```
import selenium
import pandas as pd
from selenium import webdriver
import time
from selenium.common.exceptions import
NoSuchElementException,StaleElementReferenceException
from selenium.webdriver.common.keys import Keys
import requests
import re
```

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

```
driver=webdriver.Chrome('chromedriver.exe')
url='https://www.amazon.in/'
driver.get(url)

userip = input('Enter the product you want to search: ')
```

```
Enter the product you want to search: speaker
```

```
search_bar = driver.find_element_by_id("twotabsearchtextbox")
search_bar.clear()
search_bar.send_keys(userip)
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 vertical has less than 3 pages in search results then scrape all the products available under that product vertical. Details to be scraped are: "Brand Name", "Name of the Product", "Rating", "No. of Ratings", "Price", "Return/Exchange", "Expected Delivery", "Availability", "Other Details" and "Product URL". In case, if any of the details are missing for any of the product then replace it by "-".

```
urls = []
for page in range(0,3):
    try:
        page_urls = driver.find_elements_by_xpath('//a[@class="a-link-
normal s-no-outline"]')
```

```
        for url in page_urls:
            url = url.get_attribute('href')
            if url[0:4]=='http':
                urls.append(url)
        print("Product urls of page {} has been
scraped.".format(page+1))

        nxt_button = driver.find_element_by_xpath('//li[@class="a-
last"]/a')
        if nxt_button.text == 'Next→':

            nxt_button.click()

            time.sleep(5)

        elif driver.find_element_by_xpath('//li[@class="a-disabled a-
last"]/a').text == 'Next→':
            print("No new pages exist. Breaking the loop")
            break

    except StaleElementReferenceException as e:
        print("Stale Exception")
        next_page = nxt_button.get_attribute('href')
        driver.get(next_page)
```

```
Product urls of page 1 has been scraped.
Product urls of page 2 has been scraped.
Product urls of page 3 has been scraped.
```

```
Brand_name=[]
Product_name=[]
Product_rating=[]
Product_NoOfRating=[]
Product_price=[]
Product_return=[]
Product_delivery=[]
Product_availability=[]
Product_details=[]
Product_url=[]
for url in urls:
    driver.get(url)

    try:
        brand = driver.find_element_by_xpath('//a[@id="bylineInfo"]')

        Brand_name.append(brand.text)
    except NoSuchElementException:
        Brand_name.append('-')
    try:
```

```
        name = driver.find_element_by_xpath('//h1[@id="title"]/span')

        Product_name.append(name.text)
    except NoSuchElementException:
        Product_name.append('-')
    try:
        rating =
driver.find_element_by_xpath('//span[@id="acrPopover"]')
        Product_rating.append(rating.get_attribute("title"))
    except NoSuchElementException:
        Product_rating.append('-')
    try:
        n_rating =
driver.find_element_by_xpath('//a[@id="acrCustomerReviewLink"]/span')

        Product_NoOfRating.append(n_rating.text)
    except NoSuchElementException:
        Product_NoOfRating.append('-')
    try:
        price =
driver.find_element_by_xpath('//span[@id="priceblock_ourprice"]')
        Product_price.append(price.text)
    except NoSuchElementException:
        Product_price.append('-')
    try:

        ret = driver.find_element_by_xpath('//div[@data-
name="RETURNS_POLICY"]/span/div[2]/a')
        Product_return.append(ret.text)
    except NoSuchElementException:
        Product_return.append('-')
    try:
        delivry =
driver.find_element_by_xpath('//div[@id="ddmDeliveryMessage"]/b')

        Product_delivery.append(delivry.text)
    except NoSuchElementException:
        Product_delivery.append('-')
    try:
        avl =
driver.find_element_by_xpath('//div[@id="availability"]/span')
        Product_availability.append(avl.text)
    except NoSuchElementException:
        Product_availability.append('-')
    try:

        dtls = driver.find_element_by_xpath('//ul[@class="a-unordered-
list a-vertical a-spacing-mini"]')
        Product_details.append('  ||  '.join(dtls.text.split('\n')))
    except NoSuchElementException:
```

```
        Product_details.append('-')
    Product_url.append(url)


product=pd.DataFrame({})
product['Brand']=Brand_name
product['Name']=Product_name
product['Rating']=Product_rating
product['No.of rating']=Product_NoOfRating
product['Price']=Product_price
product['Delivery']=Product_delivery
product['Availability']=Product_availability
product['Details']=Product_details
product['URL']=Product_url

product
```

```
                         Brand  \
0                   Brand: AYG
1      Visit the Tribit Store
2   Visit the ZEBRONICS Store
3        Visit the boAt Store
4   Visit the ZEBRONICS Store
..                        ...
64          Brand: Modernista
65             Brand: Generic
66              Brand: KLUZIE
67  Visit the ZEBRONICS Store
68  Visit the ZEBRONICS Store

                                                 Name
Rating  \
0   AYG Super Ultra Mini Boost Wireless Portable B...
-
1   [Upgraded Version]Tribit XSound Go Wireless Bl...  4.4 out of 5
stars
2   Zebronics Zeb-County Bluetooth Speaker with Bu...  4.0 out of 5
stars
3   boAt Stone 200 IPX6 Waterproof 3W Speaker with...  4.4 out of 5
stars
4   Zebronics Zeb-Warrior 2.0 Multimedia Speaker w...  4.1 out of 5
stars
..                                                ...
...
64  Modernista Sound Box 100 Wireless Bluetooth Sp...  3.8 out of 5
stars
65  Zebion 4W Speaker with Aux Connectivity,USB Po...  5.0 out of 5
stars
66  Best Buy WS-04 Wireless Super 3D BASS Led Ligh...  2.8 out of 5
stars
```

```
67  Zebronics BT6860RUCF 5.1 Bluetooth Speakers (B...  3.8 out of 5
stars
68  Zebronics Zeb-County Bluetooth Speaker with Bu...  4.0 out of 5
stars

      No.of rating      Price          Delivery Availability  \
0                -    ₹549.00       Aug 24 - 27    In stock.
1    1,064 ratings          -   Tuesday, Aug 24    In stock.
2   20,721 ratings          -   Tuesday, Aug 24    In stock.
3   54,052 ratings  ₹1,199.00   Tuesday, Aug 24    In stock.
4    9,441 ratings    ₹699.00   Tuesday, Aug 24    In stock.
..             ...         ...               ...          ...
64   2,377 ratings          -  Thursday, Aug 26    In stock.
65       2 ratings    ₹340.00       Aug 26 - 27    In stock.
66       7 ratings          -    Sunday, Aug 29    In stock.
67   5,448 ratings  ₹3,899.00   Tuesday, Aug 24    In stock.
68  20,721 ratings          -   Tuesday, Aug 24    In stock.

