

CostOBest

- An effective data visualization tool

1. OBJECTIVE

To help students with the best possible available options of books on various websites.

2. INTRODUCTION

The project is aimed to provide the ease and comfort with which the students can browse the books available on various sites and can buy the same accordingly. The project deals with extracting data from various sites through particular tools like web scraping and then filtering all the extracted results and showing the best possible results altogether at the same place by one click/search. This accounts for the ease with which the students can access search results of different sites and comparing them at the same place on the basis of price , reviews , ratings etc . This will help students to buy the books at best possible price and ratings . As we know that the customers find it convenient to shop online but presence of many websites makes it a bit confusing task for the customer to judge. This project aims at resolving this ambiguity.

3. BLOCK DIAGRAMS AND METHODOLOGY

1. Web sites are picked and API calls are made in step 1 .
2. The response received is to be scraped using beautiful soup.
3. Important information retrieved is to stored in server and rendered.



Fig 1

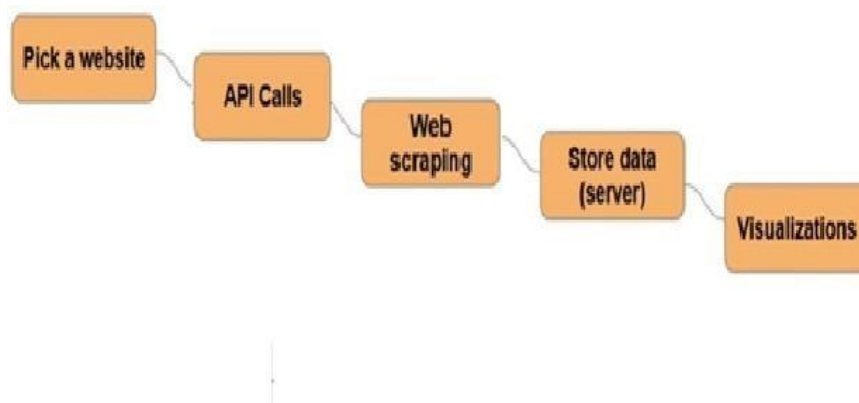


Fig 2

3.1. Websites Considered

We considered 5 websites for finding the book and their prices. The websites taken into consideration are as follows :

1. <https://www.amazon.com/>
2. <https://www.flipkart.com/>
3. <https://www.bookswagon.com/>
4. <https://www.sapnaonline.com/>
5. <https://pustakkosh.com/>

The results from amazon and flipkart have been given more priority than the other three websites because amazon and flipkart provide us with vast variety and range of trusted sellers for every book .

3.1. API Calls

API calls were made to these websites using request module of python . Search query URL of book is generated for each website manually by string processing.

```
pip install requests
```

Installed requests module using above command in terminal.

3.1.1. AMAZON

The search URL for this site: <https://www.amazon.in/s?k=> + book_name

3.1.2. FLIPKART

The search URL for this site: "<https://www.flipkart.com/search?q=>" + book_name + "&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=on&as=off"

3.1.3. BOOKSWAGON

The search URL for this site: "<https://www.bookswagon.com/search-books/>" + book_name

3.1.4. PUSTAKKOSH

The search URL for this site: "https://pustakkosh.com/rent_or_buy_books.php?s=" + book_name

3.1.5. SAPNAONLINE

The search URL for this site: "<https://www.sapnaonline.com/search?keyword=>" + book_name

3.1.6. HEADERS:

We have to attach HEADERS in search URLs because they represent the meta-data associated with the API request and response. Headers carry information for: Request and Response Body.

```
HEADERS = ({'User-Agent':  
            'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,  
            like Gecko) Chrome/44.0.2403.157 Safari/537.36',  
            'Accept-Language': 'en-US, en;q=0.5'})
```

HEADER

```
webpage = requests.get(url, headers=HEADERS)
```

MAKING REQUEST FROM REQUEST MODULE

Webpage above has received the complete HTML code for the requested webpage URL. It can now be used to fetch useful information of the webpage .

