|| Volume 9 || Issue 10 || October 2025 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH
AND ENGINEERING TRENDS

Retail Purchase Intelligence System

Sakshi Shivaji Godse[1],Vishakha Rajendra Ganore2,Vyankatesh Gopaldas Bairagi[3],Darshan Yogesh Kangane[4],Abhay Gaidhani[5]
Student, Computer Engineering,Sandip Institute Of Technology and Research Center Nashik(SITRC) [1 2 3 4]
Prof, Computer Engineering,Sandip Institute Of Technology and Research Center Nashik(SITRC)[5]
sakshigodse19@gmail.com[1],vishakhaganore14@gmail.com[2],Vyankateshbairagi.dev@gmail.com[3],kanganedarshan06@gmail.com[4], abhay.gaidhani@sitrc.org[5]

Abstract: The growing popularity of e-commerce platforms has transformed consumer behavior, with modern buyers increasingly relying on digital channels to compare prices before making purchases. However, manual price checking across multiple websites remains inefficient, time-consuming, and error-prone. This research presents a Retail Purchase Intelligence System, an automated price comparison framework that aggregates product pricing information from various e-commerce sources and displays it in a unified interface. The system utilizes web scraping techniques through Python libraries such as Beautiful Soup and Requests, combined with a centralized MySQL database for structured data storage. A lightweight front-end interface built with HTML, CSS, and JavaScript enables intuitive search and quick visualization of comparative results. Experimental validation demonstrates that the system can accurately extract and normalize pricing data across multiple online retailers, significantly reducing consumer effort and time in finding optimal deals. The proposed model also outlines scalability for dynamic websites through Selenium-based scraping and highlights future extensions such as price-trend analysis, alert notifications, and browser integration. Overall, the system provides an effective, low-cost solution for real-time price intelligence and contributes to advancing consumer-centric automation in digital retail.

Keywords:*E-commerce, Web Scraping, Price Comparison, Python, Data Aggregation, Consumer Intelligence, Online Retail, Automation*

I.INTRODUCTION:

The explosive growth of e-commerce has revolutionized the way consumers interact with products and sellers worldwide. According to global retail reports, more than 80% of online buyers compare prices across multiple stores before finalizing a purchase. This behavioral pattern highlights a key consumer priority—**price optimization**—which continues to shape competition among online retailers.

In traditional practice, buyers manually browse through numerous e-commerce platforms such as Amazon, Flipkart, or Snapdeal to verify product prices, reviews, and delivery options. While this method ensures user control, it is inherently **time-consuming**, prone to **human error**, and lacks **real-time visibility** of market fluctuations. The need for an automated solution that streamlines this process has become evident as online inventories and product variations increase rapidly.

Recent studies on digital purchasing behavior reveal that consumers spend up to 30% of their decision time comparing prices rather than evaluating quality or reviews [1], [2]. Platforms like *Google Shopping* and *Honey* partially address this gap, yet they often face limitations when it comes to region-specific retailers, product normalization, and scraping dynamically rendered pages.

The proposed work—**Retail Purchase Intelligence System**—aims to automate and enhance this process through a scalable, lightweight, and easily deployable solution. The system employs **web scraping and HTML parsing techniques** to extract structured data such as product names, pricing, and URLs from multiple sources. Extracted data are stored in a centralized **MySQL database** and presented to users via a responsive **web-based interface** built with HTML, CSS, and JavaScript.

II.LITERATURE REVIEW

Recent research in the e-commerce domain emphasizes automation of price aggregation to improve user decision-making. Beranek and Remes [1] proposed a comparative price-network model highlighting cross-site synchronization challenges in multi-vendor systems. Wen [2] analyzed big-data-driven pricing discrimination and showed how sellers dynamically adjust product costs using online analytics. Bao *et al.* [3] explored pricing coordination between manufacturers and retailers, emphasizing the impact of supply-chain service levels on end-user prices.

Shaikh and Patel [4] implemented a Python-based price-comparison website using **Beautiful Soup** for static HTML parsing, establishing the feasibility of lightweight data-extraction tools. Alam *et al.* [5] extended this by designing **UPOMA**, a localized comparison platform for Bangladeshi markets that integrates live scraping with a responsive interface.

Several studies examined **product matching** and **entity normalization**, key to aligning identical goods across heterogeneous sites. Li *et al.* [6] and Tóth *et al.* [15] introduced deep-learning frameworks using multimodal and text-based embeddings to enhance accuracy in cross-platform product identification.

Further, Choudhary and Thakur [8] and Patil *et al.* [6] compared multiple web-scraping frameworks, revealing that hybrid static–dynamic extraction via Selenium improves success rates on JavaScript-rendered pages. Complementary works by Harshitha

et al. [14] and Harini *et al.* [15] emphasized integrating user-review analytics and alert systems to enrich consumer insight.

