e-ISSN 3108-026X
International Multidisciplinary Research Journal Reviews (IMRJR)
A Peer-reviewed journal
Volume 3, Issue 4, April 2026 | DOI 10.17148/IMRJR.2026.030402

WEB-BASED ECOMMERCE PRICE COMPARISON SYSTEM USING WEB SCRAPING

Adejumo, Samuel Olujimi[1,2], Alade, Samuel Mayowa[1,2], Godwin O. Osakwe[3], Onabanjo, Oluwatobiloba John[4]

Department of Cybersecurity, Nnamdi Azikiwe University, Awka, Nigeria[1]

Department of Computer Science, Nnamdi Azikiwe University, Awka, Nigeria[2]

Department of Cyber Security, Southern Delta University, Ozoro, Delta State, Nigeria[3]

Department of Computer Science, Abiola Ajimobi Technical University, Ibadan, Nigeria[4]

Abstract*:* The rapid expansion of e-commerce has created a highly competitive marketplace where consumers are often faced with the challenge of finding the best deals across multiple online platforms. Manually comparing product prices is time-consuming, inefficient, and prone to inaccuracies due to inconsistent data across different websites. This project presents the design and development of a Web-Based Ecommerce Price Comparison System using web scraping techniques to provide real-time price comparisons for products from multiple online retailers, including Jumia, Slot, and Jiji. The system integrates a responsive user interface built with React.js and a robust backend powered by Django REST Framework. Web scraping was implemented using Python libraries such as BeautifulSoup, enabling efficient extraction of product details including prices, descriptions, and source platforms. Testing and evaluation showed that the system successfully met its objectives, offering a clean, intuitive, and responsive interface with accurate data retrieval. While challenges such as anti-bot measures, dynamic content handling and inconsistent data formats were encountered, the system provides a functional prototype with significant potential for scalability, future integration of more platforms, and adoption of advanced features such as machine learning-based product matching. This work contributes to improving consumer decision-making, enhancing online shopping efficiency, and promoting transparency in e-commerce pricing.

Keywords: E-commerce, Web Scrapping, Price comparison system, Machine learning, Pricing

I. INTRODUCTION

The evolution of e-commerce has drastically changed consumer buying behaviour, making shopping more convenient and granting access to an extensive range of products from virtually anywhere. According to Statista (2023), global e-commerce sales are projected to reach $6.3 trillion by 2024, driven by increasing internet penetration and the proliferation of online marketplaces such as Amazon, eBay, and Alibaba. However, this exponential growth has also led to an overwhelming number of choices for consumers, making it challenging to identify the best products at the most competitive prices.

In today's digital marketplace, consumers often face difficulties comparing prices across multiple platforms due to the fragmented nature of e-commerce websites. Each platform operates independently with its pricing strategies, discounts, and product listings. This fragmentation forces consumers to manually visit multiple websites to compare prices, which is time-consuming and inefficient. To address these challenges, price comparison systems have become essential tools that enable consumers to make informed purchasing decisions. These systems aggregate prices from many e-commerce platforms and using structural approach to present them, allowing users to identify the best deals available quickly.

The emergence of web scraping technologies has significantly enhanced the effectiveness of price comparison systems. Web scraping, a technique for extracting data from websites, enables the automatic collection of product information from various online marketplaces, including prices, specifications, and user reviews (Mitchell, 2015). Web scraping allows for real-time data retrieval, ensuring that consumers can access up-to-date and accurate information making it different from traditional data collection approaches prone to human error and a waste of time.

Despite the benefits, developing an efficient e-commerce price comparison system presents several challenges, including data inconsistency, website structure variability, and ethical considerations related to web scraping (Daniel Glez-Peña et

al., 2013). Moreover, ensuring data accuracy and maintaining compliance with website policies are critical aspects that must be addressed in the system's implementation.

Studies have shown that the rapid development of online retailers has created a highly competitive e-commerce landscape, offering consumers an overwhelming array of choices. However, this abundance of options has introduced significant challenges for shoppers whose focus is getting the best prices for the products they want to purchase. The comparison of prices across multiple e-commerce platforms using a manual method is not only time-consuming but also inefficient, as it requires navigating multiple websites with varying user interfaces and product listings. Furthermore, many existing price comparison tools fail to provide real-time updates, resulting in outdated or inaccurate pricing information that undermines consumer trust and decision-making. Another critical issue is the inconsistency of product data across ecommerce platforms. Variations in product descriptions, naming conventions, and availability make it difficult for consumers to compare items accurately, leading to potential mismatches and dissatisfaction. To address these challenges, this research proposes the development of an integrated web-based e-commerce price comparison system. Leveraging real-time web scraping techniques, the system will provide users with up-to-date pricing information. By combining these functionalities, the system aims to enhance decision-making efficiency, reduce the time spent on manual searches, and enhance the overall online shopping experience for consumers.