3.2. WEB SCRAPING

3.2.1. Beautiful soup

```
pip install BeautifulSoup
```

INSTALLING BEAUTIFULSOUP

The HTML code received above in webpage variable has the complete information about the page. To extract information from it we use beautiful soup module. It converts HTML data into dictionaries in which key values are html tags (div, span , a etc.) and values associated with it are the text content , and classes of those containers . Now we can access the important data values using the key values of the dictionary created by the soup.

```
soup = BeautifulSoup(webpage.text, 'html.parser' )
```

Now soup container has data in mannered form of individual sites.

```
book_classes = 'grid-product__wrap-inner'

books_set_lis = soup.find_all('div', {'class': book_classes})
pstsize = int(len(books_set_lis))
print(pstsize)
# books_data= BeautifulSoup(books_set_lis[0], 'html.parser')
ls = []

def func(i):
    price_class = 'grid-product__price-amount'
    price = books_set_lis[i].find('div', {'class': price_class})
    link_class = 'grid-product__image'
```

```

grid_img_class = 'grid-product__image-wrap'
grid_img_src = books_set_lis[i].find('div', {'class': grid_img_class})
img_src = grid_img_src.find('img')
linkk = books_set_lis[i].find('a', {'class': link_class})

book_title_class = 'grid-product__image'
book_title = books_set_lis[i].find('a', {'class': book_title_class})

# print("title : " , book_title['title'])
# print("price: " , "₹" + price.text)
# print("rating : ",rating.text)
# print("link :", linkk['href'] )
price = price_text_to_stringoing(price.text)

```

EXTRACTING DIFFERENT CHARACTERISTICS OF A BOOK

Now these different characteristics can be stored for processing them under various filterings and algorithms. Characteristics extracted of a particular book are as follows:

PRICE

COMPLETE NAME

RATINGS

WEBSITE

URL OF THE PRODUCT

3.3. STORING DATA

```
container.append([PRICE,RATING,TITLE,WEBSITE,URL,IMAGE_URL])
```

STORING DATA IN CONTAINER

The characteristics collected now have to be stored in some container so that they can undergo filtering and sorting algorithms. A particular result is stored in the form of a list that contains price, ratings, name of the book, url of the book, website name in this defined manner only. More emphasis is laid down to the results of flipkart and amazon as they are quite popular in Indian subcontinent. We have our container that stores all the lists of products. We have considered top results only from the web sites only because only those results guarantee us trusted sellers and better deals.

Now we have to apply some techniques to have our list in a sorted manner. Products have to be sorted on the basis of 2 parameters one being the primary and other being the secondary. Our primary key being price of the book and secondary key being the ratings available for the book. We first sort the list according to price and if two or more elements have the same price then we sort those elements on the basis of their user ratings. The sorting algorithm used is predefined in python.

```
container.sort()
```

SORTING THE CONTAINER

3.5 VISUALIZATION

Streamlit

```
pip install streamlit
```

INSTALLING STREAMLIT ON MACHINE

Streamlit is an open source python library that can turn data scripts into interactive web apps in a convenient way. It is hard task usually to have our scripts written in python to have a web apps using flask. Streamlit comes to the rescue in such situations.

Now we have our container that contains all the information about the products. By using streamlit we can display in the way we like in the form of a web app. We can interact with the user using the same web app and we took the search query from the user using this front end only.

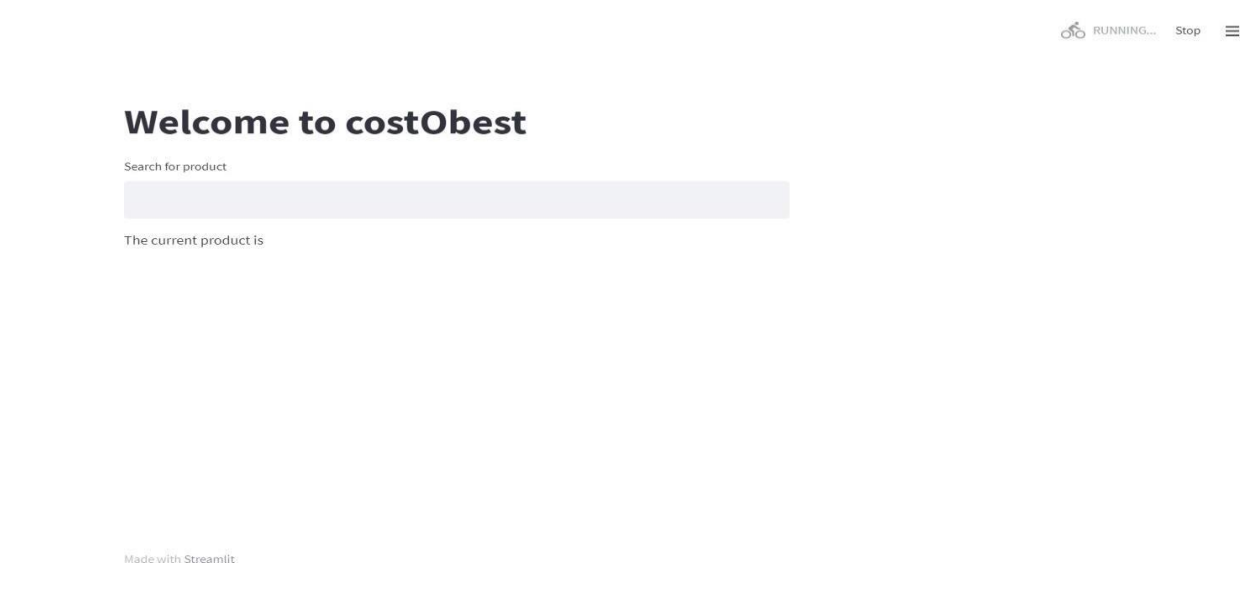
We divide the page into different sections or containers and display the data accordingly.

```
header = st.container() # THIS CONTAINER CONTAINS THE HEADER ACTION
input = st.container() # THIS CONTAINER TAKES INPUT search_results=st.container()
#THIS RENDERS RESULTS
search_results=st.container() #THIS RENDERS RESULTS
```


4. RESULTS:

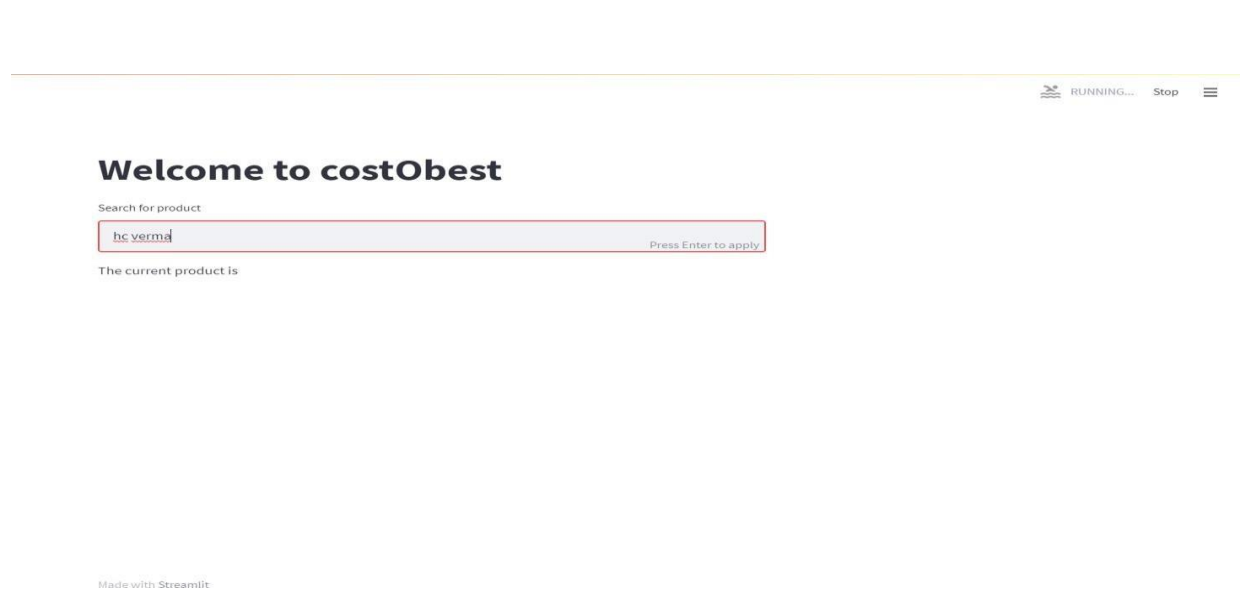
4.1. Homepage

This is the homepage that opens up and takes input from the user



4.2. TAKING INPUT

The name of the is taken as input in the search bar from the user.



4.3. Output

List of the books being rendered in the search result container.

Welcome to costObest

Search for product

concepts of physics hc verma class 11th

The current product is concepts of physics hc verma class 11th

SERIAL NO	PRICE	NAME	LINK	RATINGS	Website
1	₹230	Concepts Of Physics Class-11	L1	4.5 out of 5 stars	flipkart
2	₹240	Foundation Science Physics For Class 9 by H.C Verma Examination 2022-23	L2	4.6 out of 5 stars	amazon
3	₹270	Concepts Of Physics Class-11 (Paperback,	L3	no rating available	flipkart

Physics Class 11 (Paperback, H C VERMA)

4	₹280	Bharati Bhawan Publishers - Concept of physics Volume1 for Class 11, TextBook (By AhaGuru)	L4	4.4 out of 5 stars	amazon
5	₹283	Solution of Concepts of Physics by H.C Verma Part 1 & Part 2 lattest editions	L5	3.8 out of 5 stars	amazon
6	₹430	Concept of Physics by H.C Verma Part - I - Session 2022-23	L6	4.6 out of 5 stars	amazon
7	₹440	Concept of Physics by H.C Verma Part - II	L7	4.6 out of 5 stars	amazon

4.4. URLs being redirected to


Result from flipkart

The screenshot shows the Flipkart product page for 'Concepts of Physics Class-11 (Paperback, H C VERMA)'. The page features a blue header with the Flipkart logo, a search bar, and navigation links like 'Login', 'Become a Seller', 'More', and 'Cart'. Below the header, there's a category bar with options like 'Electronics', 'TVs & Appliances', 'Men', 'Women', 'Baby & Kids', 'Home & Furniture', 'Sports, Books & More', 'Flights', and 'Offer Zone'. The main content area displays the book cover on the left, which has a 'Best Value' tag. To the right of the cover, the product title is 'Concepts Of Physics Class-11 (Paperback, H C VERMA)'. Below the title, it says 'Be the first to Review this product'. The 'Special price' is listed as ₹230, with a crossed-out price of ₹420 and a 46% off discount. There are 'Coupons for you' and 'Available offers' listed below the price. The 'Delivery' section shows a pincode entry field and a 'Check' button. At the bottom, there are 'ADD TO CART' and 'BUY NOW' buttons. The author's name 'H C VERMA' is also visible.

Result from amazon

The screenshot shows the Amazon product page for 'Concepts of Physics Volume 1 for Class 11, TextBook (By AhaGuru) Paperback'. The page features a black header with the Amazon logo, a search bar, and navigation links like 'Hello, AJAY', 'Account & Lists', 'Returns & Orders', and 'Cart'. Below the header, there's a category bar with options like 'Books', 'Advanced Search', 'New Releases & Pre-orders', 'Best Sellers', 'Browse Genres', 'Children's & Young Adult', 'Textbooks', 'Exam Central', and 'All Indian Languages'. The main content area displays the book cover on the left, which has a '1' tag. To the right of the cover, the product title is 'Bharati Bhawan Publishers - Concept of physics Volume1 for Class 11, TextBook (By AhaGuru) Paperback – 1 January 2019'. Below the title, it says 'by Hc Verma (Author)'. The 'Paperback' price is listed as ₹280.00, with a crossed-out price of ₹435.00 and a 36% off discount. There are 'Offers' listed below the price, including 'Cashback', 'No Cost EMI', 'Bank Offer', and 'Partner Offers'. The 'Delivery' section shows a pincode entry field and a 'Check' button. At the bottom, there are 'Add to Cart' and 'Buy Now' buttons. The author's name 'H C VERMA' is also visible.

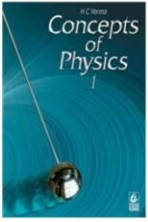
Result from bookswagon



Q
Hello, User
👤
❤️
🛒

[Book](#)
[New Arrivals](#)
[Box Sets](#)
[Best sellers](#)
[Fiction Books](#)
[Award Winners](#)
[Featured Authors](#)
🤖

[Home](#) > [Science & Mathematics](#) > [Physics](#) > [Concepts of Physics: v. 1](#)



Concepts of Physics: v. 1 (PAPERBACK)

By: H.C. Verma (Author) , H.C. Verma (Author) | Publisher Imprint: Bharati Bhavan

★★★★★ | [Write a Review](#)

₹460

₹495

Available

Ships within 8-10 Business Days

₹39 shipping in India per item and low cost Worldwide.

[Buy Now](#) [Add to Wishlist](#)

Best Seller

[SEE ALL](#)

Result from sapnaonline

Express Delivery Available on selected Pin Codes! Shop Now!

India's Largest Book Mall!

Search books by title, author, ISBN

Refine Search

👤 My Account

🛒 My Cart

SHOP BY CATEGORY

KANNADA

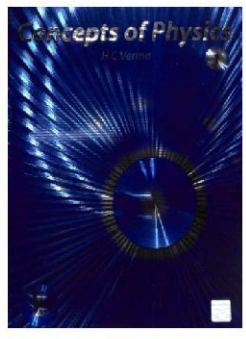
STATIONERY

TOYS

HEALTH CARE

BABY CARE

Best Seller



Home > [Academics](#) > [PCMB](#) > [Pomb to Categorize](#)

Concepts Of Physics 1

by H.C. Verma (Author), Bharati Bhawan P&D (Publisher)

★★★★★ (137 Customers)

₹495

Inclusive of all taxes

Qty

Buying in bulk?

[ADD TO CART](#) [BUY NOW](#)

Check Delivery

🚚 Enter Pincode:

Enter pincode for exact delivery dates / charges and to know if express delivery is available

[Check Availability at Stores](#)

📍 [CLICK HERE](#)

In Stock

Guaranteed Service

International Shipping [Know More](#)

Free Home Delivery [above ₹999](#)

🔗 Share

5. CONCLUSION:

We have comprehensively designed such a system where users according to their needs and demands can search for specific books and be able to obtain the best possible deals across various websites.

6. SCOPE OF IMPROVEMENT:

1. Some irrelevant search results also get fetched and rendered in the webapp.
These results can be filtered using appropriate algorithms.
2. Sites that return JS objects instead of html response could not be scraped like snapdeal.

7. REFERENCES:

1. <https://www.python.org/>
2. <https://docs.python-requests.org/en/latest/>
3. <https://beautiful-soup-4.readthedocs.io/en/latest/>
4. <https://docs.python.org/3/library/html.parser.html>
5. <https://streamlit.io/>

APPENDIX:

Code:

```
# JAI BAJARANG BALI
```

```
from re import L
import string as s
import requests
import webbrowser
import math
import streamlit as st
from bs4 import BeautifulSoup
```

```
def price_text_to_stringoing(price):
    prc=""
    for i in range(0,len(price)):
        if(price[i]>='0' and price[i]<='9'):
            prc+=price[i]
        elif(price[i]=='.'):
            break

    n=len(prc)
    a=0
    for i in range(0,n):
        a+=(int(prc[i])*(pow(10,n-i-1)))

    return a
    #utility_function
```

```
product=''
header = st.container()
inpu = st.container()
```

```

with header :
    st.title('Welcome to costObest')
with inpu :
    title = st.text_input('Search for product','')
    product=title
    st.write('The current product is', product)
search_results=st.container()

with search_results:
    stringo = product
    mn_url = "https://pustakkosh.com/rent_or_buy_books.php"
    url = "https://pustakkosh.com/rent_or_buy_books.php?s="

# amazon

    # url1="&otracker=search&otracker1=search&marketplace=FLIPKART&as-
show=on&as=off"

def price_text_to_stringoing(price):
    prc = ""
    for i in range(0, len(price)):
        if (price[i] >= '0' and price[i] <= '9'):
            prc += price[i]
        elif (price[i] == '.'):
            break

    n = len(prc)
    a = 0
    for i in range(0, n):
        a += (int(prc[i]) * (pow(10, n - i - 1)))
    return a

print("output :")

n = len(stringo)

for i in range(0, n):
    if (stringo[i] == ' '):
        url += "+"
    else:
        url += stringo[i]

