len(shoe),len(price_cm))
```

100 100 100

In [116]:

```
df7=pd.DataFrame({'Brand':brand,'Shoe description':shoe,'Price':price_cm})
df7
```

Out[116]:

	Brand	Shoe description	Price
0	Skechers	Men Go Walk Walking Shoes	Rs. 6374
1	Skechers	Men GO WALK - TERRA Shoes	Rs. 8499
2	Skechers	Men ENIGMA Running Shoes	Rs. 6649
3	Puma	Men Liberate Running Shoes	Rs. 9349
4	Skechers	GO WALK STRETCH FIT Shoes	Rs. 6374
...
95	Hush Puppies	Men Leather Formal Derbys	Rs. 8099
96	Calvin Klein	Men Sneakers	Rs. 8010
97	Hush Puppies	Solid Formal Leather Monk Shoes	Rs. 6300
98	Geox	Men Textured Leather Slip-On Sneakers	Rs. 6750
99	Saint G	Block Heeled Boots	Rs. 7439

100 rows × 3 columns

In []:

```
#Q8: Go to webpage https://www.amazon.in/  
#Enter "Laptop" in the search field and then click the search icon.  
#Then set CPU Type filter to "Intel Core i7"  
  
#After setting the filters scrape first 10 laptops data. You have to scrape 3 attributes for  
#1. Title  
#2. Ratings  
#3. Price
```

In [11]:

```
# Opening the amazon page on automated chrome browser  
  
driver.get("https://www.amazon.in/")
```

In []:

```
# Entering the laptop in the search box  
  
laptop=driver.find_element(By.CLASS_NAME, "nav-search-field ")  
laptop.send_keys("Laptop")
```

In []:

```
search=driver.find_element(By.CLASS_NAME, "nav-search-submit-text nav-sprite nav-progressive  
search.click()
```

In [62]:

```
title=[]  
rating_rt=[]  
prices=[]
```

In [63]:

```
# scraping title  
title_tags=driver.find_elements(By.XPATH, '//span[@class="a-size-medium a-color-base a-text-  
for i in title_tags[0:10]:  
    title_t=i.text  
    title.append(title_t)  
  
# scraping ratings  
rating_tags=driver.find_elements(By.XPATH, '//div[@class="a-row a-size-small"]')  
for i in rating_tags[0:10]:  
    rt=i.text  
    rating_rt.append(rt)  
  
# scraping price  
price_tags=driver.find_elements(By.XPATH, '//span[@class="a-price"]')  
for i in price_tags[0:10]:  
    pr=i.text  
    prices.append(pr)
```

In [57]:

```
title
```

Out[57]:

