Rohini Amol Shinde

Project Name : Web Scraping online gift store from ferns n petals

Project Guide : Prof. Sameer Warsolkar

Web Scraping Online gift store from ferns n petals(fnp.com)

Project Aim:

To gather comprehensive information on products from the Ferns N Petals website through web scraping, providing valuable insights into the product details, pricing, availability, and customer reviews. The collected data will be used to analyze market trends, track pricing changes, and enhance the customer shopping experience by offering detailed product comparisons and reviews.

Objectives:

1. Data Collection:
    - Extract detailed product information including names, descriptions, categories, prices, and availability.
    - Gather customer reviews and ratings for each product.
    - Capture additional product attributes such as discounts, delivery options, and special offers.
    - Store the collected data in a structured format for further analysis.
2. Pricing Analysis:
    - Analyze pricing trends across different categories and product types.
    - Monitor changes in product prices over time to identify patterns and fluctuations.
    - Compare pricing strategies of similar products to understand competitive positioning.
    - Provide insights into seasonal pricing variations and promotional offers.

3. Market Trends:
    - Identify popular product categories and trending items based on customer interest and sales data.
    - Track shifts in consumer preferences and demand over time.
    - Analyze the impact of holidays, festivals, and special occasions on product sales.
    - Forecast future market trends using historical data and patterns.
4. Competitor Analysis:
    - Compare Ferns N Petals' product offerings, pricing, and customer reviews with those of key competitors.
    - Identify unique selling points and areas for improvement relative to competitors.
    - Assess competitors' promotional strategies and their effectiveness.
    - Evaluate market positioning and share insights on competitive advantages.
5. Data Quality Assurance:
    - Implement data validation techniques to ensure accuracy and completeness of collected data.
    - Regularly update and clean the dataset to maintain its relevance and integrity.
    - Verify data consistency by cross-referencing with other reliable sources.
    - Document the data collection and processing methods to ensure reproducibility and transparency.

Outlines:

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

1. Item_name
2. Price
3. Rating
4. Reviews

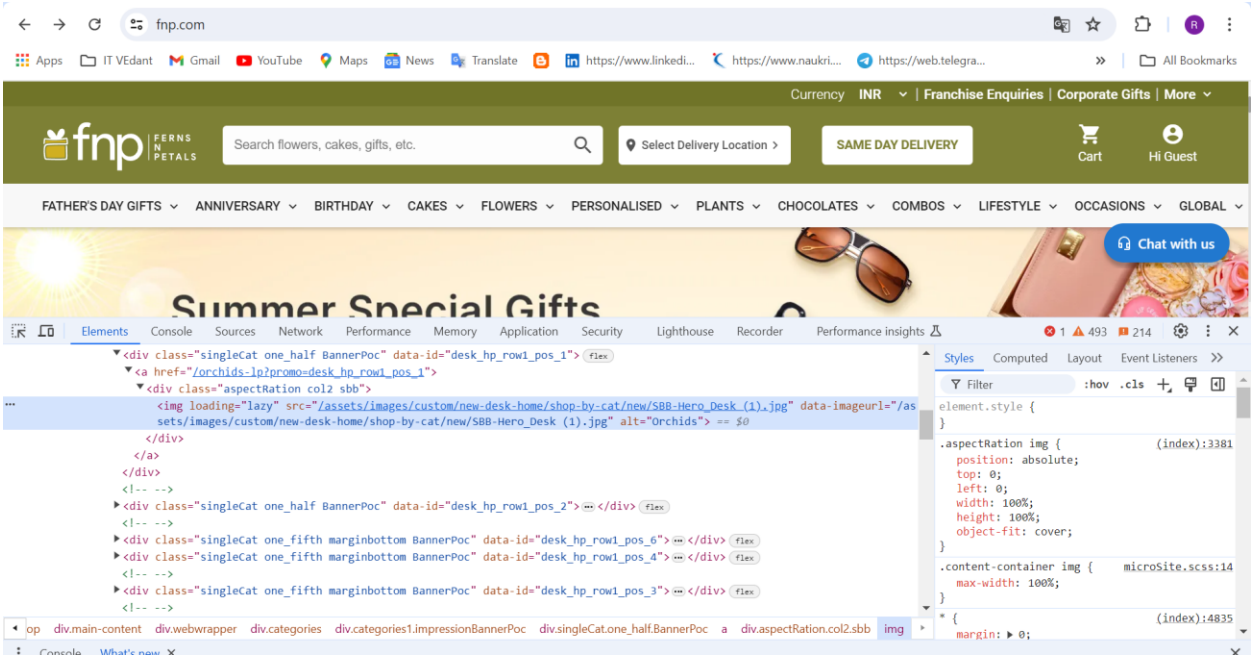
Steps:

Choose the Website and Webpage URL:

- The first step is to select the website you want to scrape. We will try to extract data of Ferns N Petals from www.fnp.com.

1.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.



2. Installing the important libraries:

Python has several web scrapping libraries. We will use the following libraries:

➢ Requests – for making HTTP requests to website

➢ BeautifulSoup – for parsing the HTML code

➢ Pandas – for storing the scraped data in data frame

3. Write the Python source code:

We'll write the main python code. The code will perform the following steps:

➢ Using requests to send an HTTP GET requests

➢ Using BeautifulSoup to parse the HTML code

➢ Extracting the required data from the HTML code

4. Exporting the extracted data:

We'll export the data as a CSV file. We will use the pandas library. We'll use the pandas library.

5. Benefits:

➢ Access to valuable data for analysis or research.

➢ Automation of data collection, saving time and effort.

➢ Stay up to date with change changes on the target websites.

6. Risk:

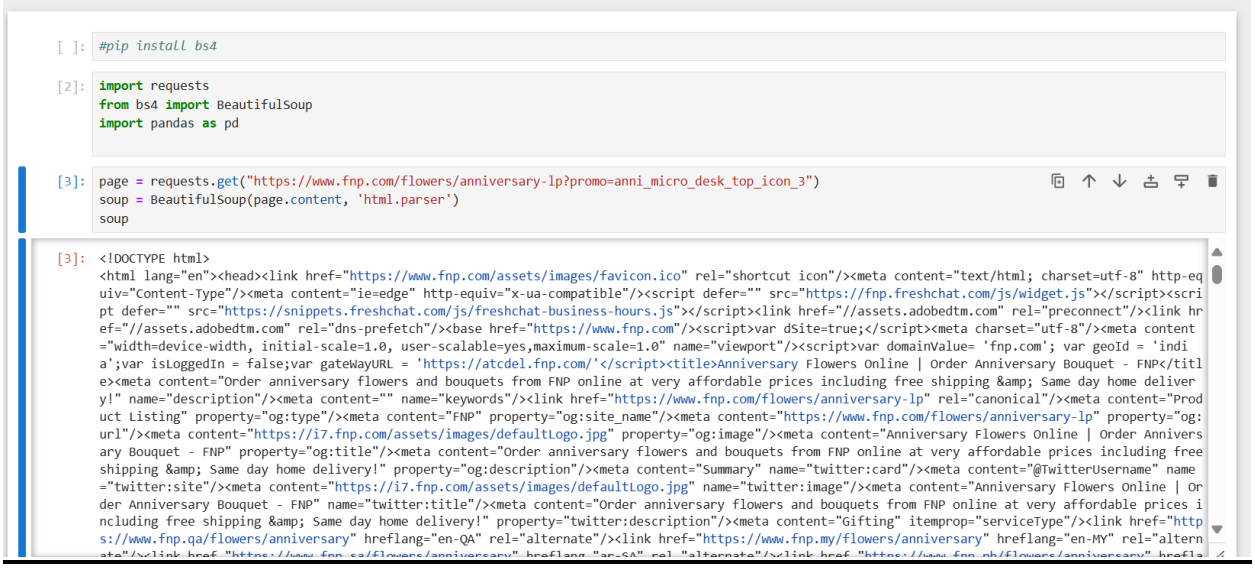
➢ Legal issues related to web scraping.

➢ Technical challenges due to website changes

➢ Store the information in a pandas DataFrame

7. Ferns N Petals web scraping coding command steps:

1. Accessing Ferns N Petals website:



```
[ ]: #pip install bs4

[2]: import requests
     from bs4 import BeautifulSoup
     import pandas as pd

[3]: page = requests.get("https://www.fnp.com/flowers/anniversary-lp?promo=anni_micro_desk_top_icon_3")
     soup = BeautifulSoup(page.content, 'html.parser')
     soup

[3]: <!DOCTYPE html>
     <html lang="en"><head><link href="https://www.fnp.com/assets/images/favicon.ico" rel="shortcut icon"/><meta content="text/html; charset=utf-8" http-eq
     uiv="Content-Type"/><meta content="ie=edge" http-equiv="x-ua-compatible"/><script defer="" src="https://fnp.freshchat.com/js/widget.js"></script><scri
     pt defer="" src="https://snippets.freshchat.com/js/freshchat-business-hours.js"></script><link href="//assets.adobedtm.com" rel="preconnect"/><link hr
     ef="//assets.adobedtm.com" rel="dns-prefetch"/><base href="https://www.fnp.com"/><script>var dSite=true;</script><meta charset="utf-8"/><meta content
     ="width=device-width, initial-scale=1.0, user-scalable=yes,maximum-scale=1.0" name="viewport"/><script>var domainValue= 'fnp.com'; var geoId = 'indi
     a';var isLoggedIn = false;var gateWayURL = 'https://atcdel.fnp.com/'</script><title>Anniversary Flowers Online | Order Anniversary Bouquet - FNP</titl
     e><meta content="Order anniversary flowers and bouquets from FNP online at very affordable prices including free shipping &amp;amp; Same day home deliver
     y!" name="description"/><meta content="" name="keywords"/><link href="https://www.fnp.com/flowers/anniversary-lp" rel="canonical"/><meta content="Prod
     uct Listing" property="og:type"/><meta content="FNP" property="og:site_name"/><meta content="https://www.fnp.com/flowers/anniversary-lp" property="og:
     url"/><meta content="https://i7.fnp.com/assets/images/defaultLogo.jpg" property="og:image"/><meta content="Anniversary Flowers Online | Order Annivers
     ary Bouquet - FNP" property="og:title"/><meta content="Order anniversary flowers and bouquets from FNP online at very affordable prices including free
     shipping &amp;amp; Same day home delivery!" property="og:description"/><meta content="Summary" name="twitter:card"/><meta content="@TwitterUsername" name
     ="twitter:site"/><meta content="https://i7.fnp.com/assets/images/defaultLogo.jpg" name="twitter:image"/><meta content="Anniversary Flowers Online | Or
     der Anniversary Bouquet - FNP" name="twitter:title"/><meta content="Order anniversary flowers and bouquets from FNP online at very affordable prices i
     ncluding free shipping &amp;amp; Same day home delivery!" property="twitter:description"/><meta content="Gifting" itemprop="serviceType"/><link href="http
     s://www.fnp.qa/flowers/anniversary" hreflang="en-QA" rel="alternate"/><link href="https://www.fnp.my/flowers/anniversary" hreflang="en-MY" rel="altern
     ate"/><link href="https://www.fnp.sa/flowers/anniversary" hreflang="ar-SA" rel="alternate"/><link href="https://www.fnp.ph/flowers/anniversary" hrefla
```