```

```

# url+=url1
print(url)

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

book_classes = 'grid-product__wrap-inner'

books_set_lis = soup.find_all('div', {'class': book_classes})
pssize = int(len(books_set_lis))
print(pssize)
# books_data= BeautifulSoup(books_set_lis[0], 'html.parser')
ls = []

def func(i):
    price_class = 'grid-product__price-amount'
    price = books_set_lis[i].find('div', {'class': price_class})
    link_class = 'grid-product__image'
    grid_img_class = 'grid-product__image-wrap'
    grid_img_src = books_set_lis[i].find('div', {'class': grid_img_class})
    img_src = grid_img_src.find('img')
    linkk = books_set_lis[i].find('a', {'class': link_class})

    book_title_class = 'grid-product__image'
    book_title = books_set_lis[i].find('a', {'class': book_title_class})

    # print("title : " , book_title['title'])
    # print("price: " , "₹" + price.text)
    # print("rating : ", rating.text)
    # print("link :", linkk['href'] )
    price = price_text_to_stringoing(price.text)

    ls.append([price, "no rating available" , book_title['title'], linkk['href'],
"pushtkosh", img_src['src']])

mn_url_amazon="https://www.amazon.in"
url_amazon="https://www.amazon.in/s?k="
# stringo=input()
for i in range(0, len(stringo)):
    if(stringo[i]==' '):
        url_amazon+=" "
    else:

```



```

url_amazon+=stringo[i]

HEADERS = ({'User-Agent':
            'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
            Chrome/44.0.2403.157 Safari/537.36',
            'Accept-Language': 'en-US, en;q=0.5'})

# Making the HTTP Request
webpage_amazon = requests.get(url_amazon, headers=HEADERS)

# Creating the Soup Object containing all data
soup_amazon = BeautifulSoup(webpage_amazon.content, "lxml")

print(url_amazon)

# print(soup)

# s-result-item s-asin sg-col-0-of-12 sg-col-16-of-20 sg-col s-widget-spacing-small
sg-col-12-of-16
book_set_Class_amazon='s-result-item s-asin sg-col-0-of-12 sg-col-16-of-20 sg-col s-
widget-spacing-small sg-col-12-of-16'
book_set_list_amazon=soup_amazon.find_all('div',{'class': book_set_Class_amazon})

amazon_size=len(book_set_list_amazon)
print(amazon_size)

def func5(i):

    book_price_class='a-price-whole'
    book_price=book_set_list_amazon[i].find('span',{'class' : book_price_class})
    if(book_price==None):
        return
    # print(book_price.text)
    book_rate=price_text_to_stringoing(book_price.text)

    book_title_class='a-size-medium a-color-base a-text-normal'
    book_title=book_set_list_amazon[i].find('span',{'class':book_title_class})
    # print(book_title.text)
    link_class='a-link-normal s-underline-text s-underline-link-text s-link-style a-
text-normal'
    link=book_set_list_amazon[i].find('a',{'class':link_class})
    # print(mn_url_amazon+link['href'])
    img_class='s-image'
    img_link=book_set_list_amazon[i].find('img',{'class':img_class})
    # print(img_link['src'])

```

```

        book_rating_class="a-icon-alt"
        #
        book_rating_div=book_set_list_amazon[i].find('div',{'class':book_rating_class})
        book_rating=book_set_list_amazon[i].find('span',{'class':book_rating_class})
        rat=""
        if(book_rating==None):
            rat="NO rating available"
        else:
            rat=(book_rating.text)

        ls.append([book_rate,rat,book_title.text,mn_url_amazon+link['href'], "amazon",img
_link['src']])

# flipkart
mn_urlflp = "https://www.flipkart.com"
urlflp = "https://www.flipkart.com/search?q="
url1flp = "&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=on&as=off"

# print("output :")

# n=len(stringo)

for i in range(0, n):
    if (stringo[i] == ' '):
        urlflp += "%20"
    else:
        urlflp += stringo[i]

urlflp += url1flp
print(urlflp)

pageflp = requests.get(urlflp)

soupflp = BeautifulSoup(pageflp.text, 'html.parser')

# page_classflp='_1YokD2 _3Mn1Gg'

# pgeflp=soupflp.find_all('div',{'class' : page_classflp})

book_classesflp = '_4ddWXP'

books_set_lisflp = soupflp.find_all('div', {'class': book_classesflp})

# books_dataflp= BeautifulSoup(books_set_lisflp[0], 'html.parser')

```

```

flipsize = int(len(books_set_lisflp))
print("flip", flipsize)

def func1(i):
    price_class = '_30jeq3'
    price = books_set_lisflp[i].find('div', {'class': price_class})

    rating_class='_3LWZlK'
    rating=books_set_lisflp[i].find('div',{'class': rating_class})
    rat=""
    if(rating is None):
        rat="no rating available"
    else:
        rat=rating.text
        rat=rat+" out of 5 stars"

    link_class = 's1Q9rs'

    linkk = books_set_lisflp[i].find('a', {'class': link_class})

    book_title_class = 's1Q9rs'
    book_title = books_set_lisflp[i].find('a', {'class': book_title_class})
    img_class = '_396cs4 _3exPp9'
    img_src = books_set_lisflp[i].find('img', {'class': img_class})

    # print("title : " , book_title['title'])
    # print("price: " ,price.text)
    # print("rating : ",rating.text)
    # print("link :", mn_url+linkk['href'] )

    price = price_text_to_stringoing(price.text)
    ls.append([price,rat, book_title['title'], mn_urlflp + linkk['href'],
"flipkart", img_src['src']])

# now adding booksswagun.com
# https://www.bookswagun.com/search-books/concept-of-physics-hc-verma

url_booksswagun = "https://www.bookswagun.com/search-books/"

for i in range(0, n):
    if (stringo[i] == ' '):
        url_booksswagun += "-"
    else:
        url_booksswagun += stringo[i]

```

```

print(url_booksswagon)

page_bookswagon = requests.get(url_booksswagon)

soup_bookswagon = BeautifulSoup(page_bookswagon.text, 'html.parser')

book_classes_bookswagon = 'list-view-books'

books_set_lis_bookswagon = soup_bookswagon.find_all('div', {'class':
book_classes_bookswagon})

int_total_no_of_books = int(len(books_set_lis_bookswagon))
print("book swagon", int_total_no_of_books)

def func2(i):
    price_class = 'sell'
    price = books_set_lis_bookswagon[i].find('div', {'class': price_class})
    # rating_class='_3LWZLK'
    # rating=books_set_lisflp[i].find('div',{'class': rating_class})

    bookswagon_name_link_class = 'title'

    linkk = books_set_lis_bookswagon[i].find('div', {'class':
bookswagon_name_link_class})

    # book_title_class=''
    book_title = linkk.find('a')

    img_src = books_set_lis_bookswagon[i].find('img')

    # print("title : " , book_title['title'])
    # print("price: " ,price.text)
    # print("rating : ",rating.text)
    # print("link :", mn_url+linkk['href'] )
    price = price_text_to_stringoing(price.text)
    ls.append([price,"no rating available", book_title.text, book_title['href'],
"bookswagon", img_src['src']])

# sapna
url_sapna_main = "https://www.sapnaonline.com"
url_sapna = "https://www.sapnaonline.com/search?keyword="

for i in range(0, n):
    if (stringo[i] == ' '):
        url_sapna += "%20"

```

```

        else:
            url_sapna += stringo[i]

print(url_sapna)

page_sapna = requests.get(url_sapna)

soup_sapna = BeautifulSoup(page_sapna.text, 'html.parser')

book_classes_sapna = 'sc-AxirZ CategoryTabInner__ProductBox-qaa80s-0 jZjvfA'

books_set_lis_sapna = soup_sapna.find_all('div', {'class': book_classes_sapna})

int_total_no_of_books_sapna = int(len(books_set_lis_sapna))
print("sapna", int_total_no_of_books_sapna)

def func3(i):
    price_class = 'ProductCard__PrcieText-sc-10n3822-7 hnbQgS'
    price = books_set_lis_sapna[i].find('h3', {'class': price_class})
    # rating_class='_3LWZlK'

    # rating=books_set_lisflp[i].find('div',{'class': rating_class})

    sapna_name_link_class = 'ProductCard__AboutText-sc-10n3822-2 k0Zyab link'

    book_title = books_set_lis_sapna[i].find('h2', {'class': sapna_name_link_class})

    # book_title_class=''
    # book_title=linkk.find('a')
    linkk = books_set_lis_sapna[i].find('a')
    img_class = 'bookImage'
    img_src = books_set_lis_sapna[i].find('img')
    # print("title : " , book_title['title'])
    # print("price: " ,price.text)
    # print("rating : ",rating.text)
    # print("link :", mn_url+linkk['href'] )

    price = price_text_to_stringoing(price.text)
    ls.append([price,"no rating available", book_title.text, url_sapna_main +
    linkk['href'], "sapna", img_src['src']])

val1 = min(flipsize, 10)
val2 = min(2, pstsize)
val3 = min(2, int_total_no_of_books)
val4 = min(2, int_total_no_of_books_sapna)

```

```

val5 = min(10, amazon_size)
# val5=min(5,int_total_no_of_books_snapdeal)

if ((val1 + val2 + val3 + val4+val5) == 0):
    print("SORRY! NO ITEM FOUND")
for i in range(0, val1):
    # print(len(ls))
    # print("serial no ",i)
    func1(i)
for i in range(0, val4):
    # print(len(ls))
    # print("serial no ",i)
    func3(i)

# print("pushtakosh")


# val3=0
for i in range(0, val3):
    func2(i)

for i in range(0, val5):
    func5(i)

    # for i in range(0,val5):
    #     func4(i)

ls.sort()


list_book_serial_no=[]
list_book_L_no=[]
list_book_price=[]
list_book_name=[]
list_book_link=[]
list_book_website_name=[]
list_book_rating=[]
c=0
val="L"
val=val+(str(c))
ls_prc=ls[0][0]
list_book_L_no.append(val)

```

```

for i in range(0, len(ls)):
    list_book_serial_no.append(i+1)
    list_book_price.append(ls[i][0])
    list_book_name.append(ls[i][2])
    list_book_link.append(ls[i][3])
    list_book_website_name.append(ls[i][4])

list_book_rating.append(ls[i][1])
if(i>=1):
    if(ls_prc!=ls[i][0]):
        c+=1
        ls_prc=ls[i][0]
    val="L"
    val=val+(str(c))
    list_book_L_no.append(val)

if stringo != '':
    cols = st.columns(6)
    cols[0].markdown("SERIAL NO")
    cols[1].markdown("PRICE")
    cols[2].markdown("NAME")
    cols[3].markdown("LINK")
    cols[4].markdown("RATINGS")
    cols[5].markdown("Website")
    for i in range(1, len(ls)):
        cols = st.columns(6)
        cols[0].write(str(list_book_serial_no[i] - 1))
        cols[1].write('₹' + str(list_book_price[i]))
        cols[2].write(list_book_name[i])
        link = '[' + list_book_L_no[i] + ']' + '(' + list_book_link[i] + ')'
        cols[3].write(link, unsafe_allow_html=True)
        cols[4].write(str(list_book_rating[i]))
        cols[5].write(list_book_website_name[i])

```