                                                 Details  \
0   SPECIAL PROMOTIONAL PRICE FOR LIMITED PERIOD. ...
1   〖KILLER AUDIO〗: With crystal highs, crisp mids...
2   Zeb-county is a compact and handy portable spe...
3   Experience the true immersive sound with a pum...
4   Zeb-Warrior is a USB powered 2.0 speaker best ...
..                                                ...
64  Superior Sound Quality- Wherever life takes yo...
65  Zebion Muze Adore is a USB powered 2.0 speaker...
66  Light Wight EASY TO ADJUST - Portable Bluetoot...
67  ZEB-BT6860RUCF is a stylish 5.1 multimedia spe...
68  Zeb-county is a compact and handy portable spe...

                                                  URL
0   https://www.amazon.in/gp/slredirect/picassoRed...
1   https://www.amazon.in/gp/slredirect/picassoRed...
2   https://www.amazon.in/Zebronics-Zeb-County-Blu...
3   https://www.amazon.in/Stone-200-Portable-Bluet...
4   https://www.amazon.in/Zebronics-Zeb-Warrior-Mu...
..                                                ...
64  https://www.amazon.in/Modernista-Wireless-Blue...
65  https://www.amazon.in/Speaker-Connectivity-Pow...
66  https://www.amazon.in/WS-04-Wireless-subwoofer...
67  https://www.amazon.in/gp/slredirect/picassoRed...
68  https://www.amazon.in/gp/slredirect/picassoRed...

[69 rows x 9 columns]
```

```
product.to_csv('product.csv')
```

3.Write a python program to access the search bar and search button on images.google.com and scrape 100 images each for keywords ‘fruits’, ‘cars’ and ‘Machine Learning’.

```
driver=webdriver.Chrome('chromedriver.exe')
url='https://images.google.com/'
driver.get(url)

img_urls = []
img_data = []
search_bar =
driver.find_element_by_xpath('//*[@id="sbtc"]/div/div[2]/input')
search_bar.send_keys("fruits")
search_button = driver.find_element_by_xpath('//*[@id="sbtc"]/button')

search_button.click()
for _ in range(500):
    driver.execute_script("window.scrollBy(0,15000)")
images = driver.find_elements_by_xpath('//img[@class="rg_i Q4LuWd"]')
for image in images:
    source= image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)
len(img_urls)
```

```
164
```

```
for i in range(len(img_urls)):
    if i >= 100:
        break
    response= requests.get(img_urls[i])
print("Scraped 100 images of fruits" .format(i+1, 100))
```

```
Scraped 100 images of fruits
```

```
img_urls = []
img_data = []
srch = driver.find_element_by_xpath('//input[@class="og3lId"]')
srch.send_keys(Keys.CONTROL+"A")
srch.send_keys(Keys.BACKSPACE)
srch.send_keys('cars')
srch.send_keys(Keys.ENTER)
for _ in range(500):
    driver.execute_script("window.scrollBy(0,25000)")
images = driver.find_elements_by_xpath('//img[@class="rg_i Q4LuWd"]')
for image in images:
    source= image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
```

```
            img_urls.append(source)
len(img_urls)
```

```
165
```

```
for i in range(len(img_urls)):
    if i >= 100:
        break
    response= requests.get(img_urls[i])
print("Scraped 100 images of cars" .format(i+1, 100))
```

```
Scraped 100 images of cars
```

```
img_urls = []
img_data = []
srch = driver.find_element_by_xpath('//input[@class="og3lId"]')
srch.send_keys(Keys.CONTROL+"A")
srch.send_keys(Keys.BACKSPACE)
srch.send_keys("Machine Learning")
srch.send_keys(Keys.ENTER)
for _ in range(500):
    driver.execute_script("window.scrollBy(0,35000)")
images = driver.find_elements_by_xpath('//img[@class="rg_i Q4LuWd"]')
for image in images:
    source= image.get_attribute('src')
    if source is not None:
        if(source[0:4] == 'http'):
            img_urls.append(source)
len(img_urls)
```

```
156
```

```
for i in range(len(img_urls)):
    if i >= 100:
        break
    response= requests.get(img_urls[i])
print("Scraped 100 images of Machine Learning" .format(i+1, 100))
```

```
Scraped 100 images of Machine Learning
```

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”, “Display Resolution”, “Processor”, “Processor Cores”, “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.

