Write a python code to scrap Samsung mobile phones name and pricelist using requests.get()

https://www.daraz.com.np/catalog/?spm=a2a0e.tm80335409.search.d_go&q=samsung%20mobile%20phones

1. **Import Libraries**

    - import requests
    - import pandas as pd

- requests → allows your Python code to **send HTTP requests** (like a browser) and get the website's data.
- pandas → allows you to **store and manipulate the scraped data** in a structured format (DataFrame) and later save it to CSV or Excel.

2. **Set the API URL**

    - API_URL = "https://www.daraz.com.np/catalog/?spm=a2a0e.tm80335409.search.d_go&q=samsung%20mobile%20phones"

- This is the **main search URL** for Daraz (Nepal) for **Samsung mobile phones**.
- This URL simulates what a browser would open when you search for "Samsung mobile phones" on Daraz.

3. **Define Query Parameters**

    - params = {   "ajax": "true",   "q": "samsung mobile phones",   "spm": "a2a0e.tm80335409.search.d_go",   "_keyori": "ss",   "from": "input"}

- These are **GET parameters** that the website expects to return **JSON data** instead of the full HTML page.
- **Explanation of key params**:
    - "ajax": "true" → tells the site to return JSON data (like an API response), not full HTML.
    - "q" → the search query ("samsung mobile phones").
    - "spm" → site-specific parameter used internally for tracking clicks.

  - "_keyori" → usually indicates search origin ("ss" = search string).
  - "from" → source of the search input.
- These parameters mimic what the browser sends when you type a query in Daraz.

4. **Send the Request**

```
response = requests.get(API_URL, params=params, headers=headers)
```

- requests.get() sends a **GET request** to the website.
- params=params → attaches the query parameters to the URL.
- headers=headers → adds the browser header.
- response → contains the server's reply (JSON in this case).

5. Parse JSON Data

  - data = response.json()
- Converts the server response into a **Python dictionary**.
- Daraz's ajax=true response is JSON, so response.json() gives us structured data.

6. Extract the List of Items

  - items = data.get("mods", {}).get("listItems", [])
- data.get("mods", {}) → looks inside the "mods" key safely (returns {} if it doesn't exist).
- .get("listItems", []) → gets the list of products.
- If either key is missing, it safely returns an **empty list**.
- items now contains **all product information** returned by the search API.

7. Initialize Lists to Store Data

  - names = []
  - prices = []

    - links = []

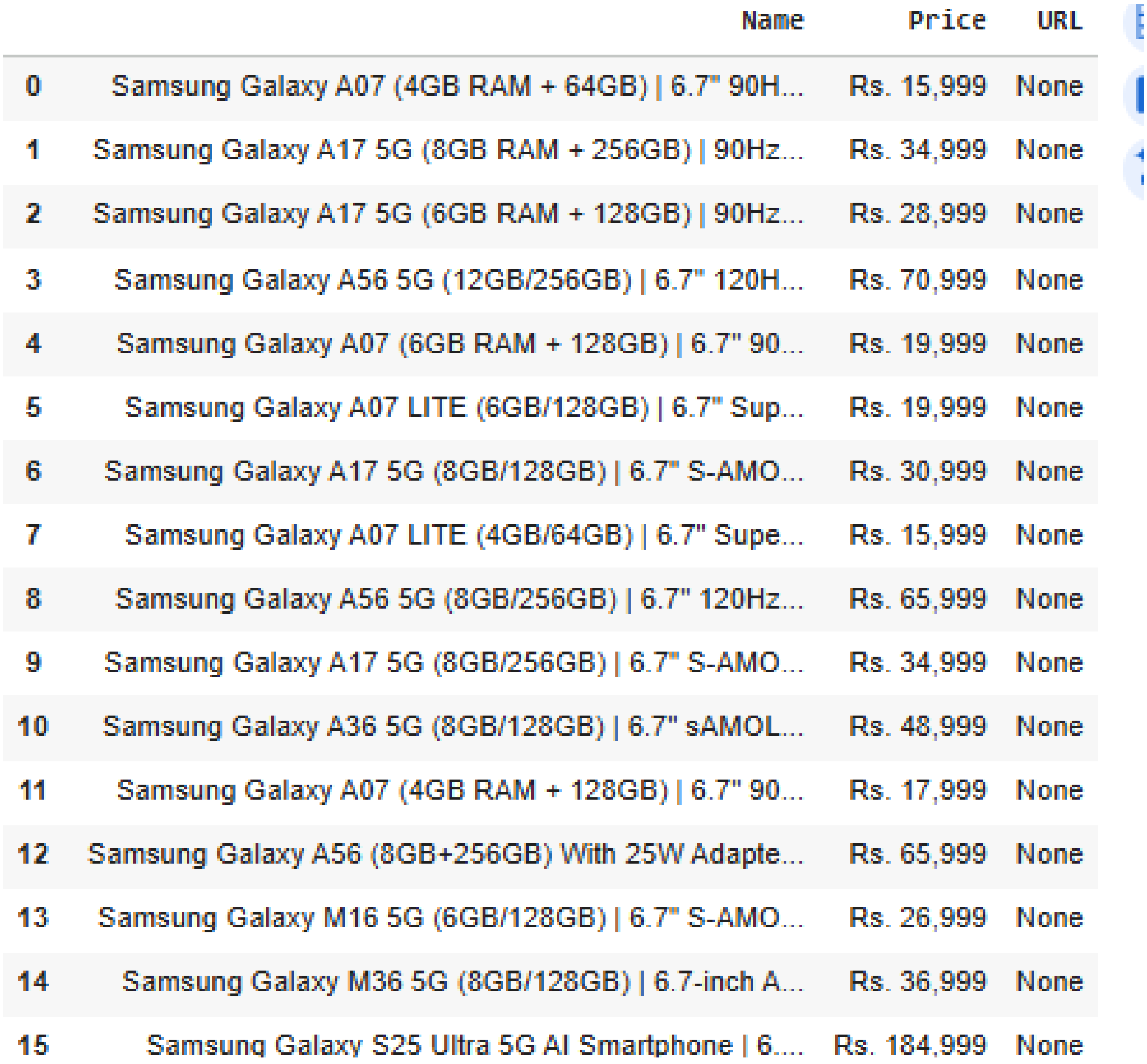
- We create empty lists to **store the product name, price, and link** for each item.
- These lists will later be combined into a DataFrame.

8. Create a Pandas DataFrame

- df = pd.DataFrame({   "Name": names,   "Price": prices,   "URL": links})
- Converts the lists into a **table-like structure** (DataFrame).
- Columns: Name, Price, URL
- Each row corresponds to **one product**.

Fullcode:

```
import requests

import pandas as pd


API_URL = "https://www.daraz.com.np/catalog/?spm=a2a0e.tm80335409.search.d_go&q=samsung%20mobile%20phones"

#query Parameters: These values simulate what the browser passes internally. Daraz uses ajax=true to return JSON response
params = {

   "ajax": "true",

   "q": "samsung mobile phones",

   "spm": "a2a0e.tm80335409.search.d_go",

   "_keyori": "ss",

   "from": "input"

}


headers = { "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
}
response = requests.get(API_URL, params=params, headers=headers)
data = response.json()
items = data.get("mods", {}).get("listItems", [])

names = []
prices = []
links = []
for it in items:
    name = it.get("name")
    price = it.get("priceShow")

    # Safe product URL handling
    url_part = it.get("productUrl")
    if url_part:
        full_url = "https:" + url_part
    else:
        full_url = None  # or "" if you prefer

    names.append(name)
    prices.append(price)
    links.append(full_url)
df = pd.DataFrame({
    "Name": names,
    "Price": prices,
    "URL": links
})
df
```



	Name	Price	URL
0	Samsung Galaxy A07 (4GB RAM + 64GB) | 6.7" 90H...	Rs. 15,999	None
1	Samsung Galaxy A17 5G (8GB RAM + 256GB) | 90Hz...	Rs. 34,999	None
2	Samsung Galaxy A17 5G (6GB RAM + 128GB) | 90Hz...	Rs. 28,999	None
3	Samsung Galaxy A56 5G (12GB/256GB) | 6.7" 120H...	Rs. 70,999	None
4	Samsung Galaxy A07 (6GB RAM + 128GB) | 6.7" 90...	Rs. 19,999	None
5	Samsung Galaxy A07 LITE (6GB/128GB) | 6.7" Sup...	Rs. 19,999	None
6	Samsung Galaxy A17 5G (8GB/128GB) | 6.7" S-AMO...	Rs. 30,999	None
7	Samsung Galaxy A07 LITE (4GB/64GB) | 6.7" Supe...	Rs. 15,999	None
8	Samsung Galaxy A56 5G (8GB/256GB) | 6.7" 120Hz...	Rs. 65,999	None
9	Samsung Galaxy A17 5G (8GB/256GB) | 6.7" S-AMO...	Rs. 34,999	None
10	Samsung Galaxy A36 5G (8GB/128GB) | 6.7" sAMOL...	Rs. 48,999	None
11	Samsung Galaxy A07 (4GB RAM + 128GB) | 6.7" 90...	Rs. 17,999	None
12	Samsung Galaxy A56 (8GB+256GB) With 25W Adapte...	Rs. 65,999	None
13	Samsung Galaxy M16 5G (6GB/128GB) | 6.7" S-AMO...	Rs. 26,999	None
14	Samsung Galaxy M36 5G (8GB/128GB) | 6.7-inch A...	Rs. 36,999	None
15	Samsung Galaxy S25 Ultra 5G AI Smartphone | 6....	Rs. 184,999	None