This study aims to develop a web-based e-commerce price comparison system adopting web scraping techniques with a view to provide consumers with real-time price comparisons across multiple online retailers. The objectives include design and development of a web-based platform, implementation web scraping, testing, and validating the platform. By leveraging web scraping tools and techniques, this project seeks to enhance the e-commerce shopping experience, ensuring that users make informed purchasing decisions efficiently. Similarly, the study focuses on the development of a web-based e-commerce price comparison system designed to enhance user decision-making in online shopping with key functionalities such as real-time comparison, web scrapping, and API integration. Also, the developed system will enable users to identify the best outlets to purchase products from by comparing prices across multiple ecommerce platforms. Moreover, the system will be accessible via mobile and desktop browsers, ensuring broad usability across devices.

The paper is organised into different sections. Section two discusses a review of the literature on previous researches related to the study, including gaps. Section three involves the materials and methods used to achieve the objectives of this study. Section four discusses the result and discussion of findings, and finally Section five gives the summary and conclusion.

II. LITERATURE REVIEW

The retail industry has changed due to the exponential rise of ecommerce, which gives customers access to a so many array of goods and services never before possible. However, this convenience has also brought about new challenges, especially when navigating a wide range of products and comparing costs across several platforms. With so much information available, consumers frequently struggle to find the best offers or make informed decisions about what to buy.

In response, price comparison tools have become essential resources that compile and present pricing and product information from multiple sources, enabling customers to make well-informed and economical choices. With an emphasis on online scraping technologies, data integration, and system usability, this section examines the body of research on the creation, approaches, and difficulties of pricing comparison systems. The analysis points up existing gaps as well as potential future developments.

Web scraping is essential to contemporary digital marketing and ecommerce because it makes it possible to automatically extract vast amounts of publicly accessible data from internet platforms. To improve market transparency and make well-informed decisions, price comparison platforms enable companies and customers to gather real-time pricing, availability, and product descriptions from rival online stores. According to Henrys (2021), adopting data-driven method gives businesses a competitive advantage by enabling them to monitor competitors' offerings, promotional strategies, and pricing models in real time. Marketers can use scraping techniques to collect information on product ratings, social media sentiment, customer reviews, and promotional content to support competition analysis, trend tracking, and dynamic pricing strategies.

In addition, web scraping facilitates lead generation and supports personalized marketing efforts by identifying customer interests and behavioural patterns across platforms (Henrys, 2021). E-marketing campaigns are guaranteed to be data-driven and adaptable to changes in the market thanks to this automation, which also minimises manual labour and expedites data collection. As a result, for businesses looking to stay competitive in quick-paced online markets, web scraping has become a crucial part of digital intelligence.

e-ISSN 3108-026X
International Multidisciplinary Research Journal Reviews (IMRJR)
A Peer-reviewed journal
Volume 3, Issue 4, April 2026 | DOI 10.17148/IMRJR.2026.030402

Furthermore, websites are classified into two namely: static and dynamic, where each form involves unique techniques of extraction (Lan et al., 2021). The static websites have a fixed data structure and display the same data or content stored on the web server side when information in form of markup language (HTML) is loaded on the client's web browser, and retrieved by sending an HTTP request. Whereas, the dynamic webpages cannot be accessed in the requested HTML of the target webpage.

Price comparison website is an online platform for a combination of product listings from many retailers, enabling users to compare prices, features, and other details for the same or similar products. These websites aim to help consumers find the best deals by providing a centralized view of pricing information across various e-commerce platforms such as Google shopping, Shopzilla, Price Grabber, Bizrate, Shopsavvy, etc. Multiple popular price comparison websites use various techniques that have their advantages and drawbacks to both consumers and retailers, including cost saving, time-efficiency, informed decision making, personalization, and convenience, while to the retailer are targeted marketing, increased visibility, cost-effectiveness, etc (Jung et al., 2014).

Lawson (2015) provides a practical guide for building Python based web scrapers. The researcher focuses on extracting data from static pages using requests and HTML parsing, and progresses to more advanced topics like concurrent downloading, handling dynamic content via JavaScript, form submission, session management, and CAPTCHA handling using popular Python tools like BeautifulSoup, Selenium, Scrapy, and user-agent rotation.

1. BeautifulSoup: This is a lightweight library ideal for parsing static HTML content. It pairs seamlessly with requests to fetch web pages and supports flexible parsing using either the built-in HTML parser or the faster lxml engine. While intuitive for beginners, it cannot execute JavaScript, making it unsuitable for dynamic websites. Its simplicity shines in small-scale projects, such as extracting headlines or product listings from basic websites. BeautifulSoup is easy to use for static HTML parsing but lacks support for JavaScript 1rendering (Sirisuriya, 2015).

2. Scrapy: Scrapy is a full-fledged framework designed for large-scale, structured web scraping. Unlike BeautifulSoup, it handles asynchronous requests natively and includes built-in features for proxy rotation, data pipelines, and middleware customization. While its learning curve is steeper, Scrapy excels in industrial-grade projects (like crawling entire e-commerce catalogs) where performance and organization are critical. For JavaScript-heavy sites, it can integrate with Splash or Playwright. Scrapy outperforms BeautifulSoup and Selenium in terms of speed and resource utilization due to its asynchronous architecture, making it the preferred option for large-scale scraping projects (Sirisuriya, 2015).