```
driver=webdriver.Chrome('chromedriver.exe')
url='https://www.flipkart.com/'
driver.get(url)

srch = driver.find_element_by_xpath('//input[@class="_3704LK"]')
srch.send_keys('OPPO F19')
clk1 = driver.find_element_by_xpath('//button[@class="L0Z3Pu"]')
clk1.click()

product_urls = []
urls = driver.find_elements_by_xpath('//a[@class="_1fQZEK"]')
for url in urls:
    product_urls.append(url.get_attribute("href"))

Brand = []
Smartphone = []
Colour = []
RAM = []
Storage = []
PrimaryCamera = []
SecondaryCamera = []
DisplaySize = []
DisplayResolution = []
Processor = []
ProcessorCores = []
BatteryCapacity = []
BatteryType = []
Price = []
URL = []
for url in product_urls:
    driver.get(url)

    URL.append(url)

    time.sleep(2)

    try:
        read_more =
driver.find_element_by_xpath('//button[@class="_2KpZ6l _1FH0tX"]')
```

```
        read_more.click()
    except NoSuchElementException:
        print("Exception Occured. Moving to next page")

    try:
        brand =
driver.find_element_by_xpath('//span[@class="B_NuCI"]')
        Brand.append(brand.text.split()[0])
    except NoSuchElementException:
        Brand.append('-')

    try:
        price = driver.find_element_by_xpath('//div[@class="_30jeq3
_16Jk6d"]')
        Price.append(price.text)
    except NoSuchElementException:
        Price.append('-')

    try:
        name = driver.find_element_by_xpath('//div[@class="_3k-BhJ"]
[1]/table/tbody/tr[3]/td[2]/ul/li')
        Smartphone.append(name.text)
    except NoSuchElementException:
        Smartphone.append('-')

    try:
        color = driver.find_element_by_xpath('//div[@class="_3k-BhJ"]
[1]/table/tbody/tr[4]/td[2]/ul/li')
        Colour.append(color.text)
    except NoSuchElementException:
        Colour.append('-')

    try:
        disp_chk = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][2]/div')
        if disp_chk.text != "Display Features" : raise
NoSuchElementException
        disp_size = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][2]/table[1]/tbody/tr[1]/td[2]/ul/li')
        DisplaySize.append(disp_size.text)
    except NoSuchElementException:
        DisplaySize.append('-')

    try:
        disp_chk = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][2]/div')
        if disp_chk.text != "Display Features" : raise
NoSuchElementException
```

```
        disp_res = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][2]/table[1]/tbody/tr[2]/td[2]/ul/li')
        DisplayResolution.append(disp_res.text)
    except NoSuchElementException:
        DisplayResolution.append('-')

    try:
        pro_chk = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][3]/table[1]/tbody/tr[2]/td[1]')
        if pro_chk.text != "Processor Type" : raise
NoSuchElementException
        processor = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][3]/table[1]/tbody/tr[2]/td[2]/ul/li')
        Processor.append(processor.text)
    except NoSuchElementException:
        Processor.append('-')

    try:

        core_chk = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][3]/table[1]/tbody/tr[3]/td[1]')
        if core_chk.text != "Processor Core" :
            core_chk =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][3]/table[1]/
tbody/tr[2]/td[1]')
            if core_chk.text != "Processor Core" :
                raise NoSuchElementException
            else :
                cores =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][3]/table[1]/
tbody/tr[2]/td[2]/ul/li')
        else :
            cores = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][3]/table[1]/tbody/tr[3]/td[2]/ul/li')
        ProcessorCores.append(cores.text)
    except NoSuchElementException:
        ProcessorCores.append('-')

    try:
        rom = driver.find_element_by_xpath('//div[@class="_3k-BhJ"]
[4]/table[1]/tbody/tr[1]/td[2]/ul/li')
        Storage.append(rom.text)
    except NoSuchElementException:
        Storage.append('-')

    try:
        ram = driver.find_element_by_xpath('//div[@class="_3k-BhJ"]
[4]/table[1]/tbody/tr[2]/td[2]/ul/li')
        RAM.append(ram.text)
```

```
    except NoSuchElementException:
        RAM.append('-')

    try:

        pri_cam = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][5]/table[1]/tbody/tr[2]/td[2]/ul/li')
        PrimaryCamera.append(pri_cam.text)
    except NoSuchElementException:
        PrimaryCamera.append('-')

    try:

        cam_chk = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][5]/table[1]/tbody/tr[6]/td[1]')
        if cam_chk != "Secondary Camera" :
            if driver.find_element_by_xpath('//div[@class="_3k-BhJ"]
[5]/table[1]/tbody/tr[5]/td[1]').text == "Secondary Camera":
                sec_cam =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][5]/table[1]/
tbody/tr[5]/td[2]/ul/li')
            else :
                raise NoSuchElementException
        else :
            sec_cam = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][5]/table[1]/tbody/tr[6]/td[2]/ul/li')
        SecondaryCamera.append(sec_cam.text)
    except NoSuchElementException:
        SecondaryCamera.append('-')

    try:
        if
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][10]/div').text !
= "Battery & Power Features" :
            if driver.find_element_by_xpath('//div[@class="_3k-BhJ"]
[9]/div').text == "Battery & Power Features" :
                bat_chk =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][9]/table/tbody/
tr/td[1]')
                if bat_chk.text != "Battery Capacity" : raise
NoSuchElementException
                bat_cap =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][9]/table/tbody/
tr/td[2]/ul/li')
            elif driver.find_element_by_xpath('//div[@class="_3k-BhJ"]
[8]/div').text == "Battery & Power Features" :
                bat_chk =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][8]/table/tbody/
tr/td[1]')
```

```
                if bat_chk.text != "Battery Capacity" : raise
NoSuchElementException
                bat_cap =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][8]/table/tbody/
tr/td[2]/ul/li')
            else:
                raise NoSuchElementException
        else :
            bat_chk = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][10]/table/tbody/tr/td[1]')
            if bat_chk.text != "Battery Capacity" : raise
NoSuchElementException
            bat_cap = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][10]/table/tbody/tr/td[2]/ul/li')
        BatteryCapacity.append(bat_cap.text)
    except NoSuchElementException:
        BatteryCapacity.append('-')

    try:
        if
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][10]/div').text !
= "Battery & Power Features" :
            if driver.find_element_by_xpath('//div[@class="_3k-BhJ"]
[9]/div').text == "Battery & Power Features" :
                bat_chk =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][9]/table/tbody/
tr[2]/td[1]')
                if bat_chk.text != "Battery Type" : raise
NoSuchElementException
                bat_typ =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][9]/table/tbody/
tr[2]/td[2]/ul/li')
            elif driver.find_element_by_xpath('//div[@class="_3k-BhJ"]
[8]/div').text == "Battery & Power Features" :
                bat_chk =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][8]/table/tbody/
tr[2]/td[1]')
                if bat_chk.text != "Battery Type" : raise
NoSuchElementException
                bat_typ =
driver.find_element_by_xpath('//div[@class="_3k-BhJ"][8]/table/tbody/
tr[2]/td[2]/ul/li')
            else:
                raise NoSuchElementException
        else :
            bat_chk = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][10]/table/tbody/tr[2]/td[1]')
            if bat_chk.text != "Battery Type" : raise
NoSuchElementException
```