Overall, the literature confirms strong demand for unified, automated, and scalable solutions. However, persistent issues remain concerning dynamic content handling, rate-limiting ethics, and structured data normalization—gaps directly addressed by the proposed **Retail Purchase Intelligence System**.

III. PROBLEM STATEMENT

In the current digital marketplace, consumers frequently rely on multiple e-commerce platforms to compare prices before making a purchase. However, this manual process of visiting several websites, searching for identical products, and tracking prices across different layouts is **inefficient, error-prone, and time-consuming**. Users must juggle numerous browser tabs, face inconsistent product naming conventions, and often miss timely deals due to fragmented information flow.

While existing solutions such as **Google Shopping** or **price-comparison extensions** offer partial automation, they often fail to include **region-specific websites**, cannot handle **dynamically rendered pages**, and lack transparency in data-sourcing. Moreover, frequent changes in web structures (DOM elements, anti-bot mechanisms) make these tools unreliable and limited in scalability.

Therefore, there exists a **clear research and practical gap**:

to design and implement an automated, accurate, and extensible price-comparison framework capable of aggregating real-time product data from multiple e-commerce websites, normalizing it into a unified structure, and presenting it through a user-friendly web interface.

The proposed **Retail Purchase Intelligence System** aims to bridge this gap by leveraging **Python-based web scraping**, **centralized data storage**, and **interactive visualization**, thereby minimizing consumer effort and enabling informed, data-driven purchase decisions.

IV. OBJECTIVES

The primary goal of this research is to design and develop an automated system that simplifies online product price comparison by integrating data extraction, aggregation, and visualization into a single unified platform. To achieve this, the following specific objectives have been defined:

1. **To design an automated data extraction engine**
   Develop a robust web-scraping module using Python libraries such as *Requests*, *Beautiful Soup*, and *Selenium* to systematically collect product names, prices, and URLs from multiple e-commerce platforms.
2. **To implement a centralized data management system**
   Utilize a structured **MySQL database** to store and organize the extracted data efficiently, enabling faster querying, updates, and historical analysis of price variations.
3. **To develop an intuitive and responsive user interface**
   Create a **web-based dashboard** using HTML, CSS, and JavaScript that allows users to enter a product query and view real-time price comparisons across different retailers in a visually clear and interactive format.
4. **To ensure seamless navigation and user interaction**
   Integrate direct redirection links to the corresponding retailer pages, ensuring users can quickly proceed to purchase from their preferred vendor.
5. **To evaluate system accuracy and performance**
   Measure the efficiency of the scraping mechanism, response time, and data consistency across multiple sites to validate the system's reliability.
6. **To establish scalability and future extensibility**
   Design the framework to support future features such as **price history tracking**, **price-drop alerts**, and **integration with browser extensions or mobile applications** for enhanced user convenience.

These objectives collectively aim to deliver a **lightweight, real-time, and user-oriented price comparison system** that contributes to smarter consumer decision-making and efficient digital retail analytics.

IV. SYSTEM ARCHITECTURE

Overview.
The Retail Purchase Intelligence System follows a modular, client–server design that transforms a user's product query into a unified price-comparison view. The pipeline comprises a browser-based **Web UI**, a Python-based **Scraper Engine** for data acquisition (static and dynamic pages), a **Normalizer & Aggregator** for schema alignment and de-duplication, and a **MySQL Data Store** for persistence. An **Admin Panel** maintains site-specific selectors and operational logs. External **E-commerce Sites** act as data sources. This decomposition aligns with the synopsis scope and technology choices (Python, HTML/CSS/JS, MySQL)

1. **Web UI (HTML/CSS/JS)** — Accepts search terms, triggers backend requests, and renders a tabular comparison of retailer prices. It also provides deep links to the original product pages for quick purchase
2. **Request Handler / API Layer** — Validates input, queues scraping jobs, applies rate limits, and returns structured responses to the UI.
3. **Scraper Engine (Python)** —
   - **Static pages:** Uses HTTP requests + HTML parsing to extract *title, price, URL*.
   - **Dynamic pages (JS-rendered):** Falls back to a headless browser driver when required.
   - **Resilience:** Selector versioning, retry with exponential back-off, and polite delays (robots awareness).
   (Choice of parsing and dynamic rendering

follows the literature on web scraping for e-commerce.) [4], [5], [12], [13]

4. **Data Normalizer & Aggregator** — Cleans and standardizes fields (currency, units), tokenizes titles, and applies lightweight product-matching rules (brand + model + key attributes). It merges duplicates and ranks offers by price/recency.
5. **MySQL Data Store** — Stores raw extracts and normalized records; supports incremental updates and audit logs. The synopsis specifies MySQL for persistence
6. **Admin Panel** — UI to manage site configurations (CSS selectors/XPaths), throttle settings, and error logs, enabling quick adaptation to retailer DOM changes
7. **External E-commerce Sites** — Authoritative sources of product and price data; accessed read-only for comparison

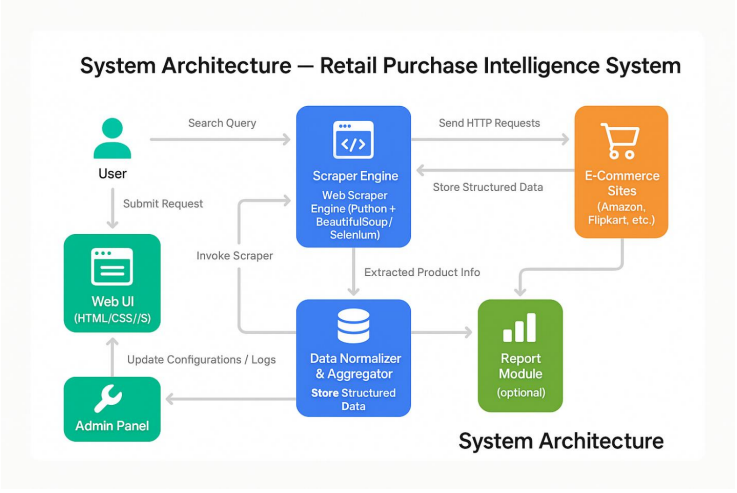

Figure 1 illustrates the high-level architecture used in implementation

V.REFERENCES

[1] L. Beranek and R. Remes, "E-commerce network with price comparator sites," in *Proc. 9th Int. Conf. on Advanced Computer Information Technologies (ACIT)*, 2019, pp. 401–404. doi: 10.1109/ACITT.2019.8779865.

[2] P. Wen, "Seller's pricing discrimination strategies under adoption of online big data technology," in *Proc. Int. Conf. on E-Commerce and E-Management (ICECEM)*, 2021, pp. 201–206. doi: 10.1109/ICECEM54757.2021.00047.

[3] F. Bao, D. Chen, H. Miao, and Y. Chen, "TPL service level and pricing models in manufacturer–retailer supply chains," in *Proc. Int. Conf. on Management of E-Commerce and E-Government (ICMeCG)*, 2010, pp. 333–337. doi: 10.1109/ICMeCG.2010.75. ijirmps.org

[4] S. M. Shaikh and A. Patel, "E-commerce price comparison website using web scraping," *Int. J. Innovative Research in Multidisciplinary and Professional Studies (IJIRMPS)*, vol. 11, no. 3, pp. 45–49, 2023. [Online]. Available: https://www.ijirmps.org/papers/2023/3/230223.pdf. doi: 10.37082/IJIRMPS.v11.i3.230223.

[5] A. Alam, A. A. Anjum, F. S. Tasin, M. R. Reyad, S. A. Sinthee, and N. Hossain, "Upoma: A dynamic online price comparison tool for Bangladeshi e-commerce websites," in *Proc. IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 194–197. doi: 10.1109/TENSYMP50017.2020.9230862. [PDF].

[6] N. Patil *et al.*, "Product price comparison across several e-commerce websites," *IJIRSET*, vol. 13, no. 5, pp. 1759–1765, 2024. [PDF].

[7] P. Rathod, M. Jadhav, and P. Pawar, "E-commerce with price comparison, price alert and fake review detection," *IJARCCE*, vol. 10, no. 1, pp. 67–72, 2021. [PDF].

[8] A. Choudhary and S. Thakur, "Web scraping-based product comparison model for e-commerce," *JETIR*, vol. 11, no. 4, pp. 238–244, 2024. [PDF].

[9] A. Kumar and S. Gupta, "Product comparison in e-commerce," *IRJET*, vol. 11, no. 3, pp. 1140–1146, 2024. [PDF].

[10] V. B. Raj *et al.*, "Price Probe: E-commerce platforms using machine learning," *IJFMR*, vol. 6, no. 3, pp. 1–9, 2024. [PDF]. IJFMR

[11] W3C, "WebDriver (Level 2)," W3C Candidate Recommendation, Sept. 2025. [Online]. Available:

[12] Selenium Project, "WebDriver documentation," Nov. 2024. [Online]. Available: https://www.selenium.dev/documentation/webdriver/ Selenium

[13] L. Richardson, "Beautiful Soup 4.13 documentation," 2025. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ crummy.com

[14] Y. Li, J. Li, Y. Suhara, A. Doan, and W.-C. Tan, "Deep entity matching with pre-trained language models," *Proc. VLDB Endowment (PVLDB)*, vol. 14, no. 1, pp. 50–60, 2021. doi: 10.14778/3421424.3421431. [PDF].

[15] S. Tóth *et al.*, "End-to-end multi-modal product matching in fashion e-commerce," *arXiv preprint* arXiv:2403.11593, 2024. [Online]. Available: https://arxiv.org/abs/2403.11593 arXiv