3. Selenium automates real browsers (Chrome, Firefox) to interact with dynamic, JavaScript-rendered content. It mimics human actions like clicking buttons, making it indispensable for scraping single-page applications (SPAs) or sites with anti-bot measures. However, its reliance on browser instances makes it slower and more resource-intensive than HTTP-based alternatives. Use cases include scraping social media platforms or complex web apps. Selenium handles dynamic content well but is slower and more resource-intensive (Sirisuriya, 2015).

4. Requests: Requests-HTML bridges the gap between static and basic dynamic scraping by combining requests with a simplified HTML parser. It includes CSS/XPath selectors and offers limited JavaScript rendering via Pyppeteer under the hood. While less powerful than Playwright, it's perfect for quick jobs where lightweight JavaScript execution is needed, such as extracting data from simple AJAX loaded elements without full browser automation.

5. Playwright: Playwright (and its Node.js counterpart, Puppeteer) modernizes browser automation with faster execution and multi-browser support (Chromium, Firefox, WebKit). Designed for reliability, it handles SPAs better than Selenium and offers a headless mode for efficiency. Its Python API simplifies tasks like screenshots or PDF generation, making it ideal for testing and scraping modern frameworks like React or Vue.js.

III. RELATED WORKS

The rapid evolution of online shopping has prompted significant research into systems that assist consumers in making more informed purchasing decisions. This has given birth to the increase in the number of online shoppers and multitudes of shopping websites in the digital space, and in turn, leading to the growth of e-commerce. Time wastage, stress and errors in decision making are associated with consumers who embark on manual searching and products comparison. One such solution is the development of e-commerce price comparison platforms, which consolidate data from multiple online stores to present real-time product information and pricing. Various studies have explored different approaches to

building these systems, particularly focusing on data aggregation techniques, scraping methodologies, and system scalability.

Mitchell's (2015) work laid the groundwork for practical web scraping in his book Web Scraping with Python, which outlines practical techniques for extracting structured data from the web using tools like BeautifulSoup and Scrapy. These tools have been widely adopted for projects requiring the automated collection of data, particularly in price comparison systems where real-time updates are crucial.

Krotov et al. (2020) examined the legal and ethical dimensions of web scraping by analyzing key court cases and legislation, such as the Computer Fraud and Abuse Act (CFAA) and notable rulings like hiQ Labs v. LinkedIn. Their work emphasizes that while scraping public data may not always be illegal, it often breaches websites' terms of service and may lead to legal consequences. The authors also stress the importance of ethical considerations, such as avoiding personal data collection, respecting robots.txt files, and minimizing server load.

Daniel Glez-Peña et al. (2013) examined the growing significance of web scraping in an API-driven world. Their work highlights how the combination of traditional scraping techniques with modern APIs can offer more reliable and ethical data access, especially for platforms with strict antibot measures. They also emphasized the importance of data normalization and handling heterogeneous data formats issues that this project also addresses.

In recent research, Shaikh et al. (2023) proposed a hybrid product comparison system that integrates machine learning with web scraping. Their system not only extracts real-time pricing and specifications from various e-commerce websites but also uses intelligent algorithms to recommend the best product based on user needs. While their work focuses on intelligent recommendations, it reinforces the value of real-time data scraping and its potential for future improvements. Similarly, Zhang et al. (2019) introduced a price comparison model that incorporates historical pricing trends and shipping costs, improving the accuracy of comparisons. Their system demonstrates the potential for enhancing user decision-making through data analytics, although it relies heavily on structured APIs rather than raw scraping.

Jankaev and Thamviset (2025) addressed the challenge of consumers wasting a lot of time due to manual search and product comparison before identifying the best deal or product price among online websites, leading to uncertainty in decision-making. In the study, the authors proposed and developed a web extraction system using web scraping methods to recognise best choices from different e-commerce websites. The system architecture was designed having three parts (Web extraction, data categorization, and application layer). The data extraction module was configured using wrapper while, the data categorization was processed using Term Frequency -Index Document frequency (TF-IDF) and Convolutional Neural Networks (CNN). The designed system was implemented in Python Programming Language with its Django web framework. Additionally, the system was evaluated by specifically using selected Lazada (lazada.co.th), Power Buy (powerbuy.co.th), Banana IT (bnn.in.th), and JIB (jib.co.th) as target websites. The results of the study obtained an accuracy of 100% in product data extraction from target websites.

These studies collectively establish a foundation for developing robust price comparison platforms. However, most existing systems either lack real-time updates, struggle with scalability, or do not cater to regional ecommerce platforms. This project builds on prior work by implementing a custom scraping solution targeted at Nigerian e-commerce websites like Jumia, Slot, and Jiji, aiming to provide localized and up-to-date price information through a responsive web interface.

Vasudevan and Megala (2019) implemented a Python-based system that extracts product data such as names, prices, and ratings from ecommerce websites and stores it in CSV format. Their approach combines data scraping with visualization techniques using tools like Matplotlib and Pandas, allowing users to analyse trends in pricing and product performance. While effective for small-scale use, the authors noted that the system struggles with scalability and dynamic JavaScript content, highlighting a key limitation for real-time or large-scale scraping tasks.

Chen (2024), using Java and Python programming tools: Selenium, BeautifulSoup, and Scrapy, addressed the challenges associated with dynamic content extraction, anti-scraping mechanisms, and the effectiveness of various scraping tools. The experiment was evaluated and compared in terms of efficiency, success rate, and data accuracy. Results of the study found that JD.com had the highest scraping efficiency when using BeautifulSoup. This research reveals how to choose the best online scraping approaches for real-time pricing comparison systems, as well as how platform-specific problems affect efficiency and data extraction accuracy.

The studies reviewed highlight the growing significance of price comparison systems within the e-commerce domain, driven by increasing consumer demand for smarter, time-saving, and cost-effective online shopping tools. Various studies have also shown that such systems enhance user decision-making by aggregating data from multiple sources and presenting accurate, up-to-date pricing and product information. The use of web scraping, data mining, and machine

e-ISSN 3108-026X
International Multidisciplinary Research Journal Reviews (IMRJR)
A Peer-reviewed journal
Volume 3, Issue 4, April 2026 | DOI 10.17148/IMRJR.2026.030402

learning has proven effective in improving system accuracy and scalability. Despite existing progress, gaps remain in areas such as real-time data updates, product matching accuracy, user personalization, and multilanguage or multi-currency support. Many systems also struggle with dynamic pricing strategies used by online retailers, resulting in outdated or inconsistent comparisons.

While comparison platforms enhance decision-making for many product types, their effectiveness depends on the nature of the product and consumers' prior expectations (Jung et al., 2014). This review establishes a strong foundation for developing an efficient and user-friendly price comparison system by adopting modern technologies. The knowledge obtained from the literature will inform the design choices and implementation strategies in this project, ensuring relevance, innovation, and practical usability.

IV. RESEARCH METHODOLOGY

The methodology adopted in this study is structured to directly address the research objectives. It involved a sequential development process beginning with planning, research, system architecture, data gathering methods, and deployment tactics used to make sure the system achieves its goals. To enable users and customers easy products' price comparison across multiple platforms, the system combines web scraping techniques and a robust web development stack.

A. REQUIREMENT ANALYSIS

Existing platforms and user expectations were evaluated to determine both functional and non-functional requirements. Important features, including cross-platform comparison, product search, filtering, and real-time updates were developed by examining effective systems like PriceSpy, Jumia Price Tracker tools, and Google Shopping.

B. FEASIBILITY STUDY

In the study, a feasibility study was conducted to evaluate the system's operational, financial, and technical viability. To guarantee a practical and long-lasting solution, elements including project cost, implementation time, skill requirements, hosting options, and long-term maintainability were taken into account.

A. SYSTEM ARCHITECTURE

The architecture of this proposed web scrapping system adopts a modular design, consisting of two components namely: front-end user interface and a back-end API layer, without persistent database storage. The web scraper functions as a core service that is triggered on user demand to extract real-time product data from selected e-commerce websites. The collected data is processed and immediately displayed to the user through the interface, ensuring fast, up-to-date comparisons without storing data long-term, which is depicted in Figure 1. The website's front-end and back-end components are put into operation throughout the development stage. This phase is separated into the subsequent phases

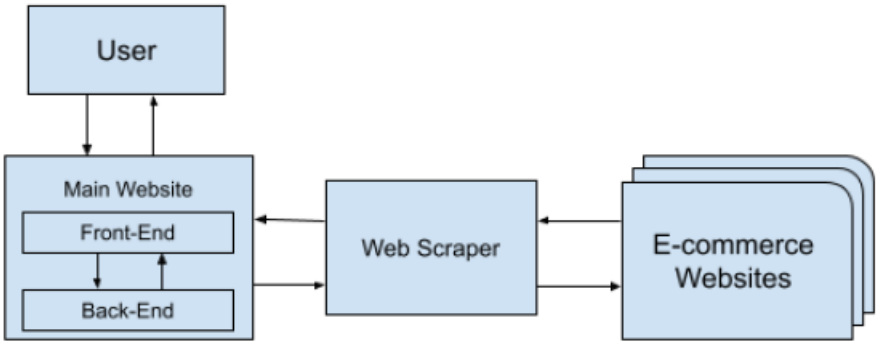

Figure 1: System Architecture of the Price

The front end was developed using React.js for building dynamic and responsive user interfaces. Its component-based architecture promotes modularity, code reusability, and maintainability, enabling a more efficient and scalable development process. Component architecture of the web scraping system or application was deconstructed into components such as:

1. Navbar: this part handles navigation across the website.
2. Search Form: This module is used to search for products.
3. Product Card: This part of the architecture shows a single product and displays useful info such as name, the platform it was acquired from, and price.

e-ISSN 3108-026X
International Multidisciplinary Research Journal Reviews (IMRJR)
A Peer-reviewed journal
Volume 3, Issue 4, April 2026 | DOI 10.17148/IMRJR.2026.030402

4. Filter panel: this is used to filter search results from various websites.

D. DESIGN AND DEVELOPMENT OF A WEB-BASED PLATFORM FOR PRICE COMPARISON

The initial phase of the study focuses on designing and developing an interactive, user-friendly web-based platform that enables users to compare product pricing across different e-commerce platforms. The development process followed these steps: UI/UX design, front end and back-end. The website platform layout and user interface (UI/UX) design were designed as a prototype using Figma, ensuring an intuitive and visually appealing user interface. Key features such as search functionality, product comparison tables, and user dashboards was prioritized to enhance usability.

Further, the front end of the platform will be developed using JavaScript frameworks such as React.js or Angular. These frameworks were chosen for their component-based architecture, which helps in the creation of dynamic and responsive user interfaces. A responsive design approach was implemented using Tailwind CSS to ensure compatibility and consistent styling across desktop and mobile browsers. Moreover, the back-end of the website platform was developed using Python frameworks such as Django or FastAPI. These frameworks provide robust tools for handling server-side logic, database management, and the integration of API. In the study, Django framework enables quick development and clear, maintainable code using the Model-view-Template (MVT) design, which guarantees separation, increasing the system's modularity and scalability. The Django REST Framework (DRF) facilitates the creation of RESTful Application programming interfaces (RESTful APIs) for seamless communication between the two tiers of the architecture. The back end was responsible for processing user requests and integrating with external e-commerce platforms. Also, JWT (JSON Web Token) authentication for securing user login and session time management. Rate limiting & request validation to prevent abusive scraping attempts.

E. IMPLEMENTATION OF WEB SCRAPING TECHNIQUES FOR PRODUCT INFORMATION EXTRACTION

The second phase involves the development of a web scraper for product information extraction from various e-commerce platforms. The web scraper was built using Python libraries such as BeautifulSoup and Scrapy. These tools handled the static content extraction which enabled the extraction of product details, including product names, descriptions, prices, and availability.

Webscraper is used for automatic data extraction from the website of interest by entering a search item in the system's user interface. The URL in turn performs search on the websites using a WebDriver of choice. The next step is using BeatifulSoup4 library for HTML content parsing and data extraction following pre-defined rules in the scraper configuration file.

Nevertheless, product data was collected from websites using web scraping techniques targeting selected e-commerce platforms for extracting product attributes, and updating data at regular intervals, and tools such as BeautifulSoup for the purpose of webscraping. Ethical considerations are taken into consideration as compliance with robots.txt guidelines, request throttling, and adherence to data privacy regulations are ensured. Again, the application was deployed using Front-end (Vercel) and Render for its backend.

F. SYSTEM TESTING AND VALIDATION

The web-based e-commerce price comparison system was tested using a variety of techniques to make sure it meets its functional and non-functional requirements and operates dependably. Validating essential functionality, evaluating system behaviour in various scenarios, and guaranteeing device and browser compatibility were the main goals of these tests. The findings of functional, usability, and performance testing are shown in this section, emphasising both the system's general efficacy and any areas that need work.

G. FUNCTIONAL AND NON-FUNCTIONAL TESTING

Each e-commerce website was tested independently and in combination with others. Table 1 summarizes some of the major functionalities tested. Also, the following non-functional testing was carried out on the webscrapping system, including:
1. Performance Testing: This test was carried out during performance, where the testing average response time varied based on the amount of product entries being scraped. For instance, when the keyword "*Laptop*" was used for searching and testing, the average response time was 8.6seconds, while the keyword "*Gtx 1660ti*" was used with an average response time was 4.5seconds
2. Usability Testing: Participants found the interface intuitive and easy to navigate, rating it 9/10.

e-ISSN 3108-026X
International Multidisciplinary Research Journal Reviews (IMRJR)
A Peer-reviewed journal
Volume 3, Issue 4, April 2026 | DOI 10.17148/IMRJR.2026.030402

H. PERFORMANCE METRICS

Based on the webscrapping operation carried out on these three popular e-commerce websites platforms like Jumia, Slot, and Jiji. Each platform was tested and assessed based on key performance indicators such as the functional and non-functional parameters like the user interface, authentication, search procedure, usability and response time. Other metrics include average scrapping time, success rate and efficiency of data (i.e., data accuracy).

Similarly, when evaluating the effectiveness of product information retrieval on different e-commerce platforms, this study conducted data retrieval assessments on the three major platforms, primarily examining the following aspects: (i) Scraping time: This is the entire amount of time needed to get product pricing information after submitting a request.(ii) Success rate: This is the proportion of accurate price information that is successfully captured.(iii) Data accuracy: This refers to how closely the prices the crawler retrieved match the prices that are shown on the website.(iv) Anti-crawler measures: Through CAPTCHA, request frequency limits, and other methods, the platform restricts crawlers.

V. RESULTS AND DISCUSSION OF FINDINGS

A. Result

The outcome of the webscrapping system testing, the comparison of the system performance was carried out using the metrics presented in the previous section on the online platforms. The relevant product data captured include: product name and specification, image, product price, and listing. The experiment setup involved crawling, scraping, and evaluating the efficiency of a product from three website platforms over 100 times. Tables 1 and 2 display experimental outcomes.

Table 1. Experiment Results

S/N	Features	Expected result	Actual result	Status
1	User Interface	Account successfully created	Account successfully created	Pass
2	Login/Authentication	Redirect to scrapper	Redirect to scrapper	Pass
3	Product Search	Return Product List	Return Product List	Pass

Table 2. Experiment results

Platforms Used	TOOLS/LIBRARIES	Average Scrapping time	Success rate	Data Accuracy	Measures of Anti srcaping
Jumia	Tailwind CSS, React-router-dom, react icons, axios, Selenium	7s	90%	99%	CAPTCHA, dynamic loading, IP blocking Jumia.
Slot	Asgiref, Django-cors-headers, Djangorest Framework, Djangorestframework simplejwt, PyJWT, Pytz, sqlparse, Psycopg2-binary, Python-dotenv, requests, bs4, gunicorn, lxml, Selenium	6.5s	90%	99%	CAPTCHA, dynamic loading, IP blocking Jumia.
Jiji	Selenium	8s	90%	99%	CAPTCHA, dynamic loading, IP blocking Jumia.

B. USER INTERFACE RESULTS

The application features a clean and responsive user interface, including the various key pages and components.

e-ISSN 3108-026X
International Multidisciplinary Research Journal Reviews (IMRJR)
A Peer-reviewed journal
Volume 3, Issue 4, April 2026 | DOI 10.17148/IMRJR.2026.030402

B1. Home Page

The homepage of the price comparison system offers a clean and intuitive interface designed to help users quickly understand the platform's purpose and functionality. It introduces the system's core objective (to help users find the best deals across various e-commerce platforms) through a prominent headline: "Find the Best Deals Across the Web." A central call-to-action button (Get Started) directs users toward the core comparison functionality of the application. Beneath this, the "How It Works" section offers a concise overview of the system's operation using minimal text and intuitive icons. It explains the two main features as depicted in Figure 2.

1. Search: This is a feature where users input a product keyword, and the system initiates a real-time search across multiple online retailers using integrated web scraping techniques.

2. Compare: This feature is one which helps to retrieve product information, including names and prices, to be displayed in a structured format to allow users to easily compare prices across different sources. The homepage features a contact form where people can submit inquiries and feedback. The form captures essential details such as the user's name, email address, and message. A visual representation of the system's responsive design is displayed using mock ups of desktop and mobile views. This highlights the platform's cross-device compatibility, ensuring usability on mobile and desktop browsers.

B2. Compare Page

The Compare Page as shown in Figure 3 provides the core functional interface of the website, enabling users to view real-time product listings retrieved from multiple ecommerce websites. Upon entering a search keyword into the input field and then clicking the Submit button, the system initiates a scraping process that fetches relevant product data from platforms such as Jumia, Jiji and Slot. Figure 3 is displayed in a responsive grid layout, with each product card featuring: (1) product image, (2) Product name and specifications, (3) Price (in Naira), (4) The source platform (e.g., Jumia, Jiji, Slot), and (5) A clickable "Visit" link that redirects the user to the original product page. Moreover, the layout provides users with a streamlined view of comparable items, allowing for efficient side-by-side price and feature comparison. The platform filter (e.g., All Platforms) allows users to decide which source platform's entries they would like to review.

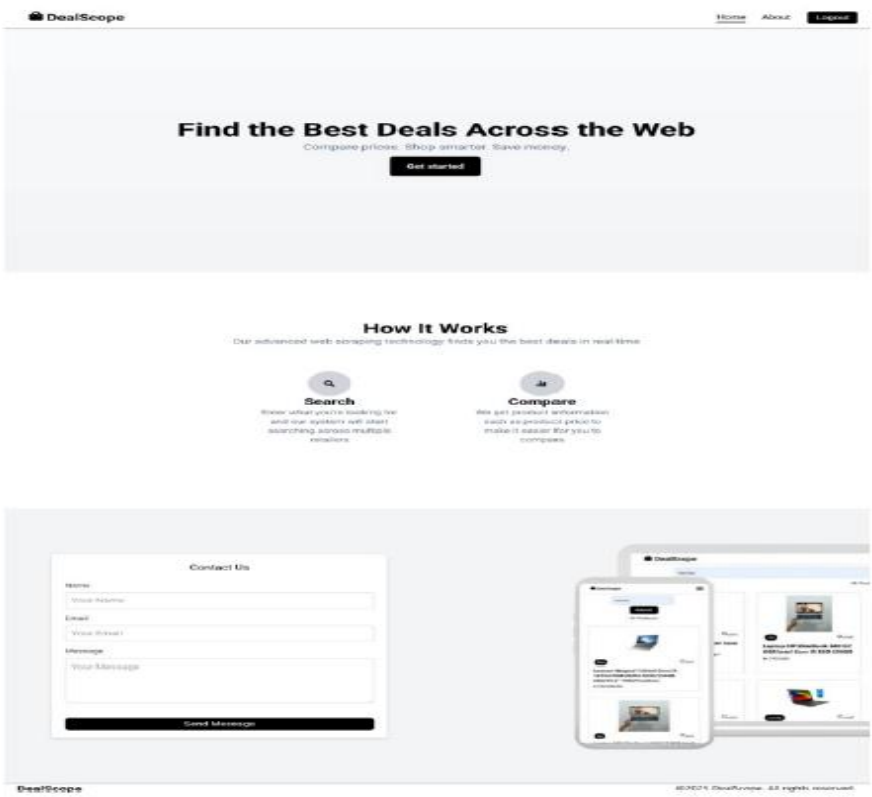

Figure 2: Home page

e-ISSN 3108-026X
International Multidisciplinary Research Journal Reviews (IMRJR)
A Peer-reviewed journal
Volume 3, Issue 4, April 2026 | DOI 10.17148/IMRJR.2026.030402

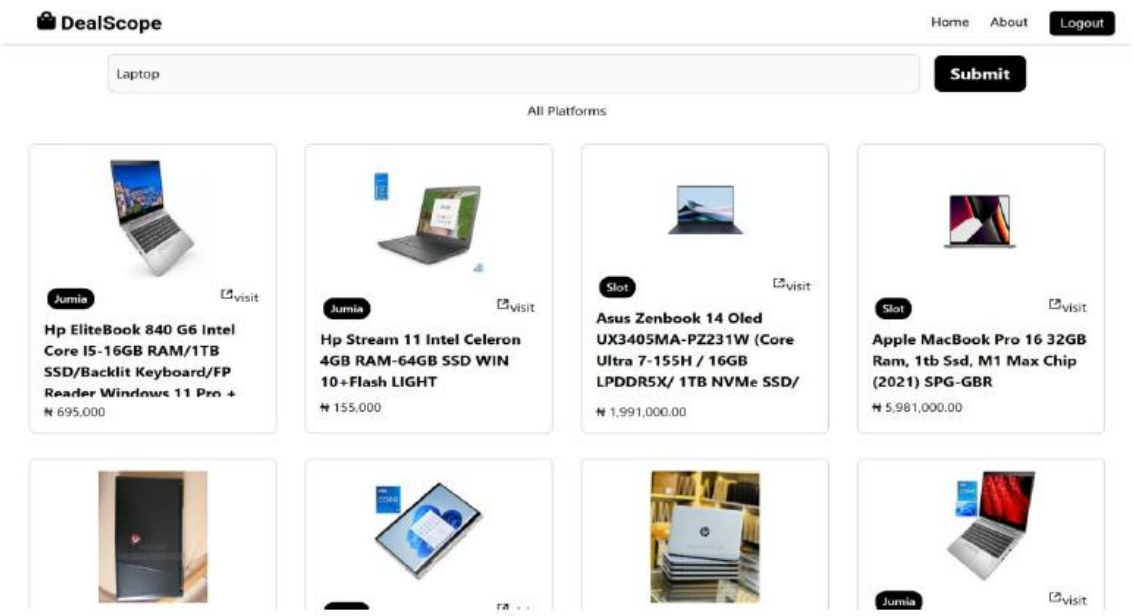

Figure 3: Compare page

B3. About Page

The About Page of the DealScope platform provides users with background information about the project, its purpose, and the value it offers to online shoppers. The page is designed to be clean, accessible, and informative ensuring visitors understand the motivation behind the system and how it works. The about page also features a contact form similar to the one in the home page.

B4. Navigation Bar

The Navigation bar (Navbar) has various navigation elements (including links to the Home, About, and Logout, Signin and Signup pages) that are clearly positioned at the top of the page, providing seamless navigation across the application. It also has the website logo that is positioned at the left side of the navigation bar.

C. DISCUSSION

From the result of the webscrapping system, the overall design of the homepage emphasizes usability, minimalism, and user engagement, aligning with modern web development standards and enhancing the overall user experience as depicted in Figure 2. The data shown in Table 1 indicates that the system developed meets up the requirements for its operation. The results indicate that the system meets its core objectives in that it successfully scrapes data from multiple online stores and provides a user-friendly interface for shoppers. Table 2 provides the experimental result from the evaluation of the platforms in terms of efficiency, average time of scrapping and success rate. It can be seen that online websites (Jumia, Slot, and Jiji). Slot has a good crawling product information efficiency, including prices with the shortest average crawling time of 6.5 seconds. This is because the majority of its pages contain dynamic materials therefore can be processed fast unlike some webpages whose contents are pre-built and delivered to user exactly as stored (static). This can easily be handled using BeautifulSoup, and its anti-crawling mechanism is relatively free. This is followed by Jumia with an efficiency of data crawled as 99% with 7seconds as its average scrapping time while Jiji has 99% efficiency and 8seconds for its scrapping.

Selenium's asynchronous processing approach speeds up Slot scraping, but the corporation may occasionally load poor data, indicating that web crawl success rates are slightly lower. Jiji's crawling efficiency is low, and Selenium simulates user operations faster. Furthermore, the online platforms have stringent anti- crawling measures including IP blocking, CAPTCHA verification, and dynamic content loading that drastically lower the crawling success rate. It is evident from comparing various platforms that the effectiveness and success rate of data crawling are directly impacted by the crawler library and platform selection. The greatest option for rapidly gathering data on static pages is BeautifulSoup; Selenium

works better on dynamic pages but is slower. To guarantee real-time scraping and data accuracy, appropriate webscraper libraries should be chosen in practice depending on the features of particular platforms.

However, some limitations were observed. For instance, websites with advanced anti-bot protection (e.g., Cloudflare) posed challenges during scraping. Prices with embedded JavaScript required more complex parsing, which sometimes failed. Similarly, variations in data formats across different websites limited the consistency and depth of product descriptions that could be provided. These findings underscore the need for advanced scraping techniques and possibly API integration where available.

VI. CONCLUSION

This study demonstrated that web scraping can be effectively used to power a real-time e-commerce price comparison system that helps consumers make smarter purchasing decisions. The developed system fulfilled its main goals: collecting accurate product data, enabling real-time price comparison, and presenting information through a clean and responsive interface. Although the system faced some technical and legal constraints, it proved to be a functional prototype with significant potential for real-world application. The project contributes meaningfully to the growing field of consumer-centric online tools and sets a strong base for further advancement in this area of study. Suggestions included integrating more e-commerce platforms and offering mobile optimization, both of which are considered for future work.

REFERENCES

[1]. D. Glez-Peña, A. Lourenço, H. López-Fernández, M. Reboiro-Jatoand F. Fdez-Riverola, "Web scraping technologies in an API world." *Briefings in Bioinformatics.*, vol.15, no.5, pp.788–797, 2013. https://doi.org/10.1093/bib/bbt026.

[2]. K. Henrys, "Importance of web scraping in e-commerce and E-marketing". Available at SSRN 3769593, 2021.

[3]. J. Jung, K.N. Kwon, D. Cho and S. Lee, (2014). Online shoppers' response to price comparison sites. Journal of Business Research, vol.67, no.10, pp. 2079–2087, 2014. https://doi.org/10.1016/j.jbusres.2014.04.016.

[4]. M.A. Khder, (2021). Web scraping or web crawling: State of art, Techniques, Approaches and Application. International Journal of Advances in Soft Computing & Its Applications, vol.13, no. 3,pp.144-168, 2021

[5]. V Krotov, L. Johnson and L.Silva, "Tutorial: Legality and Ethics of Web Scraping", *Communications of the Association for Information Systems*, vol. 47, pp-pp. 2020. https://doi.org/10.17705/1CAIS.04724

[6]. R. Lawson, "Web Scraping with Python: Successfully Scrape Data from any Website with the Power of Python." Packt Publishing, 2015.

[7]. R. Mitchell, " Web scraping with Python: Collecting more data from the modern web (2nd ed.). O'Reilly Media,2020.

[8]. S.D.S. Sirisuriya, " A Comparative Study on Web Scraping". Proceedings of the 8th International Research Conference, KDU.56, 2015.

[9]. S. Vasudevan and P. Megala, (2019). Scraping and visualization of product data from e-commerce websites. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol.8, no.11S, pp.436–439, 2019.

[10]. S. Zhang, L. Yao, A. Su and Y. Tay, "Deep learning-based recommender system: A survey and new perspectives," *ACM Computing Surveys*, vol.52, no.1, pp.1-38, 2019. https://doi.org/10.1145/3285029.

[11]. A. Shaikh, R. Khan, K. Panokher,M.K. Ranjan andV. Sonaje, " E-commerce Price Comparison Website Using Web Scraping**,"** I*nternational Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences* Volume vol.11, no.3,pp.1-13, 2023. http://dx.doi.org/10.37082/IJIRMPS.v11.i3.230223.

[12]. 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.*4. no.* 2, pp.127-136, 2024. https://doi.org/10.62051/ijcsit.v4n2.18.

[13]. P. Jankaew and W. Thamviset, "Web Scraping-based System for E-commerce Price Comparison and Similar Product Segmentation," *Journal of Applied Informatics and Technology*, vol. 7. no. 2, pp.346-362, 2025. Mahasarakham University, Thailand. http://jit.it.msu.ac.th

[14]. H. Lan, D. Sha, A.S. Malarvizhi, Y. Liu, Y. Li, N. Meister, Q. Liu, Z. Wang, J. Yang and C.P. Yang, "COVID-Scraper: An open-source toolset for automatically scraping and processing global multi-scale spatiotemporal COVID-19 records," IEEE Access, 9, 84783–84798, 2021. https://doi.org/10.1109/access.2021.3085682.