2. Using BeautifulSoup:

```
[5]: print(soup.prettify())
```

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <link href="https://www.fnp.com/assets/images/favicon.ico" rel="shortcut icon"/>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <meta content="ie=edge" http-equiv="x-ua-compatible"/>
  <script defer="" src="https://fnp.freshchat.com/js/widget.js">
  </script>
  <script defer="" src="https://snippets.freshchat.com/js/freshchat-business-hours.js">
  </script>
  <link href="//assets.adobedtm.com" rel="preconnect"/>
  <link href="//assets.adobedtm.com" rel="dns-prefetch"/>
  <base href="https://www.fnp.com"/>
  <script>
   var dSite=true;
  </script>
  <meta charset="utf-8"/>
  <meta content="width=device-width, initial-scale=1.0, user-scalable=yes,maximum-scale=1.0" name="viewport"/>
```

3. Accessing the Item_name:

```
[80]: product_title=[]
for i in title_tags:
    product_title.append(i.get_text())

product_title
```

```
[80]: ['Love For Pastel Carnations Flower Bouquet',
 'Romantic Rose Radiance Bouquet',
 'Sweet Memories Pink Roses Bouquet & Chocolate Cake',
 'Mixed Roses Romantic Flower Bunch',
 'Happy Anniversary Rose Mug',
 'Endless Love 100 Roses Bouquet ',
 'Blissful Love Roses Arrangement',
 'Carnation Love Bouquet',
 'Spectacular Rose Bouquet & Truffle Combo',
 'Red Forever Rose Gift Box- Hand Delivery',
 'Ruby Romance Rose Bouquet',
 "Sweetheart's Rose Melody",
 'Elegant Gerberas Flower In Glass Vase',
 'Paradise On Earth Carnations Flower Bouquet',
 'All The Good Vibes Roses Arrangement',
 'Blooming Joy Rose Standing Bouquet',
 'Happy Times Carnations Flower Bouquet',
 'Confetti Of Love Red Color Roses Flower Bouquet',
 'Delicate Love 6 Pink Roses Bunch',
```

4. Accessing the Price:

```
[12]: product_price=[]
      for i in price:
          product_price.append(i.get_text())

      product_price = product_price[:20]
      product_price
```

```
[12]: ['\xa0599',
       '\xa01699',
       '\xa01099',
       '\xa0499',
       '\xa01699',
       '\xa0799',
       '\xa0649',
       '\xa01699',
       '\xa01499',
       '\xa0799',
       '\xa01375',
       '\xa01049',
       '\xa01049',
       '\xa0999',
       '\xa01049',
       '\xa01375',
       '\xa0599',
       '\xa0999',
```

5. Using RegEx to remove the special character(Product_Prices) from the list:

```
[13]: import re

      cleaned_price0 = [re.sub(r'\xa0', '', item) for item in product_price]
      cleaned_price = [float(cleaned_price0.strip()) for cleaned_price0 in cleaned_price0]
      cleaned_price = cleaned_price[:20]
      # Print the cleaned list
      cleaned_price
```

```
[13]: [599.0,
       1699.0,
       1099.0,
       499.0,
       1699.0,
       799.0,
       649.0,
       1699.0,
       1499.0,
       799.0,
       1375.0,
       1049.0,
       1049.0,
       999.0,
       1049.0,
       1375.0,
       599.0,
       999.0.
```

6. Accessing the Rating:

```
[14]: Rating = soup.select(".product-card_rating-sec__34VZH")
      Rating = Rating[:20]
      Rating
```

```
[14]: [<span class="product-card_rating-sec__34VZH">4.4<i class="material-icons">star_half</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">4.8<i class="material-icons">star_half</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">4.9<i class="material-icons">star_half</i></span>,
       <span class="product-card_rating-sec__34VZH">4.8<i class="material-icons">star_half</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">4.9<i class="material-icons">star_half</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">4.7<i class="material-icons">star_half</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">4.8<i class="material-icons">star_half</i></span>,
       <span class="product-card_rating-sec__34VZH">5.0<i class="material-icons">star</i></span>,
       <span class="product-card_rating-sec__34VZH">4.9<i class="material-icons">star_half</i></span>]
```

```
[16]: Rating_tags=[]
      for i in Rating:
          Rating_tags.append(i.get_text())

      Rating_tags = Rating_tags[:20]
      Rating_tags
```

```
[16]: ['4.4star_half',
       '5.0star',
       '5.0star',
       '4.8star_half',
       '5.0star',
       '5.0star',
       '5.0star',
       '5.0star',
       '5.0star',
       '4.9star_half',
       '4.8star_half',
       '5.0star',
       '4.9star_half',
       '5.0star',
       '5.0star',
       '4.7star_half',
       '5.0star',
       '4.8star_half',
       '5.0star',
       '4.9star_half']
```

7. Using RegEx to remove the special character(ratings) from the list:

```
[87]: import re

      cleaned_ratings0 = [re.sub(r'star(_half)?', '', item) for item in Rating_tags]
      cleaned_ratings = [float(cleaned_ratings0.strip()) for cleaned_ratings0 in cleaned_ratings0]

      # Print the cleaned list
      cleaned_ratings = cleaned_ratings[:20]
      cleaned_ratings
```

```
[87]: [5.0,
       4.9,
       5.0,
       4.4,
       5.0,
       5.0,
       5.0,
       5.0,
       5.0,
       5.0,
       5.0,
       4.8,
       4.9,
       4.8,
       5.0,
       4.9,
       5.0,
       5.0,
       4.9,
```

8. Accessing the Total Number of Review:

```
[91]: Review=soup.select(".product-card_product-review-info__2-RtV")
      Review = Review[:20]
      Review
```

```
[91]: [<span class="product-card_product-review-info__2-RtV">50<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">92<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">767<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">4183<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">15<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">439<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">5<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">10<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">83<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">96<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">177<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">124<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">138<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">604<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">53<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">19<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">98<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">132<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">218<!-- --> Reviews</span>,
       <span class="product-card_product-review-info__2-RtV">352<!-- --> Reviews</span>]
```

[92]:
```
Review_list=[]
for i in Review:
    Review_list.append(i.get_text())
Review_list = Review_list[:20]
Review_list
```

[92]:
```
['50\xa0 Reviews',
 '92\xa0 Reviews',
 '767\xa0 Reviews',
 '4183\xa0 Reviews',
 '15\xa0 Reviews',
 '439\xa0 Reviews',
 '5\xa0 Reviews',
 '10\xa0 Reviews',
 '83\xa0 Reviews',
 '96\xa0 Reviews',
 '177\xa0 Reviews',
 '124\xa0 Reviews',
 '138\xa0 Reviews',
 '604\xa0 Reviews',
 '53\xa0 Reviews',
 '19\xa0 Reviews',
 '98\xa0 Reviews',
 '132\xa0 Reviews',
 '218\xa0 Reviews',
 '352\xa0 Reviews']
```

9.Using RegEx to remove the special character (Total_Reviews) from the list:

[93]:
```
import re

Review_list0 = [re.sub(r'\xa0 Reviews', '', item) for item in Review_list]
cleaned_reviews = [int(Review_list0.strip()) for Review_list0 in Review_list0]

# Print the cleaned list
cleaned_reviews = cleaned_reviews[:20]
cleaned_reviews
```

[93]:
```
[50,
 92,
 767,
 4183,
 15,
 439,
 5,
 10,
 83,
 96,
 177,
 124,
 138,
 604,
 53,
 19,
 98,
 132,
 218,
```

10.Importing the pandas and creating DataFrame:

```
[94]: import pandas as pd
FNP_data = pd.DataFrame({
    "Item_name": product_title,
    "Price": cleaned_price,
    "Rating":cleaned_ratings,
    "Reviews":cleaned_reviews
})
FNP_data
```

[94]:

	Item_name	Price	Rating	Reviews
0	Love For Pastel Carnations Flower Bouquet	1049.0	5.0	50
1	Romantic Rose Radiance Bouquet	1049.0	4.9	92
2	Sweet Memories Pink Roses Bouquet & Chocolate ...	1349.0	5.0	767
3	Mixed Roses Romantic Flower Bunch	599.0	4.4	4183
4	Happy Anniversary Rose Mug	649.0	5.0	15
5	Endless Love 100 Roses Bouquet	5249.0	5.0	439
6	Blissful Love Roses Arrangement	1699.0	5.0	5
7	Carnation Love Bouquet	2299.0	5.0	10
8	Spectacular Rose Bouquet & Truffle Combo	3499.0	5.0	83
9	Red Forever Rose Gift Box- Hand Delivery	999.0	5.0	96
10	Ruby Romance Rose Bouquet	8599.0	5.0	177
11	Sweetheart's Rose Melody	899.0	4.8	124
12	Elegant Gerberas Flower In Glass Vase	999.0	4.9	138
13	Paradise On Earth Carnations Flower Bouquet	699.0	4.8	604

11.converting and Storing DataFrame in the form of a CSV file and opening the file in application:

```
[58]: FNP_data.to_csv("FNP_data.csv",index=False)
dfepd.read_csv("FNP_data.csv")
df
```

[58]:

	Item_name	Price	Rating	Reviews
0	Mixed Roses Romantic Flower Bunch	599.0	4.4	4197
1	Blissful Love Roses Arrangement	1699.0	5.0	439
2	Anniversary Roses of Love	1099.0	5.0	42
3	Delicate Love 6 Pink Roses Bunch	499.0	4.8	632
4	Velvet Whispers of Roses Bouquet	1699.0	5.0	16
5	Enchanting Orchids Flower Bouquet	799.0	5.0	7
6	Happy Anniversary Rose Mug	649.0	5.0	15
7	Sweet Expression Floral Arrangement	1699.0	5.0	18
8	Passion of Love Anniversary Gift Box	1499.0	5.0	6
9	Happy Times Carnations Flower Bouquet	799.0	4.9	139
10	Royal Orchid & Daisy Embrace	1375.0	4.8	4
11	Love For Pastel Carnations Flower Bouquet	1049.0	5.0	53
12	Romantic Rose Radiance Bouquet	1049.0	4.9	92
13	Scintillating Floral Mix Vase	999.0	5.0	352
14	Red Rose Romance	1049.0	5.0	5
15	Blue Orchid Oasis Delight	1375.0	4.7	10
16	Mixed Brilliance Gerbera Flower Bouquet	599.0	5.0	296

12.Accessing first five rows of DataFrame Using head():

```
[59]: df.head()
```

[59]:

	Item_name	Price	Rating	Reviews
0	Mixed Roses Romantic Flower Bunch	599.0	4.4	4197
1	Blissful Love Roses Arrangement	1699.0	5.0	439
2	Anniversary Roses of Love	1099.0	5.0	42
3	Delicate Love 6 Pink Roses Bunch	499.0	4.8	632
4	Velvet Whispers of Roses Bouquet	1699.0	5.0	16

```
[ ]:
```

13.Accessing Last five rows of DataFrame Using tail():

```
[60]: df.tail()
```

[60]:

	Item_name	Price	Rating	Reviews
15	Blue Orchid Oasis Delight	1375.0	4.7	10
16	Mixed Brilliance Gerbera Flower Bouquet	599.0	5.0	296
17	Infinite Love Roses	999.0	4.8	5
18	Blooming Asiatic Lilies In Black Ribbon Tied Vase	1799.0	5.0	122
19	Floral Symphony of Couple Love	1375.0	4.9	214

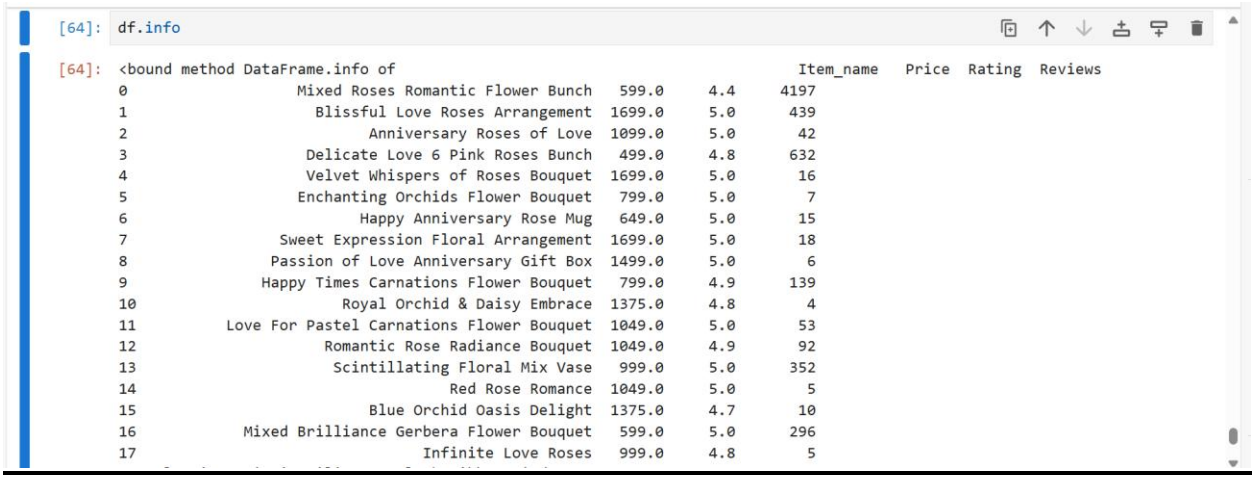
14.Using the describe() function to get information about numerical columns:

```
[61]: df.describe()
```

[61]:

	Price	Rating	Reviews
count	20.000000	20.000000	20.000000
mean	1135.400000	4.910000	333.200000
std	410.521414	0.151831	926.002251
min	499.000000	4.400000	4.000000
25%	799.000000	4.875000	9.250000
50%	1049.000000	5.000000	47.500000
75%	1406.000000	5.000000	234.500000
max	1799.000000	5.000000	4197.000000

15.Using info() to get information about the DataFrame:



```
[64]: df.info

[64]: <bound method DataFrame.info of                                          Item_name   Price  Rating  Reviews
0                    Mixed Roses Romantic Flower Bunch   599.0     4.4     4197
1                      Blissful Love Roses Arrangement  1699.0     5.0      439
2                            Anniversary Roses of Love  1099.0     5.0       42
3                     Delicate Love 6 Pink Roses Bunch   499.0     4.8      632
4                     Velvet Whispers of Roses Bouquet  1699.0     5.0       16
5                    Enchanting Orchids Flower Bouquet   799.0     5.0        7
6                          Happy Anniversary Rose Mug   649.0     5.0       15
7                  Sweet Expression Floral Arrangement  1699.0     5.0       18
8                 Passion of Love Anniversary Gift Box  1499.0     5.0        6
9                Happy Times Carnations Flower Bouquet   799.0     4.9      139
10                     Royal Orchid & Daisy Embrace  1375.0     4.8        4
11          Love For Pastel Carnations Flower Bouquet  1049.0     5.0       53
12                   Romantic Rose Radiance Bouquet  1049.0     4.9       92
13                    Scintillating Floral Mix Vase   999.0     5.0      352
14                                Red Rose Romance  1049.0     5.0        5
15                         Blue Orchid Oasis Delight  1375.0     4.7       10
16            Mixed Brilliance Gerbera Flower Bouquet   599.0     5.0      296
17                                Infinite Love Roses   999.0     4.8        5
```

8.Conclusion

By meticulously collecting and analyzing product data from the Ferns N Petals website, we can gain profound insights into pricing trends, market dynamics, and competitive positioning. This comprehensive approach will not only enhance the customer shopping experience by offering detailed product comparisons and reviews but also provide strategic advantages to Ferns N Petals. The insights derived from this analysis will empower the company to make informed decisions regarding pricing strategies, product offerings, and promotional activities, ultimately leading to a stronger market presence and increased customer satisfaction.

Thank You