In [1]:

```python
# Import the required libraries
from bs4 import BeautifulSoup  # For web scraping
import pandas as pd            # For data manipulation
import requests                # For sending HTTP requests

# Create empty lists to store the data for this page
Product_name, Price, Rating, Total_Rating,Product_link = list(), list(), list(), list(), list()

# The outputfile after extraction
output_file = 'flipkart_details2'

# Get user input for product search
user_input = input("Search for products, brands and more: ")
product = user_input.replace(" ","%20")

# URL used to get the information
base_url = f"https://www.flipkart.com/search?q={product}&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=on&as=off&page="

# Counter for page numbers
n = 0

# Function to scrape page data and save it to a CSV file
def scrape_page(url):
    global n
    # Send a GET request to the URL and get the response
    response = requests.get(url)

    # Create a BeautifulSoup object from the response text using the html.parser
    soup = BeautifulSoup(response.text, 'html.parser')

    # Increment page counter
    n += 1

    # Extract data from the page using BeautifulSoup selectors or regular expressions
    for flipkart in soup.find_all('div', class_='_1AtVbE col-12-12'):
        # Extract product name
        product_name = flipkart.find('div',class_ = '_4rR01T')
        if product_name is not None:
            Product_name.append(product_name.text)
        else:
            Product_name.append("N/A")

        # Extract product price
        price = flipkart.find('div', class_='_30jeq3 _1_WHN1')
        if price is not None:
            Price.append(price.text)
        else:
            Price.append("N/A")

        # Extract product rating
        rating = flipkart.find('div', class_='_3LWZlK')
        if rating is not None:
            Rating.append(rating.text)
        else:
            Rating.append("N/A")

        # Extract total number of ratings for the product
        t_rating = flipkart.find('span', class_='_2_R_DZ')
        if t_rating is not None:
            t_clean=((t_rating.text).split(" "))[0]
            Total_Rating.append(t_clean)
        else:
            Total_Rating.append("N/A")
```

```
        # Extract product link
        try:
            link1 = flipkart.find('a', class_='_1fQZEK')['href']
            link = link1.split("?")[0]
            pro_link = "https://www.flipkart.com" + link
            Product_link.append(pro_link)
        except:
            Product_link.append('N/A')

    # Create a Pandas DataFrame with the lists and append it to the output file
    df = pd.DataFrame({'Product Name': Product_name, 'Price': Price, 'Rating': Rating,"Total Ratings":Total_Rating,"Product Link":Product_link})
    df.to_csv(output_file+'.csv', index=False, encoding='utf-8')

    # Return the number of pages in the particular website
    return n

# Define the total number of pages to scrape
"""response = requests.get("https://www.flipkart.com/search?q=laptop&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=on&as=off&page=1")
soup2 = BeautifulSoup(response.text, 'html.parser')
num_pages = soup2.find('div', class_="_2MImiq")
max_pages=int((num_pages.span.text).split(' ')[3])"""
max_pages=3

# Scrape each page in the range of page numbers
for page_num in range(1, max_pages + 1):
    url = base_url + str(page_num)
    # Save the data to a
    print(scrape_page(url))
```

```
Search for products, brands and more: Mobile 5G above 20k
1
2
3
```

In [9]:

```
import pandas as pd
from IPython.display import HTML  # import the HTML module from the IPython.display library for displaying dataframes in Jupyter notebooks as HTML

# Read the CSV file
file = output_file+".csv"
df = pd.read_csv(file)

# Clean the data
df = df.dropna(subset=['Product Name','Price','Product Link'])
df = df.drop_duplicates()
df = df[df['Product Name'] != 'Product Name']
df.to_csv('flipkart_link.csv', index=False)

# Read the cleaned CSV file
df_new = pd.read_csv('flipkart_link.csv')

# Convert 'Total Ratings' column to integers
df_new['Total Ratings'] = df_new['Total Ratings'].str.replace(',', '').fillna('0').astype(int)

# Create a 'Highly_Ratings' column based on the 'Total_Ratings' column
df_new['Highly Ratings'] = df_new['Total Ratings'] > 1000

# Filter the DataFrame to only show highly rated items
highly_rated = df_new[df_new['Highly Ratings']]

# Sort the 'Total Ratings' column in descending order
sorted_df = highly_rated.sort_values('Total Ratings', ascending=False)

# Format the 'Product Link' column as clickable links
sorted_df.loc[:, 'Product Link'] = sorted_df['Product Link'].apply(lambda x: '<a href="{}" target="_blank">{}</a>'.format(x, x))

# Display the top 5 rows of the sorted dataframe
```

```
top_5 = sorted_df.head(5)
#display(HTML(top_5.to_html(escape=False)))
## Display only the first four columns of the top 5 rows of the sorted dataframe
display(HTML(top_5.iloc[:, :5].to_html(escape=False)))
```

	Product Name	Price	Rating	Total Ratings	Product Link
46	OPPO K10 5G (Midnight Black, 128 GB)	₹17,499	4.4	67708	https://www.flipkart.com/oppo-k10-5g-midnight-black-128-gb/p/itm28cf887931942
67	Xiaomi 11i 5G (Stealth Black, 128 GB)	₹24,999	4.2	38160	https://www.flipkart.com/xiaomi-11i-5g-stealth-black-128-gb/p/itm8f69666fd662e
14	realme 9 Pro+ 5G (Midnight Black, 256 GB)	₹24,999	4.4	30972	https://www.flipkart.com/realme-9-pro-5g-midnight-black-256-gb/p/itm15e7a06fe9352
13	realme 9 Pro+ 5G (Midnight Black, 128 GB)	₹22,999	4.4	30972	https://www.flipkart.com/realme-9-pro-5g-midnight-black-128-gb/p/itm15e7a06fe9352
20	realme 9 Pro+ 5G (Sunrise Blue, 256 GB)	₹24,999	4.4	30972	https://www.flipkart.com/realme-9-pro-5g-sunrise-blue-256-gb/p/itm15e7a06fe9352