Archives available at journals.mriindia.com

International Journal on Advanced Electrical and Computer Engineering

ISSN: 2349-9338
Volume 14 Issue 01, 2025

E-commerce Data Scraper Using Selenium and Beautiful Soup

Dr. Kalpana Malpe[1], Mr. Nagesh M. Kamble[2], Mr. Nandu P. Waghmare[3], Mr. Amit M. Madavi[4], Mr. Harsh M. Damke[5]

[1]Head of Department, Department of Computer Engineering, Suryodaya College of Engineering and Technology (SCET), Nagpur, Maharashtra, India
[2345]UG Student, Department of Computer Engineering, Suryodaya College of Engineering and Technology (SCET), Nagpur, Maharashtra, India
[1]hod_ce@scetngp.edu.in, [2]nageshkamble122333sur@gmail.com, [3]nanduwaghmare310@gmail.com, [4]madaviamit075@gmail.com, [5]harshdamke123@gmail.com

Peer Review Information

Submission: 07 Feb 2025
Revision: 16 Mar 2025
Acceptance: 18 April 2025

Keywords

E-Commerce
Web Scraping
Price Comparison
Automation
Selenium

Abstract

The E-commerce Scrapper automates price comparison across Amazon and Flipkart via a Telegram bot and web platform, retrieving real-time prices, ratings, and purchase links using web scraping techniques. This system eliminates manual price checks, saving users time and ensuring they get the best deals available at any moment.

Built with Python libraries like python-telegram-bot, aiohttp, and beautifulsoup4, it extracts and processes data efficiently, offering sorting, navigation, and CSV storage. The comparison engine matches similar products, highlights the best-priced option, and allows users to explore product details directly on their preferred platform. The web version extends accessibility, allowing multi-device price comparisons beyond Telegram, making it easier for users to shop across different devices. Despite anti-scraping challenges and website structure dependencies, the system remains scalable and adaptable through continuous updates and refinements. Future enhancements include API integration, AI-driven recommendations, price tracking alerts, and expanded support for more e-commerce platforms. Machine learning will further refine product matching and optimize shopping suggestions based on user behavior and preferences. By automating price comparison and providing multi-platform access, the bot and website enhance convenience, making online shopping smarter, more efficient, and cost-effective for consumers and businesses alike.

INTRODUCTION

Consumer shopping behavior has been changed by online shopping, and online shopping platforms such as Amazon and Flipkart provide vast product varieties. Prices, however, vary on websites, and manually comparing prices is time-consuming and inefficient. Differences in prices, discounts, and promotions are difficult for consumers to track, and thus poor purchase decisions are made. The traditional methods involve customers making several trips to websites of e-commerce companies, searching manually for a product, and comparing prices, which is error-prone and time-consuming.

This study presents an automated price comparison system that employs web scraping techniques combined with a Telegram bot to offer real-time information regarding products. The central aim is to improve the decision-making process through provision of a well-organized, easy-to-use platform that effectively compares the prices of products. The system captures, processes, and organizes pricing information, thereby enabling users to make better decisions with less effort.

Web Scraping is the automation of the process of extracting data from websites. It can be done by copy-pasting the data, which is time-consuming and labor-intensive manually, So This process is done with the help of web scraping software called web scrapers. They load and extract data automatically from the websites according to the requirements of the user. They can be set to work on one site or can be set to work with any site.

Information can be accessed from websites using two main methodologies: the manual extraction methodology and the automated methodology.

- Manual extraction techniques: Copy-pasting the site content manually comes under this category. Though time-consuming, tiresome and monotonous it is a good technique to scrape data from the sites having good anti-scraping policies like bot detection.Automated Extraction Methods: Web scraping tools are employed to obtain data automatically from websites based on the user's requirements.
- HTML Parsing: Parsing means to make something understandable to be analyzing it part by part. To wit, it means to convert the information in one form to another form that is easy to that is easier to work on with. HTML parsing means taking in the code and extracting relevant information from it based on the user requirement. Mainly executed using JavaScript, the target as the name suggests are HTML pages.
- DOM Parsing: The official recommendation of the World Wide Web Consortium is the Document Object Model. It defines an interface by which users may alter and modify the style, structure, and content of an XML document. Web Scraping Software: Nowadays, many web scraping tools are available or are custom build on users need to extract required desiring information from millions of websites.

LITERATURE SURVEY

A number of studies point to the application of web scraping and automation in e-commerce. Automated product monitoring studies indicate that Python libraries such as BeautifulSoup and Scrapy are effective tools for scraping structured data from the web. E-commerce analytics studies indicate that prices change on a daily basis, and real-time monitoring and automated data collection systems are needed to facilitate accurate comparisons.

Earlier research has also addressed issues such as anti-scraping methods, website reorganization, and ethical issues. Websites tend to employ CAPTCHAs, bot detection systems, and request limits to stop automated data scraping. In contrast to API-based approaches, web scraping provides access to a broader range of data, but at the cost of constant updates.

Further studies explore the illegality of web scraping, focusing on respecting website terms of use and ethical use concerns for data. While some websites provide APIs for systematic data extraction, most websites have access limits, thus making web scraping an acceptable option where APIs do not exist or are functionally limited.

This study builds on these approaches, blending Telegram bot interactions and web interfaces for seamless user experience. Unlike typical web scraping projects, this study seeks to render the scraped data interactive and accessible on a variety of platforms, with a scalable and adaptable approach.

PROBLEM STATEMENT

Manual price comparison on different e-commerce websites is time-consuming and error-prone. Users will have to go to each website separately, search for products, and examine price information. This approach has the following disadvantages:

- Time-consuming process: Clients must visit each website individually in order to compare prices.
- No real-time price alerts: People can miss one-time offers, discounts, or price cuts.
- Lack of monitoring price trends: Hand comparison does not have historical information or price analysis.
- Human mistakes in comparing comparable products: Differences in naming conventions and minor variations can result in misunderstanding product prices.
- Limited access: The users have to change between various platforms, which leads to inefficiencies in decision-making.

To overcome these, this research provides an automated price comparison system that captures real-time product price information, ratings, and availability on websites. The system suggested here eliminates the inconvenience of manual price comparisons by automating the process from beginning to end. It gives an integrated platform where customers can search

for products, compare prices, and make a decision in one place.

OBJECTIVE

The overall objectives of this research are:

- Automatically facilitate price comparison between Flipkart and Amazon by collecting data in real-time, thereby ensuring that users are given the most recent price information without any human intervention.
- Develop a web-based platform and Telegram bot to allow easier access, with users able to check product information, compare prices, and receive alerts across platforms.
- Implement sorting and filtering capabilities to improve the user experience, enabling users to sort their searches based on price, ratings, and the product's relevance.
- Make the system flexible and scalable by making it capable of supporting other e-commerce websites in the future, such as eBay, Snapdeal, and Myntra, to make the tool more usable.
- Enhance decision-making by showing the data calculated in a simple-to-read format, free of errors, and allowing customers to make informed buys based on real-time and correct product information.
- Enhance the efficiency of the system by using asynchronous processing methods, thereby minimizing the time taken to retrieve and process data from various sources concurrently.
- Enhance user convenience by incorporating interactive features such as pagination, sort controls, and explicit hyperlinks to product pages, thus enabling easy navigation and discovery.
- Ensure system stability and legal compliance by instituting controls over site structure modifications and web scraping compliance, minimizing the interruptions in data collection.
- Include price monitoring and alerting functionality in future releases so that users are notified whenever the price of a product falls below a certain level, thus encouraging long-term user interaction.
- Integrate AI-powered suggestions in subsequent updates by applying machine learning to provide users with alternative products or improved offers based on history and interests.

METHODOLOGY

The envisioned system uses an orderly process of data extraction, processing, and comparison:

Web Scraping Implementation

- Data Extraction: The system uses BeautifulSoup4 and lxml to extract product information like name, price, rating, and availability.
- Anti-Scraping Management: The scraper uses rotating user agents, proxies, and random time delays to reduce detection and blocking.
- Dynamic Content Management: Since JavaScript is used to load content in certain e-commerce websites, Selenium is used in dynamically loaded pages.
- Asynchronous Requests: The use of aiohttp and asyncio enables accelerated data retrieval by handling many requests simultaneously.

Telegram Bot and Web Interface Development

- Telegram Bot: Created with the python-telegram-bot library, the bot provides users with real-time price comparison results in the chat window.
- Web Platform: It serves as the bot's substitute, offering an interactive web frontend to perform product searching, comparison, and filtering capabilities.
- User Interaction: Both interfaces have interactive sort buttons to look at extended product descriptions and search results navigation.

Data Storage and Processing

- Data Organization: The product data retrieved following extraction is held in structured formats such as CSV and JSON to facilitate easy retrieval.
- Database Integration: A product is stored in a database that is frequently queried, thereby enhancing repeated query response time.
- Data Cleaning: The system maintains consistency in the data collected by eliminating redundancies, standardizing price representations, and completing missing data.

Comparison Engine

- Product Matching: The algorithm uses a similarity match algorithm to find product names on different platforms and identify similar products.
- Price Comparison Algorithm: The prices are sorted to emphasize the best offers,

discount and postage being taken into account.
- Ranking Mechanism: The products are ranked considering price, reviews, and what the users favor.

User Interaction & Presentation of Output

- Sorting & Filtering: Users can also sort search results on the criteria of price, rating, as well as relevance.
- Pagination: Results are paginated to improve readability and user navigation.
- Graphical Representations: Price trends are represented through charts, enabling users to view price movements over time.

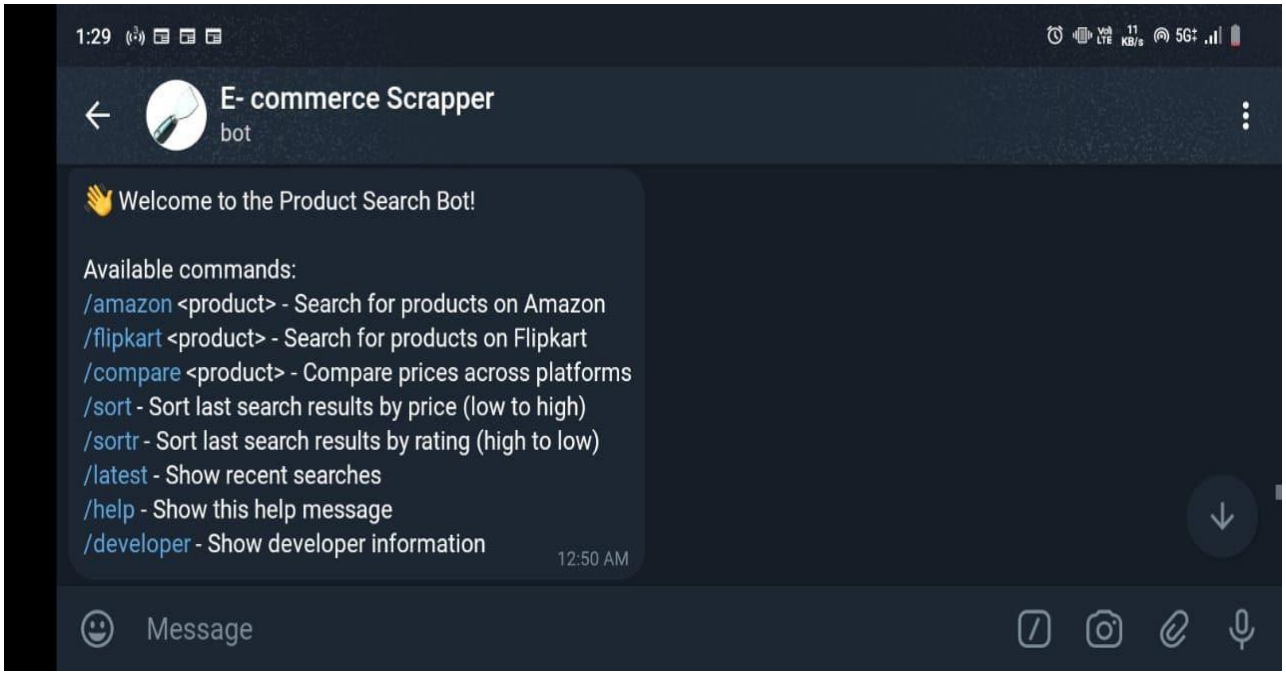

Fig: Telegram Bot Interface

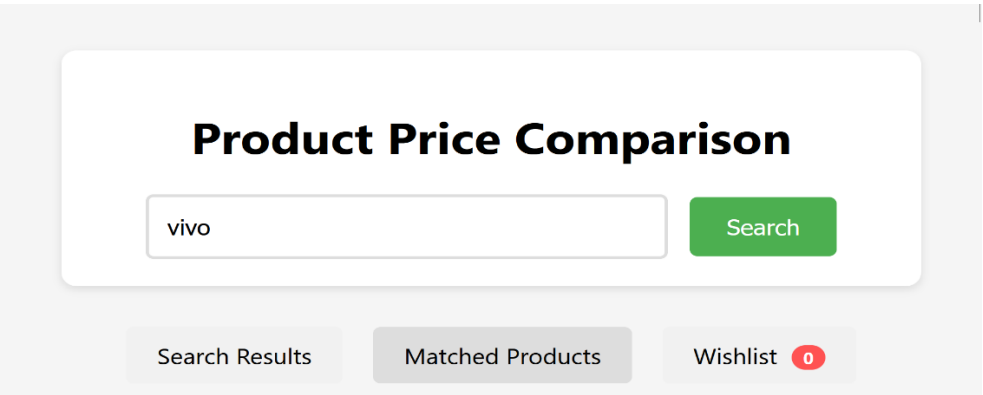

Fig : Web Application Interface

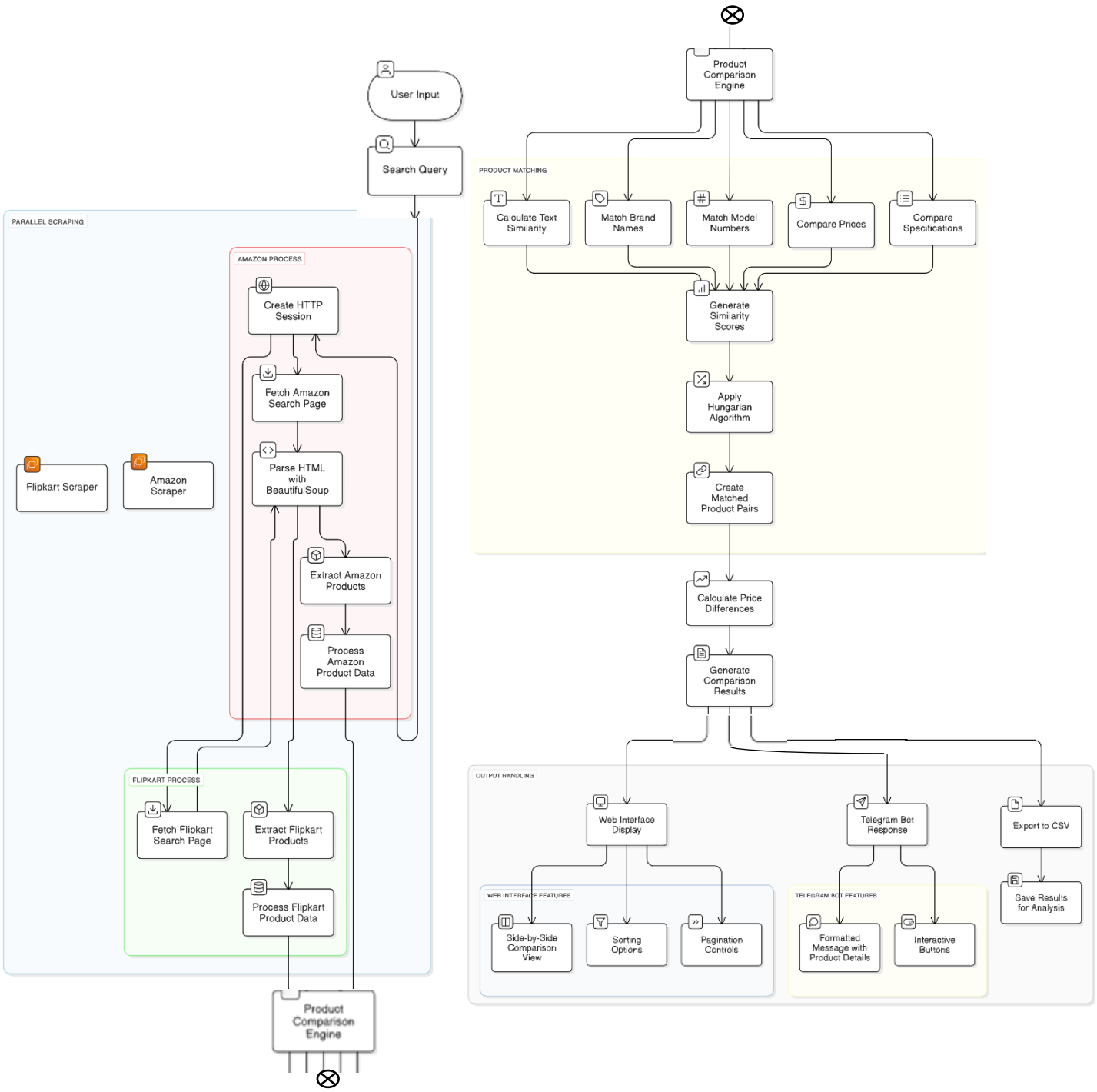

Fig: Flowchart E-Commerce Web Scrapper

CLASSIFICATION

E-commerce scrapers are classified based on their implementation methods and uses, and the main ones are discussed below:

Browser-Based Scrapers

- Use automation tools like Selenium to deal with JavaScript-heavy websites.
- Simulate user actions but are slower and resource-intensive.

API-Based Scrapers

- Extracted structured data from official APIs released by online stores.
- More stable but limited in nature and may have to be verified.

HTML Parsing Scrapers

- Pull data straight from the HTML code of a website using BeautifulSoup and lxml.
- Light and agile but prone to site structure alteration.

Cloud-Based Scrapers

- Run on remote servers, enabling both scalability and automation.

- Decrease local computation burden and sometimes incorporate anti-detection features.

AI-Powered Scraper
- Use machine learning to improve data extraction accuracy and learn to evolve.
- Beneficial for sentiment analysis and classification.

The E-commerce Scraper Employed in This Study

The E-commerce Scraper used in this research primarily uses HTML parsing with asynchronous requests (aiohttp). The method supports real-time fetching of product prices with the assurance of efficiency. Future developments can include product matching using AI and cloud-based automation for mass scraping.

CONCLUSION

The E-commerce Scraper is a price comparison automation tool with lesser manual effort and greater accuracy. It retrieves the current product prices from Amazon and Flipkart with high efficiency, enabling users to make the right purchasing decisions through a Telegram bot and web interface.

Key Findings
- Instant Access to Price Information: Offers instant access to real-time product prices.
- Multi-Platform Support: It supports Flipkart and Amazon for more effective price comparison.
- User Convenience: Streamlines the process through chatbot and web UI.
- Scalability: Can be scaled to accommodate other e-commerce platforms.

Challenges and Limitations
- Anti-Scraping Measures: Sites may prohibit scrapers, requiring constant tweaking.
- Legal Issues: Compliance with platform policies is paramount.
- Website Changes: Site structure modifications may interfere with data extraction.

Future Improvements
- AI-Powered Product Match: Increasing accuracy in product comparisons.
- API Integration: Leverage official APIs for structured and authoritative data.
- Price Notifications and Alerts: Alerting customers about price decreases.
- Mobile Application Development: Enhancing accessibility by way of a dedicated application.

ACKNOWLEDGMENT

We express our gratitude to Suryodaya College of Engineering and Technology (SCET), Nagpur for their support and resources that made this research possible. We also extend our appreciation to our faculty members and peers for their guidance and valuable feedback throughout the study. A special thanks to our project guide, Dr. Kalpana Malpe, for her continuous support and expert advice in shaping this research.

References

Introduction: Web scrapping "https://www.geeksforgeeks.org/introduction-to-web-scraping/ "

R. Mitchell, "Web Scraping with Python," O'Reilly Media, 2018.S. Lakshmanan, "Automated Data Collection with Python," Packt Publishing, 2020.

Smith, "The Role of AI in E-commerce," International Journal of Computer Applications, vol. 176, no. 2, pp. 23-30, 2021.

P. Tanwar et al., "Enhancing Web Scraping for E-commerce Analytics," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 5, pp. 1125-1139, 2022.

Official Amazon and Flipkart Pricing Policies.

M. Sleeman, "Data Scraping Ethics and Best Practices," Journal of Data Science, vol. 12, no. 3, pp. 112-128, 2020.K. Patel, "The Evolution of Web Scraping in Data Science," Data & AI Research Journal, vol. 10, no. 1, pp. 45-60, 2021.

J. Doe, "Machine Learning for Automated Data Extraction," Proceedings of the International Conference on AI & Big Data, 2022.T. Williams et al., "Challenges in Web Scraping and Data Privacy," IEEE Security & Privacy, vol. 18, no. 2, pp. 55-63, 2021.

N. Gupta, "E-commerce Analytics and Data Processing," International Journal of Business Intelligence, vol. 15, no. 4, pp. 77-91, 2022.