WEB SCRAPING PROJECT

Institute Name: ITVedant Education Pvt Ltd
Name: Sanjana Joshi
Email Address: sanjanaj1097@gmail.com

1. Aim:

To extract key product information including product image, product name, quantity, ratings, number of people who rated, old price, discount and new price. This data will facilitate market analysis, product comparison, and consumer trend insights.

2. Objectives:

● **Data Collection**: Gather detailed information on FIAMA products, including product image, product name, quantity, ratings, number of people who rated, old price, discount and new price.

● **Database Creation**: Compile the scraped data into a structured format to create a comprehensive database for easy access and analysis.

● **Price Monitoring and Analysis**: Track and analyze price trends over time to identify the best times to buy and to monitor pricing strategies.

● **Customer Feedback Analysis**: Gather and analyze customer ratings and reviews to assess product quality and customer satisfaction.

● **Structured Data Frame Generation**: Convert the extracted data into a well-organized dataframe format for efficient analysis.

3. Outline:

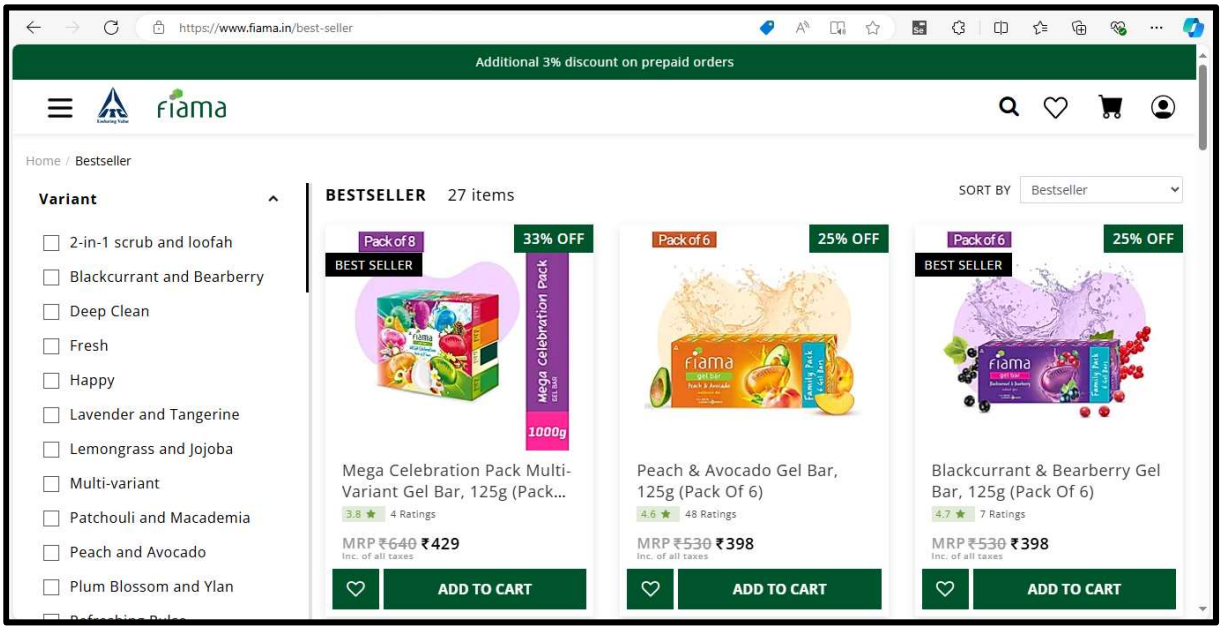
From this site, we are going to grab the following information:

- Product Image
- Product Name
- Quantity
- Rating
- Number of people who rated
- Old Price
- Discount
- New Price

