IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Price Point- Your One Step Solution

[1] Harishma KV, [2] Dr. Suresha D, [3]Sudheesh KP,[4] Pooja Hansda , [5] Priya M, [6] Radhika D, [7] Sharadhi S Shetty

[1] Assistant Professor, [2]Professor, , [3] Assistant Professor, , [4]Research Scholar, [5]Research Scholar, [6]Research Scholar, [7]Research Scholar

Department of Computer Science and Engineering

Srinivas Institute of Technology Mangalore

Abstract- Price Point is a website that comparesprices. It helps people find good deals at low prices.The site makes it simple to look at different products side by side. It grabs up-to-date pricing info from many online shops. This lets shoppersmake smart choices fast. The site uses flaskframework and connects to APIs to keep an eye on prices from lots of pages.

Keyword: Flask, Beautiful soup.

I. INTRODUCTION

A price comparison website is website where people can look at prices from different stores. This helpspeople save some cash and time and shows what products are there in the market. For stores, it is a way to get more people to see their products and sell more. People can look for specific things they want or just browse through different types of products. Thewebsite shows prices from many known shopping platforms for the same item.A functional specification document (FSD) plays a key role in systems engineering and softwaredevelopment. This document outlines the functions a system or component needs to perform. It serves as a more technical response to a corresponding requirements document. The purpose of a functionalspecification is not to explain how the proposed system works. It does not include details on how to implement

the system's functions. Instead, it concentrates on what different external agents might "see" when they interactwith the system. These agents could be people using theprogram, computer peripherals, or other computers.

In the digital age, consumers are inundated with a vast array of products and retailers online. Thisabundance of options often makes it challenging for consumers to make informed purchasing decisions quickly and efficiently. Despite the availability of numerous online shopping platforms, users face the problem of price discrepancies, lack of comprehensive product information, and difficulty in comparing

products, a price comparison website helps you find good deals on stuff you want to buy. It does this by showing prices from different online stores all in one place. You can search for what you are looking for andsee a list of options ranked by price or how popular they are. The main point is to make shopping easier andshow you all your choices without having to visit a bunch of different websites. These sites often haveextra features too, like price drops and providing more info about products. These websites use smart computer programs to keep their prices up to date, so latest deals when you look different retailers.

II. RELATED WORK

Python price comparison website project in Visual Studio, it requires few key steps. First, open Visual Studio and start a new project. Head to File > New > Project, select Python from the options, and pick atemplate that suits the needs such as Flask or Django. This lays the groundwork for the app.

Once the project is created, a virtual environment is set up, to manage dependencies. To do this, find the project in Solution Explorer by right-clicking it, select Python > Create Virtual Environment, and then by following the prompts to make a separate environment.This helps to avoid package conflicts and ensures project's dependencies are specific to it.

Now that the virtual environment is set up, we need to get the right packages in there. It can be done with two ways: using Visual Studio's Package Manager, which comes with it, or by creating and keeping an eye on a requirements.txt file. Setting up Visual Studio for a Python price comparison website needs a few key steps. First, we make a virtual environment. This keeps the project separate from another Python. It can turnthis on in Visual Studio to handle specific libraries and packages. Next, to deal with environment variables. Wecan make a .env file to keep important info like databaseURLs and secret keys. We can use libraries to

load these into the app. Then we can set up the app settings. Then put important config files. While using Flask, it might have a config.py file for the secret key and database URL. For Django projects, we will change settings.py to include database setups and other project-specific things, can set up logging. This helps to keep an eye on how the app is doing and catch errors. The use of file handlers to manage where the logs go and make sure problems get recorded. The project's structure shows how the system is set up and how its parts work together. It gives a big-picture view of how the system is designed and organized pointing out the main pieces and how they are connected. The structure makes sure the project can grow is easy to fix and does agood job meeting its goals. You can split the "Price Comparison using Flasks and Beautiful Soup" project's structure into a few parts

User Interface: This part gives users an easy-to-use screen to type in product names, start comparing prices, and see what they find. It talks to other parts to get whatusers put in and show them what comes out.

Web Scraping Modules: These modules have the jobof pulling data from different online shopping sites. Theproject uses web scraping methods to gather info about what products are called and how much they cost from places like Amazon, Flipkart, Myntra and so on. Each website needs its own special web scraping plan to figure out how the site is set up to ask for stuff and grabthe important details.

Setting up the database plays a crucial role too. We need to stick connection strings or URLs in config files and use stuff like Flask-Migrate or Django's migration thing to deal with changes in the database setup as time goes on. To make coding easier, we set up debugging options in Visual Studio. This allows us to add breakpoints, check out variables, and walk through the code bit by bit. It is also essential to link up withversion control systems like Git. This helps to track changes and team up with others. All these setups cometogether to create an environment.

Install Python on the computer. It's best to use the newest stable Python version (like Python 3.8 or a newer one).

Integrated Development Environment (IDE): An IDE can make coding a lot easier. Some popular ones are PyCharm, Visual Studio Code, or Jupyter Notebook.These tools help with s like writing code, fixing bugs, and keeping track of changes. Just make sure the IDE you pick works with your computer's operating system'

III. METHODOLOGY

Web Browsers: The price comparison system might need web browsers to show the GUI or check out product links for more info. We should put common browsers like Google Chrome, Firefox, or Edge on the computer.

If a price comparison website involves stuff happening in the browser (like running JavaScript or showing changing content), we must make sure the site works right on different browsers. Some tools to test andfix problems are:

Browser Developer Tools: New browsers have tools built right in. These let us look at HTML, CSS, and JavaScript, fix scripts when they're not working, and seehow well the site is running.

Cross-Browser Testing Tools: By using Browser Stack or Sauce Labs to check how the website looks on different browsers and gadgets.

The initial step involves installing Visual Studio andconfiguring it for Python development. This setup ensures that all necessary tools and libraries are available. During the installation of Visual Studio, it is crucial to select the Python development workload. Additionally, installing libraries such as requests, beautifulsoup4, pandas, and matplotlib through the terminal in Visual Studio will be required for web scraping, data manipulation, and visualization.

Initiating the Python Project

After the setting up of the coding space, it is timeto make a new Python project in Visual Studio. This gives a place to write and sort out all the code we will need for the price comparison tool. Use pip in the built-in terminal to get things going. Then, make folders for different parts, like web scraping script store data, and Flask spots for static files and templates. Write Python scripts for each job: a main script to run things specific scripts to scrape different online stores, and helper functions for common tasks. Start writing code to set up main functions and get thescraping and data sorting working. Use pandas to handle and look at data.

Use Visual Studio's tools to find and fix problemsby setting breakpoints and running your project. Adding a webpage that people can use to set up Flask to deal with what users type in and show them results.Make HTML templates and static files so it's easy for people to use. This whole setup gives you a strong baseto build and improve your tool. To kick things off openVisual Studio and make a new project by picking the Python application template. This sets up a workspacewhere we can write and organize all the code. Once the project is made, set up a virtual environment to handle dependencies. This makes sure to have a clean slate to start coding.The main job of the tool is to grab price info from different online shops. This means it sends requests to websites, gets their HTML, and uses something like Beautiful Soup to find and pull out the prices. By making special functions for each website to make sure they're getting the right prices.

Getting info from websites involves asking for theirHTML getting it back, and then looking through it to find what we want, like prices. This is important whenwe are trying to compare prices from a bunch of different places all at once.

Each online store's website is built, so we need to make separate functions to scrape each one. These functions should do a few things:

- Send a request to the website's search pageusing the product name to search for.
- Get the HTML content from the search resultspage.
- Go through the HTML to find and pull out theprice information

. To make a user-friendly interface for the price comparison tool, we build an HTML/CSS frontend. Thismeans designing a webpage where people can type in the product they want to check out. The interface shouldbe simple and easy to use, so folks can interact with thetool without any trouble. HTML gives the webpage its structure, while CSS makes it look good and work well.To link the Python backend with the HTML/CSS frontend, we can use a web framework like Flask. Flaskmakes it easy to route and serve the webpage, and to handle forms submissions and returning the price comparison results. The Flask application will handle user input from the web page and performs the necessary web, use HTML to create the structure.

The Flask app will handle stuff like, users type on the web page, do the web scraping and analysis needed, and send price comparisons back to show on the page. Use HTML to build the bones and meat of the web pages. Visual Studio helps in writing HTML code with smart suggestions and colored highlights. Key parts include:

- **Headers and Footers**: Make headers and footers so people can find their way around andsee the brand everywhere.
- **Forms**: Build forms to get info from users, like what products they want to compare prices for.
- **Containers**: Use div things to make sections and boxes to organize content.

Websites that allow it are scraped to get product info. We cannot access sites that block scraping without permission. The basic idea of scraping is the same, but it changes based on the site's source file. To do this, we make a URL to search the site using the product name (from object recognition or user input). Then we grab and process the URL's source file to getthe important product details. To get these details, we first figure out the pattern of the page source. The HTML tags with product info like name, seller, and price have IDs. These IDs are the same for all productson the page. Each detail has a different ID. Using theseIDs, we find where the info is in the source and get thetext near the ID. This is how we get all the product details. We do this for all products in the search results.The stuff we pull from the source file includes a list ofproducts with their names, sellers, prices, and links to buy them.

Different parts and steps team up to give us the fullpicture with spot-on results. It all kicks off with data collection. This means grabbing info on products and what they cost from a bunch of online shops and otherspots. Folks can snag this data by scraping websites, tapping into APIs, teaming up with sellers, or using outside data gatherers. Once they have got the data, they need to make it all line up. This involves getting product descriptions, groups, and other details to match.

Making a tool to compare prices with Python in Visual Studio is not rocket science. First, we have to get the Visual Studio Code and the Python add-on set up. Make sure Python is in the system PATH too. Next, create a virtual environment in VS Code by typing python -m venv myenv and turn it on.Installing some libraries like requests, beautifulsoup4, and pandas using pip. Start the script by importing these libraries. Then, write some functions to grab datafrom different online shops. Use requests to get web pages and Beautiful Soup to make sense of the HTML stuff. Pull out important info about products like names, prices, and web addresses. Clean up this data and put it in a pandas Data Frame so it's easy to work with and compare. Add features to sort and filter products based on things like price and how good people think they are., show the results in a way that's easy for people to understand, like a table or CSV file.This straightforward method mixes grabbing data
working with it and showing it off to give you a tool that works well for comparing prices.

To get data, make functions that use requests to grabweb pages and Beautiful Soup to break down the HTML stuff pulling out important product info like names, prices, and links. This info needs to be cleaned up, so it is all the same and put into a pandas Data Frame. HTML sets up how everything looks making a neat layout to show product details. CSS makes it all pretty and easy to use. Putting HTML and CSS togethermeans people can use the tool without much trouble sorting products by stuff like how much they cost or what others think of them.

Python for backend stuff and web scraping, along with HTML/CSS for the front end, helps make a reliable and productive price comparison tool. This toolgrabs and handles info from different online shops and shows it in an easy-to-use and nice-looking way making things better for users.

An API lets different apps talk to each other and share info. For example, the e-commerce API grabs product details (like pictures and costs) from shopping websites. It gets the prices and images of items. Withinthe ecommerce web site can so be mirrored within the internet application.

IV. EXPERIMENTAL RESULTS

Using Python in Visual Studio to make a price comparison tool showed some cool stuff about how wellit works. The tool uses Python libraries like requeststo grab info from websites and Beautiful Soup to make sense of the HTML. It pulls together pricing info from different online shops. The panda's library helps sort through all the data, so users can compare products by looking at things like price and ratings. Thisgives people clear info they can use to make decisions.

The tool shows it can handle big amounts of data well doing comparisons fast and right. It makes product info the same across different places, which helps make the comparisons more trustworthy. The way HTML andCSS are used for the front part shows the data in a way that's easy for people to use. This lets users look through, filter, and sort the info how they want to. The tool has an impact on how and it can work with lots of data. It also has an influence on making product details match up, no matter where they come from. This leadsto more reliable comparisons. Using HTML and CSS to create the part users see helps to present the data in a user-friendly way. So, people can find what they need, pick out certain things, and put the info in order.

Making a website to compare prices with Python inVisual Studio takes a few steps. Here's a guide to help you do this. It covers how to design the system, set up the project, create the database, design the API, test it, and put it online.

- **Find Products:** People should be able tosearch for stuff they want to buy.
- **Check Prices:** Look at how much the samething costs at different stores.
- **All-in-One Setup:** Works well for small tomedium projects. Everything is in one place.
- **Separate Services Setup:** Better for big trickyprojects. Each part does its own thing and talksto others over the internet.

API Design is about making RESTful services that let you talk to the app. You set up paths like /products to get product info, /prices for price details, and /login for users to sign in. To keep important stuffand actions safe, you use secure ways to check who is who, like systems that use tokens.

Testing and Debugging are super important to make sure the app works right. Run unit tests to checkif each part works on its own, and integration tests to see if all the pieces fit together like they should. Visual Studio has tools to help in debugging. These letus put stopping points in the code, look at what is in thevariables, and go through the code step by step to find and fix problems.

Configuring project settings has an impact on how you set environment variables and tweak configurations for your specific development setup. This might include stuff like database connection strings secret keys, and other important settings tomake the app work right and keep it secure.

After that, we should get version control going. Visual Studio works well with Git, so we can start a Gitrepository in your project, make your first commits, and handle different branches. This is key to keeptrack of changes and work together with other people.

V. CONCLUSION

To wrap up comparing prices using web-scraping and data analysis helps people and companies buy stuffsmarter. By grabbing info from lots of places and looking at it, we can spot pricing patterns see howdifferent websites stack up and snag the best bargains. We have talked about how to do it, what software to use, and what each part does. We have also shown whatpeople might get, like tables that compare things. Looking ahead, there is even more we can do with price comparison. We could use machine learning, team up with online stores, look at new stuff to buy, check out different markets, and work with stores and manufacturers. In the end, comparing prices this way issuper helpful if you want to save money or find good deals. By using data tricks and scraping, you can make smarter choices when you shop and keep more cash in your pocket over time.

PricePoint website compares prices, byfollowing a step-by-step plan to build a strong and working app. First, we need to do a deep dive into what is needed. This means figuring out what the site should do - like letting people search for stuff, compare prices, log in, and get alerts. You also need tothink about how well it'll work, if it can grow, if it's safe, and if it won't crash.

With the help of this project people can find online stuff from different websites and view the live prices at the same time over the websites. Helping people to gain more better experience in the field of finding the online products for better prices and to their comfort at one site.

REFERENCES

[1]. Henry C. B. Chan, Chris K. H. Chin and BenjaminLam, “Price-comparison agents for MAGICS*”, Hong Kong Polytechnic University, Hong Kong IEEE 2001 https://ieeexplore.ieee.org/abstract/document/95373 9/ similar/.

[2]. Grant, K.B (2008) [retrieved on June 20, 2012] – “Which Price Comparison Shopping is The Best?” at http://www.smartmoney.com/spend/familymoney/whic hc/.

[3]. Guowei Chen and Peng Zhou Zhang, “The contentextraction method of webpage information based on knowledge base”, Fifth International Joint Conference on Computational Sciences and Optimization, https://ieeexplore.ieee.org/document/6274803/,IEEE 2012.

[6]. Smith, J., Johnson, A., & Williams, R. (2023). Price Comparison of Consumer Products Using Web Scraping: A Case Study. Journal of Data Analysis and Information Science, 10(3), 123-140.

[7]. J. Thakkar, "Data Analysis of E-Commerce Websites Using Python, “International Journal of Research in Engineering, Science and Management, vol. 3, no. 7, pp. 132-138, July 2020.

[8]. N. Pandey, R. Singh, S. Arora and P. Kumar, "Price Comparison of Products in E-Commerce Using Web Scraping and Data Analytics Techniques," 2019 2nd International Conference on Computing, Communication, Control and Automation (ICCUBEA), Greater Noida, India, 2019, pp. 1-6.

[9]. S. D. N. V. K. Kiran, R. K. Bolla and M. P. S.
Rao, "Web Scraping Techniques and Applications for Data Analysis and Decision Making," Journal of Big Data, vol. 7, no. 1, pp. 1-32, Dec. 2020