

**A
MINI PROJECT
ON
PRICE COMPARATOR**

**SUBMITTED IN PARTIAL FULFILMENT FOR THE
COMPLETION OF
BE-V SEMESTER**

**IN
INFORMATION TECHNOLOGY
BY**

**M YOGITHA NANDINI (160117737030)
P ARUN RAJ (160117737034)**

**UNDER THE GUIDANCE OF
MS.E.RAMALAKSHMI,
ASSISTANT PROFESSOR,
DEPT. OF IT, CBIT.**



**DEPARTMENT OF INFORMATION TECHNOLOGY
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)**

(Affiliated to Osmania University; Accredited by NBA(AICTE) and NAAC(UGC), ISO Certified 9001:2015)

GANDIPET, HYDERABAD – 500 075

Website: www.cbit.ac.in

2018-2019

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)

DEPARTMENT OF INFORMATION TECHNOLOGY

(Affiliated to Osmania University)

GANDIPET, HYDERABAD – 500 075



CERTIFICATE

This is to certify that the project work entitled “**PRICE COMPARATOR**” submitted to **CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY**, in partial fulfilment of the requirements for the award of the completion of V semester of B.E in Information Technology, during the academic year 2018-2019, is a record of original work done by, **M.YogithaNandini (160117737030)**, **P.Arun Raj (160117737034)** during the period of study in Department of IT, CBIT, HYDERABAD, under our supervision and guidance.

Project Guide

Ms .E.RAMA LAKSHMI

Asst. Professor, Dept. of IT,
CBIT, Hyderabad.

Head of the Department

Dr. SURESH PABBOJU

Professor, Dept. of IT,
CBIT, Hyderabad.

ACKNOWLEDGMENTS

We would like to express our heartfelt gratitude to **Ms.E.Rama Lakshmi**, our project guide, for her invaluable guidance and constant support, along with her capable instruction and persistent encouragement.

We are grateful to our Head of Department, **Dr. SureshPabboju**, for his steady support and the provision of every resource required for the completion of this project.

We would like to take this opportunity to thank our Principal, **Dr.P.Ravinder Reddy**, as well as the management of the institute, for having designed an excellent learning atmosphere.

Our thanks are due to all members of the staff and our lab assistants for providing us with the help required to carry out the groundwork of this project.

DECLARATION

I hereby declare that the mini project which we have done was under the supervision of the faculty of our college.

No part of our project has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.

Date:

Name: P. Arun Raj

Signature:

Rollno: 160117737034

Name: M. Yogitha Nandini

Signature:

Rollno: 160117737030

ABSTRACT

Price Comparison project is used to compare the prices of a product from various online shopping websites to help the people to find the affordable price to buy the product and to analyse the fluctuations in prices.

Price Comparison manages to access the HTML webpage and extract useful information or data like price of product by various techniques using various modules. This project can be useful to 50-70% of consumers who tends to know the price of the product online before purchase.

This project is easy to access. This project helps users to check and compare price of same product. When the user enters the category product name and clicks on the button, it displays the product price of various online shopping websites. So, the user can know the right product at the right price.

It increases the speed to choose the product. It saves lot of time. Choosing minimum cost for same product. You can set your site integrated with multiple countries and its affiliating vendors.

LIST OF FIGURES

Figure	Description	Page no.
Fig 3.2	Flow chart of the price comparator	6
Fig 5.1	Sample code of the project	11
Fig 5.1.1	Displaying the prices for Samsung Mobiles	17
Fig 5.1.2	Displaying the prices for Sony TVs	18

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

DECLARATION

ABSTRACT

LIST OF FIGURES

1. INTRODUCTION	1
1.1 MOTIVATION	1
1.2 PROBLEM STATEMENT	1
2. EXISTING SYSTEM	2
3. PROPOSED SYSTEM	3
3.1 METHODOLOGY	3
3.2 ARCHITECTURE OF PROPOSED SYSTEM	6
4. SOFTWARE AND HARDWARE REQUIREMENTS	7
4.1 PYTHON	7
5. IMPLEMENTATION OF PROJECT	11
5.1 RESULTS	17
6. CONCLUSION AND FUTURE SCOPE	19
BIBLIOGRAPHY	20

1. INTRODUCTION

1.1 MOTIVATION

Over the past two decades a new industry of price comparison websites (PCWs) has emerged. The industry has enabled consumers to check the prices of many firms selling a particular service or product simultaneously in one place. This promises to be particularly helpful to consumers in a world where prices of even seemingly homogeneous items are typically dispersed

The purpose of this project is to retrieve the data of some websites to make our work easy in our routine life activities like shopping.

This is a simple and easy to access that helps to know product details in short period of time. This is used to ignore usage of many websites for shopping.

It increases the speed to choose the product. It saves lot of time. Choosing minimum cost for same product. You can set your site integrated with multiple countries and its affiliating vendors.

1.2 PROBLEM STATEMENT

Price Comparator is used to know the details of the products like product name, product price according to the given category and name. This project can be used by any individual.

When the category and key word of the product are given the product name and price of the given websites are retrieved through this project. This is easy way to know the price of product than spending time on various websites.

Price Comparison project is used to compare the prices of a product from various online shopping websites to help the people to find the affordable price to buy the product and to analyse the fluctuations in prices.

2. EXISTING SYSTEM

Google Shopping

Google Shopping, formerly Google Product Search, Google Products and Froogle, is a Google service invented by Craig Nevill-Manning which allows users to search for products on online shopping websites and compare prices between different vendors. Google announced at its Marketing Live event in May 2019 that the new Google Shopping will integrate the existing Google Express marketplace into a revamped shopping experience. In the U.S, Google Shopping is accessible from the web and mobile apps, available on Android and iOS. Google Shopping is also available in France, accessible from the web only. Like its predecessor, Google Shopping is free and requires a personal Google account in order to purchase from the platform. A colored price tag icon replaces the parachute icon from Google Express.

Nextag

Nextag was an independent price comparison service website for products, travel, and education. It started originally as a website where buyers and sellers could negotiate prices for computers and electronics products. Since 2000, the current business model has focused on comparison shopping. NexTag also owns Hamburg, Germany based Guentstiger.de. It provides unique and accurate functionality of tracking historical prices of a product across various sellers.

3. PROPOSED SYSTEM

3.1 METHODOLOGY

urllib.request is a python module for fetching URLs (Uniform Resource Locators). It offers a very simple interface, in the form of the `urlopen` function. This is capable of fetching URLs using a variety of different protocols. It also offers a slightly more complex interface for handling common situations - like basic authentication, cookies, proxies and so on. These are provided by objects called handlers and openers.

`urllib.request` supports fetching URLs for many “URL schemes” using their associated network protocols (e.g. FTP, HTTP). This tutorial focuses on the most common case, HTTP.

For straightforward situations `urlopen` is very easy to use. But as soon as you encounter errors or non-trivial cases when opening HTTP URLs, you will need some understanding of the HyperText Transfer Protocol. The most comprehensive and authoritative reference to HTTP is RFC 2616. This is a technical document and not intended to be easy to read. This HOWTO aims to illustrate using *urllib*, with enough detail about HTTP to help you through. It is not intended to replace the docs, but is supplementary to `urllib.request` them.

HTTP is based on requests and responses - the client makes requests and servers send responses. `urllib.request` mirrors this with a request object which represents the HTTP request you are making. In its simplest form you create a Request object that specifies the URL you want to fetch. Calling `urlopen` with this Request object returns a response object for the URL requested.

The `urllib` module in Python 3 allows you access websites via your program. This opens up as many doors for your programs as the internet opens up for you. `urllib` in Python 3 is slightly different than `urllib2` in Python 2, but they are mostly the same. Through `urllib`, you can access websites, download data, parse data, modify your headers, and do any GET and POST requests you might need to do.

Some websites do not appreciate programs accessing their data and placing weight on their servers. When they find out that a program is visiting them, they may sometimes choose to block you out, or serve you different data that a regular user might see. This can be annoying at first, but can be overcome with some simple code. To do this, you just need to modify the user-agent, which is a variable within your header that you send in. Headers are bits of data that you share with servers to let them know a bit about you. This is where Python, by default, tells the website that you are visiting with Python's urllib and your Python version. We can, however, modify this, and act as if we are a lowly Internet Explorer user, a Chrome user, or anything else really!.

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favourite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work with website data.

Web pages are structured documents, and Beautiful Soup gives you the tools to walk through that complex structure and extract bits of that information. In this guide, you will write a Python script that will scrape Craigslist for motorcycle prices. The script will be set up to run at regular intervals using a cron job, and the resulting data will be exported to an Excel spreadsheet for trend analysis. You can easily adapt these steps to other websites or search queries by substituting different URLs and adjusting the script accordingly.

The `make_soup` function makes a GET request to the target url and converts the resulting HTML into a BeautifulSoup object. The urllib3 library has excellent exception handling; if `make_soup` throws any errors, check the urllib3 docs for detailed information.

Beautiful Soup has different parsers available which are more or less strict about how the webpage is structured. The lxml parser is sufficient for the example script in this guide, but depending on your needs you may need to check the other options described in the official documentation.

An object of class BeautifulSoup is organized in a tree structure. In order to access the data you are interested in, you will have to be familiar with how the data is organized in the original HTML document. Go to the initial website in a browser, right click and select View page source (or Inspect, depending on your browser) to review the structure of the data that you would like to scrape

3.2 ARCHITECTURE OF PROPOSED SYSTEM

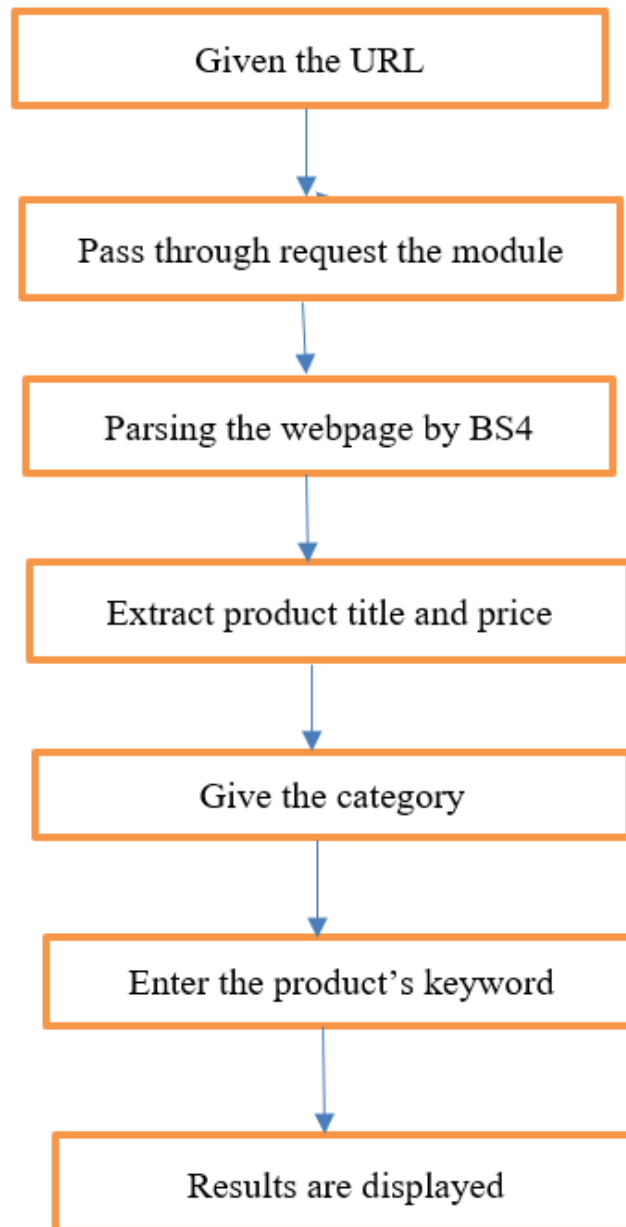


Fig.3.2 Flow chart of the project

WORKFLOW

At first the url is passed through the module to extract the items of the page. Then it parses the webpages of the given link using BS4. It only extracts the given number of pages and all these paths are contained in a python function. After this the module the price of the product and name from the webpage. After the successful compilation the category need to be entered and at last the keyword name of the product need to be entered and the result is displayed in the python idle.

4. SOFTWARE AND HARDWARE REQUIREMENTS

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective.

The purpose of software requirements specification is to provide a detail overview of the software project, its and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

4.1 PYTHON

Introduction

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released 2008,

was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3. Due to concern about the amount of code written for Python 2, support for Python 2.7 was extended to 2020. Language developer Guido van Rossum shouldered sole responsibility for the project until July 2018 but now shares his leadership as a member of a five-person steering council.

Python is a fully-functional programming language that can do *anything* almost any other language can do, at comparable speeds.

Python is capable of threading and GPU processing just like any other language. Most of the data processing modules are actually just Python wrappers around C/C++ code.

"Modules" are pre-written Python code that you "import" in your Python program. Since there are many tasks that people commonly do, we have modules that people have written that do these tasks for you, and they usually do them in the cleanest and most efficient method possible. Sometimes you will see people refer to "DRY." This stands for Don't Repeat Yourself, which often also translates into "Don't Repeat Someone Else."

The phrase "wrapper" means that someone has placed, like a wrapper, Python code over another language. So, when you have a Python wrapper around C++ code, what someone has done is written some Python code that interacts with the C++ language. This allows you to make use of various aspects of the language being wrapped, in this case C++, without actually needing to know or understand that language.

Thus, Python can be used to make games, do data analysis, control robot and hardware, create GUIs, or even to create websites.

"GUI" stands for Graphical User Interface, and is used to describe a program that incorporates graphics to make the program more interactive for the user.

Features

Python provides lots of features that are listed below.

1) Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

5) Free and Open Source

Python language is freely available at official web address. The source-code is also available. Therefore it is open source.

6) Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

8) Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

9) GUI Programming Support

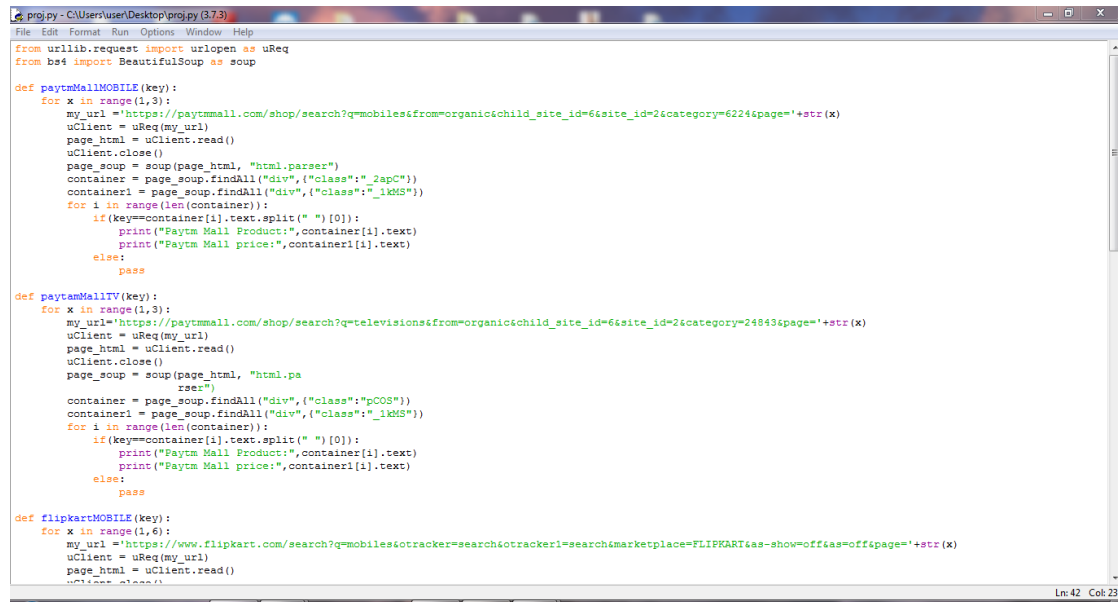
Graphical user interfaces can be developed using Python.

10) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

5. IMPLEMENTATION OF PROJECT

First of all you have to open the project in python IDLE. Then we need to run the program.



```
def paytmMallMOBILE(key):
    for x in range(1,3):
        my_url = 'https://paytmall.com/shop/search?q=mobiles&from=organic&child_site_id=66&site_id=2&category=62246&page='+str(x)
        uClient = uReq(my_url)
        page_html = uClient.read()
        uClient.close()
        page_soup = soup(page_html, "html.parser")
        container = page_soup.findAll("div", {"class": "_2apC"})
        container1 = page_soup.findAll("div", {"class": "_1kMS"})
        for i in range(len(container)):
            if (key==container[i].text.split(" ")[0]):
                print("Paytm Mall Product:",container[i].text)
                print("Paytm Mall price:",container1[i].text)
            else:
                pass

def paytmMallTV(key):
    for x in range(1,3):
        my_url = 'https://paytmall.com/shop/search?q=televisions&from=organic&child_site_id=66&site_id=2&category=24843&page='+str(x)
        uClient = uReq(my_url)
        page_html = uClient.read()
        uClient.close()
        page_soup = soup(page_html, "html.parser")
        container = page_soup.findAll("div", {"class": "pCOS"})
        container1 = page_soup.findAll("div", {"class": "_1kMS"})
        for i in range(len(container)):
            if (key==container[i].text.split(" ")[0]):
                print("Paytm Mall Product:",container[i].text)
                print("Paytm Mall price:",container1[i].text)
            else:
                pass

def flipkartMOBILE(key):
    for x in range(1,6):
        my_url = 'https://www.flipkart.com/search?q=mobiles&otracker=search&otracker1=search&marketplace=FLIPKART&as-show=off&as=off&page='+str(x)
        uClient = uReq(my_url)
        page_html = uClient.read()
        uClient.close()
```

Fig-5.1 Project sample code

The packages used are

1. urllib.request
2. bs4

The modules used are

1. urlopen
2. BeautifulSoup

The project is based on the concept of web scraping. Web scraping is an automated method used to extract large amounts of data from websites. The data on the websites are unstructured. Web scraping helps collect these unstructured data and store it in a structured form. There are different ways to scrape websites such as online Services, APIs or writing your own code.

When you run the code for web scraping, a request is sent to the URL that you have mentioned. As a response to the request, the server sends the data and allows you to read the HTML or XML page. The code then, parses the HTML or XML page, finds the data and extracts it.

To extract data using web scraping with python, you need to follow these basic steps:

1. Find the URL that you want to scrape
2. Inspecting the Page
3. Find the data you want to extract
4. Write the code
5. Run the code and extract the data
6. Store the data in the required format

Libraries used for Web Scraping :

Selenium is a web testing library. It is used to automate browser activities.

Beautiful Soup is a Python package for parsing HTML and XML documents. It creates parse trees that is helpful to extract the data easily.

Pandas is a library used for data manipulation and analysis. It is used to extract the data and store it in the desired format.

Now let us discuss the tasks done by these packages and modules. The websites used to retrieve the data are flipkart, paytam mall, ebay. The functions define in the code are flipkartMOBILE, paytmMallMOBILE, eBayMOBILE, flipkartTV, paytamMallTV, eBayTV which consists of the specified uniform resource locator of the websites.

Description of the function flipkartMOBILE:

The flipkart mobiles uniform resource locator is passed as a string through urlopen, which actually open up the client connection and grab the flipkart web page.

It's going to dump everything out by reading the grabbed flipkart web page. As this is an open internet connection the client is closed after grabbing flipkart web page.

Parse the html by using BeautifulSoup module because right now the html is a big jumble of text. Call the soup function which was made earlier. And store it into a variable or else it's going to get lost.

Now flipkart inspect the web page to know the tag name and class name of product title to extract the product name, using findAll function which retrieves the product name of all the products in the flipkart web page and store it in a variable container.

Inspect the flipkart web page to know the tag name and class name of product price to extract the product price, using findAll function which retrieves the product price of all the products in the flipkart web page and store it in a variable container1 .

Description of the function flipkartTV:

The flipkart tv uniform resource locator is passed as a string through urlopen, which actually open up the client connection and grab the flipkart web page. It's going to dump everything out by reading the grabbed flipkart web page. As this is an open internet connection the client is closed after grabbing flipkart web page.

Parse the html by using BeautifulSoup module because right now the html is a big jumble of text. Call the soup function which was made earlier. And store it into a variable or else it's going to get lost.

Now flipkart inspect the web page to know the tag name and class name of product title to extract the product name, using findAll() function which retrieves the product name of all the products in the flipkart web page and store it in a variable container.

Inspect the flipkart web page to know the tag name and class name of product price to extract the product price, using findAll function which retrieves the product price of all the products in the flipkart web page and store it in a variable container1.

Description of the function paytmMallMOBILE:

The paytm mall mobiles uniform resource locator is passed as a string through urlopen, which actually open up the client connection and grab the paytm mall web page. It's going to dump everything out by reading the grabbed paytm mall web page. As this is an open internet connection the client is closed after grabbing paytm mall web page.

Parse the html by using BeautifulSoup module because right now the html is a big jumble of text. Call the soup function which was made earlier. And store it into a variable or else it's going to get lost.

Now inspect the paytm mall web page to know the tag name and class name of product title to extract the product name, using findAll function which retrieves the product name of all the products in the paytm mall web page and store it in a variable container.

Inspect the paytm mall web page to know the tag name and class name of product price to extract the product price, using findAll function which retrieves the product price of all the products in the paytm mall web page and store it in a variable container1 .

Description of the function paytmMallTV:

The paytm mall tv uniform resource locator is passed as a string through urlopen, which actually open up the client connection and grab the paytm mall web page. It's going to dump everything out by reading the grabbed paytm mall web page. As this is an open internet connection the client is closed after grabbing paytm mall web page.

Parse the html by using BeautifulSoup module because right now the html is a big jumble of text. Call the soup function which was made earlier. And store it into a variable or else it's going to get lost.

Now inspect the paytm mall web page to know the tag name and class name of product title to extract the product name, using findAll function which retrieves the

product name of all the products in the paytm mall web page and store it in a variable container.

Inspect the paytm mall web page to know the tag name and class name of the product price to extract the product price, using findAll function which retrieves the product price of all the products in the paytm mall web page and store it in a variable container1.

Description of the function ebayMOBILE:

The ebay mobiles uniform resource locator is passed as a string through urlopen, which actually open up the client connection and grab the ebay web page. It's going to dump everything out by reading the grabbed ebay web page. As this is an open internet connection the client is closed after grabbing ebay web page.

Parse the html by using BeautifulSoup module because right now the html is a big jumble of text. Call the soup function which was made earlier. And store it into a variable or else it's going to get lost.

Now inspect the ebay web page to know the tag name and class name of product title to extract the product name, using findAll function which retrieves the product name of all the products in the ebay web page and store it in a variable container.

Inspect the ebay web page to know the tag name and class name of the product price to extract the product price, using findAll function which retrives the product price of all the products in the ebay web page and store it in a variable container1 .

Description of the function ebayTV:

The ebay tv uniform resource locator is passed as a string through urlopen, which actually open up the client connection and grab the ebay web page. It's going to dump everything out by reading the grabbed ebay web page. As this is an open internet connection the client is closed after grabbing ebay web page.

Parse the html by using BeautifulSoup module because right now the html is a big

jumble of text. Call the soup function which was made earlier. And store it into a variable or else it's going to get lost.

Now inspect the ebay web page to know the tag name and class name of product title to extract the product name, using findAll function which retrieves the product name of all the products in the ebay web page and store it in a variable container.

Inspect the ebay web page to know the tag name and class name of product price to extract the product price, using findAll function which retrieves the product price of all the products in the ebay web page and store it in a variable container1 .

When the category is entered the functions are called automatically and enter the key word of the product name, which is passed as an argument to the functions, which is compared with the first word of the title and prints the product name and product price.

5.1 RESULTS

From the Fig-5.1 we get to know that when category is Mobiles and Product name is Samsung the output is resulted as in the output screen.


```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

Enter Category:Mobiles
Enter Product Name:Samsung
Flipkart Product: Samsung Galaxy S9 Plus (Coral Blue, 64 GB)
Flipkart price: 57,900
Flipkart Product: Samsung Galaxy S9 Plus (Burgundy Red, 64 GB)
Flipkart price: 57,900
Paytm Mall Product: Samsung Galaxy A2 Core 1 GB 16 GB Dark Grey
Paytm Mall price: 5,290
Paytm Mall Product: Samsung Galaxy A20 3 GB 32 GB (Black)
Paytm Mall price: 11,490
Paytm Mall Product: Samsung Galaxy A20 3 GB 32 GB (Blue )
Paytm Mall price: 11,490
EBay Product: Samsung Galaxy Note5 SM-N920 - 32GB - Gold Platinum (T-Mobile) Smartphone
EBay price: INR 17,767.04
EBay Product: Samsung Galaxy S8+ SM-G955U - 64GB - Arctic silver (T-Mobile) Smartphone
EBay price: INR 8,350.51
EBay Product: Samsung Galaxy S1 - 64GB - Midnight Black (Boost Mobile) Smartphone BUNDLE!
EBay price: INR 15,990.33
EBay Product: Samsung Galaxy S8 G950U 64GB - Factory Unlocked (Verizon, AT&T T-Mobile) Black..
EBay price: INR 8,527.47
EBay Product: Samsung Galaxy J7 Refine 32gb Boost Mobile
EBay price: INR 10,677.99 to INR 11,061.76
EBay Product: Samsung Nexus S SPH-D720 16GB T-Mobile Black Android Smartphone Cellphone
EBay price: INR 3,553.41
EBay Product: Samsung Galaxy A20 (Metro PCS / Metro by T-Mobile) Metro PCS ONLY
EBay price: INR 10,208.94
EBay Product: Samsung Galaxy S6 Active G890A 32/64GB AT&T T-Mobile GSM Unlocked Smartphone
EBay price: INR 4,527.75 to INR 5,433.87
EBay Product: Samsung Galaxy S4 - T-Mobile
EBay price: INR 23,878.19
EBay Product: Samsung Nexus S SPH-D720 16GB T-Mobile Black Android Smartphone Cellphone
EBay price: INR 8,386.04
EBay Product: Samsung Galaxy S6 Active G890A 32GB AT&T T-Mobile Unlocked Smartphone WHITE M575
EBay price: INR 3,552.70
EBay Product: Samsung Galaxy Note 8 SM-N950U 64GB Midnight Black Unlocked for GSM ATT T-Mobile
EBay price: INR 2,481.70 to INR 3,237.87
EBay Product: Samsung Galaxy S9+ Plus SM-G965U1, Boost Mobile (Global Unlocked)
EBay price: INR 2,770.95
EBay Product: Samsung Galaxy S10+ Plus, BOOST MOBILE!!! (Global Version)
EBay price: INR 28,426.55
```

Fig 5.1.1 Displaying the prices for samsung mobiles

From the Fig-5.2 we get to know that when hen category id TV and Product name is Sony the output is resulted as in the output screen.

```

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
type help, copyright, credits or license() for more information.
>>>
===== RESTART: C:\Users\user\Desktop\proj.py =====
Enter Category:TV
Enter Product Name:Sony
Flipkart Product: Sony Bravia W622G 80cm (32 inch) HD Ready LED Smart TV
Flipkart price: 25,999
Flipkart Product: Sony Bravia X7500F 138.8cm (55 inch) Ultra HD (4K) LED Smart Android TV
Flipkart price: 79,999
Flipkart Product: Sony Bravia W622F 80cm (32 inch) HD Ready LED Smart TV
Flipkart price: 21,999
Flipkart Product: Sony Bravia W622G 80cm (32 inch) HD Ready LED Smart TV
Flipkart price: 25,999
Paytm Mall Product: Sony Smart 80 cm (32 inch) Full HD LED TV - K ...30,25036,900-18%Buy for ₹27,830with Cashback and Discount
Paytm Mall price: 30,250
Paytm Mall Product: Sony 109.22 cm (43 inch) KD-43X8000G 4K (Ultr ...56,99084,900-33%Buy for ₹52,431with Cashback and Discount
Paytm Mall price: 56,990
Paytm Mall Product: Sony Smart 108 cm (43 inch) Full HD LED TV - ...40,39951,900-22%Buy for ₹37,167with Cashback and Discount
Paytm Mall price: 40,399
Paytm Mall Product: Sony 139.7 cm (55 inch) KD-55X7002G 4K (Ultra ...70,9801,09,900-35%Buy for ₹65,302with Cashback and Discount
Paytm Mall price: 70,980
Paytm Mall Product: Sony 109.22 cm (43 inch) KD-43X7002G 4K (Ultr ...50,75072,900-30%Buy for ₹46,690with Cashback and Discount
Paytm Mall price: 50,750
Paytm Mall Product: Sony 124.46 cm (49 inch)KD-49X8000G 4K (Ultra ...74,98494,900-21%Buy for ₹68,985with Cashback and Discount
Paytm Mall price: 74,984
Paytm Mall Product: Sony Smart 139.7 cm (55 inch) 4K (Ultra HD) L ...83,0001,24,900-34%Buy for ₹76,360with Cashback and Discount
Paytm Mall price: 83,000
EBay Product: Sony Bravia 32-Inch LCD TV
EBay price: INR 7,114.91
EBay Product: Sony ICF-DVDS7TV Kitchen Under the Cabinet Television • 9" LCD TV/DVD/CD/Clock
EBay price: INR 14,227.68
EBay Product: Sony 32" Class LCD 720 p 60Hz HDTV, KDL-32BX330
EBay price: INR 7,114.91
EBay Product: Sony WEGA 15" Model KLV-S15G10. Beautiful condition. Only lightly used.
EBay price: INR 4,624.69
EBay Product: Sony DVP-SR210P Compact Disc DVD Player-Dolby- Progressive Scan w Remote
EBay price: INR 1,565.28
EBay Product: Sony Location Free Wireless Base Station LF-B10 SS-259 simple Set Up
EBay price: INR 3,130.56
EBay Product: Sony BRAVIA A8G 65" 2160p (4K) UHD OLED Android TV
EBay price: INR 249,021.70
>>>
Ln: 43 Col: 4

```

Fig 5.1.2 Displaying the prices for sony TVs

6. CONCLUSION&FUTURE SCOPE

We try to spend as much time as we can, to check whether or not we are buying a right thing? And the most important; whether we are buying it at a reasonable rate or not? Providing the ability to compare products is at the heart of what comparison sites are trying to achieve. However, those sites are achieving this with varying levels of success. My target of website is simply try my project, learn about the products, search and browse the options and shop with satisfaction of trust and all these at your door steps.

Price Comparison project is used to compare the prices of a product from various online shopping websites to help the people to find the affordable price to buy the product and to analyse the fluctuations in prices.

We have to extend our project by comparing for more products such as electronic products, fashion products and also by providing customer reviews .

We will try to design an android application using this.

BIBLIOGRAPHY

<https://www.google.co.in/>

<https://www.geeksforgeeks.org/implementing-web-scraping-python-beautiful-soup/>

<https://www.youtube.com/watch?v=OknMZUnTyds&list=PLgCYzUzKIBEvInwQhGSdnbyJ62nixHCt>

<https://www.youtube.com/watch?v=Ot8DGZ8qfY&list=PLzV8uWUcseN91J9bLuVLnlmWIE6wIWTiY>

<https://stackoverflow.com/>

<https://www.quora.com/>

<https://github.com/>