```
['ASUS Zenbook 13 OLED, 13.3-inch (33.78 cms) FHD OLED, Intel EVO Core i7-1165G7 11th Gen, Thin and Light Laptop (16GB/1TB SSD/Iris Xe Graphics/Windows 11/Office 2021/Mist/1.14 kg), UX325EA-KG701WS',
 'Lenovo IdeaPad Slim 5 Intel Core i7 12th Gen 15.6" (39.62cm) FHD IPS Thin & Light Laptop (16GB/512GB SSD/Windows 11/Office 2021/Backlit/FPR/3months Game Pass/Storm Grey/1.85Kg), 82SF008WIN',
 'HP Pavilion x360 11th Gen Intel Core i7 14 inch(35.6 cm) FHD,IPS, Multitouch 2-in-1 Laptop(16GB RAM/512GB SSD/Backlit KB/Intel Iris Xe Graphics/Pen/Alexa Built-in/MSO/1.52Kg) 14-dy1050TU, Spruce Blue',
 'ASUS Vivobook 15, 15.6-inch (39.62 cms) FHD, Intel Core i7-1065G7 10th Gen, Thin and Light Laptop (16GB/512GB SSD/Iris Plus Graphics/Windows 11/Office 2021/Silver/1.8 kg), X515JA-EJ701WS',
 'Lenovo ThinkBook 15 Intel 11th Gen Core i7 15.6"(39.62 cm)FHD Thin and Light Laptop (16GB/512GB SSD/Windows 11 Home/MS Office H&S 2021/Iris® Xe Graphics/Backlit/Mineral Grey/1.7 Kg) 20VE00W4IH',
 'Lenovo Yoga 7i 11th Gen Intel Core i7-1165G7 14 inches FHD IPS 2-in-1 Touchscreen Business Laptop (16GB/512GB SSD/Windows 10/MS Office/Digital Pen/Fingerprint Reader/Slate Grey/1.43Kg), 82BH004HIN',
 'HP Pavilion Plus, 12th Gen Intel Core i7 16GB RAM/1TB SSD 14 inch(35.6 cm),OLED,400 nits,UWVA, Eye Safe Laptop/Intel Iris Xe Graphics/Backlit KB/B&O/FPR/Win 11/Alexa Built-in/MSO 2021, 14-eh0024TU',
 'Samsung Galaxy Book2 Pro Intel 12th Gen i7 EvoTM 39.62cm (15.6") AMOLED Thin & Light Laptop (16 GB/512 GB SSD/Windows 11/MS Office/Backlit KB/Fingerprint Sensor/Graphite/1.11Kg), NP950XED-KA1IN',
 'Hp Pavilion X360 11Th Gen Intel Core I7 14 Inches Fhd Multitouch 2In1 Laptop (16Gb Ram/512Gb Ssd/B&O/Windows 11 Home/Fpr/Backlit Kb/Intel Iris Xe Graphics/Pen/Alexa/Ms Office/Silver/1.52Kg)14-Dy1047Tu',
 'Fujitsu UH-X 11th Gen Intel Core i7 13.3" FHD IPS 400Nits 2-in1 Touch Convertible Laptop (16GB/1TB SSD/Windows 11/MS Office 2021/Iris Xe Graphics/2 Yr Warranty/Backlit Kb/Black/0.997 kg, 4ZR1F38027)']
```

In [58]:

```
rating_rt
```

Out[58]:

```
['7', '12', '4', '58', '62', '89', '39', '18', '26', '84']
```

In [64]:

prices

Out[64]:

```
[ '₹91,990',
  '₹80,000',
  '₹80,990',
  '₹64,990',
  '₹77,990',
  '₹1,12,950',
  '₹86,990',
  '₹1,14,990',
  '₹82,490',
  '₹1,00,000']
```

In [65]:

print(len(title),len(rating_rt),len(prices))

10 10 10

In [66]:

```
df8=pd.DataFrame({'Title':title,'Global_Ratings':rating_rt,'Price':prices})
df8
```

Out[66]:

	Title	Global_Ratings	Price
0	ASUS Zenbook 13 OLED, 13.3-inch (33.78 cms) FHD, Intel Core i7 12th Gen, 8GB RAM, 512GB SSD, Windows 11 Pro	7	₹91,990
1	Lenovo IdeaPad Slim 5 Intel Core i7 12th Gen 15.6-inch (39.62 cms) FHD, Intel Iris Xe Graphics, 8GB RAM, 512GB SSD, Windows 11 Pro	12	₹80,000
2	HP Pavilion x360 11th Gen Intel Core i7 14 inch (35.56 cms) FHD, Intel Iris Xe Graphics, 8GB RAM, 512GB SSD, Windows 11 Pro	4	₹80,990
3	ASUS Vivobook 15, 15.6-inch (39.62 cms) FHD, Intel Core i7 11th Gen, 8GB RAM, 512GB SSD, Windows 11 Pro	58	₹64,990
4	Lenovo ThinkBook 15 Intel 11th Gen Core i7 15.6-inch (39.62 cms) FHD, Intel Iris Xe Graphics, 8GB RAM, 512GB SSD, Windows 11 Pro	62	₹77,990
5	Lenovo Yoga 7i 11th Gen Intel Core i7-1165G7 13.3-inch (33.78 cms) FHD, Intel Iris Xe Graphics, 8GB RAM, 512GB SSD, Windows 11 Pro	89	₹1,12,950
6	HP Pavilion Plus, 12th Gen Intel Core i7 16GB RAM, 512GB SSD, 15.6-inch (39.62 cms) FHD, Intel Iris Xe Graphics, Windows 11 Pro	39	₹86,990
7	Samsung Galaxy Book2 Pro Intel 12th Gen i7 Evo 13.3-inch (33.78 cms) FHD, Intel Iris Xe Graphics, 8GB RAM, 512GB SSD, Windows 11 Pro	18	₹1,14,990
8	Hp Pavilion X360 11Th Gen Intel Core I7 14 Inch (35.56 cms) FHD, Intel Iris Xe Graphics, 8GB RAM, 512GB SSD, Windows 11 Pro	26	₹82,490
9	Fujitsu UH-X 11th Gen Intel Core i7 13.3" FHD (33.78 cms) FHD, Intel Iris Xe Graphics, 8GB RAM, 512GB SSD, Windows 11 Pro	84	₹1,00,000

In []:

#Q10: Write a python program to scrape the salary data for Data Scientist designation.
 #You have to scrape Company name, Number of salaries, Average salary, Minsalary, Max Salary
 #The above task will be, done as shown in the below steps:
 #1. First get the webpage <https://www.ambitionbox.com/>
 #2. Click on the salaries option as shown in the image.
 #3. After reaching to the following webpage, In place of "Search Job Profile" enters "Data Scientist".
 #4. Scrape the data for the first 10 companies. Scrape the company name, total salary record, minimum salary, maximum salary, experience required.
 #5. Store the data in a dataframe.

In [66]:

```
# Opening the ambition box page on automated chrome browser
driver.get("https://www.ambitionbox.com/")
```

In [57]:

```
Company=['Walmart', 'Ab Inbev', 'Optum', 'ZS', 'Fractal Analytics', 'Sigmoid Analytics', 'Tiger
         'Legato Health Technolgies', 'HSBC', 'Tredence']
Company
```

Out[57]:

```
['Walmart',
 'Ab Inbev',
 'Optum',
 'ZS',
 'Fractal Analytics',
 'Sigmoid Analytics',
 'Tiger Analytics',
 'Legato Health Technolgies',
 'HSBC',
 'Tredence']
```

In [67]:

```
total_rec=[]
avg_sal=[]
min_sal=[]
max_s=[]
exp_r=[]
```

In [73]:

```
rec_tag=driver.find_elements(By.XPATH,'//span[@class="datapoints"]')
for p in rec_tag:
    total_rec.append(p.text)
```

In [74]:

```
total_rec
```

Out[74]:

```
['(based on 24 salaries)',
 '(based on 59 salaries)',
 '(based on 49 salaries)',
 '(based on 35 salaries)',
 '(based on 118 salaries)',
 '(based on 10 salaries)',
 '(based on 70 salaries)',
 '(based on 11 salaries)',
 '(based on 10 salaries)',
 '(based on 14 salaries)']
```

In [68]:

```
avg_tag=driver.find_elements(By.XPATH,'//p[@class="averageCtc"]')  
for q in avg_tag:  
    avg_sal.append(q.text)
```

In [69]:

```
avg_sal
```

Out[69]:

```
['₹ 32.2L',  
 '₹ 19.8L',  
 '₹ 16.4L',  
 '₹ 15.9L',  
 '₹ 15.5L',  
 '₹ 14.7L',  
 '₹ 14.6L',  
 '₹ 14.5L',  
 '₹ 14.0L',  
 '₹ 13.9L']
```

In [48]:

```
max_tag=driver.find_elements(By.XPATH,'//div[@class="value body-medium"]')  
for s in max_tag:  
    max_s.append(s.text)
```

In [51]:

```
max_s
```

Out[51]:

```
['₹ 25.0L',  
 '₹ 45.0L',  
 '₹ 15.0L',  
 '₹ 26.0L',  
 '₹ 11.0L',  
 '₹ 22.6L',  
 '₹ 11.0L',  
 '₹ 22.0L',  
 '₹ 9.0L',  
 '₹ 23.0L',  
 '₹ 12.7L',  
 '₹ 19.7L',  
 '₹ 9.0L',  
 '₹ 20.0L',  
 '₹ 11.0L',  
 '₹ 20.0L',  
 '₹ 12.0L',  
 '₹ 18.0L',  
 '₹ 8.8L',  
 '₹ 17.5L']
```

In [52]:

```
a=str(max_s)
max_er=[max_s[i] for i in range(len(max_s)) if i%2!=0]
max_er
```

Out[52]:

```
['₹ 45.0L',
 '₹ 26.0L',
 '₹ 22.6L',
 '₹ 22.0L',
 '₹ 23.0L',
 '₹ 19.7L',
 '₹ 20.0L',
 '₹ 20.0L',
 '₹ 18.0L',
 '₹ 17.5L']
```

In [53]:

```
a=str(max_s)
min_er=[max_s[i] for i in range(len(max_s)) if i%2==0]
min_er
```

Out[53]:

```
['₹ 25.0L',
 '₹ 15.0L',
 '₹ 11.0L',
 '₹ 11.0L',
 '₹ 9.0L',
 '₹ 12.7L',
 '₹ 9.0L',
 '₹ 11.0L',
 '₹ 12.0L',
 '₹ 8.8L']
```

In [71]:

```
exp_tg=driver.find_elements(By.XPATH,'//div[@class="sbold-list-header"]')
for j in exp_tg:
    j=j.text.split('(')[0]
    exp_r.append(j)
```

In [72]:

exp_r

Out[72]:

```
['3-4 yrs experience ',
 '2-4 yrs experience ',
 '2-4 yrs experience ',
 '1-2 yrs experience ',
 '2-4 yrs experience ',
 '1 yr experience ',
 '2-4 yrs experience ',
 '4 yrs experience ',
 '4 yrs experience ',
 '3 yrs experience ']
```

In [75]:

```
print(len(Company),len(avg_sal),len(min_er),len(max_er),len(exp_r),len(total_rec))
```

10 10 10 10 10 10

In [76]:

```
df10=pd.DataFrame({'Company_name':Company,'Total_salary_record':total_rec,'Average_salary':avg_sal,
                   'Minimum_salary':min_er,'Maximum_salary':max_er,'Experienced_required':exp_r})
df10
```

Out[76]:

	Company_name	Total_salary_record	Average_salary	Minimum_salary	Maximum_salary	Exprience_required
0	Walmart	(based on 24 salaries)	₹ 32.2L	₹ 25.0L	₹ 45.0L	
1	Ab Inbev	(based on 59 salaries)	₹ 19.8L	₹ 15.0L	₹ 26.0L	
2	Optum	(based on 49 salaries)	₹ 16.4L	₹ 11.0L	₹ 22.6L	
3	ZS	(based on 35 salaries)	₹ 15.9L	₹ 11.0L	₹ 22.0L	
4	Fractal Analytics	(based on 118 salaries)	₹ 15.5L	₹ 9.0L	₹ 23.0L	
5	Sigmoid Analytics	(based on 10 salaries)	₹ 14.7L	₹ 12.7L	₹ 19.7L	
6	Tiger Analytics	(based on 70 salaries)	₹ 14.6L	₹ 9.0L	₹ 20.0L	
7	Legato Health Technologies	(based on 11 salaries)	₹ 14.5L	₹ 11.0L	₹ 20.0L	
8	HSBC	(based on 10 salaries)	₹ 14.0L	₹ 12.0L	₹ 18.0L	
9	Tredence	(based on 14 salaries)	₹ 13.9L	₹ 8.8L	₹ 17.5L	

In []: