PROJECT REPORT

ON

WEB SCRAPING WITH GUI

Bachelor of Technology

In

Computer Science & Engineering

Of

Indus Institute of Technology and Engineering,

Indus University, Ahmedabad

Submitted to:
Zalak Trivedi
Assistant Professor

Department of CE
IITE, Indus University

Submitted by:
Deven Patel [IU1841050031]
Het Patel [IU1841050034]

Department of CE
IITE, Indus University

CANDIDATE'S DECLARATION

I declare that the final semester report entitled "**Web Scraping with GUI**" is my own work conducted under the supervision of the guide **Assistant Prof. Zalak Trivedi.**

I further declare that to the best of my knowledge, the report for B.Tech final semester does not contain part of the work which has been submitted for the award of B.Tech Degree either in this university or any other university without proper citation.

Candidate's Signature

DEVEN PATEL (IU1841050031)

HET PATEL (IU1841050034)

Guide: **ZALAK TRIVEDI**
Assistant Professor,
Department of Computer Engineering,
Indus Institute of Technology and Engineering
INDUS UNIVERSITY– Ahmedabad,
State: Gujarat

INDUS INSTITUTE OF TECHNOLOGY AND ENGINEERING

COMPUTER ENGINEERING

2021 -2022

CERTIFICATE

Date: 19-04-2022

This is to certify that the project work entitled "**Web Scraping with GUI**" has been carried out by **Deven Patel,** and **Het Patel** under my guidance in partial fulfillment of degree of Bachelor of Technology in **COMPUTER ENGINEERING (Final Year)** of Indus University, Ahmedabad during the academic year 2021 - 2022.

ZALAK TRIVEDI	Dr. SEEMA MAHAJAN
Assistant Professor,	Head of the Department,
Department of Computer Engineering,	Department of Computer Engineering,
I.T.E, Indus University	I.T.E, Indus University
Ahmedabad	Ahmedabad

ACKNOWLEDGEMENT

Towards the successful completion of my B. Tech in Computer Engineering final year project with the help of our mentor/guide Assistant Prof. Zalak Trivedi.

I am sincerely grateful to Assistant Prof. Zalak Trivedi who provided me this opportunity and necessary support during the development of the project.

I have no other words to express my sincere thanks to all faculties of Indus University, Ahmedabad for their kind cooperation and able guidance. I would like to thank Assistant Prof. Zalak Trivedi, my internal project guide in college without whom the project could not be executed.

INDEX

Abstract

Paise Bachao is a website where users can insert any keyword of a product of their choice and receive details of numerous products from two different e-commerce sites. The result will be in the form of a table in which the name and price of myriad products will be displayed according to the keyword entered by the user.

Paise Bachao is a simple yet streamlined website which makes the life of the users a lot easier by helping them in buying things online. It depicts only key features and price that in turn makes the decision making process more convenient.

To make this happen, web scraping is the best way for extracting data from different e–commerce websites. It assists to withdraw sufficient and precise information about products. Further, we have implemented this methodology very efficiently as Paise Bachao is able to depict the results collected from different e-commerce websites within a fraction of time.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

- **Project Introduction**
- **Objective of the Project**
- **Technology Used**
- **Advantages of the Project**
- **Basic Idea of our Core Project**

1.1 Project Introduction:-

Public often spends hours surfing the net and collecting necessary information manually, which is not the most exciting task, leading to errors because of the “human factor”: tiredness or boredom.

Web scraping (also called web data extraction or data scraping) provides a solution for those who want to get access to structured web data in an automated fashion.

By implementing web scraping automation, companies or individuals can extract data more efficiently and redistribute employees to more ROI-driven and crucial business tasks.

Building a system that collects the prices of a product from two different eCommerce websites and prepares a list of them. Extracting product data for comparison shopping on Amazon and flipkart (two leading ecommerce websites on Internet).

1.2 Objective of the Project:-

The main objective of the project is to reduce the burden on the customers’ shoulders by finding the best deal from the available deals across all ecommerce platforms by extracting product data for comparison from Amazon and Flipkart ecommerce websites.

In addition, depicting all the collected data in a very streamlined manner and creating an immersive GUI.

Further, it is time efficient as it only shows pivotal features which need to be known while making the decision whether to purchase it or not.

1.3 Technology Used:-

1. Python 3.9

Python 3.9 is one of the series of Python 3. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas the other languages use punctuations.

2. Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.

3. Visual Studio 2022

Visual Studio 2022 has built-in support for Git version control to clone, create, and open your own repositories. The Git tool window has everything you need for committing and pushing changes to code, managing branches, and resolving merge conflicts. If you have a GitHub account, you can manage those repos directly within Visual Studio.

4. Flask library

Flask is a web framework, it's a Python module that lets you develop web applications easily. It has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like url routing, template engine.

5. BeautifulSoup library

Beautiful Soup is a Python library that is used for web scraping purposes to pull the data out of HTML and XML files. It creates a parse tree from page source code that can be used to extract data in a hierarchical and more readable manner.

1.4 Advantages of the Project:-

The first and foremost advantage is that customers do not need to roam or surf across different ecommerce websites to find the best deal available for a certain product.

It only includes pivotal features of the product rather than exploring features in depth which makes the decision process of the customer convenient and fast.

It saves a lot of time for the customers as nowadays there are many products available under the same series of a company, it resolves this issue.

1.5 Basic Idea of our Core of the Project:-

Web Scraping:-

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. The web scraping software may directly access the World Wide Web using the Hypertext Transfer Protocol or a web browser.

It is a form of copying in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler.

Web scraping a web page involves fetching it and extracting from it. Fetching is the downloading of a page (which a browser does when a user views a page). Therefore, web crawling is a main component of web scraping, to fetch pages for later processing. Once fetched, then extraction can take place. The content of a page may be parsed, searched, reformatted, its data copied into a spreadsheet or loaded into a database. Web scrapers typically take something out of a page, to make use of it for another purpose somewhere

else. An example would be to find and copy names and telephone numbers, or companies and their URLs, or e-mail addresses to a list (contact scraping).

Web Crawler:-

A Web crawler, sometimes called a spider or spider-bot and often shortened to crawler, is an Internet bot that systematically browses the World Wide Web and that is typically operated by search engines for the purpose of Web indexing (web spidering).

A Web crawler starts with a list of URLs to visit. Those first URLs are called the seeds. As the crawler visits these URLs, by communicating with web servers that respond to those URLs, it identifies all the hyperlinks in the retrieved web pages and adds them to the list of URLs to visit, called the crawl frontier.

URLs from the frontier are recursively visited according to a set of policies. If the crawler is performing archiving of websites (or web archiving), it copies and saves the information as it goes. The archives are usually stored in such a way they can be viewed, read and navigated as if they were on the live web, but are preserved as 'snapshots'.

The archive is known as the repository and is designed to store and manage the collection of web pages. The repository only stores HTML pages and these pages are stored as distinct files. A repository is similar to any other system that stores data, like a modern-day database.

The only difference is that a repository does not need all the functionality offered by a database system. The repository stores the most recent version of the web page retrieved by the crawler.

Different purposes of Web Scraping & Web Crawling and why they are combined to give output :-

In web scraping, it's all about the data. The data fields you want to extract from specific websites. And it's a big difference because with scraping you usually know the target websites, you may not know the specific page URLs, but you know the domains at least.

With crawling, you probably don't know the specific URLs and you probably don't know the domains either. And this is the reason you crawl: you want to find the URLs. So that

you can do something with them later. For example, search engines crawl the web so they can index pages and display them in the search results.

But another data crawling example would be when you have one website that you want to extract data from - in this case you know the domain - but you don't have the page URLs of that specific website. So you don't know what pages to scrape. So first you create a crawler that will output all the page URLs that you care about - it can be pages in a specific category on the site or in specific parts of the website. Or maybe the URL needs to contain some kind of word for example and you collect all those URLs - and then you create a scraper that extracts predefined data fields from those pages.

So with web crawling the output is a lot more simple because it's just a list of URLs — I mean you can have other fields as well but the main elements are the URLs.

And with web scraping, you usually have a lot more fields 5-10-20 or more data fields. The URL can be one, but when you scrape, you extract the data not necessarily for the URL but for other data fields that are displayed on the website which can be - depending on the business use case - product name or product price, or some text or other information from any type of website.

All in all, web scraping is about extracting the data from one or more websites. While crawling is about finding or discovering URLs or links on the web. Usually, in web data extraction projects, you need to combine crawling and scraping.

CHAPTER 2

LITERATURE SURVEY

- **Introduction to Survey**
- **Why Survey?**

2.1 Introduction to Survey:-

A literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project. It is the most important part of your report as it gives you a direction in the area of your research. It helps you set a goal for your analysis - thus giving you your problem statement.

2.2 Why Survey?

➔ This Provides foundation of knowledge on topic

.

➔ To identify areas of prior academic work and prevent duplication and give credit to other researchers.

➔ To identify inconsistencies: gaps in research, conflicts in previous studies, open questions left from other research.

➔ To identify the need for additional research (justifying your research) and identify the relationship of words in context of its contribution to the topic and to other works. Place your own research within the context of existing literature making a case for why further study is needed.

CHAPTER 3

SYSTEM REQUIREMENTS

- **Software/Hardware Requirements**

3.1 Software and Hardware Requirements

Hardware and software requirements indicate the minimum required configuration to be present in the user devices to work effectively with the system. Devices that are not able to fulfill these requirements might not be able to work with the system and it may result in inaccessibility of few functionalities of the system on that particular user device.

3.1.1 Software Requirements

Operating System	Windows, Linux
Web Browser	Chrome, Mozilla Firefox, Microsoft Edge, Safari and Opera
Compatibility with Technologies	Python 3, Jupyter Notebook, Flask

3.1.2 Hardware Requirements

Processor	Minimum Intel i3 Processor
RAM	4 GB RAM
Graphics	Intel HD Graphics or higher
Internet Speed	Minimum 50 Mbps

CHAPTER 4

PROJECT MANAGEMENT AND DESIGN PHASE

- **Project Planning and Scheduling**
- **Project Development Approach**
- **Work-flow Diagram**
- **DFD Diagram**
- **Use Case Diagram**

4.1 Project Planning and Scheduling :-

Project Planning:-

The problem is decomposed into smaller problems, software managers use historical data (as well as personal experience and intuition) to determine estimates for each.

Project planning is a method for stating how to complete a project within a certain period of time, usually with defined stages, and with designated resources.

It begins by setting the scope of a project and eventually working through each level of dependent actions, tasks, checkpoints, and deadlines.

The final estimates are typically adjusted by taking project complexity and risk into account. The resulting work product is called a project management plan.

The objective of the software planning is providing the framework that enables the manager to make a reasonable estimation of the resources, cost, and schedule.

Project planning involves estimating how much resources will be required to build a specific software system. So that project outcome can be bounded.

Project Scheduling:-

Software project scheduling is an activity that distributes estimated efforts across the planned duration by allocating the effort to specific software engineering tasks.

Proper Scheduling Requires:-

- All tasks appear in the network.
- Effort and timing are intelligently allocated to each task.
- Interdependencies between tasks are properly indicated.
- Resources are allocated for the work to be done.

4.2 Project Development Approach:-

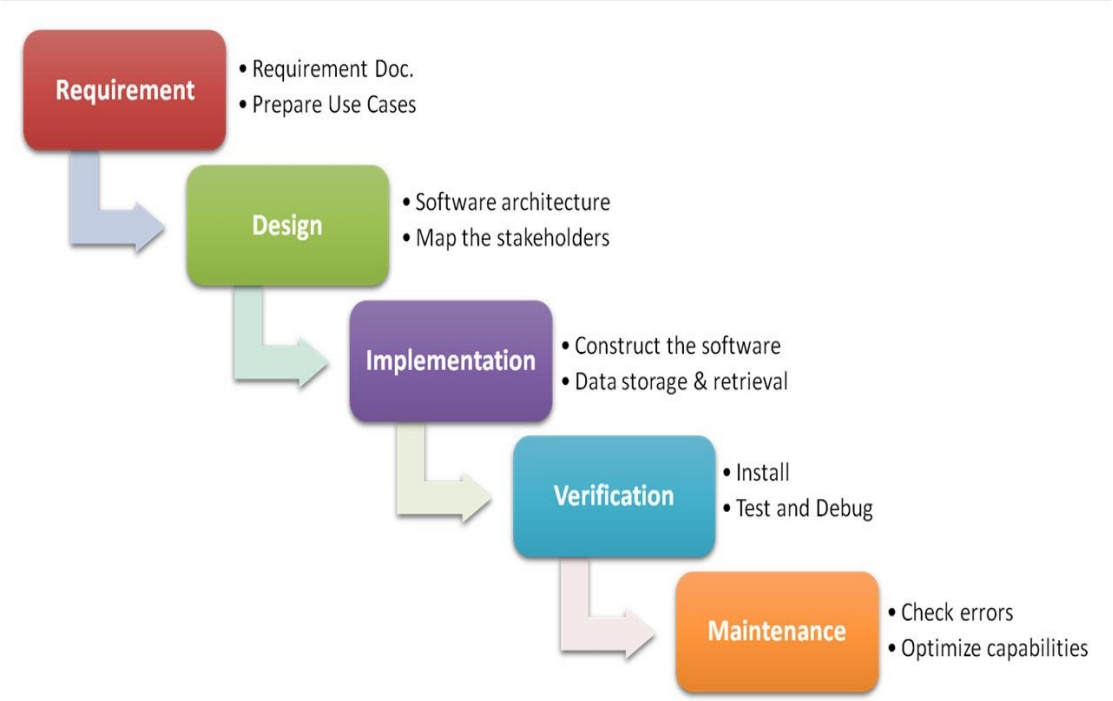


Fig. 4.2.1

Requirement Analysis

In this stage various requirements of all two, Admin, User. Admin expectseasy operation with the system, Faster and Efficient management of Product and User data, time-saving. Users require quick and on-timeDelivery of Products. Admin expects error free data of users that have registered for the Website.

Design

In this phase, various calculations were done regarding how the campus recruitment system will be made and how it will work or operate with particular hardware, software, and network infrastructure. In order to achieve the desired system, the design phase has been executed using two different parts: Logical Design in which role and functionalities of all users interacting with the system were identified and Database Design in which the structure of the database was designed to provide fast data accessibility.

Implementation

Based on the results of Requirement Analysis and Design phases, selection of appropriate technology was done and to achieve the planned system, actual coding was done in the Implementation Phase. In this phase, various software and installation files were downloaded to work with technological tools such as visual studio code. The task of designing the database and creating methods to store, update and retrieve the data from it was also done in this phase.

Testing

As the name of the phase suggests, the task of testing the fully functioning system was done in this phase. It included many steps such as accessing all its pages on multiple devices with different resolution or screen sizes to test its Responsiveness. It also includes testing all the possible ways of storing, updating and retrieving the data from the database.

Maintenance

In this stage, the well-functioning system is being looked after if it needs any changes in its design or database planning. Such changes are usually required at certain intervals of time to ensure that the system works properly without causing any errors or data losses.

Implementing Project Planning & Scheduling & Development Model:-

➔ According to the project planning and scheduling we have divided our whole project into four different phases. So, we could plan and execute the whole project meticulously. We developed our project in the waterfall model manner.

Phase 1:- Search & Extract Data **Deadline:- January**

This phase is all about searching and gathering information from the sources and doing literary surveys. This phase is also similar to the requirement and design phase of the waterfall model mentioned above because in this phase we also finalized the design of our project.

Phase 2:- Web scraper & crawler program **Deadline:- March(1st week)**

This phase is the pivotal phase of the whole project, as in this we code our main software programs according to which the whole system will react. This phase is comparable to the implementation step of the waterfall model.

Phase 3:- Designing GUI **Deadline:- March**

This is the second most important phase in our project. During it, we designed a minimal but immersive GUI with the help of which our user can interact with the backend part of our project. This is also covered in the implementation part.

Phase 4:- Testing and Improvising **Deadline:- April(1st week)**

This is the last phase of our project. It is similar to the verification and maintenance part of the model. In this phase, we tested our project with various possible inputs. Further, according to the advice of the mentor we also improvised it.

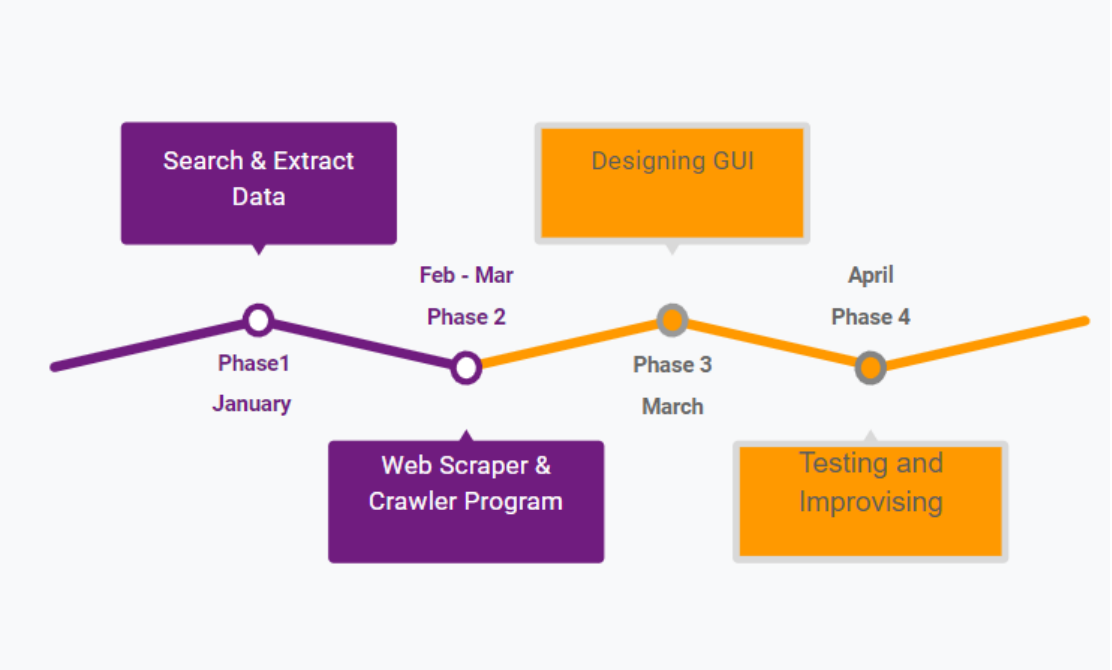


Fig. 4.2.2

4.3 Work-flow Diagram:-

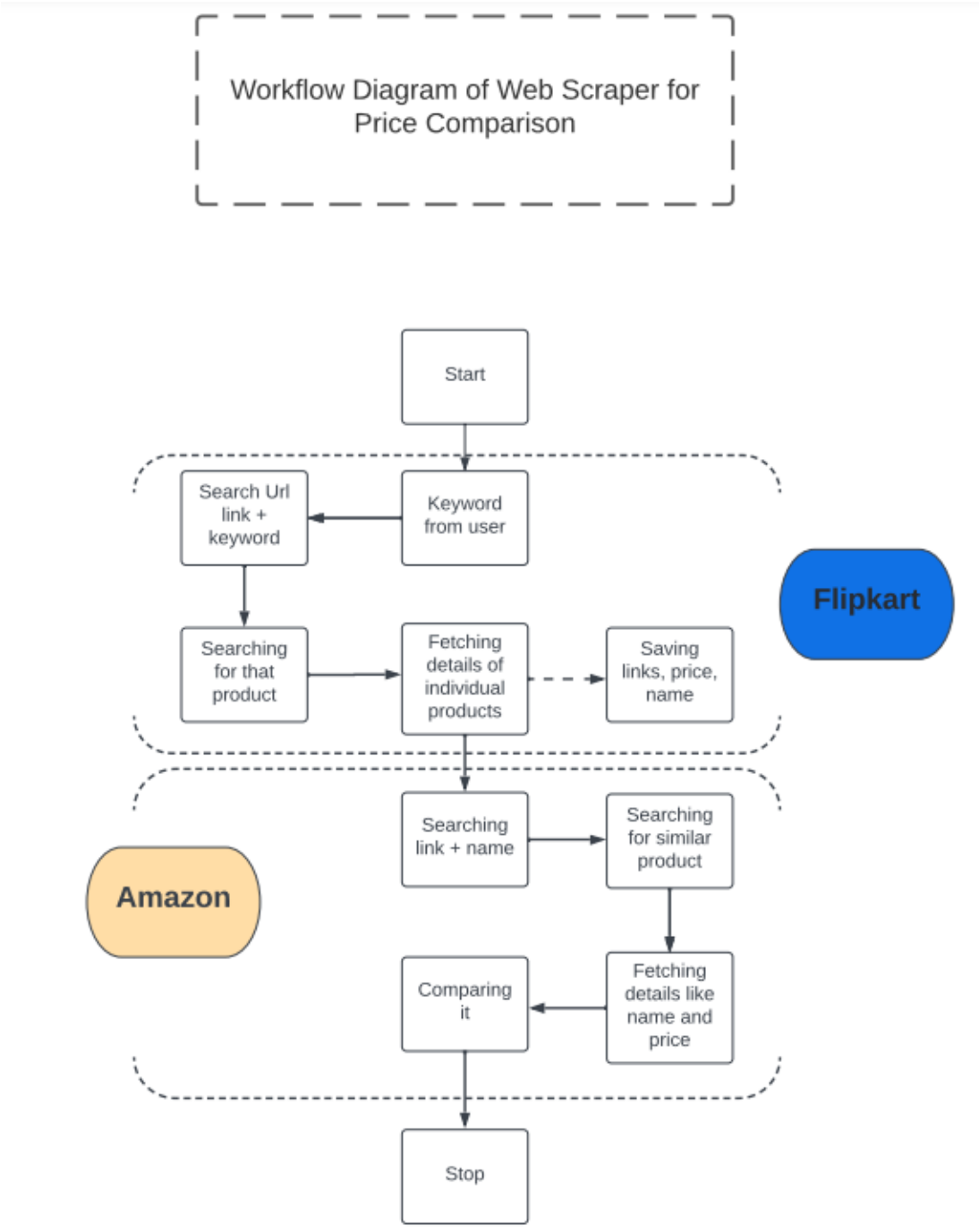


Fig. 4.3.1

4.4 DFD Diagram:-

Level 0 :

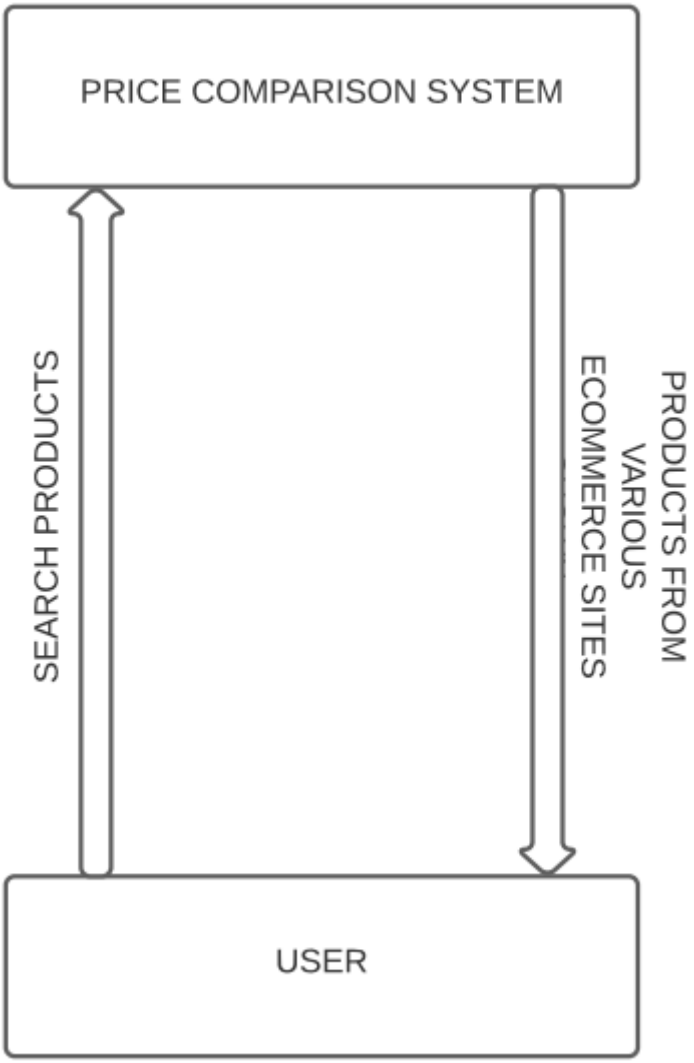


Fig. 4.4.1

Level 1:

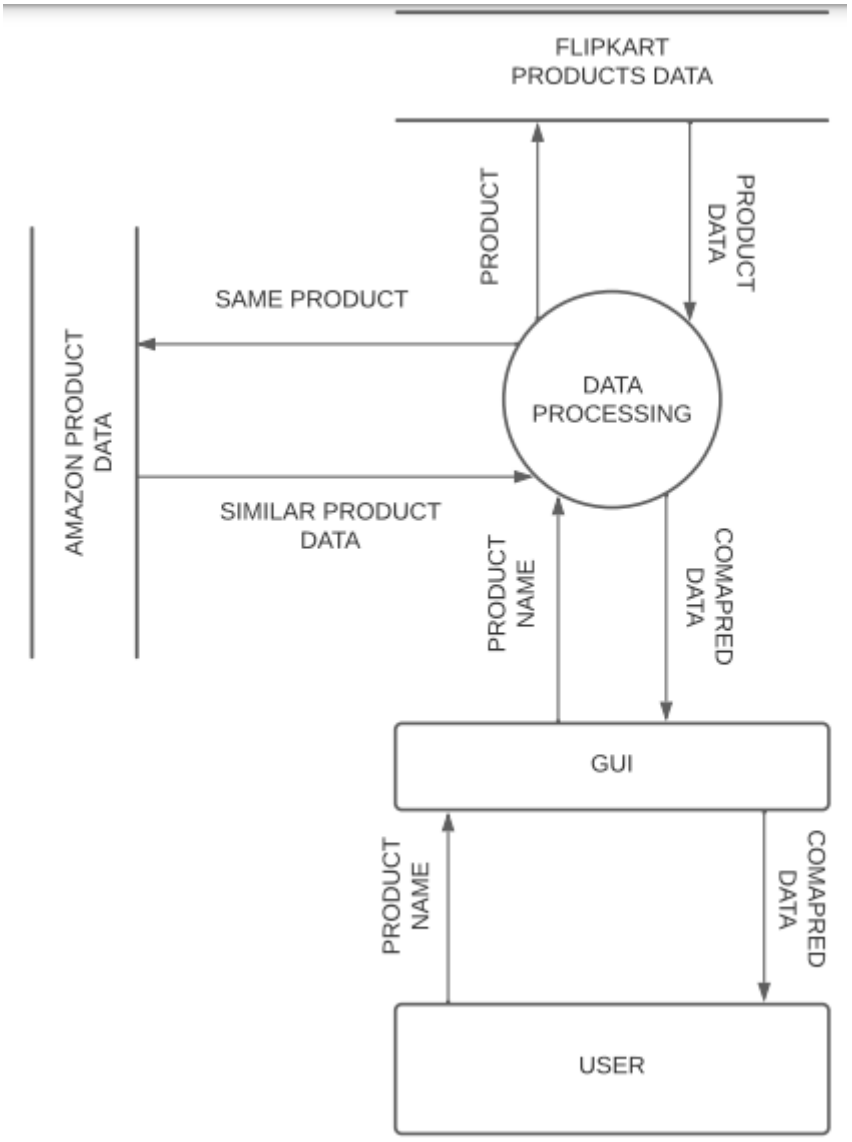


Fig. 4.4.2

Level 2:

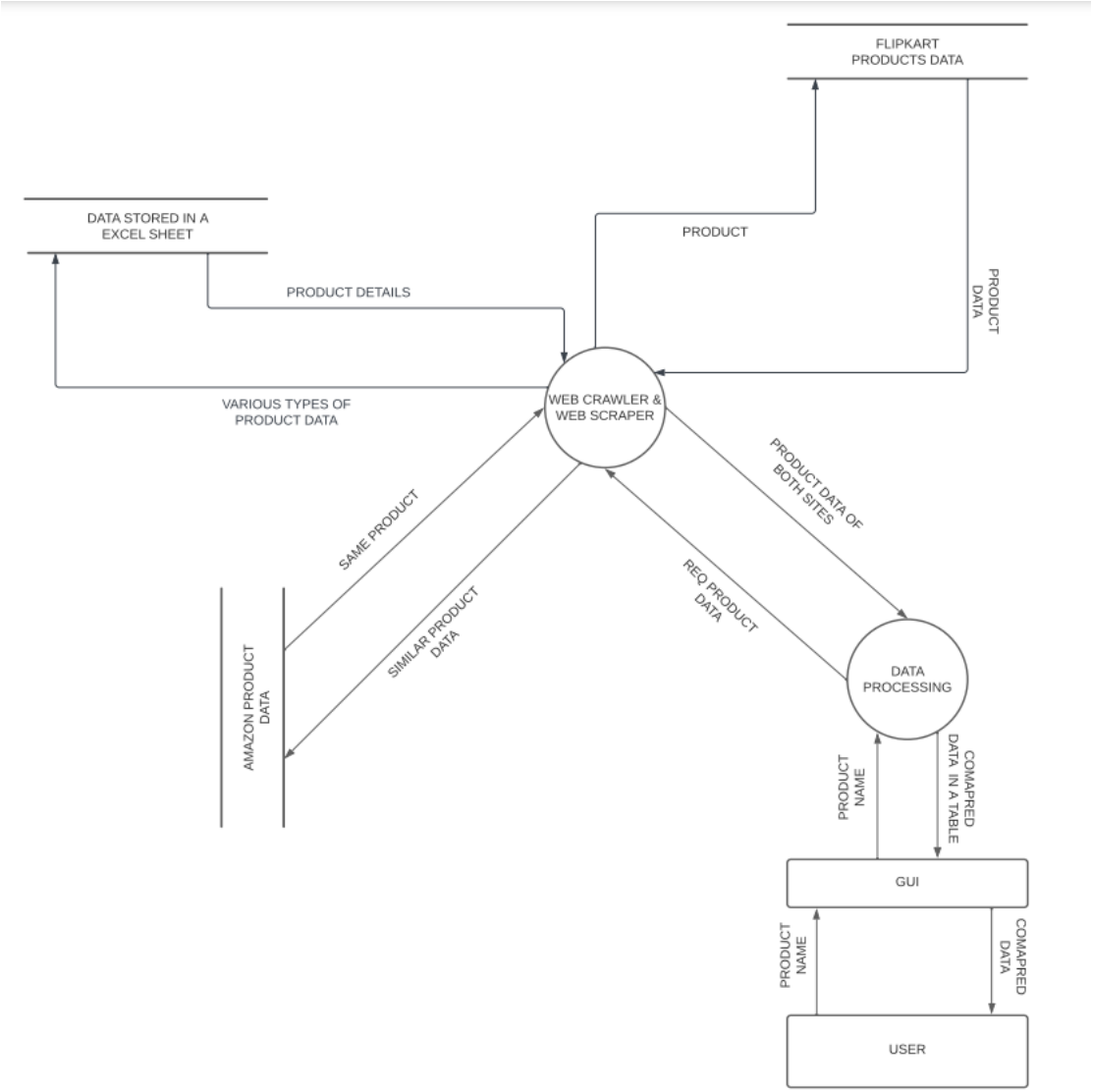


Fig. 4.4.3

4.5 Use Case Diagram:-

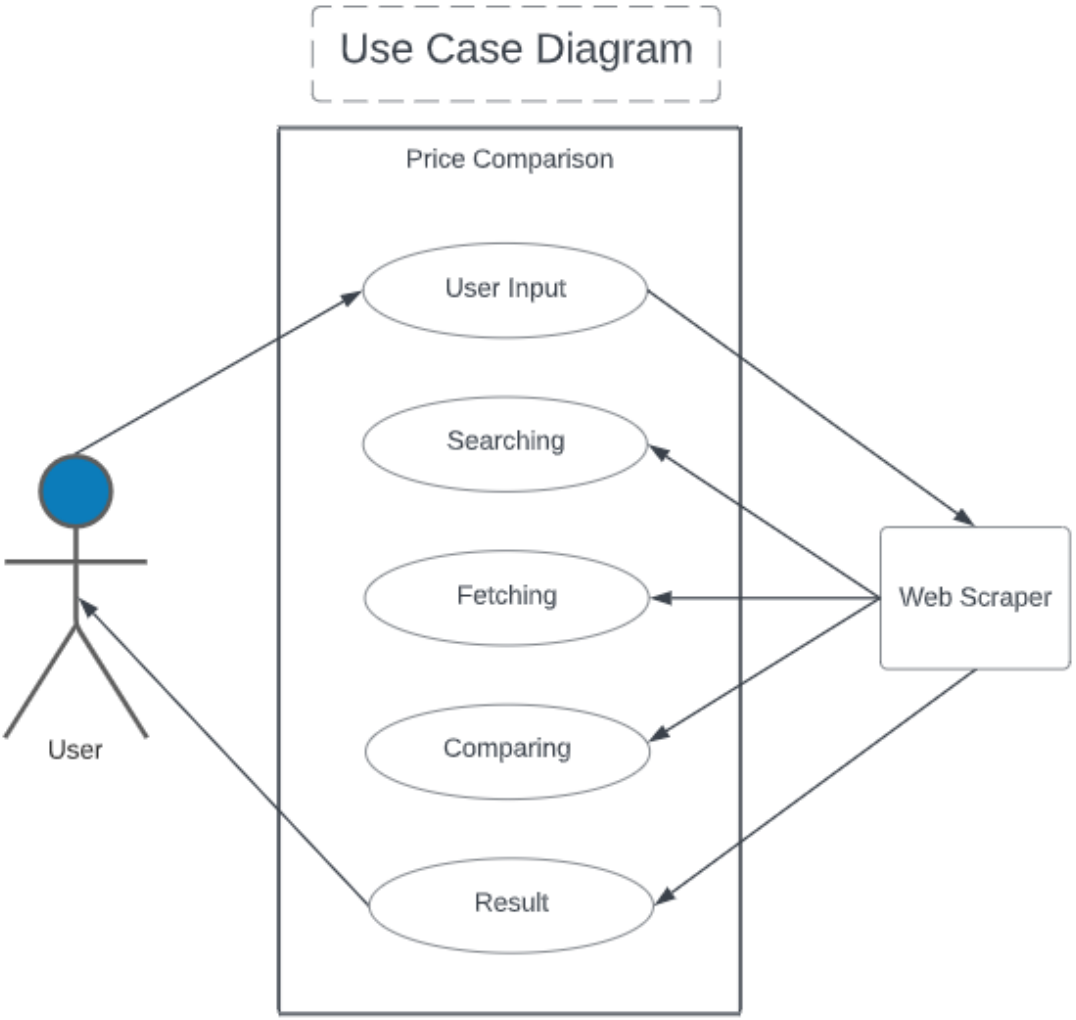


Fig. 4.5.1

CHAPTER 5

IMPLEMENTATION PHASE

- **Screenshots of Code on Visual Studio**
- **Output with GUI**
- **Code Snippets from Jupyter Notebook**

5.1 Screenshots of Code on Visual Studio

Imported Libraries

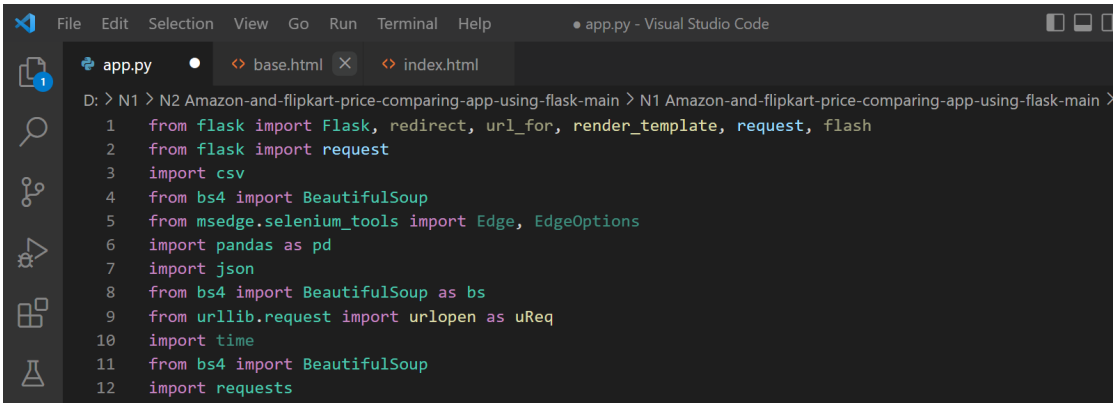


Fig. 5.1.1

Code for extracting data from websites like flipkart

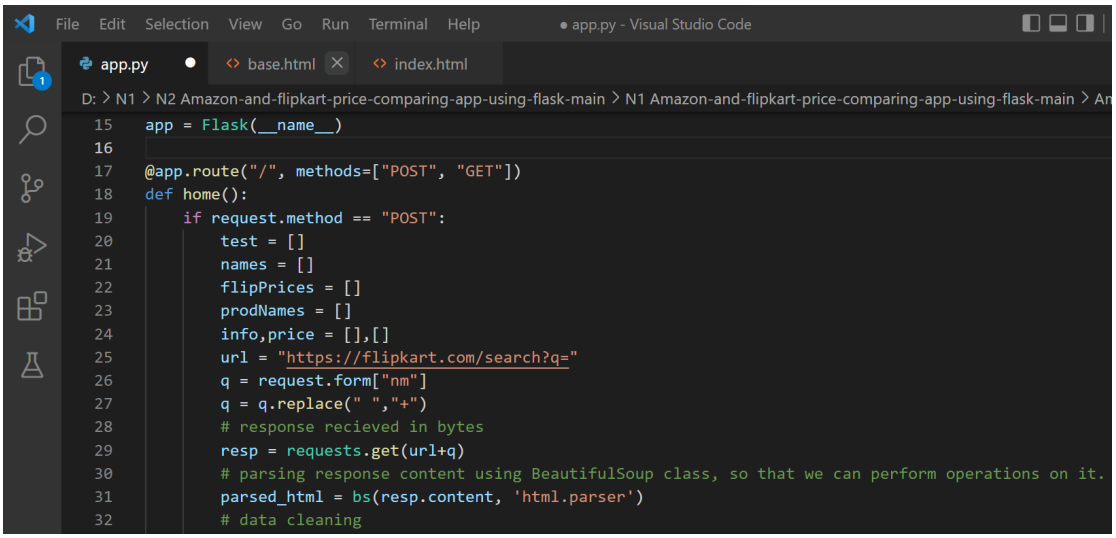


Fig. 5.1.2

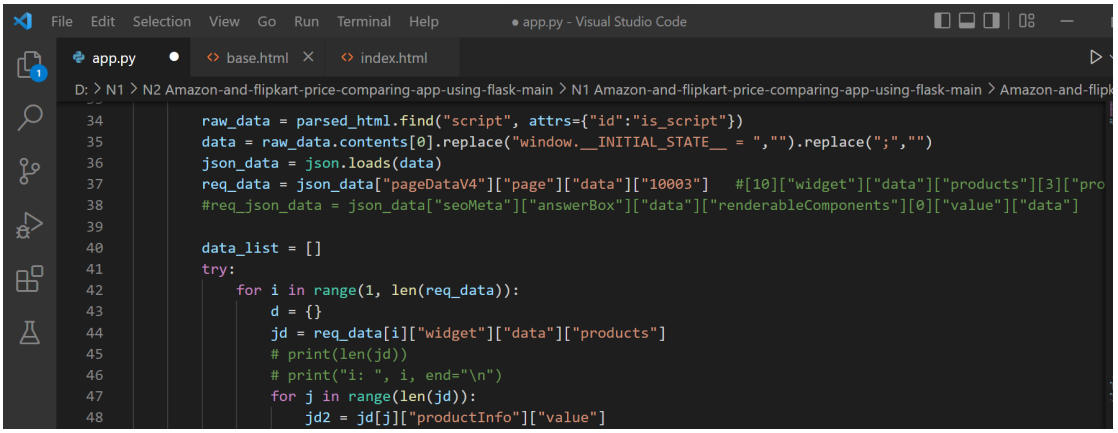


Fig. 5.1.3

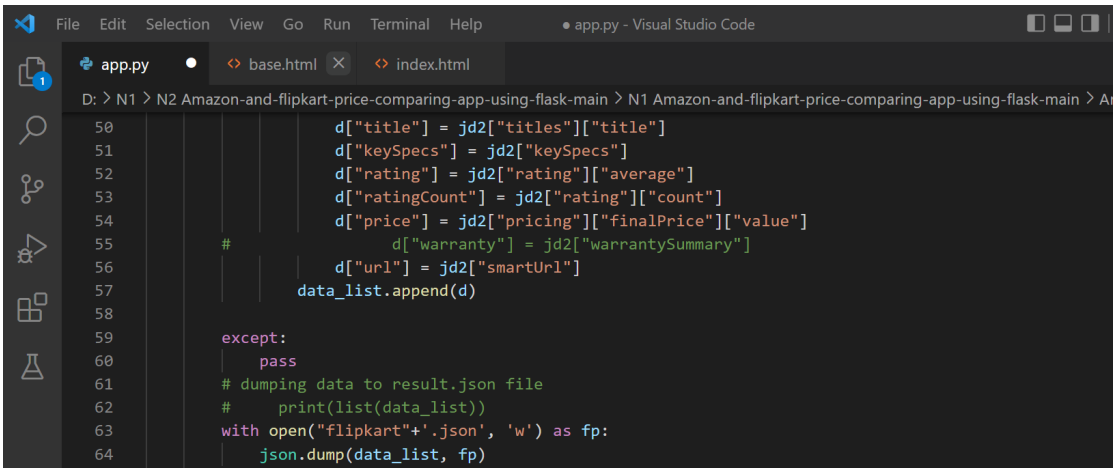


Fig. 5.1.4

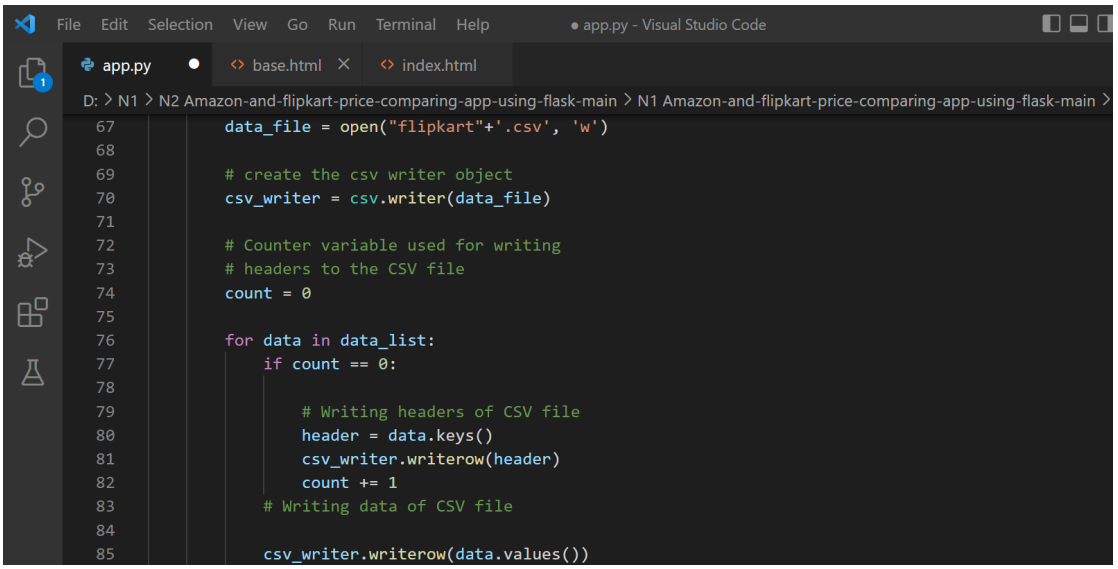


Fig. 5.1.5

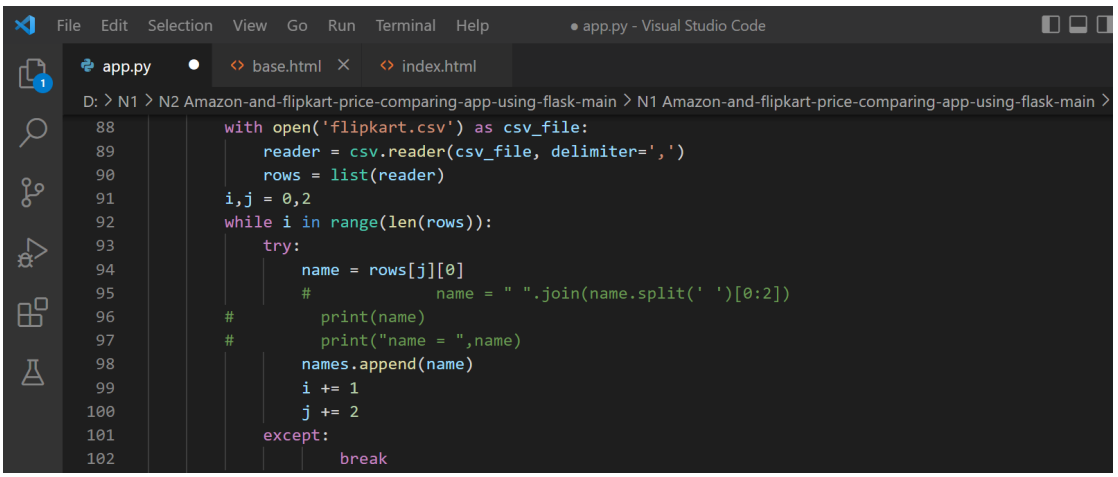


Fig. 5.1.6

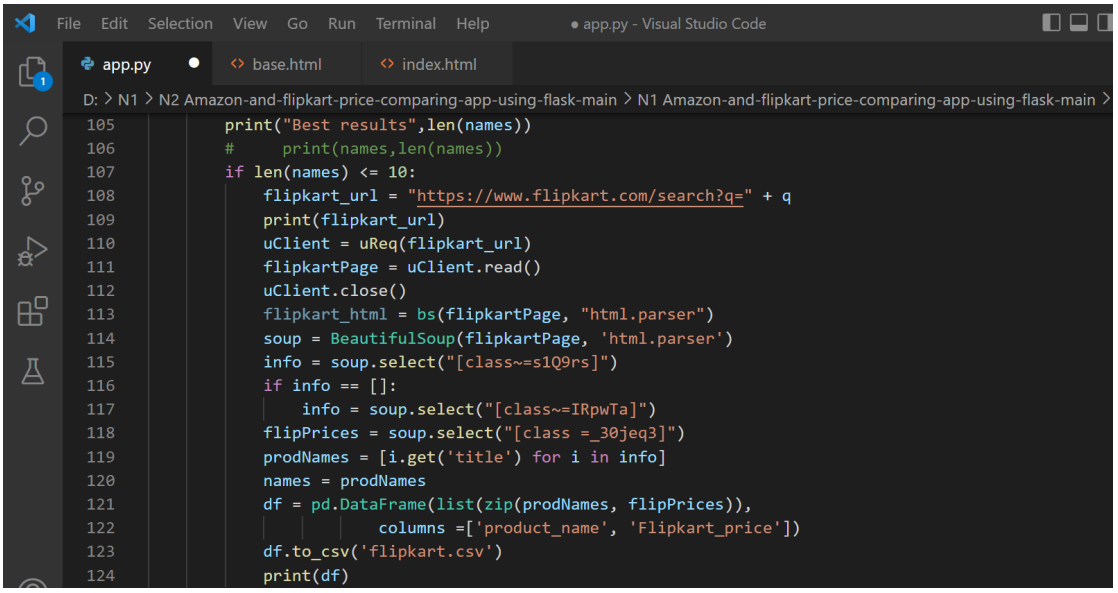


Fig. 5.1.7

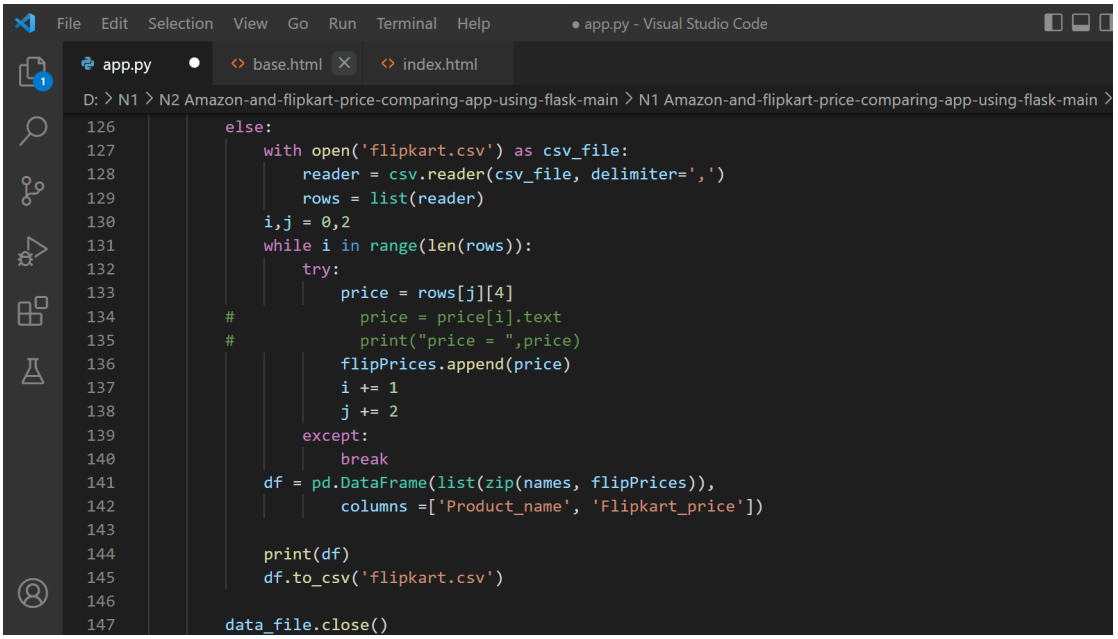


Fig. 5.1.8

Code for finding similar product from amazon

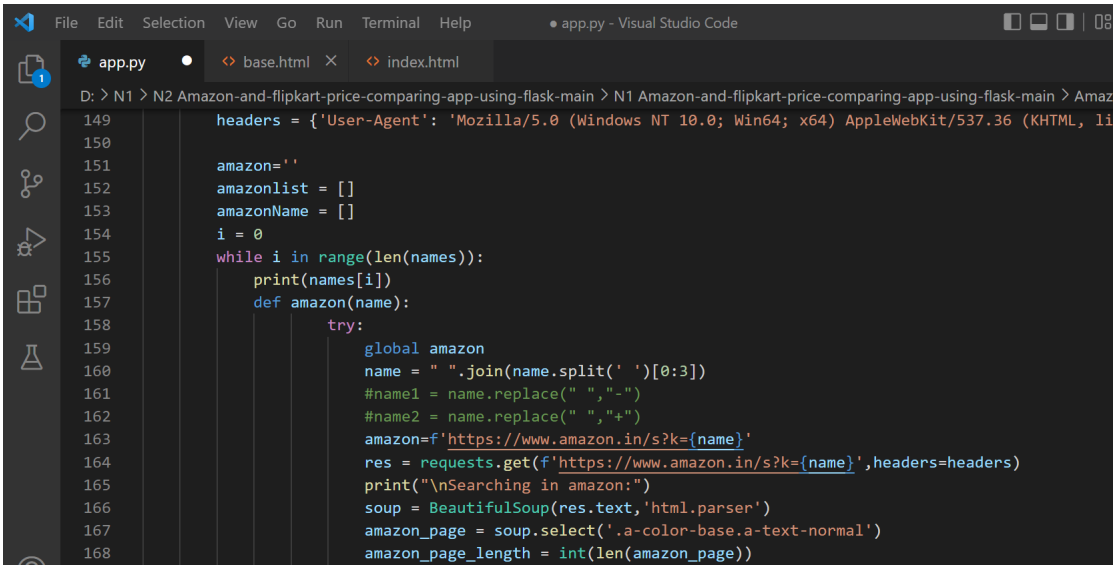


Fig. 5.1.9

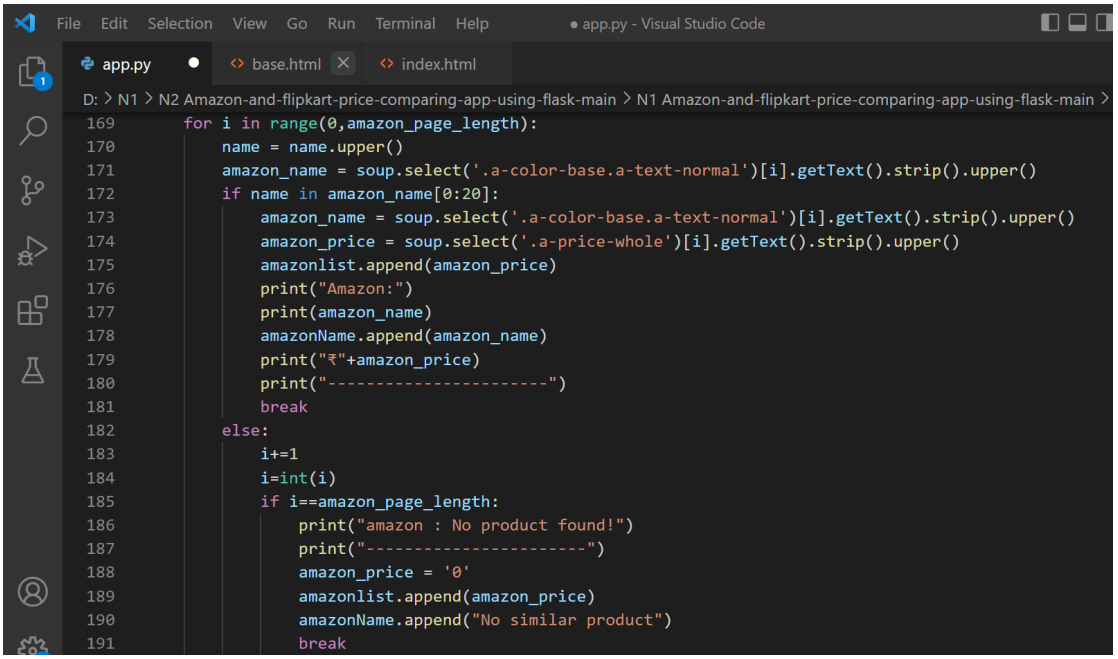


Fig. 5.1.10

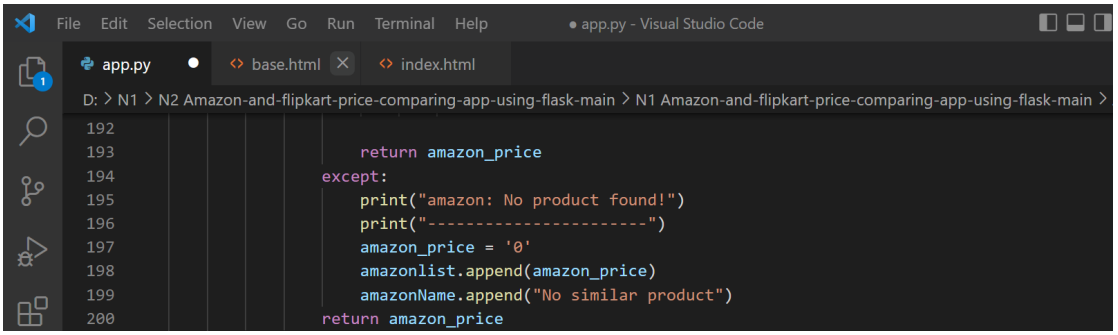


Fig. 5.1.11

Code for representing the product in tabular form

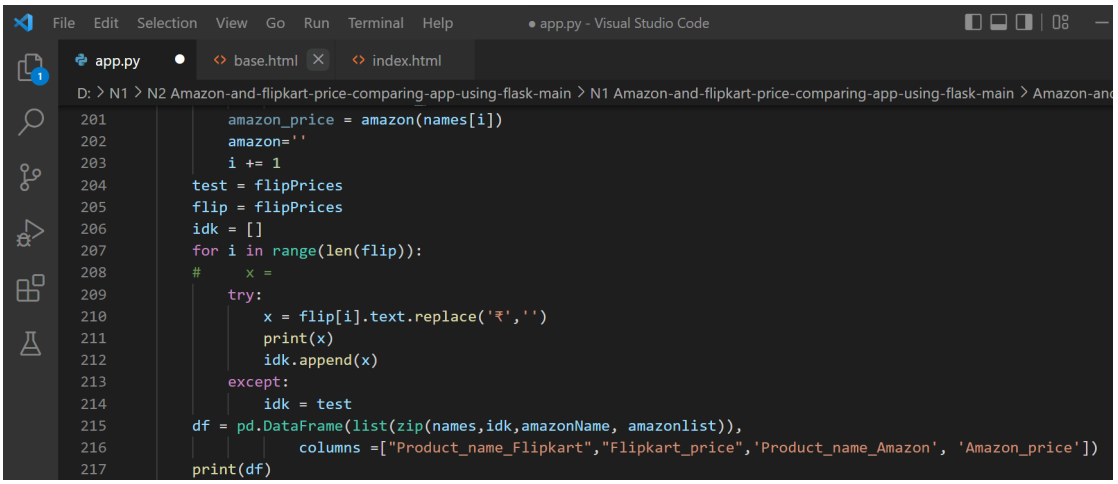


Fig. 5.1.12

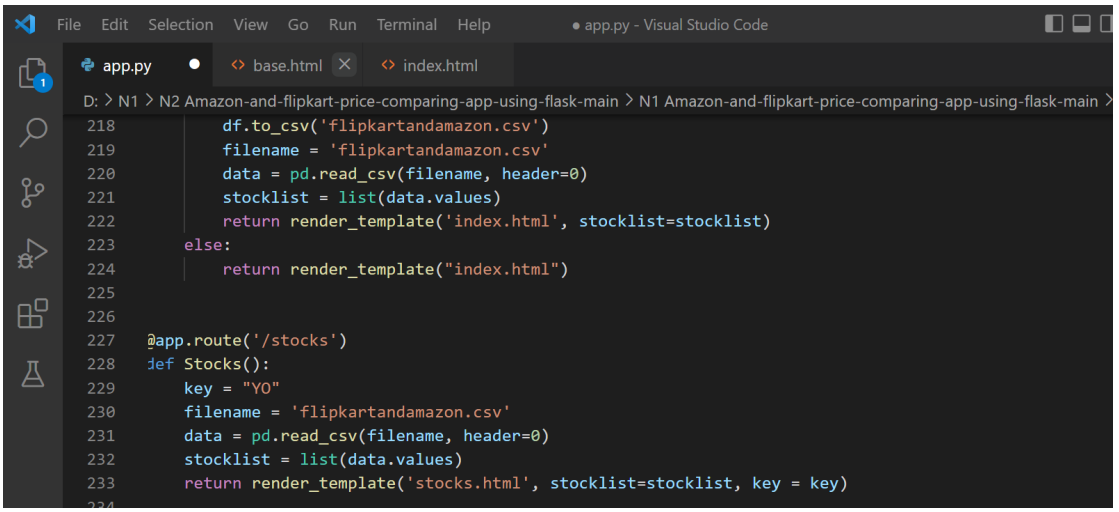


Fig. 5.1.13

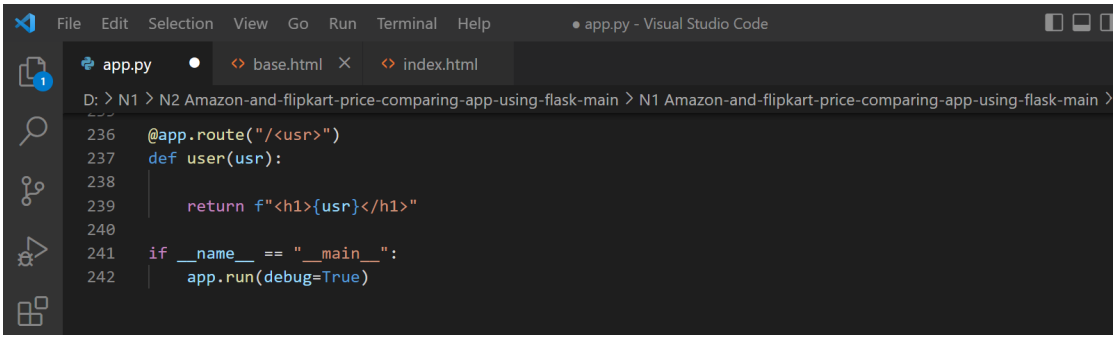


Fig. 5.1.14

Code of GUI (base.html)

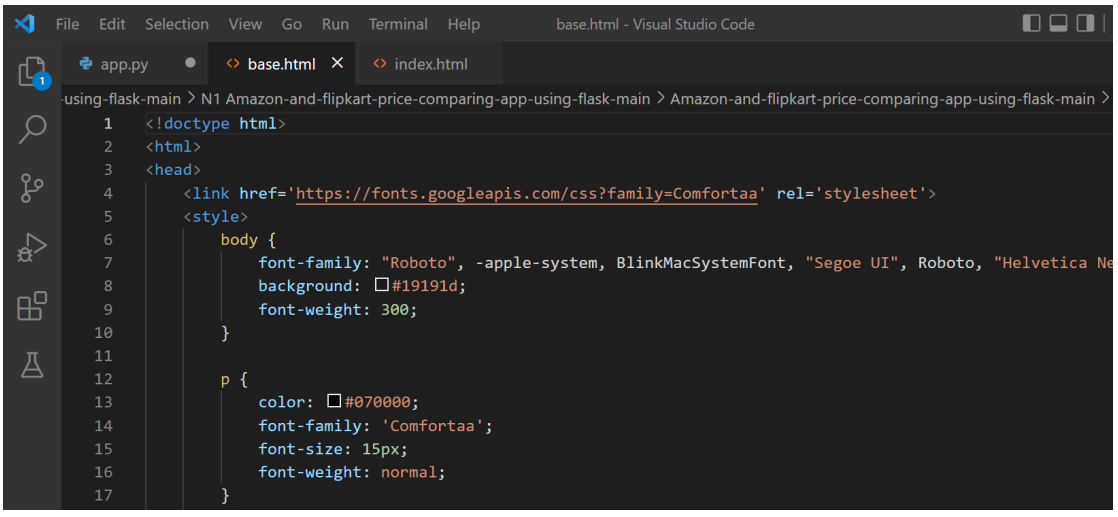


Fig. 5.1.15

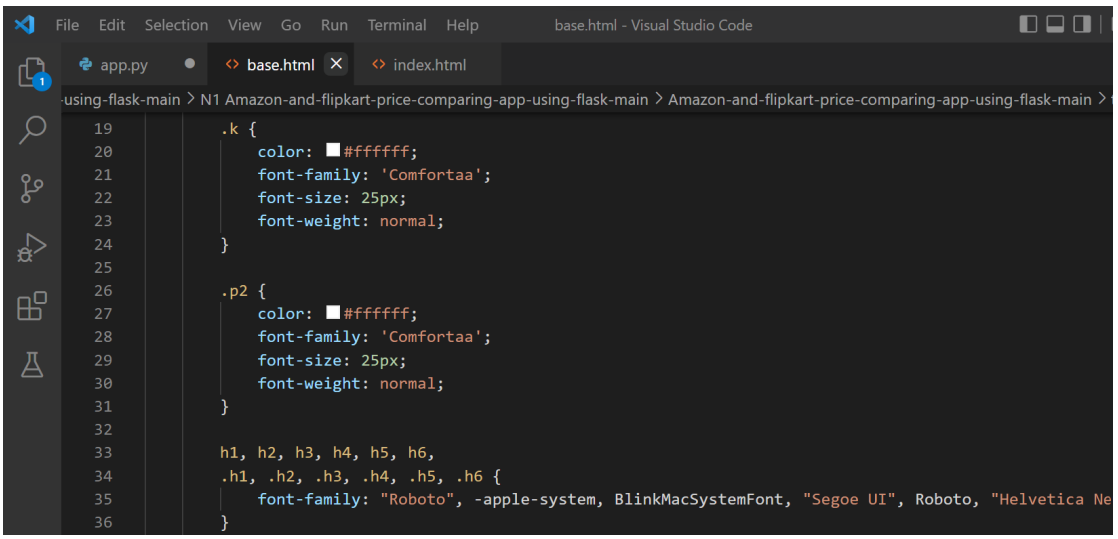


Fig. 5.1.16

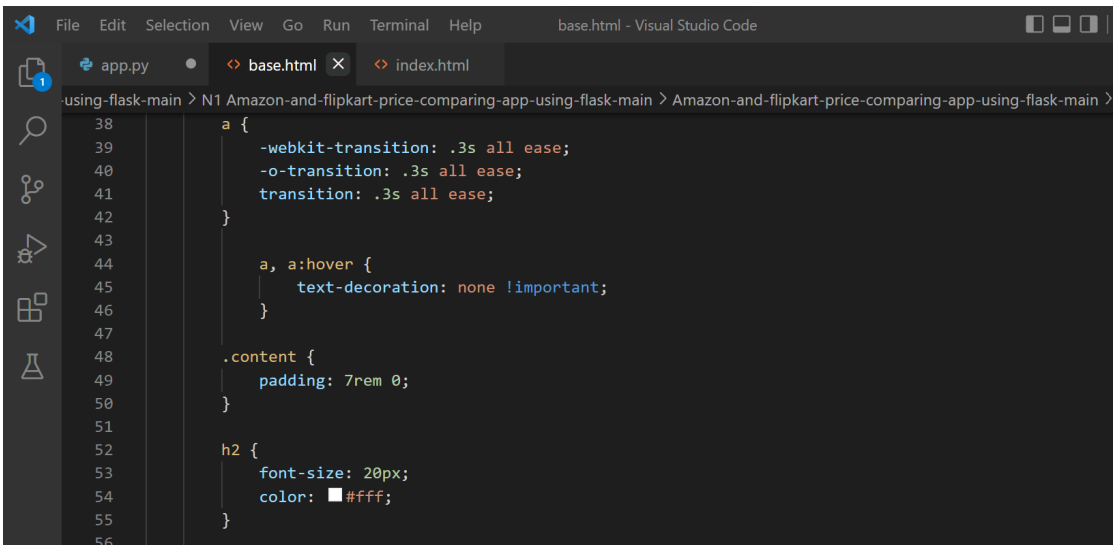


Fig. 5.1.17

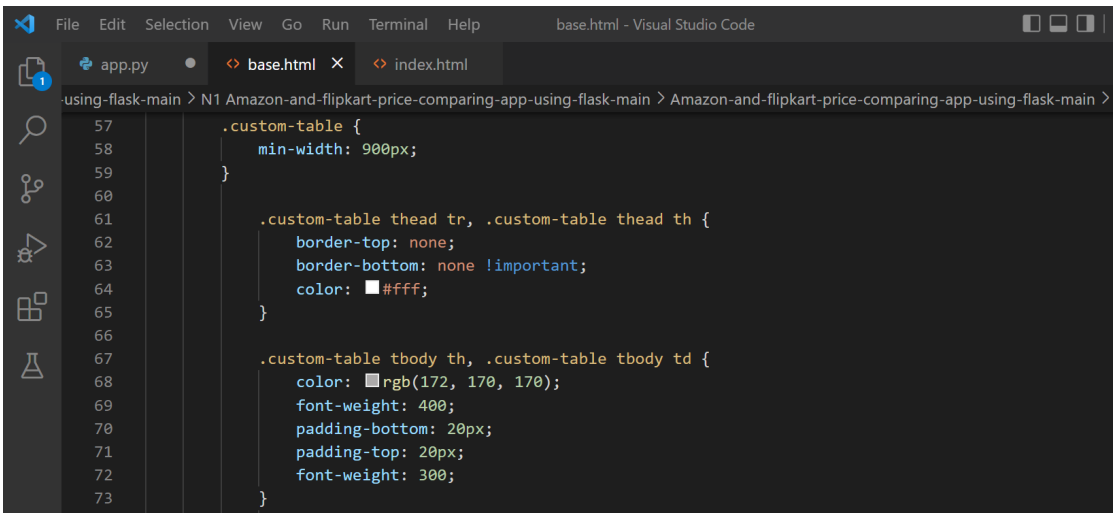


Fig. 5.1.18

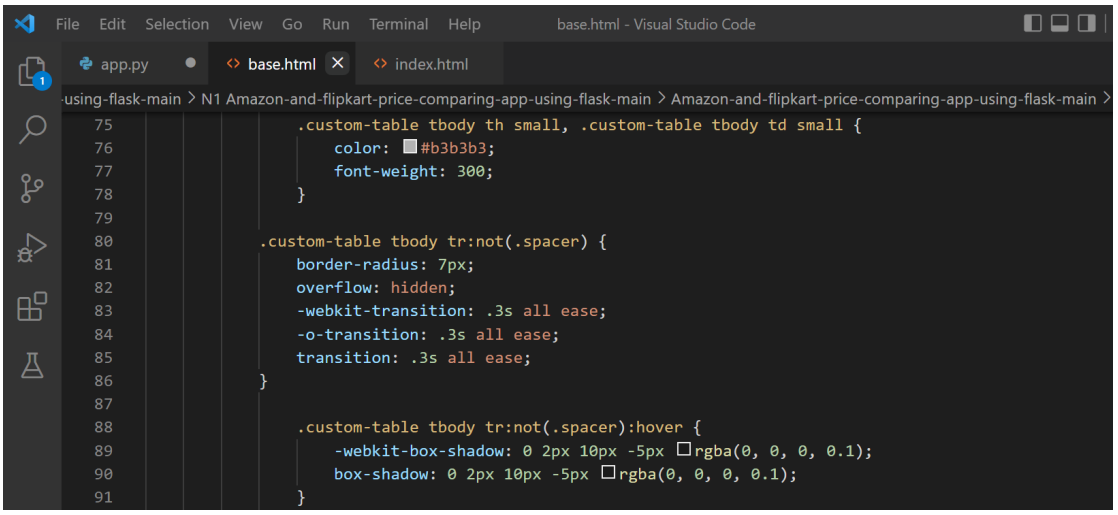


Fig. 5.1.19

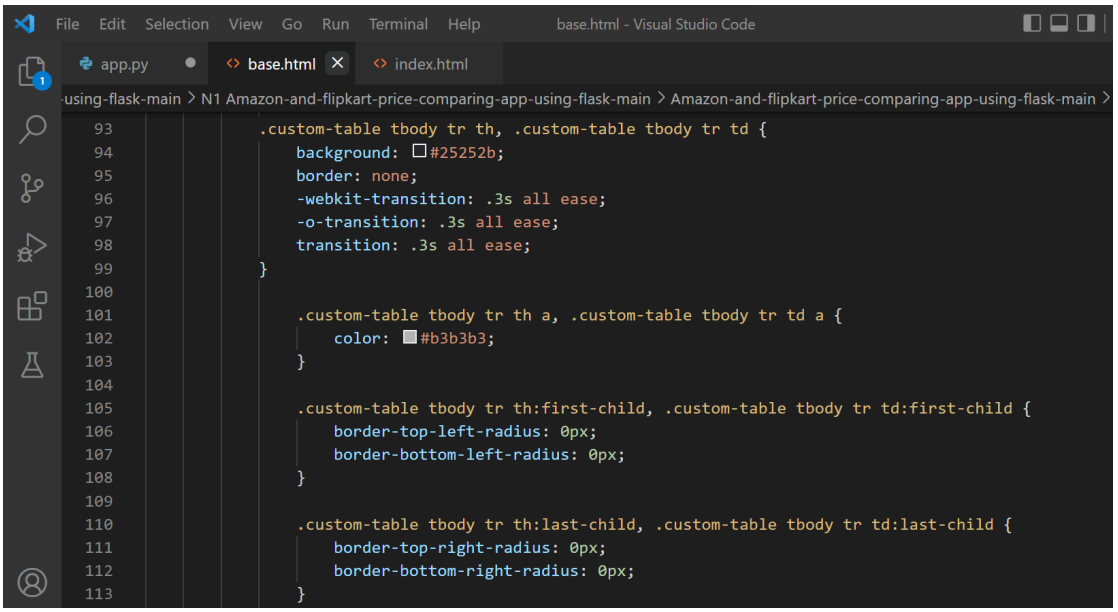


Fig. 5.1.20

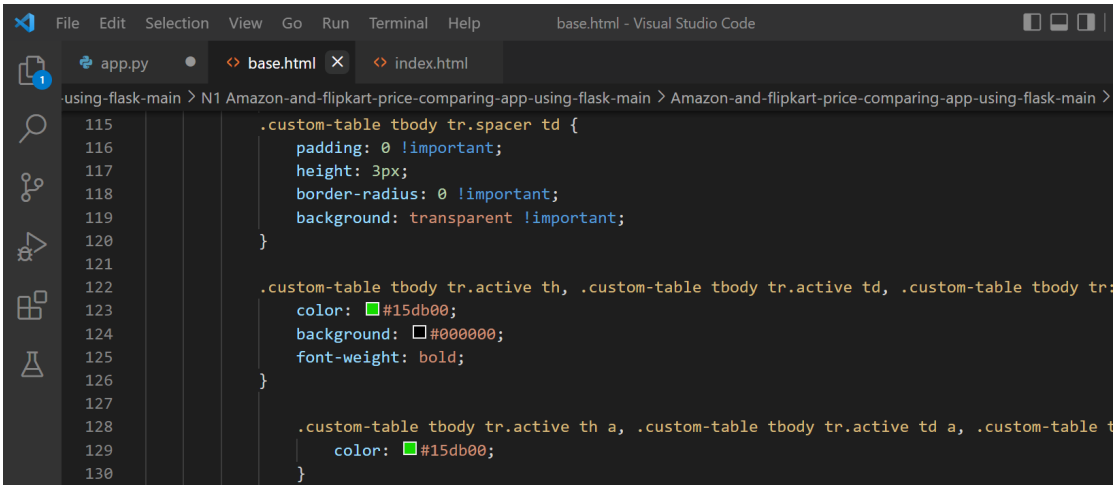


Fig. 5.1.21

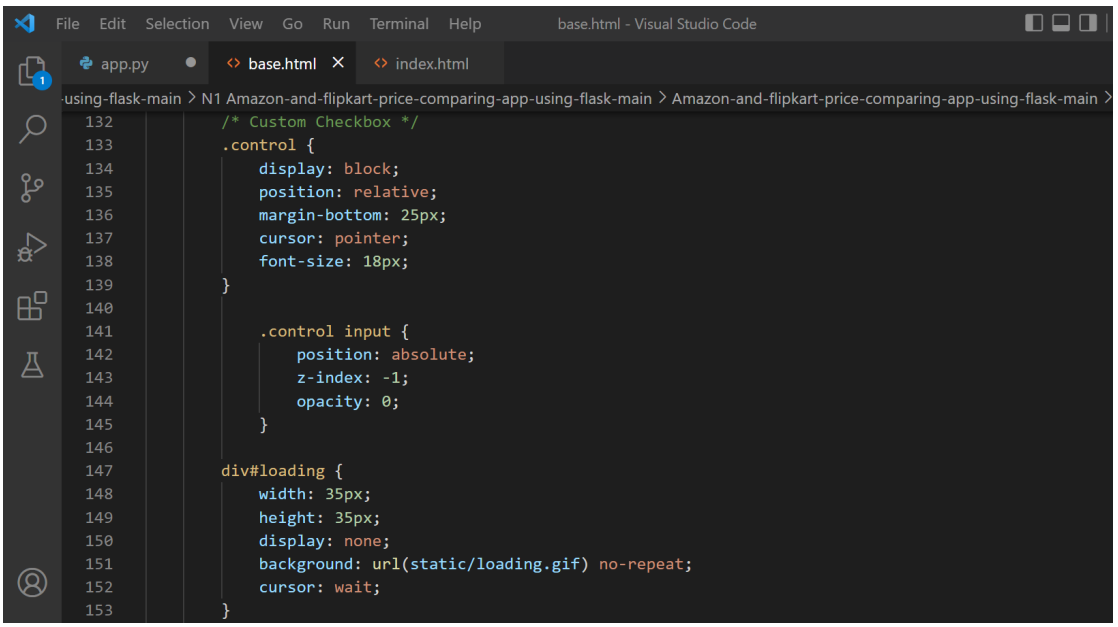


Fig. 5.1.22

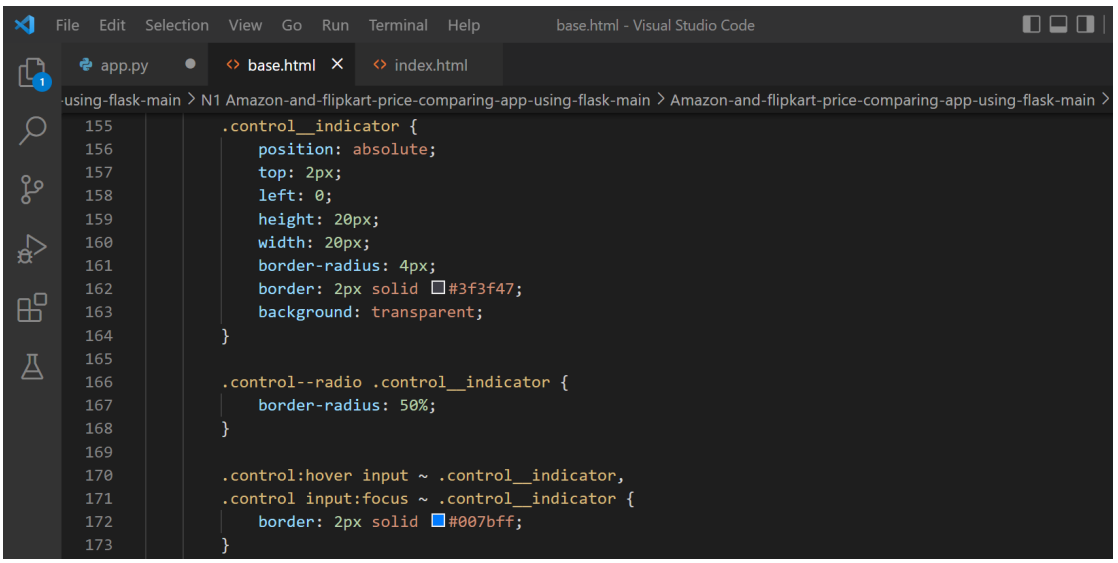


Fig. 5.1.23

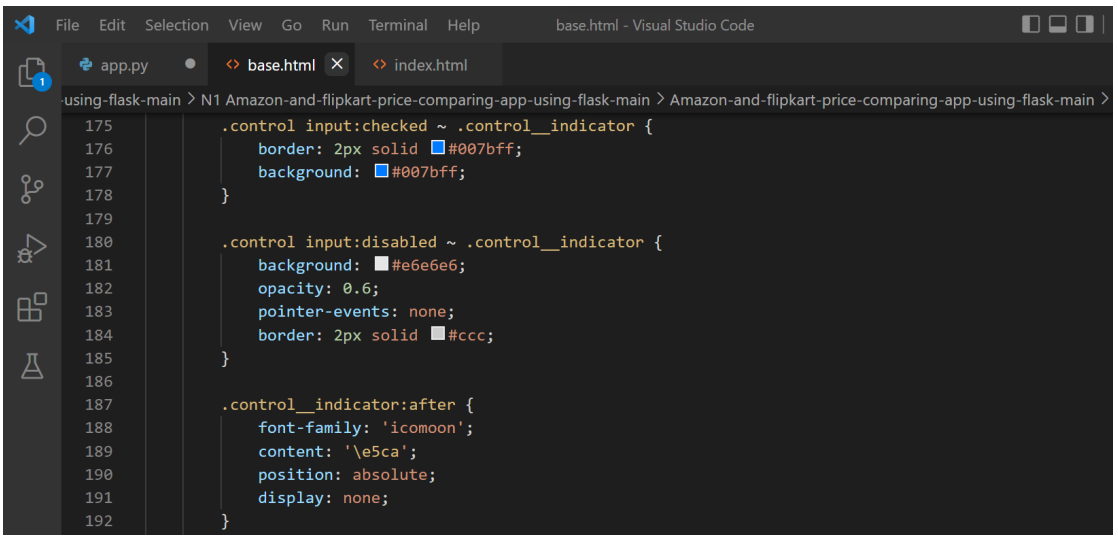


Fig. 5.1.24

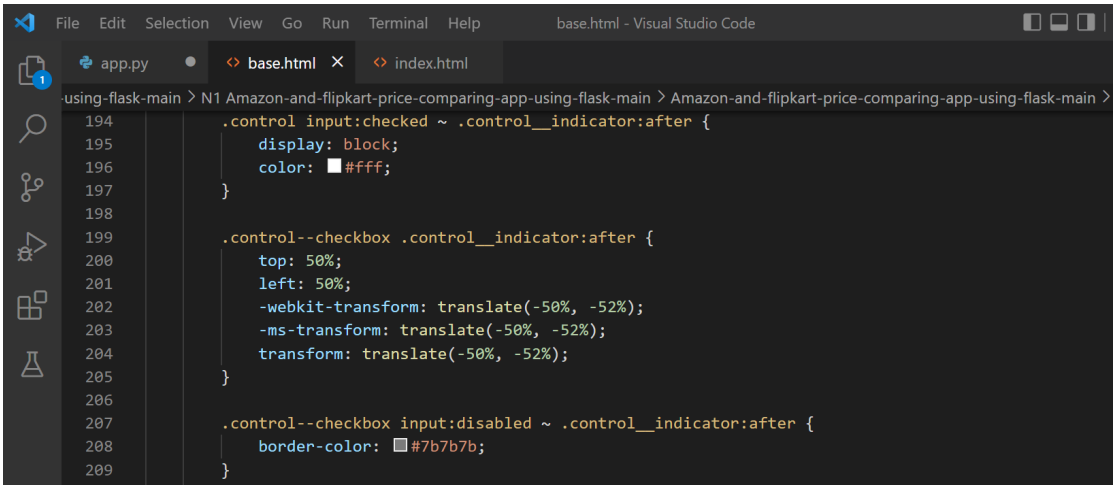


Fig. 5.1.25

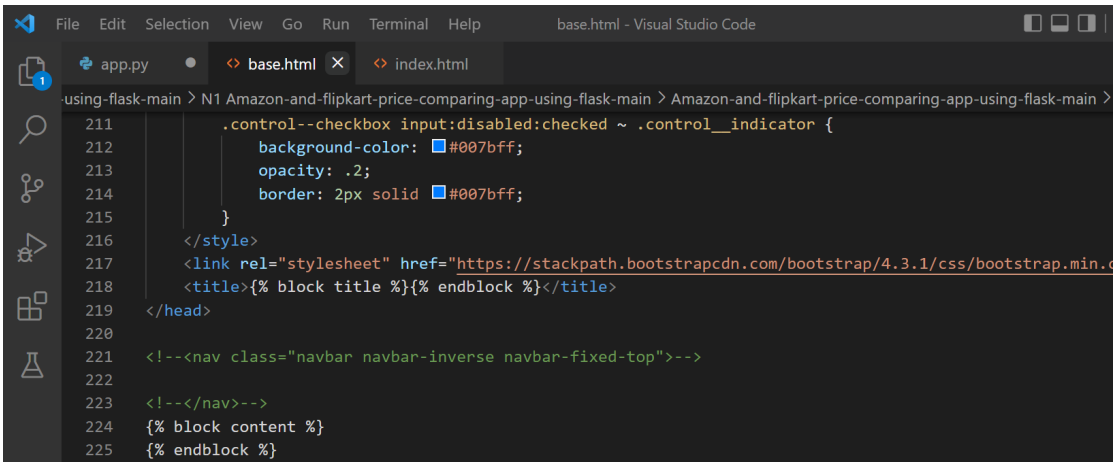


Fig. 5.1.26

Code of GUI (index.html)

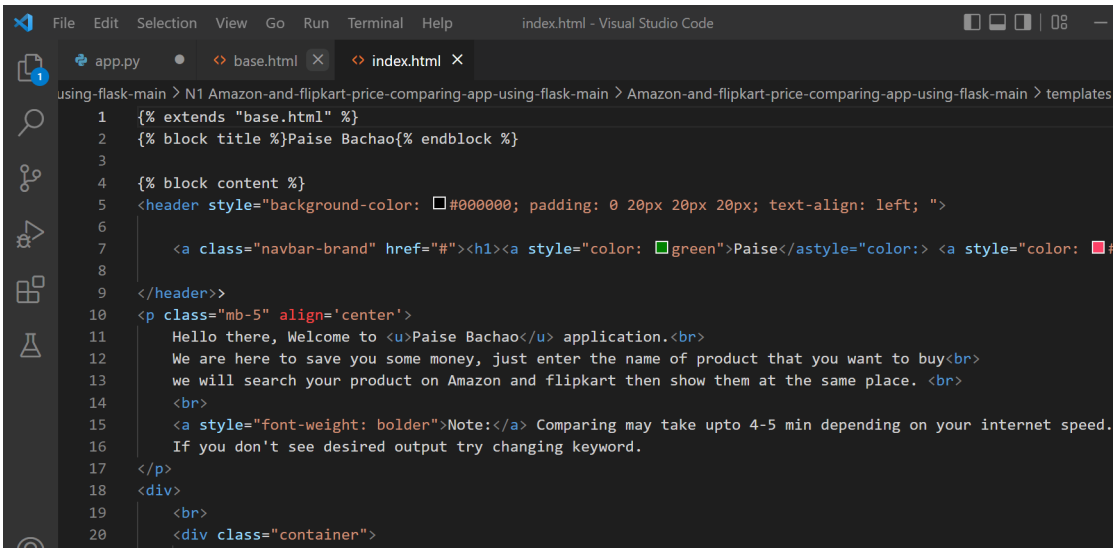


Fig. 5.1.27

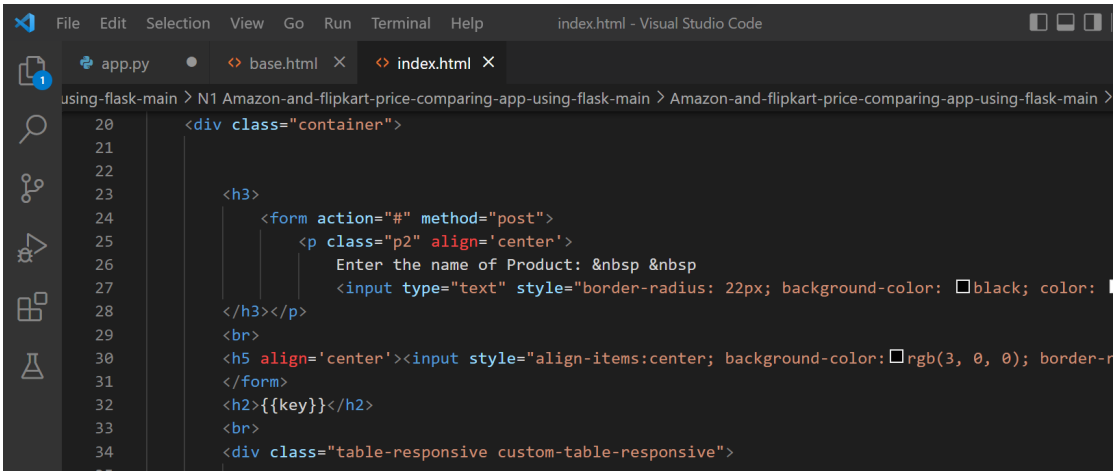


Fig. 5.1.28

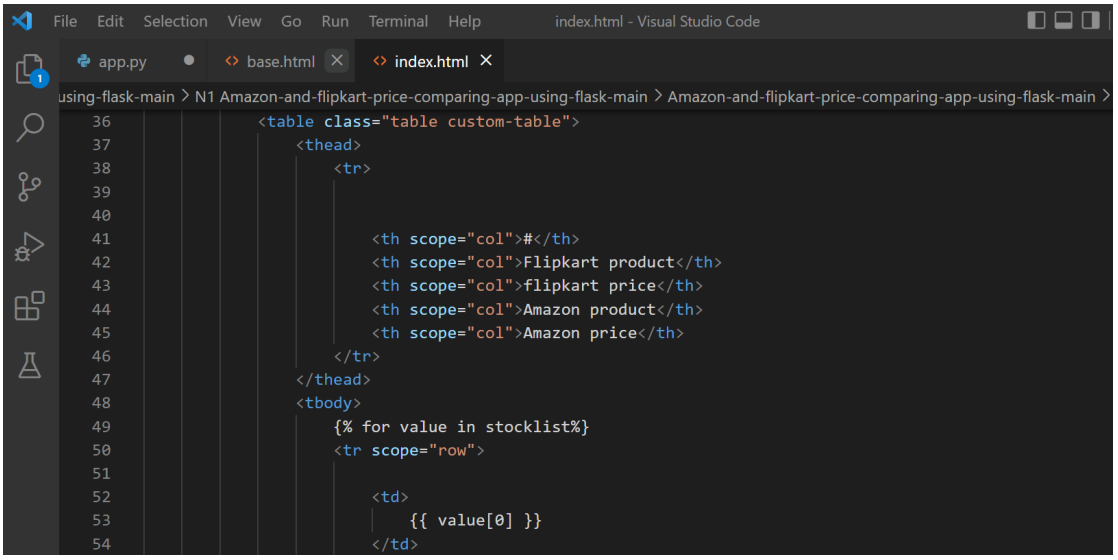


Fig. 5.1.29

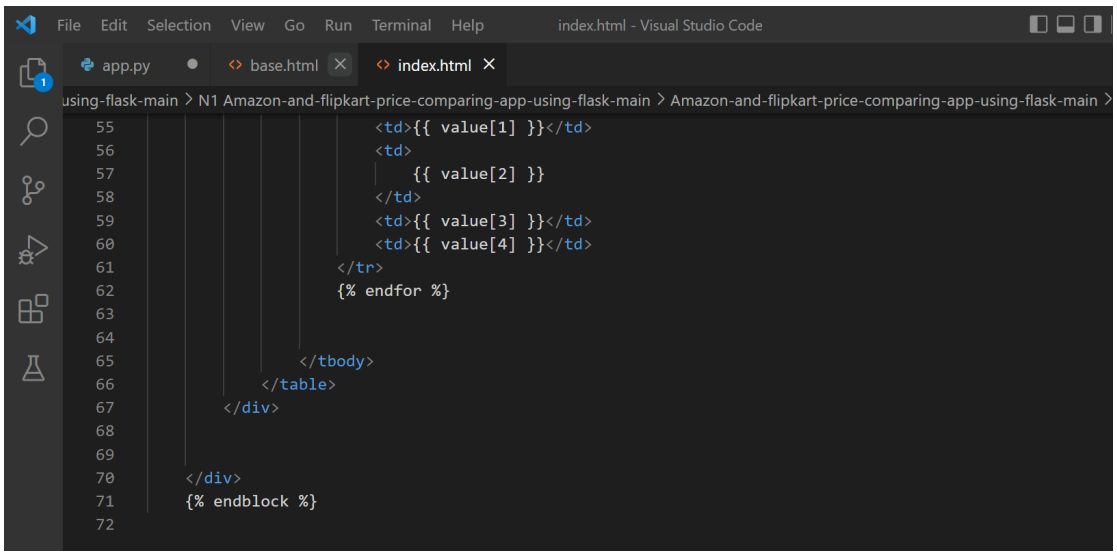


Fig. 5.1.30

5.2 Output with GUI

Running on Command Prompt

The link shown in fig. 3.1.1 is the main link of our website for price comparison.

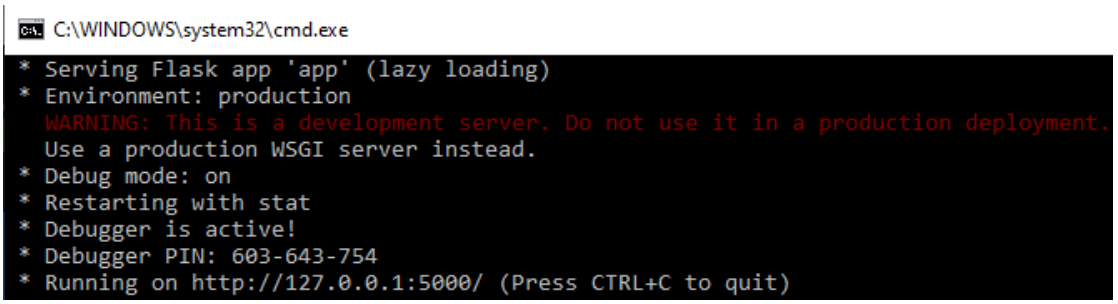


Fig. 5.2.1

Output of GUI

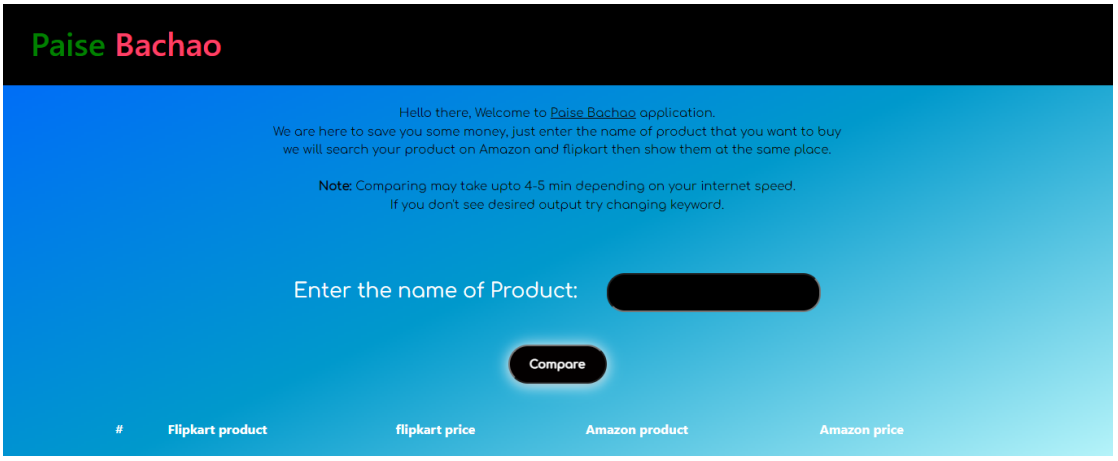


Fig. 5.2.2

Through this GUI user can interact with our project, users just have to input the keyword in the "ENTER THE NAME OF PRODUCT" section.

1	REDMI 10 (Pacific Blue, 64 GB)	10999	No similar product	0
2	POCO C31 (Royal Blue, 64 GB)	8999	POCO C31 (ROYAL BLUE, 64 GB) (4 GB RAM)	13,499
3	**realme C11 2021 (Cool Blue, 32 GB)**	**7499**	**REALME C11 2021 (COOL BLUE, 4GB RAM, 64GB STORAGE), MEDIUM (RMX3231)**	**9,549**
4	realme C11 2021 (Cool Grey, 64 GB)	8999	REALME C11 2021 (COOL BLUE, 4GB RAM, 64GB STORAGE), MEDIUM (RMX3231)	9,549
5	realme C11 2021 (Cool Blue, 64 GB)	8999	REALME C11 2021 (COOL BLUE, 4GB RAM, 64GB STORAGE), MEDIUM (RMX3231)	14,999
6	realme C11 2021 (Cool Grey, 32 GB)	7499	REALME C11 2021 (COOL BLUE, 4GB RAM, 64GB STORAGE), MEDIUM (RMX3231)	9,549
7	MOTOROLA G60 (Moonless, 128 GB)	17999	No similar product	0

Fig. 5.2.3

These are the results which are shown according to the keyword entered by the user.

5.3 Code Snippet on Jupyter with selected outputs

```
In [1]: import csv
        from bs4 import BeautifulSoup
        import pandas as pd
        import csv
        import json
        import requests
        from bs4 import BeautifulSoup as bs
        from urllib.request import urlopen as uReq
        import pandas as pd
        import time
```

Fig. 5.3.1

```
In [2]: q = input("Enter product name (Searching on flipkart)")
        q = q.replace(" ","+")
        test = []
        names = []
        flipPrices = []
        prodNames = []
        info,price = [],[]
        url = "https://flipkart.com/search?q="
        file_name = q.replace(" ", "_")

        resp = requests.get(url+q)
        parsed_html = bs(resp.content, 'html.parser')

        raw_data = parsed_html.find("script", attrs={"id":"is_script"})
        data = raw_data.contents[0].replace("window.__INITIAL_STATE__ = ","").replace(";","")
        json_data = json.loads(data)
        req_data = json_data["pageDataV4"]["page"]["data"]["10003"]

        data_list = []
        try:
            for i in range(1, len(req_data)):
                d = {}
                jd = req_data[i]["widget"]["data"]["products"]
                for j in range(len(jd)):
                    jd2 = jd[j]["productInfo"]["value"]

                    d["title"] = jd2["titles"]["title"]
                    d["keySpecs"] = jd2["keySpecs"]
                    d["rating"] = jd2["rating"]["average"]
                    d["ratingCount"] = jd2["rating"]["count"]
                    d["price"] = jd2["pricing"]["finalPrice"]["value"]
                    d["url"] = jd2["smartUrl"]
                data_list.append(d)
```

Fig. 5.3.2

```
            i += 1
            j += 2
        except:
            break
    df = pd.DataFrame(list(zip(names, flipPrices)),
                columns =['Product_name', 'Flipkart_price'])

    df.to_csv('test.csv')
    print(df)


data_file.close()
```

```
Enter product name (Searching on flipkart)Laptops
Best results 18
                                          Product_name Flipkart_price
0   HP Core i3 11th Gen - (8 GB/512 GB SSD/Windows...          42990
1   ASUS Celeron Dual Core - (4 GB/1 TB HDD/Window...          24990
2   ASUS VivoBook 14 Pentium Silver - (4 GB/256 GB...          28990
3   Infinix INBook X1 Core i3 10th Gen - (8 GB/256...          34990
4   Lenovo IdeaPad 3 Core i3 10th Gen - (8 GB/256 ...          37490
5   acer Aspire 7 Core i5 10th Gen - (8 GB/512 GB ...          52990
6   HP Pavilion Ryzen 5 Hexa Core 5600H - (8 GB/51...          57990
7   DELL Vostro Core i3 10th Gen - (8 GB/1 TB HDD/...          40990
8   ASUS VivoBook 15 (2021) Core i3 10th Gen - (8 ...          40990
9   MSI GF63 Thin Core i5 10th Gen - (8 GB/512 GB ...          59990
10  HP Ryzen 3 Dual Core 3250U - (8 GB/256 GB SSD/...          37990
11  ASUS Core i3 11th Gen - (8 GB/256 GB SSD/Windo...          38990
12  HP Core i3 11th Gen - (8 GB/256 GB SSD/Windows...          40990
13  acer Aspire 3 Ryzen 3 Dual Core 3250U - (8 GB/...          36990
14  Lenovo IdeaPad 3 Celeron Dual Core 4th Gen - (...          29990
15  DELL Vostro Core i3 11th Gen - (8 GB/256 GB SS...          38990
16  DELL Vostro Core i3 10th Gen - (8 GB/256 GB SS...          37990
17  MSI GF65 Thin Core i7 10th Gen - (16 GB/1 TB S...          84990
```

Fig. 5.3.3

In [4]:
```
import pandas as pd
df = pd.read_csv("test.csv", sep=",")
df.head(100)
```

Out[4]:

	Unnamed: 0	Product_name	Flipkart_price
0	0	HP Core i3 11th Gen - (8 GB/512 GB SSD/Windows...	42990
1	1	ASUS Celeron Dual Core - (4 GB/1 TB HDD/Window...	24990
2	2	ASUS VivoBook 14 Pentium Silver - (4 GB/256 GB...	28990
3	3	Infinix INBook X1 Core i3 10th Gen - (8 GB/256...	34990
4	4	Lenovo IdeaPad 3 Core i3 10th Gen - (8 GB/256 ...	37490
5	5	acer Aspire 7 Core i5 10th Gen - (8 GB/512 GB ...	52990
6	6	HP Pavilion Ryzen 5 Hexa Core 5600H - (8 GB/51...	57990
7	7	DELL Vostro Core i3 10th Gen - (8 GB/1 TB HDD/...	40990
8	8	ASUS VivoBook 15 (2021) Core i3 10th Gen - (8 ...	40990
9	9	MSI GF63 Thin Core i5 10th Gen - (8 GB/512 GB ...	59990
10	10	HP Ryzen 3 Dual Core 3250U - (8 GB/256 GB SSD/...	37990
11	11	ASUS Core i3 11th Gen - (8 GB/256 GB SSD/Windo...	38990
12	12	HP Core i3 11th Gen - (8 GB/256 GB SSD/Windows...	40990
13	13	acer Aspire 3 Ryzen 3 Dual Core 3250U - (8 GB/...	36990
14	14	Lenovo IdeaPad 3 Celeron Dual Core 4th Gen - (...	29990
15	15	DELL Vostro Core i3 11th Gen - (8 GB/256 GB SS...	38990
16	16	DELL Vostro Core i3 10th Gen - (8 GB/256 GB SS...	37990
17	17	MSI GF65 Thin Core i7 10th Gen - (16 GB/1 TB S...	84990

Fig. 5.3.4

In [6]:
```
from bs4 import BeautifulSoup
import requests
import time
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 S

flipkart=''
ebay=''
amazon=''
olx=''

amazonlist = []
amazonName = []
i = 0
while i in range(len(names)):
    print(names[i])
    def amazon(name):
            try:
                global amazon
                name = " ".join(name.split(' ')[0:2])
                name1 = name.replace(" ","-")
                name2 = name.replace(" ","+")
                amazon=f'https://www.amazon.in/{name1}/s?k={name2}'
                res = requests.get(f'https://www.amazon.in/{name1}/s?k={name2}',headers=headers)
                print("\nSearching in amazon:")
                soup = BeautifulSoup(res.text,'html.parser')
                amazon_page = soup.select('.a-color-base.a-text-normal')
                amazon_page_length = int(len(amazon_page))
                for i in range(0,amazon_page_length):
                    name = name.upper()
                    amazon_name = soup.select('.a-color-base.a-text-normal')[i].getText().strip().upper()
                    if name in amazon_name[0:20]:
                        amazon_name = soup.select('.a-color-base.a-text-normal')[i].getText().strip().upper()
                        amazon_price = soup.select('.a-price-whole')[i].getText().strip().upper()
                        amazonlist.append(amazon_price)
                        print("Amazon:")
```

Fig. 5.3.5

```
flip = flipPrices
idk = []
for i in range(len(flip)):
#     x =
    try:
        x = flip[i].text.replace('₹','')
        print(x)
        idk.append(x)
    except:
        idk = test
df = pd.DataFrame(list(zip(names,idk,amazonName, amazonlist)),
                  columns =["Product_name_Flipkart","Flipkart_price",'Product_name_Amazon', 'Amazon_price'])

df.to_csv('flipkartandamazon.csv')
df
```

```
ASUS VivoBook 14 Pentium Silver - (4 GB/256 GB SSD/Windows 11 Home) X415KA-EK111WS Thin and Light Laptop

Searching in amazon:
Amazon:
ASUS VIVOBOOK 14 (2021), INTEL CORE I5-1135G7 11TH GEN, 14-INCH (35.56 CMS) FHD THIN AND LIGHT LAPTOP (8GB/1TB HDD + 256GB SS
D/OFFICE 2021/WINDOWS 11/IRIS XE GRAPHICS/SILVER/1.6 KG), X415EA-EK572WS
₹38,844
----------------------
Infinix INBook X1 Core i3 10th Gen - (8 GB/256 GB SSD/Windows 11 Home) XL11 Thin and Light Laptop

Searching in amazon:
amazon : No product found!
----------------------
Lenovo IdeaPad 3 Core i3 10th Gen - (8 GB/256 GB SSD/Windows 11 Home) 15IML05 Thin and Light Laptop

Searching in amazon:
Amazon:
LENOVO IDEAPAD SLIM 5 11TH GEN INTEL CORE I5 15.6" FHD IPS THIN & LIGHT LAPTOP(16GB/512GB SSD/WINDOWS 11/OFFICE 2021/BACKLIT/
```

Fig. 5.3.6

In [7]:
```
df = pd.DataFrame(list(zip(amazonName, amazonlist)),
                  columns =['Product_name', 'Amazon_price'])

print(df)
```

	Product_name	Amazon_price
0	No similar product	0
1	No similar product	0
2	ASUS VIVOBOOK 14 (2021), INTEL CORE I5-1135G7 ...	38,844
3	No similar product	0
4	LENOVO IDEAPAD SLIM 5 11TH GEN INTEL CORE I5 1...	48,710
5	ACER ASPIRE 3 AMD ATHLON SILVER 3050U DUAL-COR...	38,011
6	HP PAVILION 15 AMD RYZEN 5- 8GB/512GB SSD 15.6...	67,690
7	DELL VOSTRO 3400 INTEL I3-1115G4 14 INCHES FHD...	41,490
8	ASUS VIVOBOOK 14 (2021), INTEL CORE I5-1135G7 ...	38,844
9	MSI GF63 THIN INTEL CORE I5 10TH GEN 15.6 INCH...	58,499
10	VICTUS BY HP RYZEN 7-5800H 16.1 INCH(40.9 CM) ...	58,990
11	ASUS CORE I5 10TH GEN - (8 GB + 32 GB OPTANE/5...	33,988
12	No similar product	0
13	ACER ASPIRE 5 INTEL 11TH GEN CORE I5 15.6 INCH...	28,990
14	LENOVO IDEAPAD SLIM 5 11TH GEN INTEL CORE I5 1...	48,710
15	DELL VOSTRO 3400 INTEL I3-1115G4 14 INCHES FHD...	41,490
16	DELL VOSTRO 3400 INTEL I3-1115G4 14 INCHES FHD...	41,490
17	MSI GF65 THIN INTEL CORE I7 10TH GEN 15.6 INCH...	83,990

In [15]:
```
import csv

file = open('flipkartandamazon.csv', 'w+', newline ='')
with file:
    write = csv.writer(file)
    write.writerows(names)
    write.writerows(idk)
    write.writerows(amazonName)
    write.writerows(amazonlist)
```

In [16]:
```
df = pd.DataFrame(list(zip(names,idk,amazonName, amazonlist)),
                  columns =["Product_name_Flipkart","Flipkart_price",'Product_name_Amazon', 'Amazon_price'])

df
```

Fig. 5.3.7

Out[16]:

	Product_name_Flipkart	Flipkart_price	Product_name_Amazon	Amazon_price
0	HP Core i3 11th Gen - (8 GB/512 GB SSD/Windows...	42990	No similar product	0
1	ASUS Celeron Dual Core - (4 GB/1 TB HDD/Window...	24990	No similar product	0
2	ASUS VivoBook 14 Pentium Silver - (4 GB/256 GB...	28990	ASUS VIVOBOOK 14 (2021), INTEL CORE I5-1135G7 ...	38,844
3	Infinix INBook X1 Core i3 10th Gen - (8 GB/256...	34990	No similar product	0
4	Lenovo IdeaPad 3 Core i3 10th Gen - (8 GB/256 ...	37490	LENOVO IDEAPAD SLIM 5 11TH GEN INTEL CORE I5 1...	48,710
5	acer Aspire 7 Core i5 10th Gen - (8 GB/512 GB ...	52990	ACER ASPIRE 3 AMD ATHLON SILVER 3050U DUAL-COR...	38,011
6	HP Pavilion Ryzen 5 Hexa Core 5600H - (8 GB/51...	57990	HP PAVILION 15 AMD RYZEN 5- 8GB/512GB SSD 15.6...	67,690
7	DELL Vostro Core i3 10th Gen - (8 GB/1 TB HDD/...	40990	DELL VOSTRO 3400 INTEL I3-1115G4 14 INCHES FHD...	41,490
8	ASUS VivoBook 15 (2021) Core i3 10th Gen - (8 ...	40990	ASUS VIVOBOOK 14 (2021), INTEL CORE I5-1135G7 ...	38,844
9	MSI GF63 Thin Core i5 10th Gen - (8 GB/512 GB ...	59990	MSI GF63 THIN INTEL CORE I5 10TH GEN 15.6 INCH...	58,499
10	HP Ryzen 3 Dual Core 3250U - (8 GB/256 GB SSD/...	37990	VICTUS BY HP RYZEN 7-5800H 16.1 INCH(40.9 CM) ...	58,990
11	ASUS Core i3 11th Gen - (8 GB/256 GB SSD/Windo...	38990	ASUS CORE I5 10TH GEN - (8 GB + 32 GB OPTANE/5...	33,988
12	HP Core i3 11th Gen - (8 GB/256 GB SSD/Windows...	40990	No similar product	0
13	acer Aspire 3 Ryzen 3 Dual Core 3250U - (8 GB/...	36990	ACER ASPIRE 5 INTEL 11TH GEN CORE I5 15.6 INCH...	28,990
14	Lenovo IdeaPad 3 Celeron Dual Core 4th Gen - (...	29990	LENOVO IDEAPAD SLIM 5 11TH GEN INTEL CORE I5 1...	48,710
15	DELL Vostro Core i3 11th Gen - (8 GB/256 GB SS...	38990	DELL VOSTRO 3400 INTEL I3-1115G4 14 INCHES FHD...	41,490
16	DELL Vostro Core i3 10th Gen - (8 GB/256 GB SS...	37990	DELL VOSTRO 3400 INTEL I3-1115G4 14 INCHES FHD...	41,490
17	MSI GF65 Thin Core i7 10th Gen - (16 GB/1 TB S...	84990	MSI GF65 THIN INTEL CORE I7 10TH GEN 15.6 INCH...	83,990

Fig. 5.3.8

5.4 Screenshots of Code on Visual Code

app.py

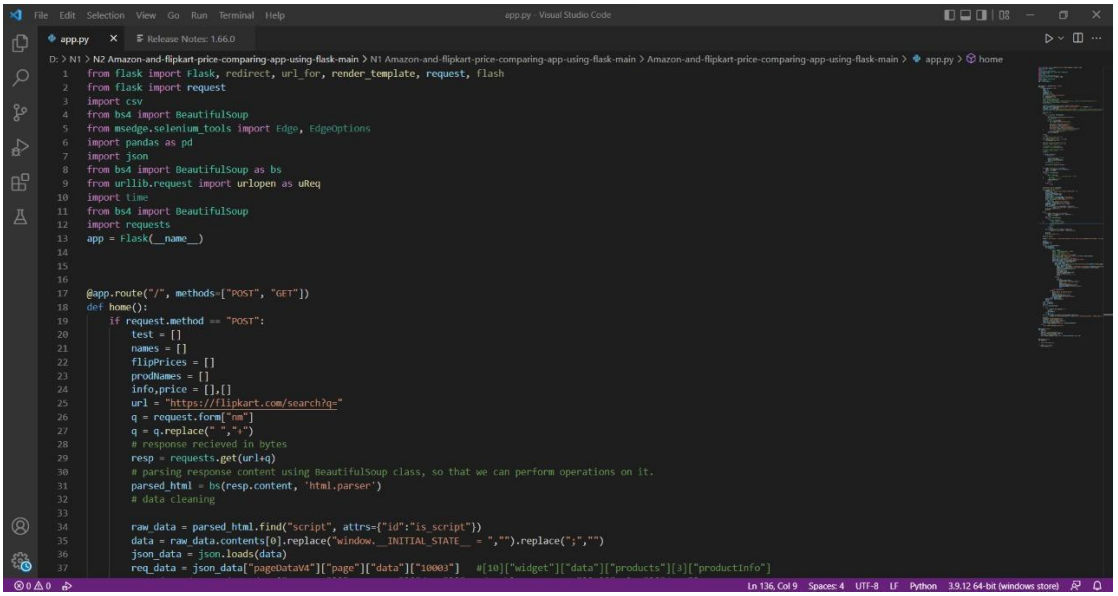


Fig. 5.4.1

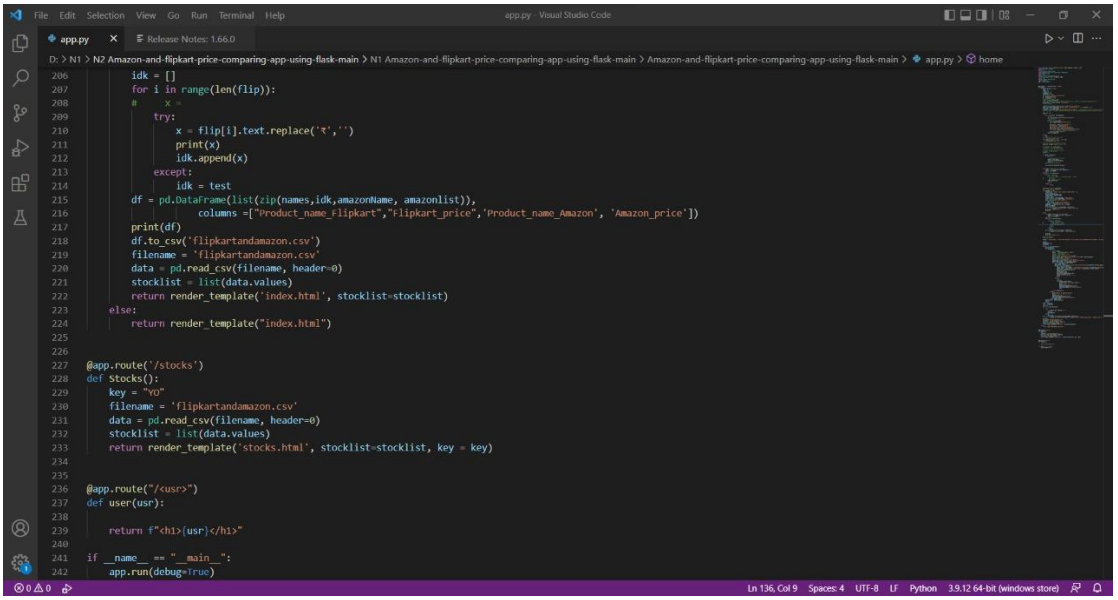
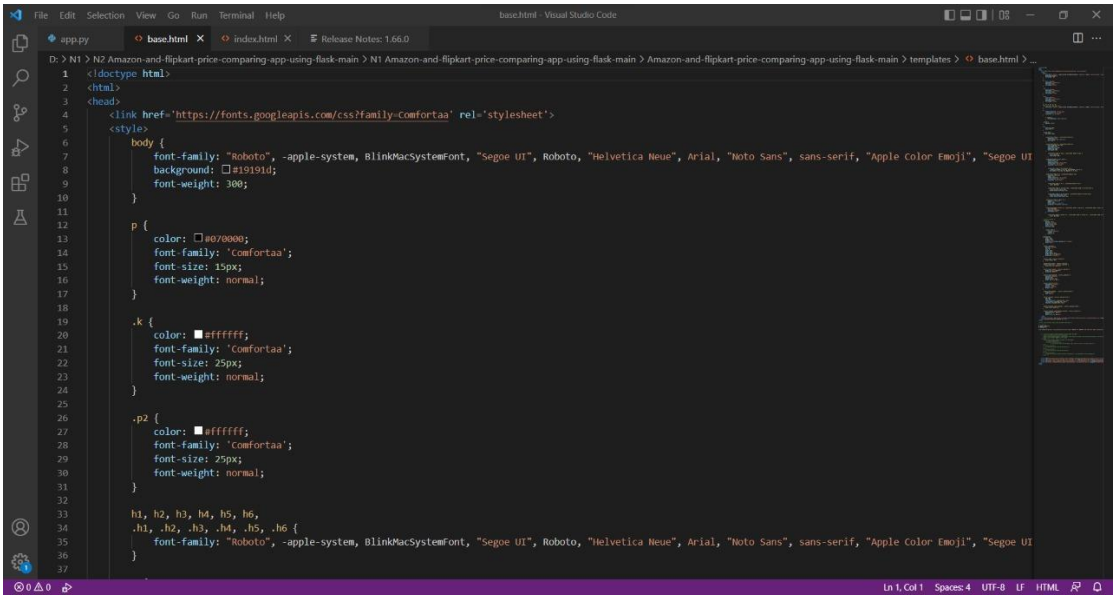


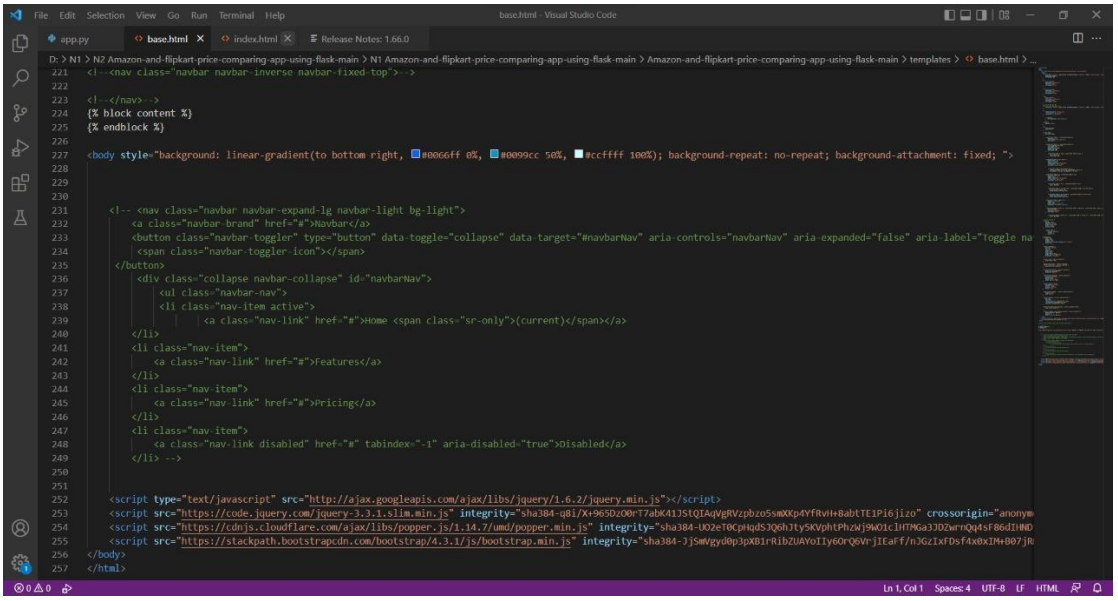
Fig. 5.4.2

base.html



```
<!doctype html>
<html>
<head>
    <link href='https://fonts.googleapis.com/css?family=Comfortaa' rel='stylesheet'>
    <style>
        body {
            font-family: "Roboto", -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif, "Apple Color Emoji", "Segoe UI
            background: #19191d;
            font-weight: 300;
        }

        p {
            color: #070000;
            font-family: 'Comfortaa';
            font-size: 15px;
            font-weight: normal;
        }

        .k {
            color: #ffffff;
            font-family: 'Comfortaa';
            font-size: 25px;
            font-weight: normal;
        }

        .p2 {
            color: #ffffff;
            font-family: 'Comfortaa';
            font-size: 25px;
            font-weight: normal;
        }

        h1, h2, h3, h4, h5, h6,
        .h1, .h2, .h3, .h4, .h5, .h6 {
            font-family: "Roboto", -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif, "Apple Color Emoji", "Segoe UI
        }
```

Fig. 5.4.3



```
<!--<nav class="navbar navbar-inverse navbar-fixed-top">-->

<!--</nav>-->
{% block content %}
{% endblock %}

<body style="background: linear-gradient(to bottom right, #0066ff 0%, #0099cc 50%, #ccffff 100%); background-repeat: no-repeat; background-attachment: fixed; ">



<!-- <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle na
      <span class="navbar-toggler-icon"></span>
  </button>
    <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
        <li class="nav-item active">
            <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="#">Features</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="#">Pricing</a>
    </li>
    <li class="nav-item">
        <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>
    </li> -->


<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js"></script>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonym
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHND
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jR
</body>
</html>
```

Fig. 5.4.4

index.html

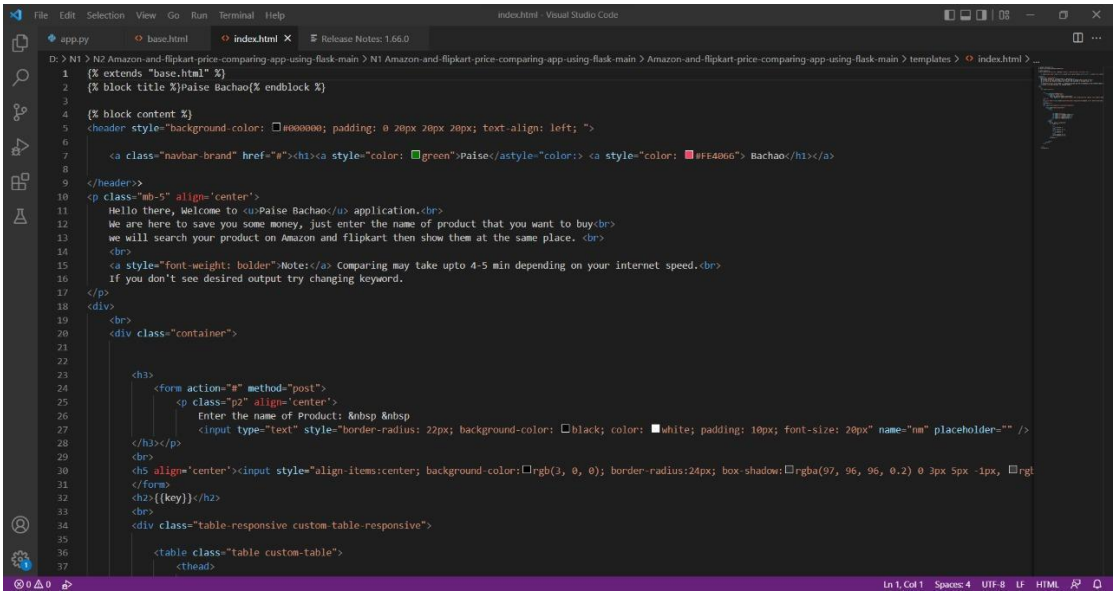


Fig. 5.4.5

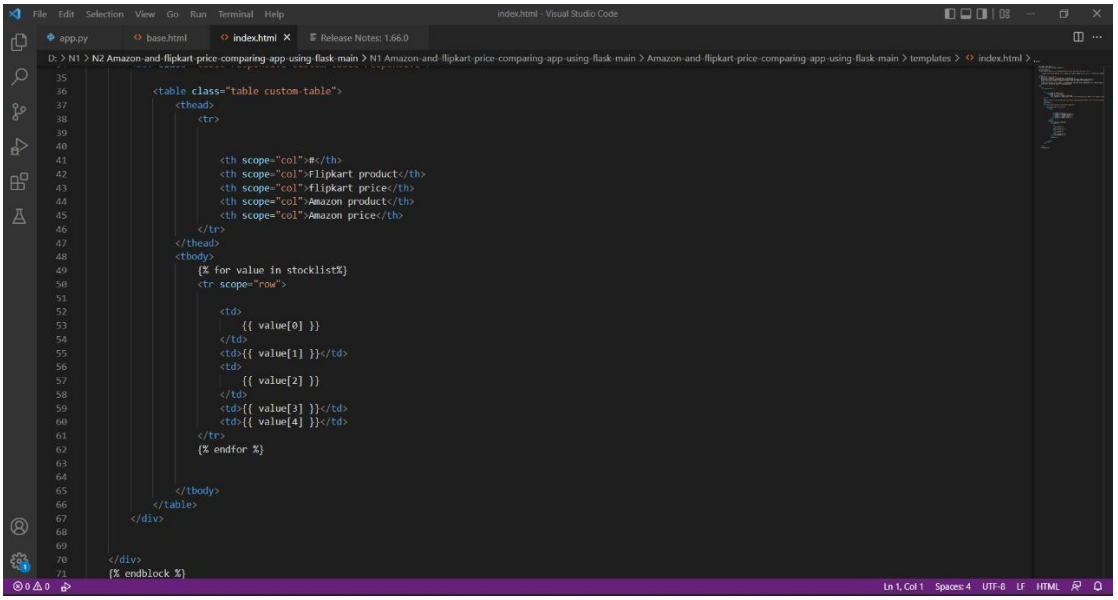


Fig. 5.4.6

CHAPTER 6

LIMITATIONS & FUTURE SCOPE

6.1 Limitations:-

➔ It does not record activity of the user yet.
➔ Sometimes due to technical problems it takes a lot of time to load data.
➔ Direct URLs of the products shown, are not provided yet.
➔ GUI does not have many features.
➔ A little change in the keyword might cause a huge difference into the results.

6.2 Future Scope:-

➔ It is highly scalable.
➔ If the activity of the user is recorded and used with AI, then it could also provide valuable opinions and suggestions to the user about a certain product.
➔ GUI could be expanded and also add some more features such as sorting on the basis of different features of the product.

CHAPTER 7

CONCLUSION

7. Conclusion

With the help of this project, the burden on the customers' shoulders is reduced by finding the best deal from the available deals across all ecommerce platforms by extracting product data for comparison from Amazon and Flipkart ecommerce websites.

In addition, the collected data is depicted using a very streamlined and immersive GUI.

Further, it is time efficient as it only shows pivotal features which need to be known while making the decision whether to purchase it or not.

Hence, it helps a lot, especially, to the customers who surf a lot to find a 'value for money' product. Moreover, the decision making process is made a lot easier with the help of our project, as only features which influence the decision the most are shown compared to the other websites which depict a lot of unnecessary features.

CHAPTER 8

BIBLIOGRAPHY

8. Bibliography

- https://electroneek.com/blog/rpa/rpa-101-what-is-web-scraper/
- https://www.zyte.com/learn/difference-between-web-scraping-and-web-crawling/
- https://webscraper.io/blog/brief-history-of-web-scraping
- https://en.wikipedia.org/wiki/Web_scraping