```
            bat_typ = driver.find_element_by_xpath('//div[@class="_3k-
BhJ"][10]/table/tbody/tr[2]/td[2]/ul/li')
        BatteryType.append(bat_typ.text)
    except NoSuchElementException:
        BatteryType.append('-')

product=pd.DataFrame()
product["Brand"] = Brand
product["Smartphone"] =Smartphone
product["Colour"] =Colour
product["RAM"] =RAM
product["Storage(ROM)"] =Storage
product["Primary Camera"] =PrimaryCamera
product["Secondary Camera"] =SecondaryCamera
product["Display Size"] =DisplaySize
product["Display Resolution"] =DisplayResolution
product["Processor"] =Processor
product["Processor Cores"] =ProcessorCores
product["Battery Capacity"] =BatteryCapacity
product["Battery Type"] =BatteryType
product["Price"] =Price
product["URL"] =URL
product
```

	Brand	Smartphone	Colour	\
0	OPPO	F19	Space Silver	
1	OPPO	F19	Midnight Blue	
2	OPPO	F19	Prism Black	
3	OPPO	F19 Pro	Fantastic Purple	
4	OPPO	F19 Pro	Fantastic Purple	
5	OPPO	F19 Pro	Fluid Black	
6	OPPO	F19 Pro	Crystal Silver	
7	OPPO	F19 Pro	Crystal Silver	
8	OPPO	F19 Pro+ 5G	Space Silver	
9	OPPO	F19 Pro+ 5G	Fluid Black	
10	OPPO	F19 Pro	Fluid Black	
11	OPPO	F1S	Gold	
12	OPPO	F3	Rose Gold	
13	OPPO	F3 Deepika Padukone Limited Edition	Rose Gold	
14	OPPO	F3	Gold	
15	OPPO	F3	Black	
16	OPPO	-	-	
17	OPPO	F1S	Grey	

	RAM	Storage(ROM)	Primary Camera	\
0	6 GB	128 GB	48MP + 2MP + 2MP	
1	6 GB	128 GB	48MP + 2MP + 2MP	

```
2                       6 GB                128 GB        48MP + 2MP +
2MP
3                       8 GB                128 GB  48MP + 8MP + 2MP +
2MP
4                       8 GB                256 GB  48MP + 8MP + 2MP +
2MP
5                       8 GB                128 GB  48MP + 8MP + 2MP +
2MP
6                       8 GB                128 GB  48MP + 8MP + 2MP +
2MP
7                       8 GB                256 GB  48MP + 8MP + 2MP +
2MP
8                       8 GB                128 GB  48MP + 8MP + 2MP +
2MP
9                       8 GB                128 GB  48MP + 8MP + 2MP +
2MP
10                      8 GB                256 GB  48MP + 8MP + 2MP +
2MP
11                      3 GB                 32 GB        13MP Rear
Camera
12                      4 GB                 64 GB        13MP Rear
Camera
13                      4 GB                 64 GB        13MP Rear
Camera
14                      4 GB                 64 GB        13MP Rear
Camera
15                      4 GB                 64 GB        13MP Rear
Camera
16  Qualcomm Snapdragon 616  Android Lollipop 5.1                    3
GB
17                      3 GB                 32 GB        13MP Rear
Camera

                  Secondary Camera          Display Size  Display
Resolution  \
0              16MP Front Camera  16.33 cm (6.43 inch)  2400 x 1080
Pixels
1              16MP Front Camera  16.33 cm (6.43 inch)  2400 × 1080
Pixels
2              16MP Front Camera  16.33 cm (6.43 inch)  2400 × 1080
Pixels
3              16MP Front Camera  16.33 cm (6.43 inch)  2400 x 1080
Pixels
4              16MP Front Camera  16.33 cm (6.43 inch)  2400 x 1080
Pixels
5              16MP Front Camera  16.33 cm (6.43 inch)  2400 x 1080
Pixels
6              16MP Front Camera  16.33 cm (6.43 inch)  2400 x 1080
Pixels
```

```
7               16MP Front Camera  16.33 cm (6.43 inch)  2400 x 1080
Pixels
8               16MP Front Camera  16.33 cm (6.43 inch)  2400 x 1080
Pixels
9               16MP Front Camera  16.33 cm (6.43 inch)  2400 x 1080
Pixels
10              16MP Front Camera  16.33 cm (6.43 inch)  2400 x 1080
Pixels
11                              -   13.97 cm (5.5 inch)   1280 x 720
Pixels
12  16MP + 8MP Dual Front Camera   13.97 cm (5.5 inch)  1920 x 1080
pixels
13  16MP + 8MP Dual Front Camera   13.97 cm (5.5 inch)  1920 x 1080
pixels
14  16MP + 8MP Dual Front Camera   13.97 cm (5.5 inch)  1920 x 1080
pixels
15  16MP + 8MP Dual Front Camera   13.97 cm (5.5 inch)  1920 x 1080
pixels
16                              -                     -
-
17                              -   13.97 cm (5.5 inch)          720 x
1280

                               Processor Processor Cores Battery
Capacity  \
0               Qualcomm Snapdragon 662       Octa Core         5000
mAh
1               Qualcomm Snapdragon 662       Octa Core         5000
mAh
2               Qualcomm Snapdragon 662       Octa Core         5000
mAh
3         MediaTek Helio P95 (MT6779V)       Octa Core         4310
mAh
4         MediaTek Helio P95 (MT6779V)       Octa Core         4310
mAh
5         MediaTek Helio P95 (MT6779V)       Octa Core         4310
mAh
6         MediaTek Helio P95 (MT6779V)       Octa Core         4310
mAh
7         MediaTek Helio P95 (MT6779V)       Octa Core         4310
mAh
8               MediaTek Dimensity 800U       Octa Core         4310
mAh
9               MediaTek Dimensity 800U       Octa Core         4310
mAh
10        MediaTek Helio P95 (MT6779V)       Octa Core         4310
mAh
11                                  EOL       Octa Core         3075
mAh
```

```
12                    Mediatek MT6750T       Octa Core          3200
mAh
13  Mediatek MT6750T Octa Core 1.5 GHz       Octa Core          3200
mAh
14   Mediatek MT6750T Octa Core 1.5GHz       Octa Core          3200
mAh
15   Mediatek MT6750T Octa Core 1.5GHz       Octa Core          3200
mAh
16                                     -               -
-
17          Mediatek MT6755 or MTK7650     Single Core          3075
mAh

   Battery Type    Price
URL
0              -  ₹18,990  https://www.flipkart.com/oppo-f19-space-
silver...
1              -  ₹18,990  https://www.flipkart.com/oppo-f19-midnight-
blu...
2              -  ₹18,900  https://www.flipkart.com/oppo-f19-prism-
black-...
3              -  ₹21,990  https://www.flipkart.com/oppo-f19-pro-
fantasti...
4              -  ₹25,990  https://www.flipkart.com/oppo-f19-pro-
fantasti...
5              -  ₹21,990  https://www.flipkart.com/oppo-f19-pro-fluid-
bl...
6              -  ₹21,990  https://www.flipkart.com/oppo-f19-pro-
crystal-...
7              -  ₹23,490  https://www.flipkart.com/oppo-f19-pro-
crystal-...
8              -  ₹25,990  https://www.flipkart.com/oppo-f19-pro-5g-
space...
9              -  ₹25,990  https://www.flipkart.com/oppo-f19-pro-5g-
fluid...
10             -  ₹23,490  https://www.flipkart.com/oppo-f19-pro-fluid-
bl...
11   Li-Polymer  ₹10,990  https://www.flipkart.com/oppo-f1s-gold-32-
gb/p...
12             -  ₹18,000  https://www.flipkart.com/oppo-f3-rose-gold-
64-...
13             -  ₹17,000  https://www.flipkart.com/oppo-f3-deepika-
paduk...
14             -  ₹18,000  https://www.flipkart.com/oppo-f3-gold-64-
gb/p/...
15             -  ₹20,990  https://www.flipkart.com/oppo-f3-black-64-
gb/p...
16             -  ₹15,739  https://www.flipkart.com/oppo-f1-gold-16-
gb/p/...
```

```
17   Li-Polymer  ₹12,990  https://www.flipkart.com/oppo-f1s-grey-32-
gb/p...

product.to_csv('oppo.csv')
```

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

```
driver=webdriver.Chrome('chromedriver.exe')
time.sleep(5)
driver.get('https://www.google.com/maps')
time.sleep(5)
driver.find_element_by_xpath('//input[@id="searchboxinput"]').send_key
s(input('Enter city to search: '))
driver.find_element_by_xpath('//button[@class="searchbox-
searchbutton"]').click()
time.sleep(5)
url = driver.current_url
pattern = re.compile(r'\-?[0-9.]+\,\-?[0-9.]+')
matchy = pattern.finditer(url)
for m in matchy:
    loc=str(m).split()[-1][7:-2]
    print('latitude:',loc.split(',')[0])
    print('longtitude:',loc.split(',')[1])

Enter city to search: Mumbai
latitude: 19.0514876
longtitude: 72.8893351
```

6.Write a program to scrap details of all the funding deals for second quarter (i.e. July 20 –September 20) from trak.in.

```
driver=webdriver.Chrome('chromedriver.exe')
url='https://trak.in/'
driver.get(url)

button =
driver.find_element_by_xpath('//li[@id="menu-item-51510"]/a').get_attr
ibute('href')
driver.get(button)

Date=[]
Startup=[]
Industry=[]
SubVertical=[]
Location=[]
Investor=[]
Investment=[]
Amount=[]
```

```
for i in range(57,60):
    dt = driver.find_elements_by_xpath('//table[@id="tablepress-
{}"]/tbody/tr/td[2]'.format(i))
    for d in dt:
        Date.append(d.text)

    sn = driver.find_elements_by_xpath('//table[@id="tablepress-
{}"]/tbody/tr/td[3]'.format(i))
    for n in sn:
        Startup.append(n.text)

    ind = driver.find_elements_by_xpath('//table[@id="tablepress-
{}"]/tbody/tr/td[4]'.format(i))
    for n in ind:
        Industry.append(n.text)

    sv = driver.find_elements_by_xpath('//table[@id="tablepress-
{}"]/tbody/tr/td[5]'.format(i))
    for s in sv:
        SubVertical.append(s.text)

    loc = driver.find_elements_by_xpath('//table[@id="tablepress-
{}"]/tbody/tr/td[6]'.format(i))
    for l in loc:
        Location.append(l.text)

    inv = driver.find_elements_by_xpath('//table[@id="tablepress-
{}"]/tbody/tr/td[7]'.format(i))
    for n in inv:
        Investor.append(n.text)

    invt = driver.find_elements_by_xpath('//table[@id="tablepress-
{}"]/tbody/tr/td[8]'.format(i))
    for n in invt:
        Investment.append(n.text)

    amt = driver.find_elements_by_xpath('//table[@id="tablepress-
{}"]/tbody/tr/td[9]'.format(i))
    for a in amt:
        Amount.append(a.text)

fund=pd.DataFrame()
fund['Date']=Date
fund['Startup']=Startup
fund['Industry']=Industry
fund['SubVertical']=SubVertical
fund['Location']=Location
fund['Investor']=Investor
fund['Investment']=Investment
```

```
fund['Amount']=Amount
fund
```

```
         Date  Startup                Industry
SubVertical  \
0  01/04/2021   BYJU’S                Edu-tech             Online
tutoring
1  05/04/2021   Meesho             E-commerce  Online reselling
platform
2  14/04/2021   Swiggy  Online Food Delivery       Online Food
Delivery
3  07/04/2021    Groww                FinTech        Investment
platform
4  14/04/2021  Beldara             E-commerce     Global B2B
marketplace

    Location                                              Investor
Investment  \
0  Bengaluru                                      Innoven Capital
Series F
1  Bengaluru                                SoftBank Vision Fund 2
Series E
2  Bengaluru  Amansa Holdings, Carmignac, Falcon Edge Capita...
Series J
3  Bengaluru         MC Global Edtech, B Capital, Baron, others
Series D
4     Mumbai                            Hindustan Media Ventures
Venture

        Amount
0  460,000,000
1  300,000,000
2  343,000,000
3   83,000,000
4    7,400,000
```

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

```
driver=webdriver.Chrome('chromedriver.exe')
url='https://www.digit.in/'
driver.get(url)

top_10=driver.find_element_by_xpath("/html/body/div[1]/div/div[4]/ul/
li[4]/a")
top_10.click()
time.sleep(2)
laptops=driver.find_element_by_xpath("/html/body/div[3]/div/div/div[2]
/div[5]/div[1]/div/button[2]")
laptops.click()
```

```
best_gaming=driver.find_element_by_xpath("//div[@id='laptops']//
div[3]//a")
driver.get(best_gaming.get_attribute('href'))

name = []
Price = []
OS = []
display = []
processor = []
HDD = []
RAM = []
weight = []
dimension = []
GPU = []
time.sleep(1)
names=driver.find_elements_by_xpath("//div[@class='right-container']/
div/a/h3")
for i in names:
    name.append(i.text)
time.sleep(1)
os=driver.find_elements_by_xpath("//div[@class='product-detail']/div/
ul/li[1]/div/div")
for i in os:
    OS.append(i.text)
time.sleep(1)
displays=driver.find_elements_by_xpath("//div[@class='product-
detail']/div/ul/li[2]/div/div")
for i in displays:
    display.append(i.text)
time.sleep(1)
processors=driver.find_elements_by_xpath("//div[@class='product-
detail']/div/ul/li[3]/div/div")
for i in processors:
    processor.append(i.text)
time.sleep(1)
memories=driver.find_elements_by_xpath("//div[@class='Spcs-details']
[1]/table/tbody/tr[6]/td[3]")# extrat HDD and RAM form xpath
for i in memories:
    HDD.append(i.text.split("/")[0])
    RAM.append(i.text.split("/")[1])
time.sleep(1)
weights=driver.find_elements_by_xpath("//div[@class='Spcs-details']
[1]/table/tbody/tr[7]/td[3]")# extrat weight form xpath
for i in weights:
    weight.append(i.text)
time.sleep(1)
dimension=[]
dimensions=driver.find_elements_by_xpath("//div[@class='Spcs-details']
[1]/table/tbody/tr[8]/td[3]")
for i in dimensions:
```

```
    dimension.append(i.text)
time.sleep(1)
GPUs=driver.find_elements_by_xpath("//div[@class='Spcs-details'][1]/
table/tbody/tr[9]/td[3]")
for i in GPUs:
    GPU.append(i.text)
time.sleep(1)
price=driver.find_elements_by_xpath("//table[@id='summtable']//tr//
td[3]")
for i in price:
    Price.append(i.text)

df=pd.DataFrame({"Name":name,
                 "price":Price,
                 "OS":OS,
                 "Display":display,
                 "HDD":HDD,
                  "RAM":RAM,
                 "processor":processor,
                 "weight":weight,
                 "Dimension":dimension,
                 "Graphical processor":GPU})
df
```

```
                        Name    price                OS
Display  \
0     ALIENWARE AREA 51M R2      N/A  WINDOWS 10 HOME  17.3" (1920 X
1080)
1          ALIENWARE M15 R3  ₹341990  WINDOWS 10 HOME  15.6" (3840 X
2160)
2    ASUS ROG STRIX SCAR 15      N/A  WINDOWS 10 HOME  15.6" (1920 X
1080)
3     ASUS ROG ZEPHYRUS G14  ₹164990  WINDOWS 10 HOME    14" (1920 X
1080)
4          LENOVO LEGION 5I   ₹76988   WINDOWS 10 PRO  15.6" (1920 X
1080)
5  ASUS ROG ZEPHYRUS DUO 15  ₹185000       WINDOWS 10  15.6" (3840 X
1100)
6      ACER ASPIRE 7 GAMING   ₹64370  WINDOWS 10 HOME  15.6" (1920 X
1080)

           HDD            RAM  \
0    1 TB SSD  16 GBGB DDR4
1    1 TB SSD  16 GBGB DDR4
2    1 TB SSD  16 GBGB DDR4
3    1 TB SSD  16 GBGB DDR4
4    1 TB SSD  16 GBGB DDR4
5  512 GB SSD   4 GBGB DDR4
6  512 GB SSD   8 GBGB DDR4
```

```
                                              processor weight  \
0   10TH GENERATION INTEL® CORE™ I7-10700 | 2.90 GHZ    4.1
1       10TH GENERATION INTEL® CORE™ I9-10980HK | NA     NA
2                     AMD RYZEN™ 9 5900HX | 3.3 GHZ   2.30
3               AMD 3RD GENERATION RYZEN 9 | 3.3 GHZ   1.65
4  10TH GENERATION INTEL® CORE™ I5-10300H | 2.50 GHZ    2.3
5                 INTEL CORE I7 10TH GEN 10875H | NA    2.4
6                  AMD RYZEN™ 5-5500U HEXA-CORE | NA   2.15

                  Dimension            Graphical processor
0   27.65 x 402.6 x 319.14        Intel® UHD Graphics 630
1                       NA                             NA
2       35.4 x 25.9 x 2.26      NVIDIA® GeForce RTX™ 3070
3        32.5 x 22.1 x 1.8        NVIDIA GeForce RTX 2060
4  363.06 x 259.61 x 23.57  NVIDIA® GeForce® GTX 1650 4GB
5  268.30 x 360.00 x 20.90  NVIDIA GeForce RTX 2070 Max-Q
6       2.29 x 36.3 x 25.4      NVIDIA® GeForce® GTX 1650
```

8.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”.

