

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in Information Technology

By

D DIYA MARY
REG. NO. 39120026



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

NOVEMBER 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI- 600119
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONA FIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **D DIYA MARY** (Reg. No: 39120026) who carried out the project entitled "**Reading raw text from web in JSON, CSV, TSV format**" under my supervision from June 2021 to November 2021.

Internal Guide

Dr. Bevish Jinila. Y M.E., Ph.D

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **D DIYA MARY** hereby declare that the project report entitled **Reading raw text from web in JSON, CSV, TSV format** done by me under the guidance of **Dr. Bevish Jinila. Y** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

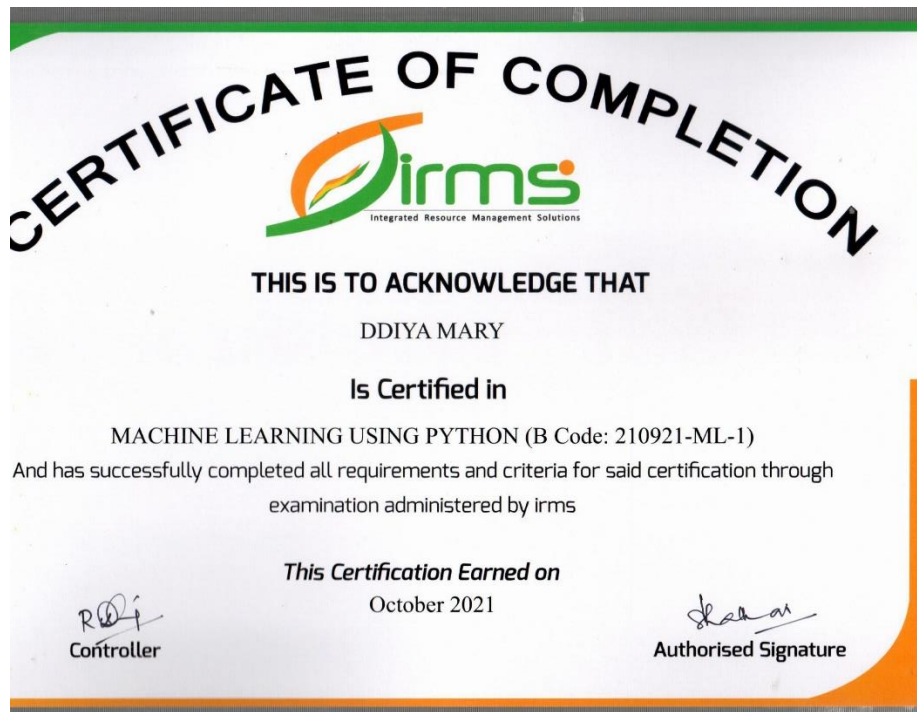
I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr. R. Subhashini, M.E., Ph.D., Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Bevish Jinila. Y, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



ABSTRACT

When we collect information from website we usually copy and paste the data which can be time consuming and tedious. Web scraping technique can be used to extract large amount of data from websites and is saved to a local file in the computer. It is a process of automation. Web scraping begins with loading the document, parsing, extracting and transforming the data.

In this project, we use BeautifulSoup package which is a python library for parsing structured data to explore the document. We scrape flipkart website and fetch details of the products and store it in a file.

TABLE OF CONTENTS		
CHAPTER	TITLE	PAGE NO
	ABSTRACT	06
1	INTRODUCTION	09
2	AIM AND SCOPE	10
3	MATERIALS AND METHODS	11
	3.1 MATERIALS	11
	3.1.1 HARDWARE SPECIFICATION	11
	3.1.2 SOFTWARE SPECIFICATION	11
	3.2 METHODS AND MODULES	12
	3.2.1 SYSTEM ARCHITECTURE	12
	3.2.2 MODULES	13
4	RESULT AND DISCUSSION	14
	4.1 RESULT	15
	4.2 DISCUSSION	16
6	CONCLUSION	17
	REFERENCES	18
	APPENDIX A: SNAP SHOTS	19
	APPENDIX B: SOURCE CODE	23

Fig No	Figure Number	Page No.
1	3.1	12
2	3.2	13
3	3.3	14
4	3.4	14
5	4.1	15
6	4.2	16
7	4.3	16

CHAPTER 1

INTRODUCTION

In today's competitive world everybody is looking for ways to innovate and make use of new technologies. Web scraping provides a solution for those who want to get access to structured web data in an automated fashion.

In general, web data extraction is used by people and businesses who want to make use of the vast amount of publicly available web data to make smarter decisions. If you've ever copied and pasted information from a website, you've performed the same function as any web scraper, only on a microscopic, manual scale. Unlike the mundane, mind-numbing process of manually extracting data, web scraping uses intelligent automation to retrieve hundreds, millions, or even billions of data points from the internet's seemingly endless frontier.

Web scraping refers to extraction of data from a website. The information collected is exported in the format the user requires. First, the web scraper will be given URL to load before scraping. Then scraper extracts the required data from the page. Lastly, the web scraper will output the data which is collected into file.

In this project, we inspect a website by viewing its html code. We use beautifulsoup package to fetch the required details from the website. This way, collecting raw text is simplified and can be easily viewed in a file.

CHAPTER 2

AIM AND SCOPE

Web scraping has a huge range of applications. A data scraping tool can help you automate the process of extracting information from other websites, quickly and accurately. It can also make sure the data you've extracted is neatly organized, making it easier to analyze and use for other purposes.

So, we will use this technique to gather the required data and organize and save it in formats that we desire.

First, beginning with inspecting the webpage, we open a connection with the website and access the html code. Followed by parsing the html with BeautifulSoup package using the prettify function. Then we extract the required data, product name, price and its ratings. Lastly, we save it in the file in three formats: json, csv and tsv.

The purpose of this project is to extract data from website, most of the data in website is in html format and hence it has to be parsed and data has to be retrieved from it.

CHAPTER 3

MATERIALS AND METHODS

3.1 MATERIALS

3.1.1 HARDWARE SPECIFICATION

Processor : Intel Core i3 or AMD Ryzen CPU
Internet Connection : Fiber net, Wifi
RAM : 4GB

3.1.2 SOFTWARE SPECIFICATION

Operating System : Windows, Linux, Mac
Language : Python 3
Software : Jupyter Notebook (Anaconda3)

Python is an interpreted high-level general purpose programming language. It was developed at Centrum Wiskunde & Informatica (CWI) in the Netherlands in the late 1980s. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible (with modules). This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Python strives for a simpler, less-clustered syntax and grammar while giving developers a choice in their coding methodology.

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating notebook documents. It is a browser based REPL containing an ordered list of input/output cells. Underneath the interface, a notebook is a JSON document, ending with “.ipynb” extension.

3.2 METHOD AND MODULES

3.2.1 SYSTEM ARCHITECTURE

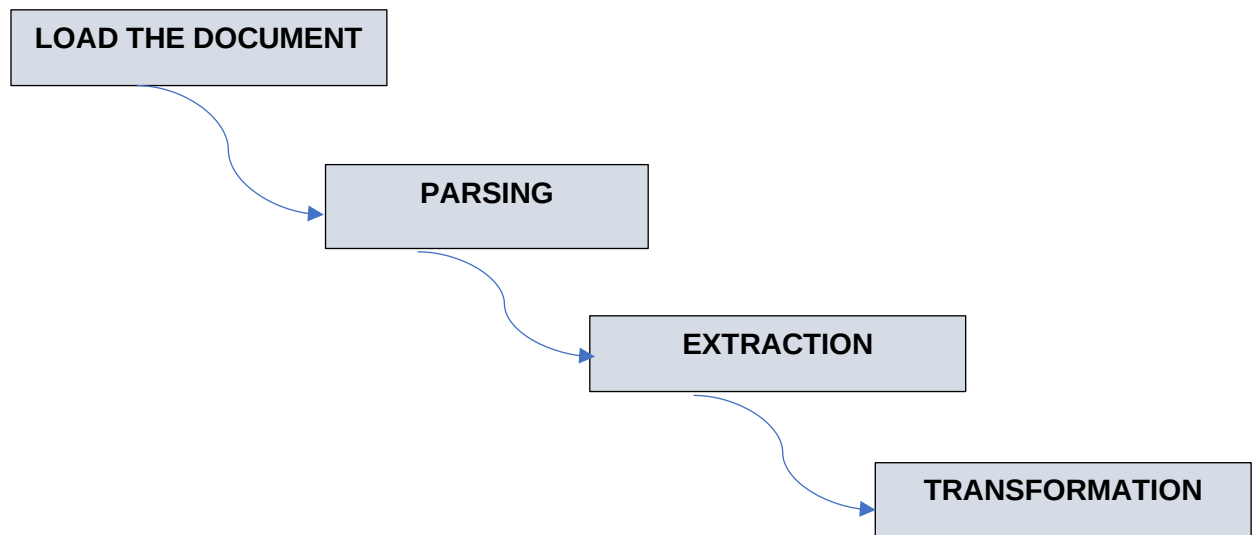


Fig.3.1 Architecture Diagram

The fig 3.1 shows the architectural diagram which shows the process of how this project is carried out. The first step is Load the Document, which is the html page. Followed by parsing, which is interpreting the document to make our search possible. Next is extraction that is the data we require which can be like price or reviews of the product. The last step is transformation, where data is converted into useful formats.

3.2.2 MODULES

There are four modules:

- Load the Document
- Parsing
- Extraction
- Transformation

3.2.2.1 LOADING DOCUMENT

Inspecting your data source should be the first step before extracting data from website. To better understand the source, use developer tools to explore site's document object model. Right click on the element on the page and click on inspect, it will display the HTML text. When you hover over the HTML text, you can see corresponding elements get highlighted. The URL of the website can be used to proceed with further steps.

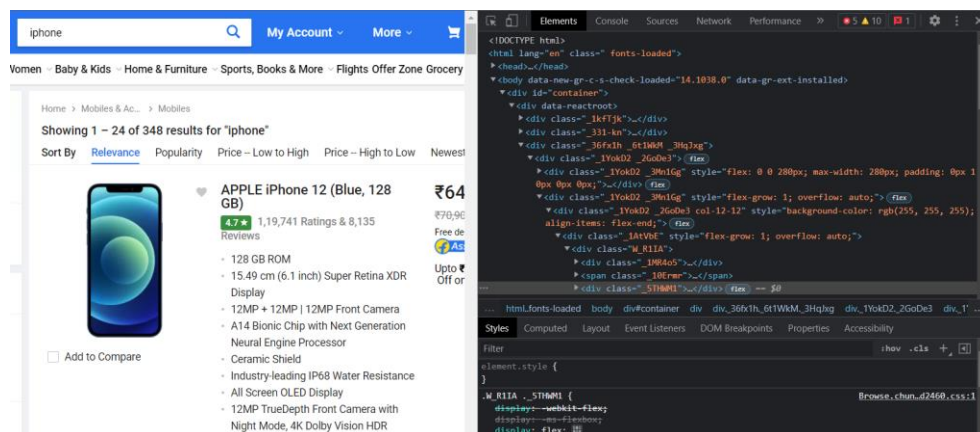


Fig 3.2 HTML code of the website

3.2.2.2 PARSING

We use BeautifulSoup which is a python library for parsing structured data. Parsing is done to make it more accessible and pick the data we want. We interpret the document which makes searching of data possible. In our case, we'll be using HTML document. Using for loop, we can get all the data from all div classes.

PARSE THE HTML FILE

```
In [ ]:  page_soup = soup(page_html,"html.parser")
```

Fig 3.3 Parsing the HTML file

3.2.2.3 EXTRACTION

In this step, we'll extract the required data from the website. Here, we extract product name, price, and review. A container which is the main div class will be used which contains all the elements present. Using the respective class from the HTML code, we print the required data.

3.2.2.4 TRANSFORMATION

After gathering the data, we will store it in three formats: JSON, CSV, and TSV. We'll create a file and use the open function and instantiate the mode write and store it in the file.

PRINT DATA IN CSV FILE

```
In [10]:  filename = "C:\\Users\\Diya\\OneDrive\\Desktop\\ML\\outputs\\products.csv"
          f = open(filename,"w")
          headers = "Product_Name,Pricing,Ratings\n"
          f.write(headers)

Out[10]:  29
```

Fig 3.4 retrieving data in CSV format

CHAPTER 4

RESULT AND DISCUSSION

4.1 RESULT

After we extract the data that is product name, price, and ratings, it is saved in file in CSV, TSV and JSON formats. As displayed below, it can be seen that the first column is product name followed by its price and rating.

	A	B	C	D	E	F
1	Product_N	Pricing	Ratings			
2	APPLE iPhc	Rs.64999	4.71	19	741	
3	APPLE iPhc	Rs.59999	4.71	19	741	
4	APPLE iPhc	Rs.29999	4.51	31	76	
5	APPLE iPhc	Rs.29999	4.51	31	76	
6	APPLE iPhc	Rs.59999	4.71	19	741	
7	APPLE iPhc	Rs.64999	4.71	19	741	
8	APPLE iPhc	Rs.59999	4.71	19	741	
9	APPLE iPhc	Rs.49999	4.558	115		
10	APPLE iPhc	Rs.59999	4.71	19	741	
11	APPLE iPhc	Rs.64999	4.71	19	741	
12	APPLE iPhc	Rs.49999	4.558	115		
13	APPLE iPhc	Rs.29999	4.51	31	76	
14	APPLE iPhc	Rs.49999	4.558	115		
15	APPLE iPhc	Rs.59999	4.71	19	741	
16	APPLE iPhc	Rs.64999	4.71	19	741	
17	APPLE iPhc	Rs.59999	4.71	19	741	
18	APPLE iPhc	Rs.49999	4.558	115		
19	APPLE iPhc	Rs.56999	4.558	115		
20	APPLE iPhc	Rs.49999	4.558	115		
21	APPLE iPhc	Rs.54999	4.558	115		
22	APPLE iPhc	Rs.56999	4.558	115		
23	APPLE iPhc	Rs.34999	4.51	31	76	
24	APPLE iPhc	Rs.54999	4.558	115		
25	APPLE iPhc	Rs.54999	4.558	115		
26						

Fig 4.1 Output in CSV format

```

1 Product_Name,Pricing,Ratings
2 APPLE iPhone 12 (Blue| 128 GB),Rs.64999,4.71,19,741
3 APPLE iPhone 12 (Black| 64 GB),Rs.59999,4.71,19,741
4 APPLE iPhone SE (White| 64 GB),Rs.29999,4.51,31,076
5 APPLE iPhone SE (Black| 64 GB),Rs.29999,4.51,31,076
6 APPLE iPhone 12 (Purple| 64 GB),Rs.59999,4.71,19,741
7 APPLE iPhone 12 (Red| 128 GB),Rs.64999,4.71,19,741
8 APPLE iPhone 12 (Red| 64 GB),Rs.59999,4.71,19,741
9 APPLE iPhone 12 Mini (White| 64 GB),Rs.49999,4.558,115
10 APPLE iPhone 12 (White| 64 GB),Rs.59999,4.71,19,741
11 APPLE iPhone 12 (Black| 128 GB),Rs.64999,4.71,19,741
12 APPLE iPhone 12 Mini (Purple| 64 GB),Rs.49999,4.558,115
13 APPLE iPhone SE (Red| 64 GB),Rs.29999,4.51,31,076
14 APPLE iPhone 12 Mini (Black| 64 GB),Rs.49999,4.558,115
15 APPLE iPhone 12 (Green| 64 GB),Rs.59999,4.71,19,741
16 APPLE iPhone 12 (Green| 128 GB),Rs.64999,4.71,19,741
17 APPLE iPhone 12 (Blue| 64 GB),Rs.59999,4.71,19,741
18 APPLE iPhone 12 Mini (Blue| 64 GB),Rs.49999,4.558,115
19 APPLE iPhone 12 Mini (Blue| 128 GB),Rs.56999,4.558,115
20 APPLE iPhone 12 Mini (Green| 64 GB),Rs.49999,4.558,115
21 APPLE iPhone 12 Mini (Black| 128 GB),Rs.54999,4.558,115
22 APPLE iPhone 12 Mini (Red| 128 GB),Rs.56999,4.558,115
23 APPLE iPhone SE (Black| 128 GB),Rs.34999,4.51,31,076
24 APPLE iPhone 12 Mini (Purple| 128 GB),Rs.54999,4.558,115
25 APPLE iPhone 12 Mini (White| 128 GB),Rs.54999,4.558,115

```

Fig 4.2 Output in TSV format

	A	B	C	D	E	F	G
1	Product_N	Pricing	Ratings				
2	APPLE iPhc	Rs.64999	4.71	19	741		
3	APPLE iPhc	Rs.59999	4.71	19	741		
4	APPLE iPhc	Rs.29999	4.51	31	76		
5	APPLE iPhc	Rs.29999	4.51	31	76		
6	APPLE iPhc	Rs.59999	4.71	19	741		
7	APPLE iPhc	Rs.64999	4.71	19	741		
8	APPLE iPhc	Rs.59999	4.71	19	741		
9	APPLE iPhc	Rs.49999	4.558	115			
10	APPLE iPhc	Rs.59999	4.71	19	741		
11	APPLE iPhc	Rs.64999	4.71	19	741		
12	APPLE iPhc	Rs.49999	4.558	115			
13	APPLE iPhc	Rs.29999	4.51	31	76		
14	APPLE iPhc	Rs.49999	4.558	115			
15	APPLE iPhc	Rs.59999	4.71	19	741		
16	APPLE iPhc	Rs.64999	4.71	19	741		
17	APPLE iPhc	Rs.59999	4.71	19	741		
18	APPLE iPhc	Rs.49999	4.558	115			
19	APPLE iPhc	Rs.56999	4.558	115			
20	APPLE iPhc	Rs.49999	4.558	115			
21	APPLE iPhc	Rs.54999	4.558	115			
22	APPLE iPhc	Rs.56999	4.558	115			
23	APPLE iPhc	Rs.34999	4.51	31	76		
24	APPLE iPhc	Rs.54999	4.558	115			
25	APPLE iPhc	Rs.54999	4.558	115			

Fig 4.3 Output in JSON format

4.2 DISCUSSION

From these outputs, we can organize it and use it for further purposes like analyzing, reviewing. This helps in getting data in a structured format compared to how we copy and paste the data and makes the job more simplified.

CHAPTER 5

CONCLUSION

Whether we want to start a new project or churn out a new strategy for existing business, we need to access and analyze a vast amount of data. There maybe data in websites which cannot be copy and pasted, in such cases also web scraping can help in extracting data which we want. Even if we could copy paste, saving it in our desired format is not possible, but with web scraping you could save it in the format required and further analyze the data.

Therefore, web scraping simplifies the process of extracting data, speeds it up by automating it and creates easy access to the scrapped data by providing it in required formats.

This project helps in getting the required data from a website using Beautifulsoup package and save it in json, csv and tsv formats. This can further be extended to adapt to more websites and more packages can be used to enhance web scraping and make it more efficient.

REFERENCES

1. What is Web scraping, imperva.com
2. Web scraping by Colm Kenny, zyte.com
3. Web scraping by Martin Perez, [parsehub](http://parsehub.com) blog
4. Beautiful soup by Martin Breuss, realpython.com
5. Implementing web scraping with python, [geeksforgeeks](http://geeksforgeeks.com)

APPENDIX A

SNAP SHOTS

Jupyter PT-1 PROJECT Last Checkpoint: 2 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

IMPORT REQUIRED LIBRARIES

```
In [1]: from bs4 import BeautifulSoup as soup
        from urllib.request import urlopen as uReq
```

SPECIFY THE URL OF THE WEBSITE

```
In [2]: my_url = "https://www.flipkart.com/search?q=iphone&otracker=start&as-show=on&as=off"
```

OPEN A CONNECTION AND GATHER THE DATA

```
In [3]: uClient = uReq(my_url)
        page_html = uClient.read()
        uClient.close()
```

PARSE THE HTML FILE

```
In [4]: page_soup = soup(page_html, "html.parser")
```

FIND ALL THE DIV WHICH IS A CLASS

```
In [44]: containers = page_soup.findAll("div", {"class": "_2kH4tA"})
        print(len(containers))
```

24

STRUCTURE THE HTML FILE

```
In [45]: print(soup.prettify(containers[0]))
```

```
<div class="_2kH4tA">
  <a class="_1fQZEK" href="/apple-iphone-12-blue-128-gb/p/itm02853ae92e90a?pid=MOBFwBYZKPTZF9VG&lid=LSTMOBFwBYZKPTZF9VGJDSAM0&marketplace=FLIPKART&q=iphone&store=tyy%2F4io&srno=s_1_1&otracker=search&fm=organic&iid=2dfa1d50-5eb2-4355-b13a-6b70f52fc27f.MOBFwBYZKPTZF9VG.SEARCH&ppt=None&ppn=None&ssid=s8bznrdudc0000001636542318221&qh=0b3f45b266a97d70" rel="noopener noreferrer" target="_blank">
    <div class="MIX0uux">
      <div class="_2QcLo">
        <div>
          <div class="CXw8mj" style="height:200px;width:200px">
            
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="3wLduG">
    <div class="_3PzNI">
      <span class="f3A4_V">
        <label class="_2iDkf8">
          <input class="_30VH1S" readonly="" type="checkbox"/>
        </div>
      </div>
    </div>
  </div>
```

PRINT PRODUCT NAME

```
In [7]: container = containers[0]
print(container.div.img["alt"])

APPLE iPhone 12 (Blue, 128 GB)
```

PRINT PRODUCT PRICE

```
In [8]: price = container.findAll("div",{"class":"col col-5-12 n1I3QM"})
print(price[0].text)

₹64,999₹70,9008% offUpto ₹14,250 Off on Exchange
```

PRINT PRODUCT RATINGS

```
In [9]: ratings = container.findAll("div",{"class":"gUuXy-"})
print(ratings[0].text)

4.71,19,741 Ratings & 8,135 Reviews
```

PRINT DATA IN CSV FILE

```
In [10]: filename = "C:\\Users\\Diya\\OneDrive\\Desktop\\ML\\outputs\\products.csv"
f = open(filename,"w")
headers = "Product_Name,Pricing,Ratings\n"
f.write(headers)
```

Out[10]: 29

GET DATA FOR ALL DIV CLASSES

```
In [11]: for container in containers:
    product_name = container.div.img["alt"]

    price_container = container.findAll("div",{"class":"col col-5-12 n1I3QM"})
    price = price_container[0].text.strip()

    rating_container = container.findAll("div",{"class":"gUuXy-"})
    rating = rating_container[0].text

    #print("product_name:" + product_name)
    # print("price:" + price)
    # print("rating:" + rating)

    #string parsing
    trim_price = ''.join(price.split(','))
    rm_rupee = trim_price.split("₹")
    #print(rm_rupee)
    add_rs_price = "Rs." + rm_rupee[1]
    split_price = add_rs_price.split('E')
    final_price = split_price[0]

    split_rating = rating.split(" ")
    final_rating = split_rating[0]

    print(product_name.replace(",","|") + "," + final_price + "," + final_rating + "\n")
    f.write(product_name.replace(",","|") + "," + final_price + "," + final_rating + "\n")
f.close()
```

APPLE iPhone 12 (Blue| 128 GB),Rs.64999,4.71,19,741
APPLE iPhone 12 (Black| 64 GB),Rs.59999,4.71,19,741
APPLE iPhone 12 (Purple| 64 GB),Rs.59999,4.71,19,741
APPLE iPhone SE (Black| 64 GB),Rs.29999,4.51,31,076
APPLE iPhone 12 (Red| 128 GB),Rs.64999,4.71,19,741
APPLE iPhone SE (White| 64 GB),Rs.29999,4.51,31,076
APPLE iPhone 12 (White| 64 GB),Rs.59999,4.71,19,741
APPLE iPhone 12 (Red| 64 GB),Rs.59999,4.71,19,741
APPLE iPhone 12 Mini (White| 64 GB),Rs.49999,4.558,115
APPLE iPhone 12 (Black| 128 GB),Rs.64999,4.71,19,741
APPLE iPhone 12 Mini (Purple| 64 GB),Rs.49999,4.558,115
APPLE iPhone SE (Red| 64 GB),Rs.29999,4.51,31,076
APPLE iPhone 12 Mini (Black| 64 GB),Rs.49999,4.558,115
APPLE iPhone 12 (Blue| 64 GB),Rs.59999,4.71,19,741
APPLE iPhone 12 (Green| 64 GB),Rs.59999,4.71,19,741
APPLE iPhone 12 Mini (Blue| 64 GB),Rs.49999,4.558,115
APPLE iPhone 12 Mini (Green| 64 GB),Rs.49999,4.558,115
APPLE iPhone 12 Mini (Red| 128 GB),Rs.54999,4.558,115
APPLE iPhone 12 Mini (Purple| 128 GB),Rs.54999,4.558,115
APPLE iPhone 12 Mini (Black| 128 GB),Rs.54999,4.558,115
APPLE iPhone 12 Mini (White| 128 GB).Rs.54999.4.558.115

PRINT DATA IN TSV FORMAT

```
In [81]: #tsv
filename = "C:\\Users\\Diya\\OneDrive\\Desktop\\ML\\outputs\\products.tsv"
f = open(filename, "w")
headers = "Product_Name,Pricing,Ratings\n"
f.write(headers)
```

Out[81]: 29

```
In [82]: #for container in containers:
product_name = container.div.img["alt"]

price_container = container.findAll("div", {"class": "col col-5-12 n1I3QM"})
price = price_container[0].text.strip()

rating_container = container.findAll("div", {"class": "gUuXy-"})
rating = rating_container[0].text

#print("product_name:" + product_name)
# print("price:" + price)
# print("rating:" + rating)

#string parsing
trim_price = ''.join(price.split(','))
rm_rupee = trim_price.split("₹")
#print(rm_rupee)
add_rs_price = "Rs." + rm_rupee[1]
split_price = add_rs_price.split('E')
final_price = split_price[0]

split_rating = rating.split(" ")
final_rating = split_rating[0]

f.write(product_name.replace(",", "|") + "," + final_price + "," + final_rating + "\n")
f.close()
```

PRINT DATA IN JSON FORMAT

```
In [14]: #json
filename = "C:\\Users\\Diya\\OneDrive\\Desktop\\ML\\outputs\\.json"
f = open(filename, "w")
headers = "Product_Name,Pricing,Ratings\n"
f.write(headers)
```

Out[14]: 29

```
In [15]: #for container in containers:
product_name = container.div.img["alt"]

price_container = container.findAll("div", {"class": "col col-5-12 n1I3QM"})
price = price_container[0].text.strip()

rating_container = container.findAll("div", {"class": "gUuXy-"})
rating = rating_container[0].text

#print("product_name:" + product_name)
# print("price:" + price)
# print("rating:" + rating)

#string parsing
trim_price = ''.join(price.split(','))
rm_rupee = trim_price.split("₹")
#print(rm_rupee)
add_rs_price = "Rs." + rm_rupee[1]
split_price = add_rs_price.split('E')
final_price = split_price[0]

split_rating = rating.split(" ")
final_rating = split_rating[0]

print(product_name.replace(",", "|") + "," + final_price + "," + final_rating + "\n")
f.write(product_name.replace(",", "|") + "," + final_price + "," + final_rating + "\n")
f.close()
```

APPENDIX B

SOURCE CODE

IMPORT REQUIRED LIBRARIES

```
from bs4 import BeautifulSoup as soup
from urllib.request import urlopen as uReq
```

SPECIFY THE URL OF THE WEBSITE

```
my_url="https://www.flipkart.com/search?q=iphone&otracker=start&as-  
show=on&as=off"
```

OPEN A CONNECTION AND GATHER THE DATA

```
uClient = uReq(my_url)
page_html = uClient.read()
uClient.close()
```

PARSE THE HTML FILE

```
page_soup = soup(page_html,"html.parser")
```

FIND ALL THE DIV WHICH IS A CLASS

```
containers = page_soup.findAll("div", {"class": "_2kHMtA"})
#print(len(containers))
```

STRUCTURE THE HTML FILE

```
print(soup.prettify(containers[0]))
```

PRINT PRODUCT NAME, PRICE AND RATING

```
container = containers[0]  
print(container.div.img["alt"])
```

```
price = container.findAll("div",{"class":"col col-5-12 nll3QM"})  
print(price[0].text)
```

```
ratings = container.findAll("div",{"class":"gUuXy-"})  
print(ratings[0].text)
```

PRINT DATA IN CSV FILE

```
filename = "C:\\Users\\Diya\\OneDrive\\Desktop\\ML\\outputs\\products.csv"  
f = open(filename,"w")  
headers = "Product_Name,Pricing,Ratings\n"  
f.write(headers)
```

```
for container in containers:
```

```
    product_name = container.div.img["alt"]
```

```
    price_container = container.findAll("div",{"class":"col col-5-12 nll3QM"})  
    price = price_container[0].text.strip()
```

```
    rating_container = container.findAll("div",{"class":"gUuXy-"})  
    rating = rating_container[0].text
```

```

    #print("product_name:" + product_name)
    # print("price:" + price)
    # print("rating:" + rating)

    #string parsing
    trim_price = ".join(price.split(','))
    rm_rupee = trim_price.split("₹")
    #print(rm_rupee)
    add_rs_price = "Rs." + rm_rupee[1]
    split_price = add_rs_price.split('E')
    final_price = split_price[0]

    split_rating = rating.split(" ")
    final_rating = split_rating[0]

    print(product_name.replace(",","|") + "," +final_price + "," +final_rating + "\n")
    f.write(product_name.replace(",","|") + "," +final_price + "," +final_rating + "\n")
f.close()

```

PRINT DATA IN TSV FILE

```

#tsv
filename = "C:\\Users\\Diya\\OneDrive\\Desktop\\ML\\outputs\\products.tsv"
f = open(filename,"w")
headers = "Product_Name,Pricing,Ratings\n"
f.write(headers)

for container in containers:
    product_name = container.div.img["alt"]

    price_container = container.findAll("div",{"class":"col col-5-12 nll3QM"})

```

```

price = price_container[0].text.strip()

rating_container = container.findAll("div",{"class":"gUuXy-"})
rating = rating_container[0].text

#print("product_name:" + product_name)
# print("price:" + price)
# print("rating:" + rating)

#string parsing
trim_price = ".join(price.split(','))
rm_rupee = trim_price.split("₹")
#print(rm_rupee)
add_rs_price = "Rs." + rm_rupee[1]
split_price = add_rs_price.split('E')
final_price = split_price[0]

split_rating = rating.split(" ")
final_rating = split_rating[0]

print(product_name.replace(",","|") + "," +final_price + "," +final_rating + "\n")
f.write(product_name.replace(",","|") + "," +final_price + "," +final_rating + "\n")
f.close()

```

PRINT DATA IN JSON FILE

```

#json
filename = "C:\\Users\\Diya\\OneDrive\\Desktop\\ML\\outputs\\.json"
f = open(filename,"w")
headers = "Product_Name,Pricing,Ratings\n"
f.write(headers)

```

```
for container in containers:
```

```
    product_name = container.div.img["alt"]
```

```
    price_container = container.findAll("div",{"class":"col col-5-12 nll3QM"})
```

```
    price = price_container[0].text.strip()
```

```
    rating_container = container.findAll("div",{"class":"gUuXy-"})
```

```
    rating = rating_container[0].text
```

```
    #print("product_name:" + product_name)
```

```
    # print("price:" + price)
```

```
    # print("rating:" + rating)
```

```
    #string parsing
```

```
    trim_price = ".join(price.split(','))
```

```
    rm_rupee = trim_price.split("₹")
```

```
    #print(rm_rupee)
```

```
    add_rs_price = "Rs." + rm_rupee[1]
```

```
    split_price = add_rs_price.split('E')
```

```
    final_price = split_price[0]
```

```
    split_rating = rating.split(" ")
```

```
    final_rating = split_rating[0]
```

```
    print(product_name.replace(",","|") + "," +final_price + "," +final_rating + "\n")
```

```
    f.write(product_name.replace(",","|") + "," +final_price + "," +final_rating + "\n")
```

```
f.close()
```