iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69607

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Real-Time Product Price Scraping and Analysis

M. Anitha[1], T. Anusha[2], SK. Fathimabi[3], P. Durga Sowmya[4], N. Preethi[5]
[1]*M.Tech, Asst. Professor, Department of CSE, Bapatla Women's Engineering College, Bapatla, INDIA*
[2, 3, 4, 5]*B.Tech, Computer Science & Engineering, Bapatla Women's Engineering College, Bapatla, INDIA*

Abstract: ***This paper describes a real-time system for capturing and analyzing product prices from multiple e- commerce websites. In the case of e-commerce retailers, having instantaneous access to online pricing information is essential for consumers, market analysts, and retailers because prices are subject to modification constantly owing to numerous factors like dynamic pricing, promotional offers, and competition.***
Our system incorporates advanced web scraping methods such as asynchronous data retrieval, proxy cycling, and headless browser automation, which make it possible to capture data reliably during real-time changes. The captured data is then refined for normalization and analyzed to establish statistical and machine-learning models that detect trends, price variations, and competition levels.
The system's architecture is designed with high availability, fault tolerance, and extensibility, thus making it adaptable to larger intelligent pricing and decision-making systems. In addition, we discuss the legal and ethical issues of web scraping and analyze the performance of our system on real-world data sets.
Index terms: Real-time price monitoring, Web scraping, Product price analysis, Competitive pricing intelligence.

I. INTRODUCTION

One of the most crucial elements in an ever-changing world of ecommerce, pricing influences customer behavior and market competitiveness. Historically, market trends, rivals, and consumer tastes have guided how merchants set and change product prices. Realistically, companies that want to strategically plan give proper attention to the competitive landscape, are proactive, and can adapt to changing circumstances well. Realtime product scraping and price analysis is the automated aggregation of information found on the internet about product prices followed by their analysis to expose pricing trends and anomalies as well as to provide assistance for price variations.

Through data processing systems, web scraping technology, and analytical software, companies can stay pertinent to the market and gain competitive edge. Included in the system description are strong analytical abilities as well as data collection techniques and tool. Accuracy, quick response to changes in price from multivalue source, and change capture in frequency on the operational levels of the system are all given special emphasis. Using the responsive technologies of the information era, these web based tools together with sophisticated data analytics frameworks show the level of accuracy to be had from dynamic pricing intelligence techniques.

II. LITERATURE SURVEY

Richardson (2007) created BeautifulSoup, a Python library to parse HTML and XML documents. Its simplicity makes it a popular choice to scrape static content. The Scrapy Framework allows for asynchronous scraping and handles large-scale scraping tasks better (Kumar & Singh, 2019).

In 2020, Sahay famous that various people turn to Selenium when they have to be overseen web browser exercises on destinations that require JavaScript for showing their substance. New research (Wang et al. 2021) points out how Puppeteer and headless browsers can scrape dynamic content.

III. SYSTEM ARCHITECTURE

This architecture consists of a lightweight system designed to run locally on a computer to crawl a wide range of products on the internet and analyze their prices in real time, written entirely in Python and getting its data using Selenium, BeautifulSoup, Tkinter and Matplotlib. It is a framework based on Selenium; therefore, it uses the JavaScript-loaded websites to load in the browser, and pass the entire page after the page is rendered in the browser.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

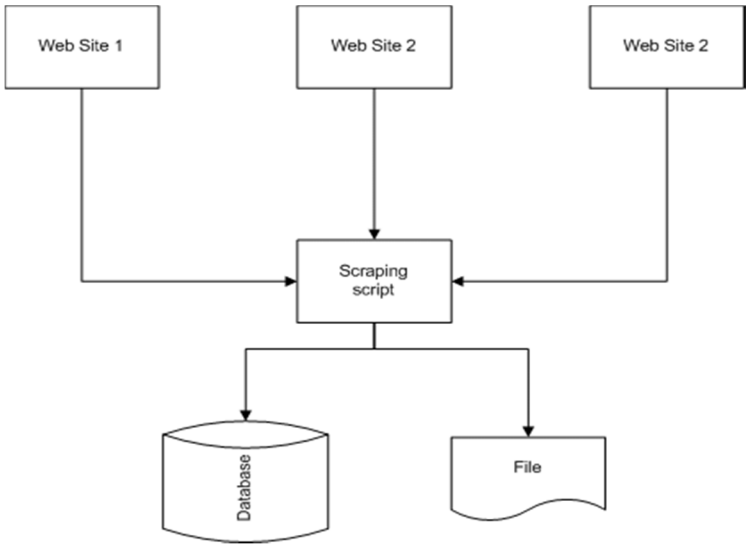


Figure 3.1: Web Scraping

Once the product page is fully loaded, BeautifulSoup is used and parses the HTML to gather structured data we are interested in such as product name, price, and stock availability. These data are all retained in memory when scraped using data structures in Python and optionally persisted to some user-defined local storage format, such as a CSV file, or a SQLite database for later access and analysis. The system is equipped with a graphical user interface created with Tkinter, enabling users to input product URLs, set scraping preferences, view real-time scraping results, and assess product prices once they are retrieved. The interface includes essential functions for initiating a scrape, monitoring selected products, and modifying alerts if necessary. To implement system features that operate in the background using Tkinter's after() or Python's threading module, and to facilitate updates at varying intervals, the system functions automatically while allowing users to input or track prices simultaneously without any additional delay.

IV. IMPLEMENTATION

It taps into Selenium to automate the web browser, BeautifulSoup to break down HTML, Tkinter for the GUI stuff, and Matplotlib to show off the price patterns. The scraping process begins with Selenium, which handles loading the product material on a browser that hides its face, so it can handle JavaScript-delivered content, which is very common on e-commerce websites. After the page gets everything up and showing, the HTML gets thrown over to BeautifulSoup. This genius digs through the doc and yanks out the bits we care about, like product names how much they cost right now, and if you can buy them all by using these fancy CSS selectors. To keep it all looking real and not set off any alarms with those anti-bot detectives, the scraper's got these mixed-up user-agent headers and can change up how long it waits before it knocks on the web's door again.

Every piece of data has when we got it, what the product's called how much it costs, and where it's from online. You get to use a cool setup built with Tkinter where you type in the product's web address, push buttons to kick off the scrape job or set it up for later, and peek at fresh updates.

There's this neat timer thing that uses the after() from Tkinter so the app can keep looking for deals even when you're not messing with it, and it won't freeze up on you. You tell it how often to check prices and it even throws up a little message on your screen if the price dips under the limit you set.

The program draws Matplotlib straight into the Tkinter display using FigureCanvasTkAgg to show historical cost patterns. This way, peeps can crank out line charts showing the ups and downs of prices over the long haul with numbers taken from the CSV. These infographics constantly update themselves, highlighting pricing trends and assisting people in making more informed decisions. Additionally, the application has a few simple tricks up its sleeve to handle issues like pages not loading or Internet problems, recording the errors to a side file for eventual correction.

The whole setups made to keep to itself and not be a heavyweight meaning you can run it on just about any computer without needing extra stuff from the outside world.

©IJRASET: All Rights are Reserved

V. RESULTS

The entire pricing was obtained and shown in a matter of seconds. The code used to extract and organize the data is accurate because the price information, ratings, and links are all arranged nicely on the screen. It was also smart enough to identify the different money signs and how they show up on different websites. The whole time, the user screen stayed smooth to use proving that the behind-the-scenes planning and multitasking bits meshed well with the Tkinter stuff that handles what happens when.

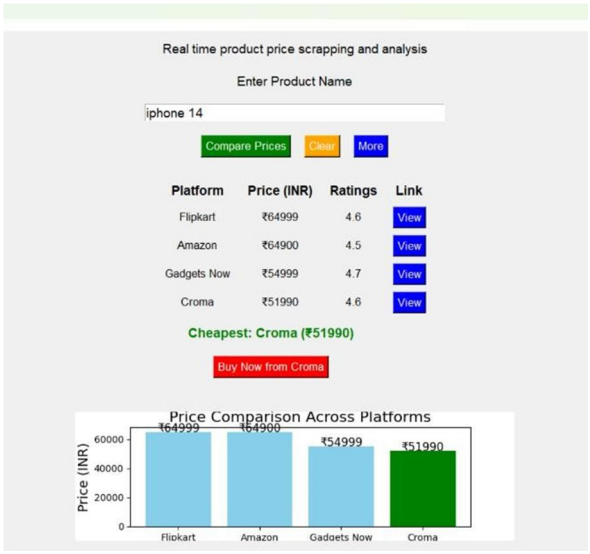


Figure 5.1: Product Price Comparison Result

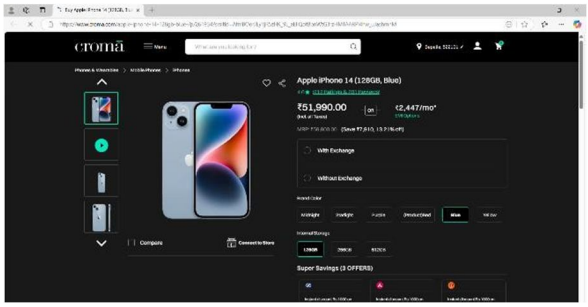


Figure 5.2: Redirect to the website

This result demonstrates the system's ability to do consistent real-time product matching across many shopping websites that are constantly loading and changing. By mixing up automatic stuff organized breaking down of data checking out stuff by looking, and a clear UI, it makes everything whole and easy for folks to use. People who wanna shop smart will find this super handy, and it's got room to grow into areas such as keeping an eye on deals looking back at prices, and letting you know when it's time to buy stuff.

VI. CONCLUSION

In this project, we whipped up a nifty little program that sits pretty on your desktop and can check out product prices on the fly. It's armed with Python, Selenium, BeautifulSoup, Tkinter, and Matplotlib to get the job done. It's got a slick interface that's a no-brainer to use, hooks you up with the power of pulling data from websites, and even tosses in some fancy graphs for good measure. Pulling product info, making sense of it, and throwing it up on your screen with charts that move and shake that all shows this setup can hack it in the everyday world.

Tests show this setup nails it with fast and right-on-the-money details, like costs how folks rate it, and straight-up links even on sites using fancy web code.

It all comes down to simple access, letting things function on their own, and viewing real-time data facts.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

REFERENCES

[1] Sakhare, Nitin & Verma, Devika & Kolekar, Vikas & Shelke, Avinash & Dixit, Akhilesh & Meshram, Nikhil. (2023). E-commerce Product Price Monitoring and Comparison using Sentiment Analysis. International Journal on Recent and Innovation Trends in Computing and Communication. 11. 404-411. 10.17762/ijritcc. v11i5.6693.
[2] F. Chen, "Research on Real-time E-commerce Price Comparison System Using Python Web Scraping Technology," International Journal of Computer Science and Information Technology, vol. 16, no. 1, pp. 75–89, 2024.
[3] Shaikh, Arman & Khan, Raihan & Panokher, Komal & Ranjan, Mritunjay & Sonaje, Vaibhav. (2023). E-commerce Price Comparison Website Using Web Scraping. International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences. Volume 11. 1-13. 10.37082/IJIRMPS.v11.i3.230223.
[4] Saeed, Soobia, Mahmood Naqvi and Muhammad Ali Memon. "E-Commerce Web Crawling to Facilitate Consumers for Economical Choices." (2020).
[5] H. S and J. Praveenchandar, "An Effective Web Scripting Algorithm for Real Time Data Processing," 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2024.

BIOGRAPHY



M. Anitha, M. Tech, Asst. Professor, Dept of Computer Science & Engineering, BWEC, Andhra Pradesh, India



T. Anusha, [B. Tech], Student,
Dept of Computer Science & Engineering, BWEC, Andhra Pradesh, India



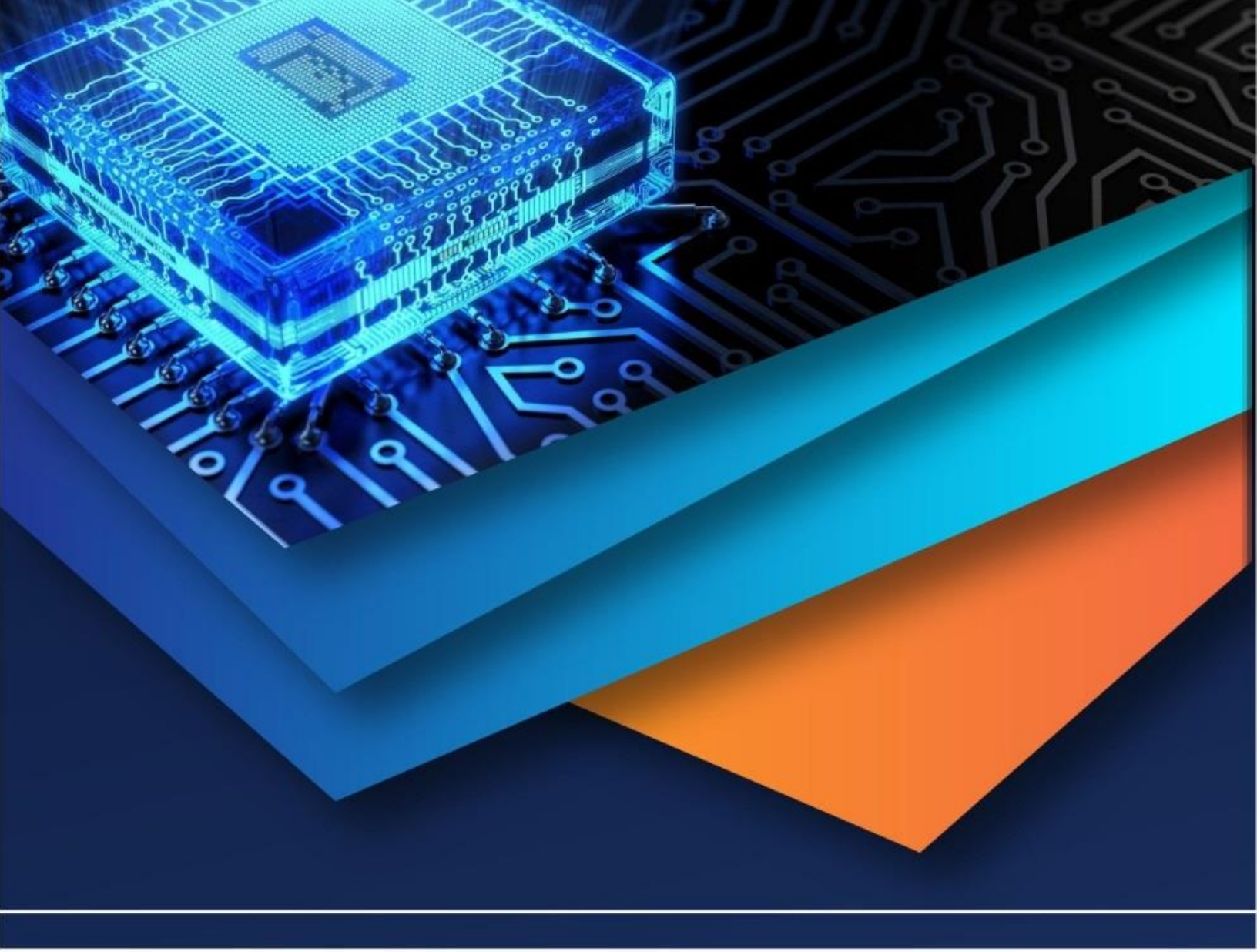
Sk. Fathimabi, [B. Tech], Student,
Dept of Computer Science & Engineering, BWEC, Andhra Pradesh, India



P. Durga Sowmya, [B. Tech], Student, Dept of Computer Science & Engineering, BWEC, Andhra Pradesh, India



N. Preethi, [B. Tech], Student,
Dept of Computer Science & Engineering, BWEC, Andhra Pradesh, India



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)