4. Steps:

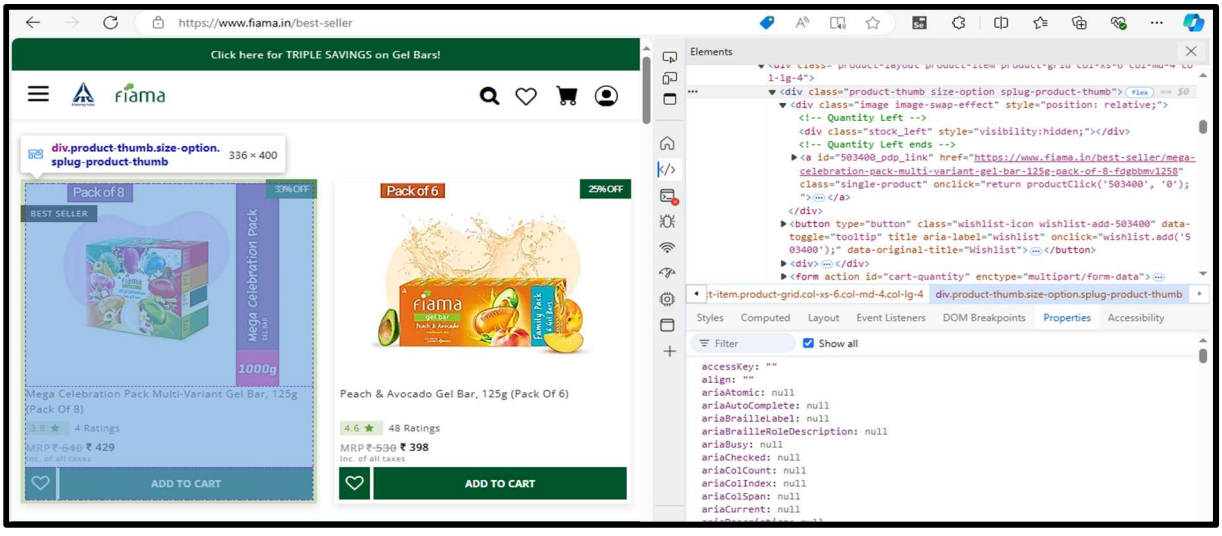
1. Choose the Website and Webpage URL:

- The first step is to select the website you want to scrape. We will extract data of Best-selling Fiama products from https://www.fiama.in/best-seller



2. Inspect the website:

- Now the next step is to understand the website structure. Understand what the attributes of the elements that are of your interest are. Right click on the website to select “Inspect”. This will open HTML code. Use the inspector tool to see the name of all the elements to use in the code.



3. Installing the important libraries:

Python has several web scraping libraries. In this project, following libraries are used i.e Requests – for making HTTP requests to website :

- **BeautifulSoup** – for parsing the HTML code
- **Pandas** – for storing the scraped data in data frame

4. Write the Python source code:

In python main code, we will be:

- Using requests to send an HTTP GET requests
- Using BeautifulSoup to parse the HTML code
- Extracting the required data from the HTML code
- Storing the information in a pandas DataFrame

5. Exporting the extracted data:

- Export the data as a CSV file.

6. Benefits:

- Access to Valuable Data
- Automation of Data Collection
- Real-Time Insights

7. Risks:

- Legal Issues
- IP Blocking and Rate Limiting
- Data Accuracy and Integrity

5. Web Scraping Python Coding Commands:

a. Accessing Best-selling Fiama products on the FIama website using requests and BeautifulSoup libraries:

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

```
url = 'https://www.fiama.in/best-seller&sort=sort_order&order=desc&page=1'
```

```
page = requests.get(url)
soup = BeautifulSoup(page.content,'html.parser')
soup
```

```
<!DOCTYPE html>

<!--[if IE]><![endif]-->
<!--[if IE 8 ]><html dir="ltr" lang="en" class="ie8"><![endif]-->
<!--[if IE 9 ]><html dir="ltr" lang="en" class="ie9"><![endif]-->
<!--[if (gt IE 9)|!(IE)]><!-->
<html class="responsive" dir="ltr" lang="en">
<!--<![endif]-->
<head>
<meta charset="utf-8"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" name="viewport">
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<title>Bestseller</title>
<base href="https://www.fiama.in/"/>
<meta content="Online Shopping for bestseller, Shop for bestseller from Fiama. *COD *Easy returns and exchanges. *Free Shipp
ing* " name="description">
<meta content="Online Shopping for bestseller, bestseller collection online, Fiama online shopping, Fiama online" name="keywo
```

b. Accessing Product Images:

```
ProductImage = soup.findAll('img',{'class':'swap-image'})
#ProductImage
img_urls = []
for img in ProductImage:
    img_url = img.get('data-src')
    if img_url:
        img_urls.append(img_url)

img_urls
```

```
['https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDGBBMV1258_1.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDPA1256_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDBB1256_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/1-June-2023/FDSGCP125x4_1.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/10-Oct-2023/FDMSGDC500_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDED115_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDMD110_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDCS110_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDLF125_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDREFSHPLS125_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/2-Mar-2023/FDCS1153_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDMD1103_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/2-Mar-2023/FDED1153_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/9-Feb-2023/FDGBBACP1253_2.png',
 'https://cdn.staticans.com/image/tr:e-sharpen-01,h-250,w-290,cm-pad_resize/data/Fiama/2-Mar-2023/FDREFSHPLS1253_2.png']
```

c. Accessing Product Names:

```
ProductName = soup.findAll('h4',{'class':'product-name'})
ProductName
```

```
[<h4 class="product-name">
 <a class="single-product" href="https://www.fiama.in/best-seller/mega-celebration-pack-multi-variant-gel-bar-125g-pack-of-8-fd
gbbmv1258" onclick="return productClick('503400', '0');">
                                        Mega Celebration pack Multi-variant Gel Bar, 125g (Pack of 8)
</a>
 </h4>,
 <h4 class="product-name">
 <a class="single-product" href="https://www.fiama.in/best-seller/peach-avocado-gel-bar-125g-pack-of-6-fdpa1256" onclick="retur
n productClick('503344', '1');">
                                        Peach &amp;amp; Avocado Gel Bar, 125g (Pack of 6)                                    </a>
 </h4>,
 <h4 class="product-name">
 <a class="single-product" href="https://www.fiama.in/best-seller/blackcurrant-bearberry-gel-bar-125g-pack-of-6-fdbb1256" oncli
ck="return productClick('503345', '2');">
                                        Blackcurrant &amp;amp; Bearberry Gel Bar, 125g (Pack of 6)
</a>
 </h4>,
 <h4 class="product-name">
 <a class="single-product" href="https://www.fiama.in/best-seller/shower-gel-men-celebration-pack-125ml-pack-of-4loofah-fdsgcp1
25x4" onclick="return productClick('510673', '3');">
                                        Shower Gel Men Celebration Pack, 125ml (Pack of 4) + Loofah
</a>
 </h4>,
```

Product Name and Quantity which were combined have been split into two separate columns.

Also we replaced all the empty spaces and new lines to get the product name.

```
Product_Name = []
for i in ProductName:
    Product_Name.append(i.get_text().replace("\n","").strip().split(",")[0])
Product_Name
```

```
['Mega Celebration pack Multi-variant Gel Bar',
 'Peach & Avocado Gel Bar',
 'Blackcurrant & Bearberry Gel Bar',
 'Shower Gel Men Celebration Pack',
 'Deep Clean Men Shower Gel',
 'Blackcurrant & Bearberry Gel Bar',
 'Peach & Avocado Gel Bar',
 'Lemongrass & Jojoba Gel Bar',
 'Patchouli & Macademia Gel Bar',
 'Refreshing Pulse Men Gel Bar',
 'Lemongrass & Jojoba Gel Bar',
 'Peach & Avocado Gel Bar',
 'Blackcurrant & Bearberry Gel Bar',
 'Active Celebration Pack Men Multi-variant Gel Bar',
 'Refreshing Pulse Men Gel Bar']
```

d. Accessing Quantities of the Product:

Here, we used RegEx to remove the texts after g/ml and also removed all the unnecessary spaces to get the quantity.

```
import re

def remove_after_g_ml(item):
    return re.sub(r'(g|ml).*', r'\1', item)

Quantity = []
for i in ProductName:
    qa = i.get_text().replace("\n","").split(",")[1].strip().replace(" ","")
    cleaned_items = remove_after_g_ml(qa)
    Quantity.append(cleaned_items)
Quantity
```

```
['125g',
 '125g',
 '125g',
 '125ml',
 '500ml',
 '125g',
 '125g',
 '125g',
 '125g',
 '125g',
 '125g',
 '125g',
 '125g',
 '125g',
 '125g']
```

e. Accessing Ratings

Here, some products did not have ratings, so those products which did not have ratings were given 'NA' to make the length of the rows even.

```
caption_section = soup.findAll('div',{'class':'caption'})
caption_section

star_ratings = []
for r in caption_section:
    rating = r.find('span',{'class':'rating-star'})
    if rating:
        star_ratings.append(rating.text)
    else:
        star_ratings.append("NA")

star_ratings
```

```
['3.8',
 '4.6',
 '4.7',
 'NA',
 'NA',
 '4.7',
 '4.6',
 '4.8',
 '4.9',
 '4.7',
 '4.8',
 '4.6',
 '4.7',
 '4.3',
 '4.3']
```

f. Accessing number of people who rated

```
caption_section = soup.findAll('div',{'class':'caption'})
#caption_section
No_of_Ratings = []
for r in caption_section:
    rating = r.find('div',{'class':'rating-wrp'})
    if rating:
        ra = rating.text
        No_of_Ratings.append(ra)
    else:
        No_of_Ratings.append("NA")

No_of_Ratings
```

```
['3.84 Ratings',
 '4.648 Ratings',
 '4.77 Ratings',
 'NA',
 'NA',
 '4.77 Ratings',
 '4.648 Ratings',
 '4.821 Ratings',
 '4.914 Ratings',
 '4.716 Ratings',
 '4.821 Ratings',
 '4.648 Ratings',
 '4.77 Ratings',
 '4.33 Ratings',
 '4.34 Ratings']
```

Here, the star ratings and number of people who rated were extracted as merged texts as shown above. For eg. 3.8 & 4 Ratings were extracted as '3.84 Ratings'. So, we split using split(".") and removed the 'Ratings' text.

```
caption_section = soup.findAll('div',{'class':'caption'})
#caption_section
No_of_Ratings = []
for r in caption_section:
    rating = r.find('div',{'class':'rating-wrp'})
    if rating:
        ra = rating.text.split(".")[1].replace("Ratings","")
        No_of_Ratings.append(ra)
    else:
        No_of_Ratings.append("NA")

No_of_Ratings
```

```
['84 ',
 '648 ',
 '77 ',
 'NA',
 'NA',
 '77 ',
 '648 ',
 '821 ',
 '914 ',
 '716 ',
 '821 ',
 '648 ',
 '77 ',
 '33 ',
 '34 ']
```

In the above result, we added a function to remove the first digit to get the actual number of people who rated.

```
caption_section = soup.findAll('div',{'class':'caption'})

def remove_first_digit(s):
    return s[1:] if s.isdigit() else s

No_of_Ratings = []
for r in caption_section:
    rating = r.find('div',{'class':'rating-wrp'})
    if rating:
        ra = rating.text.split(".")[1].replace("Ratings","").strip()
        cleaned_data = remove_first_digit(ra)
        No_of_Ratings.append(cleaned_data)
    else:
        No_of_Ratings.append("NA")

No_of_Ratings
```

```
['4',
 '48',
 '7',
 'NA',
 'NA',
 '7',
 '48',
 '21',
 '14',
 '16',
 '21',
 '48',
 '7',
 '3',
```

g. Accessing Old Price:

```
OldPrice = soup.findAll('span',{'class':'price-old'})
OldPrice
```

```
[<span class="price-old"><i class="fa fa-inr rs-sym"></i> 640</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 530</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 530</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 520</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 499</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 95</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 95</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 95</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 95</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 95</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 265</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 265</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 265</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 265</span>,
 <span class="price-old"><i class="fa fa-inr rs-sym"></i> 295</span>]
```

```
Old_Price = []
for o in OldPrice:
    Old_Price.append(o.getText().strip())
Old_Price
```

```
['640',
 '530',
 '530',
 '520',
 '499',
 '95',
 '95',
 '95',
 '95',
 '95',
 '265',
 '265',
 '265',
 '265',
 '295']
```

h. Accessing Discount:

```
Offer = soup.findAll('span',{'class':'sale sale-percentage'})
Offer
```

```
[<span class="sale sale-percentage">33% Off</span>,
 <span class="sale sale-percentage">25% Off</span>,
 <span class="sale sale-percentage">25% Off</span>,
 <span class="sale sale-percentage">25% Off</span>,
 <span class="sale sale-percentage">35% Off</span>,
 <span class="sale sale-percentage">11% Off</span>,
 <span class="sale sale-percentage">11% Off</span>,
 <span class="sale sale-percentage">11% Off</span>,
 <span class="sale sale-percentage">11% Off</span>,
 <span class="sale sale-percentage">11% Off</span>,
 <span class="sale sale-percentage">10% Off</span>,
 <span class="sale sale-percentage">10% Off</span>,
 <span class="sale sale-percentage">10% Off</span>,
 <span class="sale sale-percentage">10% Off</span>,
 <span class="sale sale-percentage">10% Off</span>]
```

```
Discount = []
for o in Offer:
    Discount.append(o.getText().replace("Off","").strip())
Discount
```

```
['33%',
 '25%',
 '25%',
 '25%',
 '35%',
 '11%',
 '11%',
 '11%',
 '11%',
 '11%',
 '10%',
 '10%',
 '10%',
 '10%',
 '10%']
```

i. Accessing New Price:

```
NewPrice = soup.findAll('span',{'class':'price-new'})
NewPrice
```

```
[<span class="price-new"><i class="fa fa-inr rs-sym"></i> 429 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 398 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 398 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 390 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 324 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 85 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 85 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 85 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 85 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 85 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 239 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 239 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 239 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 239 </span>,
 <span class="price-new"><i class="fa fa-inr rs-sym"></i> 266 </span>]
```

```
New_Price = []
for n in NewPrice:
    New_Price.append(n.getText().strip())
New_Price
```

```
['429',
 '398',
 '398',
 '390',
 '324',
 '85',
 '85',
 '85',
 '85',
 '85',
 '239',
 '239',
 '239',
 '239',
 '266']
```

j. Importing pandas and creating dataframe:

```
import pandas as pd
Fiama_Products = pd.DataFrame({
    "Fiama_Product_Image":img_urls,
    "Fiama Product Name":Product_Name,
    "Quantity":Quantity,
    "Rating":star_ratings,
    "No of Ratings":No_of_Ratings,
    "Old Price(Rs)": Old_Price,
    "Discount Offer":Discount,
    "New Price(Rs)":New_Price,

})

Fiama_Products
```

	Fiama_Product_Image	Fiama Product Name	Quantity	Rating	No of Ratings	Old Price(Rs)	Discount Offer	New Price(Rs)
0	https://cdn.staticans.com/image/tr:e-sharpen-0...	Mega Celebration pack Multi-variant Gel Bar	125g	3.8	4	640	33%	429
1	https://cdn.staticans.com/image/tr:e-sharpen-0...	Peach & Avocado Gel Bar	125g	4.6	48	530	25%	398
2	https://cdn.staticans.com/image/tr:e-sharpen-0...	Blackcurrant & Bearberry Gel Bar	125g	4.7	7	530	25%	398
3	https://cdn.staticans.com/image/tr:e-sharpen-0...	Shower Gel Men Celebration Pack	125ml	NA	NA	520	25%	390
4	https://cdn.staticans.com/image/tr:e-sharpen-0...	Deep Clean Men Shower Gel	500ml	NA	NA	499	35%	324
5	https://cdn.staticans.com/image/tr:e-sharpen-0...	Blackcurrant & Bearberry Gel Bar	125g	4.7	7	95	11%	85
6	https://cdn.staticans.com/image/tr:e-sharpen-0...	Peach & Avocado Gel Bar	125g	4.6	48	95	11%	85
7	https://cdn.staticans.com/image/tr:e-sharpen-0...	Lemongrass & Jojoba Gel Bar	125g	4.8	21	95	11%	85
8	https://cdn.staticans.com/image/tr:e-sharpen-0...	Patchouli & Macademia Gel Bar	125g	4.9	14	95	11%	85
9	https://cdn.staticans.com/image/tr:e-sharpen-0...	Refreshing Pulse Men Gel Bar	125g	4.7	16	95	11%	85
10	https://cdn.staticans.com/image/tr:e-sharpen-0...	Lemongrass & Jojoba Gel Bar	125g	4.8	21	265	10%	239
11	https://cdn.staticans.com/image/tr:e-sharpen-0...	Peach & Avocado Gel Bar	125g	4.6	48	265	10%	239
12	https://cdn.staticans.com/image/tr:e-sharpen-0...	Blackcurrant & Bearberry Gel Bar	125g	4.7	7	265	10%	239
13	https://cdn.staticans.com/image/tr:e-sharpen-0...	Active Celebration Pack Men Multi-variant Gel Bar	125g	4.3	3	265	10%	239
14	https://cdn.staticans.com/image/tr:e-sharpen-0...	Refreshing Pulse Men Gel Bar	125g	4.3	4	295	10%	266

k. Converting and Storing DataFrame in the form of CSV file and opening the file in application:

```
Fiama_Products.to_csv("Fiama_Products_Web_Scraping_Final.csv",index=False)
```

```
Read = pd.read_csv("Fiama_Products_Web_Scraping_Final.csv")
Read
```

	Fiama_Product_Image	Fiama Product Name	Quantity	Rating	No of Ratings	Old Price(Rs)	Discount Offer	New Price(Rs)
0	https://cdn.staticans.com/image/tr:e-sharpen-0...	Mega Celebration pack Multi-variant Gel Bar	125g	3.8	4.0	640	33%	429
1	https://cdn.staticans.com/image/tr:e-sharpen-0...	Peach & Avocado Gel Bar	125g	4.6	48.0	530	25%	398
2	https://cdn.staticans.com/image/tr:e-sharpen-0...	Blackcurrant & Bearberry Gel Bar	125g	4.7	7.0	530	25%	398
3	https://cdn.staticans.com/image/tr:e-sharpen-0...	Shower Gel Men Celebration Pack	125ml	NaN	NaN	520	25%	390
4	https://cdn.staticans.com/image/tr:e-sharpen-0...	Deep Clean Men Shower Gel	500ml	NaN	NaN	499	35%	324
5	https://cdn.staticans.com/image/tr:e-sharpen-0...	Blackcurrant & Bearberry Gel Bar	125g	4.7	7.0	95	11%	85
6	https://cdn.staticans.com/image/tr:e-sharpen-0...	Peach & Avocado Gel Bar	125g	4.6	48.0	95	11%	85
7	https://cdn.staticans.com/image/tr:e-sharpen-0...	Lemongrass & Jojoba Gel Bar	125g	4.8	21.0	95	11%	85
8	https://cdn.staticans.com/image/tr:e-sharpen-0...	Patchouli & Macademia Gel Bar	125g	4.9	14.0	95	11%	85
9	https://cdn.staticans.com/image/tr:e-sharpen-0...	Refreshing Pulse Men Gel Bar	125g	4.7	16.0	95	11%	85
10	https://cdn.staticans.com/image/tr:e-sharpen-0...	Lemongrass & Jojoba Gel Bar	125g	4.8	21.0	265	10%	239
11	https://cdn.staticans.com/image/tr:e-sharpen-0...	Peach & Avocado Gel Bar	125g	4.6	48.0	265	10%	239
12	https://cdn.staticans.com/image/tr:e-sharpen-0...	Blackcurrant & Bearberry Gel Bar	125g	4.7	7.0	265	10%	239
13	https://cdn.staticans.com/image/tr:e-sharpen-0...	Active Celebration Pack Men Multi-variant Gel Bar	125g	4.3	3.0	265	10%	239
14	https://cdn.staticans.com/image/tr:e-sharpen-0...	Refreshing Pulse Men Gel Bar	125g	4.3	4.0	295	10%	266

6. Conclusion:

In this FIAMA web scraping python project, we successfully developed a web scraper using Python's Beautiful Soup and Requests libraries. The scraper was designed to extract specific data from a target website, demonstrating the following key points:

1. **Understanding Web Scraping**: We explored the fundamentals of web scraping, including the ethical considerations and legal implications.
2. **Tool Utilization**: The project showcased the effective use of Beautiful Soup for parsing HTML and Requests for handling HTTP requests.
3. **Data Extraction and Processing:** We implemented techniques to locate and extract relevant data, followed by processing and storing it in a structured format.
4. **Storing data in DataFrame and exporting in csv:** We stored extracted data in DataFrame and exported to csv file.

Overall, this project provided valuable insights into the practical applications of web scraping, highlighting its potential for automating data collection tasks.