```
driver=webdriver.Chrome('chromedriver.exe')
url='https://www.forbes.com/?sh=1d7fec302254'
driver.get(url)

driver.find_element_by_xpath('//button[@class="icon--
hamburger"]').click()
time.sleep(1)

from selenium.webdriver.common.action_chains import ActionChains
action = ActionChains(driver)
hov = driver.find_element_by_xpath('//a[@class="header__title"]')
action.move_to_element(hov).perform()
time.sleep(1)
clk = driver.find_element_by_xpath('//a[@class="section__link"]')
clk.click()

rank = []
name = []
networth = []
age = []
citizenship = []
source = []
industry = []
for _ in range(0,15):
    nm = driver.find_elements_by_xpath('//div[@class="personName"]')
```

```
    rnk = driver.find_elements_by_xpath('//div[@class="rank"]')
    wor = driver.find_elements_by_xpath('//div[@class="netWorth"]')
    ag = driver.find_elements_by_xpath('//div[@class="age"]')
    citi =
driver.find_elements_by_xpath('//div[@class="countryOfCitizenship"]')
    sou = driver.find_elements_by_xpath('//div[@class="source-
column"]')
    ind = driver.find_elements_by_xpath('//div[@class="category"]')
    for r in rnk:
        rank.append(r.text.replace('.',''))
    for n in nm:
        name.append(n.text)
    for w in wor:
        networth.append(w.text)
    for a in ag:
        age.append(a.text)
    for c in citi:
        citizenship.append(c.text)
    for s in sou:
        source.append(s.text)
    for i in ind:
        industry.append(i.text)
    try:
        nxt =
driver.find_element_by_xpath('//button[@class="pagination-btn
pagination-btn--next "]')
        nxt.click()
    except:
        pass

billionaires = pd.DataFrame()
billionaires['Rank'] = rank
billionaires['Name'] = name
billionaires['NetWorth'] = networth
billionaires['Age'] = age
billionaires['Citizenship']  = citizenship
billionaires['Source'] =  source
billionaires['Industry'] = industry
billionaires
```

	Rank	Name	NetWorth	Age	Citizenship	\
0	1	Jeff Bezos	$177 B	57	United States	
1	2	Elon Musk	$151 B	49	United States	
2	3	Bernard Arnault & family	$150 B	72	France	
3	4	Bill Gates	$124 B	65	United States	
4	5	Mark Zuckerberg	$97 B	36	United States	
...	...	...	...	..	...	
2905	2674	Daniel Yong Zhang	$1 B	49	China	
2906	2674	Zhang Yuqiang	$1 B	65	China	
2907	2674	Zhao Meiguang	$1 B	58	China	

```
2908  2674            Zhong Naixiong     $1 B  58          China
2909  2674          Zhou Wei family     $1 B  54          China

              Source          Industry
0             Amazon        Technology
1     Tesla, SpaceX        Automotive
2               LVMH  Fashion & Retail
3          Microsoft        Technology
4           Facebook        Technology
...              ...               ...
2905     e-commerce        Technology
2906     Fiberglass     Manufacturing
2907    gold mining   Metals & Mining
2908   conglomerate       Diversified
2909       Software        Technology

[2910 rows x 7 columns]
```

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

```
driver=webdriver.Chrome('chromedriver.exe')
url='https://www.youtube.com'
driver.get(url)

search_bar = driver.find_element_by_id('search')
search_bar.send_keys("Gojo Satoru vs Jogo | Fight Scene")  #entering
Video name
time.sleep(1)

search_btn = driver.find_element_by_id("search-icon-legacy")
search_btn.click()
time.sleep(1)

link_click = driver.find_element_by_xpath("//yt-formatted-
string[@class ='style-scope ytd-video-renderer']")
link_click.click()

for _ in range(1000):
    driver.execute_script("window.scrollBy(0,10000)")

comments = []
comment_time = []
Time = []
Likes = []
No_of_Likes = []

comments = []
comment_time = []
Time = []
```

```
Likes = []
No_of_Likes = []
cm_tags = driver.find_elements_by_id("content-text")
for cm in cm_tags:
    if cm.text is None:
        comments.append("--")
    else:
        comments.append(cm.text)
time.sleep(5)

tm_tags = driver.find_elements_by_xpath("//a[contains(text(),'ago')]")
for tm in tm_tags:
    Time.append(tm.text)

for i in range(0,len(Time),2):
    comment_time.append(Time[i])
time.sleep(5)

like_tags = driver.find_elements_by_xpath("//span[@class='style-scope
ytd-comment-action-buttons-renderer']")
for like in like_tags:
    Likes.append(like.text)

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

Youtube=pd.DataFrame()
Youtube['Comments'] = comments
Youtube['Comment_time'] = comment_time
Youtube['Comment upvotes'] = No_of_Likes

Youtube
```

```
                                                Comments
Comment_time  \
0                It wasn't a fight, it was a massacre.           2
months ago
1   The funny things is Jogo is powerful af he can...   1 month ago
(edited)
2   "it's ironic isn't it? when granted everything...  2 months ago
(edited)
3   Everyone is talking about how good is the anim...           2
months ago
4   I mean jogo’s not weak he just chose the wrong...           2
months ago
5   Gojo finished all side quests, contracts, mino...           5
months ago
6   Okay but let's be real, Jogo's lineup of attac...  2 months ago
(edited)
7   I'm fucking obsessed with this guy. He's so OP...           3
```

```
months ago
8         Gojou: takes off mask\n\n“Oh no he’s hot!!”           2
months ago
9   Gojo merely pulled out his Domain and this man...           2
months ago
10  Gojo is like a main character from another sto...           6
months ago
11  Gojo's domain is so powerful. When a person en...            1
month ago
12  Everybody gangsta till you realize the whole u...           2
months ago
13  Jogo is probably one of the strongest characte...   1 month ago
(edited)
14  I will never get over the fact that Gojo's lip...           2
months ago
15  Fan theory: Gojou only opens his blindfold whe...           5
months ago
16  When you’re max level & going through old miss...           3
months ago
17  Gojo can literally end this whole series in hi...           2
months ago
18                  2:11 gojo be like: THIS IS MY SON           2
months ago
19  Jogos like one of the most OP characters it’s ...              1
day ago

   Comment upvotes
0              11K
1             6.4K
2             6.1K
3               4K
4             2.7K
5              14K
6             2.9K
7             1.9K
8             4.5K
9               1K
10             28K
11             762
12             947
13             481
14             347
15             14K
16             519
17             218
18             600
19               1
```

10.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.

```
driver=webdriver.Chrome("chromedriver.exe")
time.sleep(3)
#get the web page with given url
url = "https://www.hostelworld.com/"
driver.get(url)
time.sleep(5)

search = driver.find_element_by_id('search-input-field')
# write Lonodn in search bar
search.send_keys("London")
time.sleep(5)
#select london
london =
driver.find_element_by_xpath('/html/body/div[1]/div/div/div[1]/div[1]/
div/div[2]/div[4]/div/div[2]/div/div[1]/div/div/ul/li[2]/div')
london.click()
time.sleep(5)
# do click on search button
search_btn = driver.find_element_by_id('search-button')
search_btn.click()

hostel_name = []
distance = []
pvt_prices = []
dorms_price = []
rating = []
reviews = []
over_all = []
facilities = []
description =[]
product_url = []
for i in driver.find_elements_by_xpath("//div[@class = 'pagination-
item pagination-current' or @class='pagination-item']"):
    i.click()
    time.sleep(4)
    #fetching 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('-')
    #fetching distance from city centre

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

    for i in driver.find_elements_by_xpath("//div[@class='prices-
col']"):
    #fetch 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('-')
    #fetching dorms from price
    for i in driver.find_elements_by_xpath("//div[@class='prices-
col']"):
        try:
            dorms =
driver.find_element_by_xpath("//a[@class='prices']//div[2]//div")
            dorms_price.append(dorms.text)
        except NoSuchElementException:
            dorms_price.append('-')
            #fetching 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('-')
    #lets fetch url of each hostel
    p_url = driver.find_elements_by_xpath("//div[@class='prices-
col']//a[2]")
    for i in p_url:
        product_url.append(i.get_attribute('href'))

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

```
    #lets click on show more button for description
    try:
        driver.find_element_by_xpath("//a[@class='toggle-
content']").click()
        time.sleep(5)
    except NoSuchElementException:
        pass
    #fetching 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('-')
    #fetching total reviews

    try:
        rws = driver.find_element_by_xpath("//div[@class='reviews']")
        reviews.append(rws.text.replace('Total Reviews',''))
    except NoSuchElementException:
        reviews.append('-')
        #fetch overall review
    try:
        overall_rw =
driver.find_element_by_xpath("//div[@class='keyword']//span")
        over_all.append(overall_rw.text)
    except NoSuchElementException:
        over_all.append('-')
    #fetch property description
    try:
        disc = driver.find_element_by_xpath("//div[@class='content']")
        description.append(disc.text)
    except NoSuchElementException:
        over_all.append('-')

df = pd.DataFrame()
df['Hostel_Name'] = hostel_name
df['Distance fron city centre'] = distance
df['Ratings'] = rating
df['Total_reviews'] = reviews
df['Overall Reviews'] = over_all
df['Privates from price'] = pvt_prices
df['Dorms from price'] = dorms_price
df['Description'] = description
df
```

```
                        Hostel_Name  \
0            London Waterloo Hostel
1          Barmy Badger Backpackers
2   Wombat's The City Hostel London
```

```
3           Prime Backpackers Angel
4         St Christopher's Village
..                             ...
81                      The Dover
82                Park Hotel Essex
83                 Cranbrook Hotel
84                      St. Athans
85                Aron Guest House

                      Distance fron city centre Ratings Total_reviews
\
0                        0.7km from city centre     7.5          2409

1                        5.5km from city centre      10          1652

2                        3.6km from city centre     9.2         13142

3                        3.6km from city centre      10           513

4                        1.8km from city centre     8.9         10829

..                                          ...     ...           ...

81               Hotel - 1.9km from city centre       -             0

82              Hotel - 24.1km from city centre       -             0

83              Hotel - 14.8km from city centre       -             0

84   Bed and Breakfast - 2.9km from city centre       -             0

85  Bed and Breakfast - 13.1km from city centre       -             0


   Overall Reviews Privates from price Dorms from price  \
0        Very Good              Rs6477           Rs1593
1           Superb              Rs6477           Rs1593
2           Superb              Rs6477           Rs1593
3           Superb              Rs6477           Rs1593
4         Fabulous              Rs6477           Rs1593
..             ...                 ...              ...
81       No Rating              Rs4426           Rs1723
82       No Rating              Rs4426           Rs1723
83       No Rating              Rs4426           Rs1723
84       No Rating              Rs4426           Rs1723
85       No Rating              Rs4426           Rs1723

                                        Description
0            73 Lambeth Walk, London, London, England
1     17 Longridge Road, Earls Court, London, England
2                    7 Dock Street, London, England
```

```
3       333 City Road, 333 City Road, London, England
4            165 Borough High Street, London, England
..                                                 ...
81                  44 Belgrave Road, London, England
82        327 Cranbrook Road, Ilford, London, England
83       22/24 Coventry Road, Ilford, London, England
84  20 Tavistock Place, Russell Square, London, En...
85                   27 South Ealing, London, England

[86 rows x 8 columns]
```