```
Name : Dhadge Yash Kailas
Roll No : 9019
Class : BE-IT
Course : Information Storage and Retrieval

Group:C (Assignment-01)
Problem Statement :
    Build the web crawler to pull product information and links from an e-
commerce website. (Python)
```

Importing the necessary libraries

In [4]:

```python
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

URL

In [5]:

```python
baseurl = "https://www.flipkart.com/laptops/~buyback-guarantee-on-laptops-/pr?si
```

Header

In [6]:

```python
#Creating a user agent to make every request (Default user agent may get blocked

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
```

Type *Markdown* and LaTeX: α^2

Request

In [7]:

```
req = requests.get(baseurl).text
soup=BeautifulSoup(req,'html.parser')
print(soup)
```

```
<!DOCTYPE html>
<html lang="en"><head><link href="https://rukminim1.flixcart.com" rel="precon
nect"/><link href="//static-assets-web.flixcart.com/fk-p-linchpin-web/fk-cp-z
ion/css/app_modules.chunk.905c37.css" rel="stylesheet"/><link href="//static-
assets-web.flixcart.com/fk-p-linchpin-web/fk-cp-zion/css/app.chunk.104e9a.cs
s" rel="stylesheet"/><meta content="text/html; charset=utf-8" http-equiv="Con
tent-type"/><meta content="IE=Edge" http-equiv="X-UA-Compatible"/><meta conte
nt="102988293558" property="fb:page_id"/><meta content="658873552,624500995,1
00000233612389" property="fb:admins"/><meta content="noodp" name="robots"/><l
ink href="https://static-assets-web.flixcart.com/www/promos/new/20150528-1405
47-favicon-retina.ico" rel="shortcut icon"/><link href="/osdd.xml?v=2" rel="s
earch" type="application/opensearchdescription+xml"/><meta content="website"
property="og:type"/><meta content="Flipkart.com" name="og_site_name" property
="og:site_name"/><link href="/apple-touch-icon-57x57.png" rel="apple-touch-ic
on" sizes="57x57"/><link href="/apple-touch-icon-72x72.png" rel="apple-touch-
icon" sizes="72x72"/><link href="/apple-touch-icon-114x114.png" rel="apple-to
uch-icon" sizes="114x114"/><link href="/apple-touch-icon-144x144.png" rel="ap
ple-touch-icon" sizes="144x144"/><link href="/apple-touch-icon-57x57.png" rel
="apple-touch-icon"/><meta content="app" name="twitter:card"/><meta content
```

Concating reference URL with BaseURL

In [8]:

```
url_list = soup.find_all('a')
for url in url_list:
  link = url.get('href')
  print(baseurl + link)
```

```
https://www.flipkart.com/laptops/~buyback-guarantee-on-laptops-/pr?sid=6bo%2C
b5g&uniqBStoreParam1=val1&wid=11.productCard.PMU_V2/ (https://www.flipkart.co
m/laptops/~buyback-guarantee-on-laptops-/pr?sid=6bo%2Cb5g&uniqBStoreParam1=va
l1&wid=11.productCard.PMU_V2/)
https://www.flipkart.com/laptops/~buyback-guarantee-on-laptops-/pr?sid=6bo%2C
b5g&uniqBStoreParam1=val1&wid=11.productCard.PMU_V2/plus (https://www.flipkar
t.com/laptops/~buyback-guarantee-on-laptops-/pr?sid=6bo%2Cb5g&uniqBStoreParam
1=val1&wid=11.productCard.PMU_V2/plus)
https://www.flipkart.com/laptops/~buyback-guarantee-on-laptops-/pr?sid=6bo%2C
b5g&uniqBStoreParam1=val1&wid=11.productCard.PMU_V2/account/login?ret=/laptop
s/~buyback-guarantee-on-laptops-/pr%3Fsid%3D6bo%252Cb5g%26uniqBStoreParam1%3D
val1%26wid%3D11.productCard.PMU_V2 (https://www.flipkart.com/laptops/~buyback
-guarantee-on-laptops-/pr?sid=6bo%2Cb5g&uniqBStoreParam1=val1&wid=11.productC
ard.PMU_V2/account/login?ret=/laptops/~buyback-guarantee-on-laptops-/pr%3Fsi
d%3D6bo%252Cb5g%26uniqBStoreParam1%3Dval1%26wid%3D11.productCard.PMU_V2)
https://www.flipkart.com/laptops/~buyback-guarantee-on-laptops-/pr?sid=6bo%2C
b5g&uniqBStoreParam1=val1&wid=11.productCard.PMU_V2https://seller.flipkart.co
m/sell-online/?utm_source=fkwebsite&utm_medium=websitedirect (https://www.fli
pkart.com/laptops/~buyback-guarantee-on-laptops-/pr?sid=6bo%2Cb5g&uniqBStoreP
```

Lists to store name,price,rating of product

In [10]:

```
products=[]
prices=[]
ratings=[]
```

In [11]:

```
product_list = soup.find_all("div",{"class":"_4rR01T"})
for product in product_list:
  products.append(str(product.text))
```

In [12]:

```
price_list = soup.find_all("div",{"class":"_30jeq3 _1_WHN1"})
for price in price_list:
  p = price.text
  prices.append(p)
```

In [13]:

```
rating_list = soup.find_all("div",{"class":"_3LWZlK"})
for rating in rating_list:
  ratings.append(str(rating.text))
```

Displaying Results

In [14]:

```
print("\t\t\tProduct Name" + "\t\t\t\t\t\t\t     "+ "Product price"+"\t "+ "Produ
for i in range(0,len(products)):
  print(products[i],end="\t\t")
  print(prices[i],end="\t\t\t")
  print(ratings[i])
  print("\n")
```

```
                        Product Name
Product price    Product rating
DELL Vostro 3000 Core i5 7th Gen - (8 GB/1 TB HDD/Ubuntu/2 GB Graphics) 3568 La
ptop            ₹42,999                 4.3


Lenovo Core i5 7th Gen - (8 GB/1 TB HDD/DOS/2 GB Graphics) IP 320-15IKB Laptop
₹40,990                 4.2
```

In [ ]: