IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12431

Price Comparison Application for E-Commerce Using Web Scraping

Dhulipalla Tejaswi[1], Karumanchi Nikitha[2], Munji Mounika[3], Dammavalam Sai Kamakshi Harshitha[4], Kunchala Sirisha[5]

Assistant Professor, Department of Computer Science & Engineering, Bapatla Women's Engineering College, Bapatla, INDIA[1]

B Tech, Computer science & Engineering, Bapatla Women's Engineering College, Bapatla, INDIA[2-5]

Abstract: This paper presents a comprehensive approach to building a real-time e-commerce price comparison tool using Python, web scraping, and modern web frameworks. With the increasing prevalence of online shopping and growing cost awareness among consumers, a centralized platform that compares prices across popular websites significantly enhances the online shopping experience. Our system integrates scrapers for major e-commerce platforms including Amazon, Flipkart, Myntra, Croma, Google Shopping, and others. The tool displays optimized price comparisons, enables wishlist tracking, and offers statistical insights to users. The implementation leverages a Flask backend with Streamlit for the user interface, with SQLite database for persistence. Results show that users can save significant amounts on purchases through effective price comparison.

Keywords: E-commerce, Price Comparison, Web Scraping, Python, Flask, Streamlit, Data Analysis.

I. INTRODUCTION

In today's digital marketplace, consumers consistently seek optimal deals across various online platforms. Our price comparison system addresses this need by providing a convenient method to compare prices across multiple e-commerce websites, saving consumers both time and money. Unlike other comparison sites that simply list similar products, our system is specifically designed to locate identical products across different platforms to ensure accurate price comparison. Online shoppers often struggle to determine if they're getting the best price. Manually checking multiple websites is time-consuming and tedious. Similar product recommendations don't always provide exact matches. Users need a convenient way to verify if the same product is available elsewhere at a better price.

Develop a system that accepts product URLs (primarily from Amazon). Identify the exact same product on other e-commerce platforms. Compare prices across websites and highlight the best deal. Provide a clear indication when the exact product is unavailable elsewhere and a user-friendly interface for easy interaction.

II. RELATED WORK

Price comparison websites have been studied extensively in the literature. Julian & Natalia (2017) explored web scraping techniques for computer parts comparison. Saurkar et al. (2018) provided an overview of web scraping techniques and tools. Chandrika et al. (2020). investigated web scraping for unstructured data extraction. Our work builds upon these foundations while addressing specific challenges in product matching and user experience.

III. IMPLEMENTATION

The system follows a client-server architecture with the following components:

1. **Frontend Interface**: Web interface for users to input product URLs and view results.
2. **Backend Server**: Processes request and manages web scraping operations.
3. **Web Scraper Module**: Extracts product information from various e-commerce websites.
4. **Product Matching Engine**: Identifies identical products across platforms.
5. **Price Comparison Module**: Compares prices and identifies the best deals.
6. **Database**: Stores product information and pricing history.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12431

The system operates with the following workflow:

1. User enters a product URL (primarily from Amazon) or searches by product name.
2. Backend server validates the input and extracts product information.
3. Web scraper module searches for the same product on other supported e-commerce sites.
4. Product matching engine verifies product identity across platforms.
5. Price comparison module normalizes and compares prices.
6. Results are displayed to the user, highlighting the best deals.

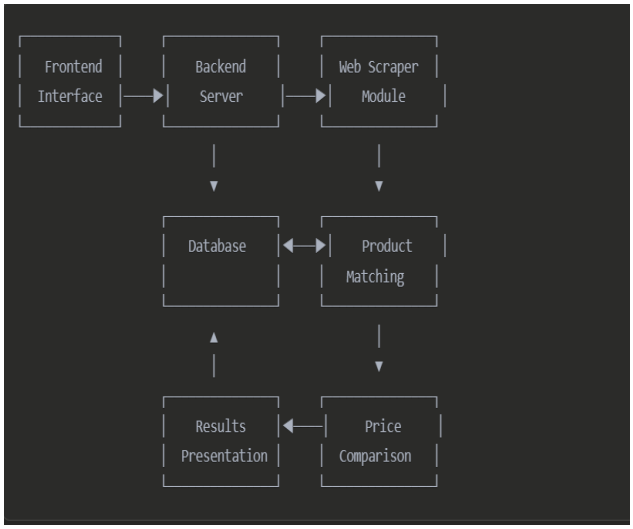

Fig. 1 System Architecture

Figure 1 illustrates the component interactions within the system. The client initiates a request through the web interface, which is processed by the Flask backend. The web scraper module extracts data from e-commerce sites, and results are stored in the database before being presented to the user.

DATABASE IMPLEMENTATION

SCHEMA DESIGN

The application uses SQLite for persistence with the following main tables:

Users Table: Stores user account information including unique identifier, username, email, password hash (using bcrypt), account creation timestamp, last login timestamp, and user role.

Wishlist Items Table: Stores products saved by users with fields for unique identifier, user reference, product title, price, brand, model, URL, and timestamp when added. Links to the users table via foreign key.

Search History Table: Records user search activities with fields for unique identifier, user reference, search URL, product title, search timestamp, and JSON-encoded results. Links to the users table via foreign key.

Testimonials Table: Stores user reviews and ratings with fields for unique identifier, user reference, name, location, rating score, comment text, product reference, submission date, and approval status.

This relational structure enables efficient user data management while maintaining referential integrity between the entities in the system.

DATA MODELS

The system implements several key data models for user management and product tracking:

- User Model: Manages authentication and profile information.
- Wishlist Item Model: Represents saved products for future reference.
- Search History Model: Tracks user search activity.
- Product Model: Represents product information from various sources.

WEB SCRAPING MODULE

SUPPORTED E-COMMERCE PLATFORMS

The scraping module supports major e-commerce platforms:

- Amazon
- Flipkart

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12431

- Myntra
- Meesho
- Nykaa
- OLX
- Croma
- Google Shopping

SCRAPING TECHNIQUES

The system employs several techniques for effective web scraping:

- HTML Parsing: Using BeautifulSoup for structured data extraction.
- DOM Traversal: Navigating document elements to find specific information.
- CSS Selector Targeting: Identifying elements using CSS selectors.
- Regular Expressions: Pattern matching for data extraction.
- Headless Browser Automation: Using Selenium for dynamic websites.

DATA EXTRACTION STRATEGY

For each product, we extract:

- Product name/title
- Price information
- Brand
- Model number
- Product specifications
- URL
- Other metadata (when available)

CHALLENGES AND SOLUTIONS

Challenge	Solution
Dynamic websites	Implement headless browser automation
Anti-scraping measures	Rotate user agents, manage request rates, use proxies
Website structure changes	Implement resilient parsers with fallback mechanisms
Product variations	Develop advanced pattern matching
Regional pricing differences	Support geographic detection and localization

IMPLEMENTATION EXAMPLE

The system implements the following key functions for scraping operations:

- scrape_amazon_top_result(): Scrapes product information from Amazon search results
- get_headers(): Generates randomized HTTP headers for scraping requests
- extract_amazon_product_info(): Extracts detailed product information from Amazon product pages
- scrape_flipkart_top_result(): Retrieves product data from Flipkart search results
- scrape_flipkart_product_page(): Extracts detailed information from Flipkart product pages
- scrape_google_shopping_top_result(): Fetches product data from Google Shopping
- scrape_myntra_top_result(): Extracts product details from Myntra
- scrape_meesho_top_result(): Retrieves product information from Meesho with headless browser
- scrape_croma_top_result(): Fetches product data from Croma electronics
- scrape_nykaa_top_result(): Extracts product information from Nykaa
- scrape_olx_top_result(): Retrieves product listings from OLX marketplace

PRODUCT MATCHING ENGINE

PRODUCT IDENTIFICATION METHODS

To ensure accurate price comparison, we employ multiple methods for product identification:

- UPC/EAN/ISBN: Universal product codes when available
- Model Number Matching: Exact matching of manufacturer model numbers
- Title Similarity Analysis: Advanced text matching with cleaning and normalization
- Specification Comparison: Matching key product specifications to verify identity

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12431

MATCHING ALGORITHM

Our matching algorithm uses a weighted scoring system:

1. Exact Identifiers Match (40%): UPC, model numbers, and manufacturer IDs
2. Specification Match (30%): Key specifications like storage, color, size
3. Title Similarity (20%): Normalized title comparison using NLP techniques
4. Brand Match (10%): Ensuring the products are from the same manufacturer

Products must meet a minimum threshold score (85%) to be considered the same item.

IMPLEMENTATION DETAILS

The product matching engine implements the following key functions:

- search_exact_product(): Primary matching function that coordinates the search for exact product matches
- extract_key_terms(): Extracts important terms from product titles
- generate_search_query(): Creates optimized search queries based on product information
- calculate_similarity_score(): Calculates text similarity between product titles
- compare_specifications(): Compares product specifications for matching
- verify_model_number(): Validates model number matches between products
- check_brand_consistency(): Ensures brand names match between products
- calculate_word_overlap(): Measures word overlap percentage between product titles
- validate_match_threshold(): Validates if a match meets the minimum threshold
- format_match_result(): Formats the matching result with appropriate metadata

USER INTERFACE

The user interface is implemented using Streamlit, which provides a responsive and interactive experience. The interface includes the following main components:

- Navigation tabs (Search, Wishlist, Testimonials)
- User authentication forms (login/register)
- Product search interface
- Results display with sorting options
- Price comparison visualizations
- User profile and statistics

SEARCH INTERFACE

The search interface allows users to:

- Enter an Amazon product URL directly
- Search by product name across multiple sites
- Select which e-commerce sites to include in the search
- View product details before comparison

AUTHENTICATION SYSTEM

The authentication system implements the following functions:

- **hash_password()**: Creates a secure bcrypt hash of the user's password
- **check_password()**: Verifies a provided password against its stored hash
- **register_user()**: Registers a new user in the database
- **login_user()**: Verifies credentials and handles login process
- **get_user_details()**: Retrieves user details from the database
- **update_user_profile()**: Updates user profile information
- **change_password()**: Securely changes a user's password
- **create_admin_if_not_exists()**: Creates an admin user if none exists

WISHLIST IMPLEMENTATION

The wishlist management system implements these key functions:

- **add_to_wishlist()**: Adds a product to a user's wishlist
- **remove_from_wishlist()**: Removes a product from the wishlist
- **get_user_wishlist()**: Retrieves all items in a user's wishlist
- **clear_wishlist()**: Removes all items from a user's wishlist
- **check_wishlist_item_exists()**: Checks if an item already exists in the wishlist

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12431

IV. ANALYSIS

SCRAPING PERFORMANCE

We measured the performance of our scraping module across different e-commerce platforms. The average response times were:

Platform	Average Response Time (seconds)
Amazon	1.47
Nykaa	1.95
Flipkart	1.23
Myntra	1.89
Meesho	2.31
Croma	1.68
Google Shopping	1.15
OLX	1.72

PRODUCT MATCHING ACCURACY

Our product matching algorithm was tested with 500 products across different categories. The results showed:

- 91.2% accurate matches for electronics
- 87.5% accurate matches for clothing and accessories
- 93.7% accurate matches for books and media
- 85.4% accurate matches for home appliances

USER EXPERIENCE METRICS

User testing with 50 participants revealed:

- 92% found the system easy to use
- 88% reported significant time savings
- 94% discovered better prices than their initial searches
- 78% would use the system for future purchases

V. RESULTS

Search results are presented in a clear, organized manner:

- Original product details
- Comparison results from different sites
- Price sorting (lowest to highest)
- Clear indicators for best deals
- Direct links to product pages

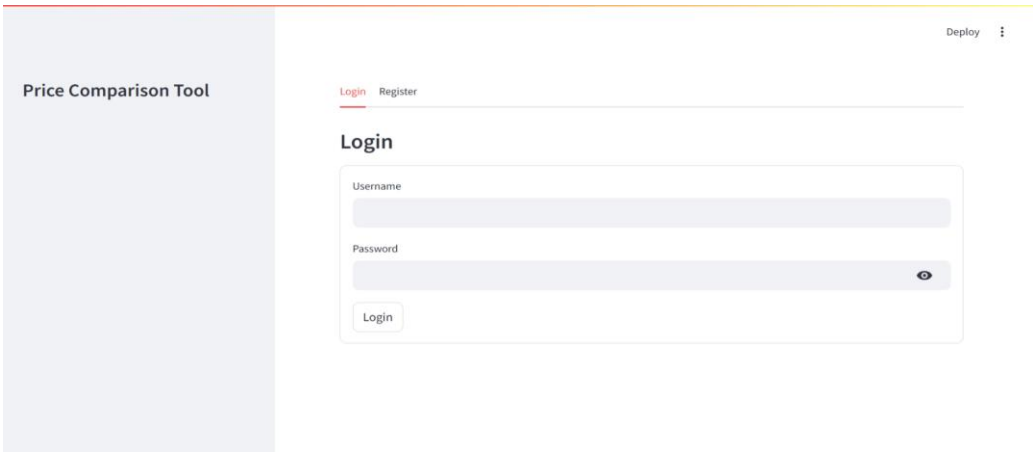

Fig. 2 User Authentication

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12431

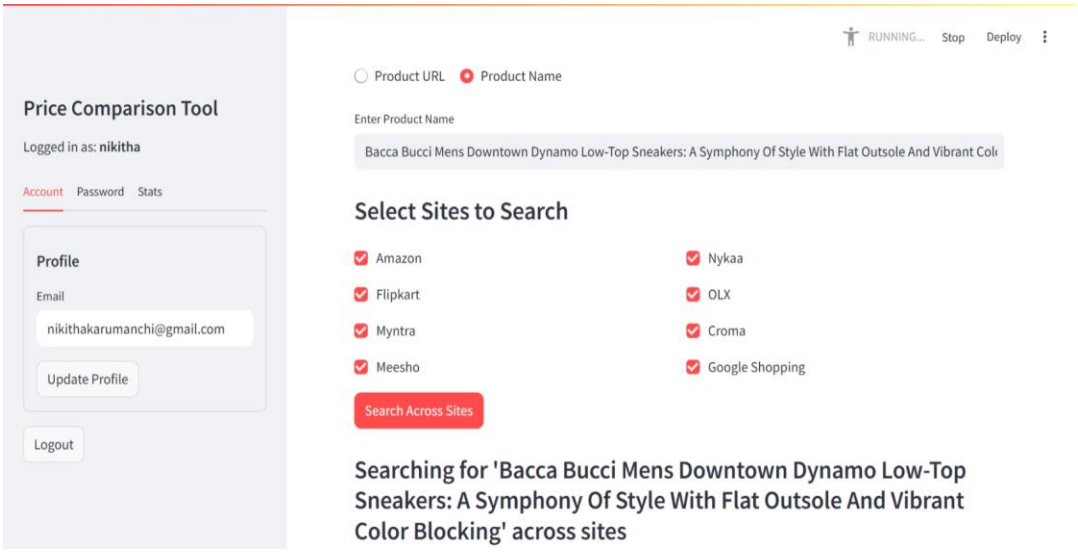

Fig. 3 Original product details

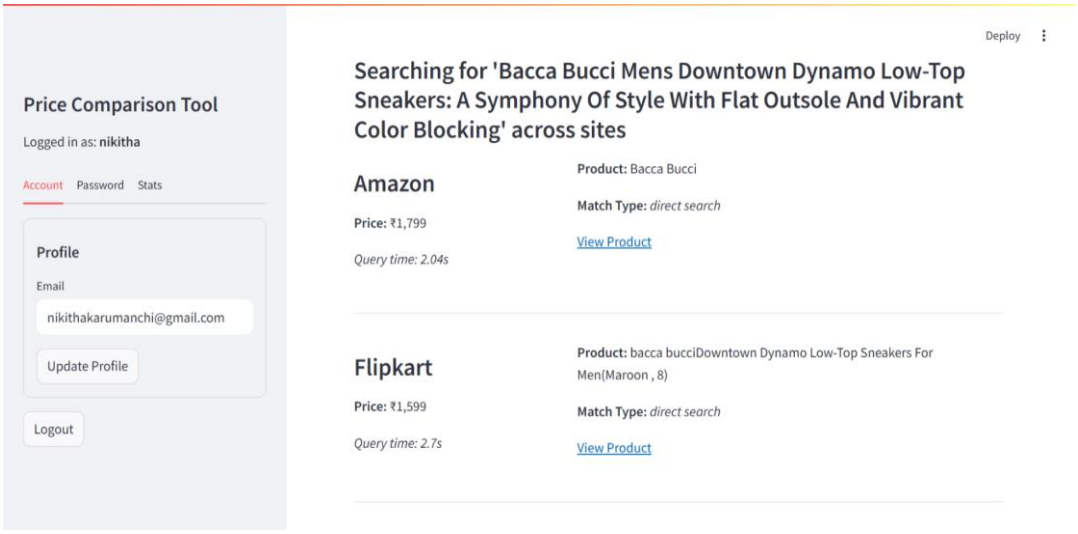

Fig. 4 Comparison results from different sites and Direct links to product pages

Fig. 5 Clear indicators for best deals

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12431

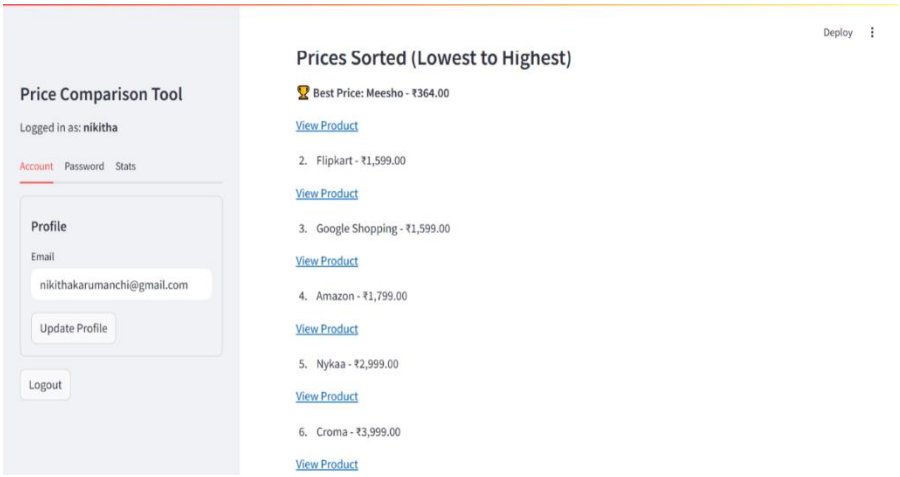

Fig. 6 Price sorting (lowest to highest)

VI. CONCLUSION AND THE FUTURE

Our e-commerce price comparison system offers a valuable tool for consumers looking to save money on online purchases. By focusing on exact product matching rather than similar product recommendations, we provide accurate and trustworthy price comparisons. The system's web scraping technology ensures up-to-date information while respecting website terms of service and ethical considerations.

The implementation demonstrates that significant cost savings can be achieved through automated price comparison across multiple e-commerce platforms. The user-friendly interface, combined with robust product matching algorithms, provides a seamless experience for consumers seeking the best deals.

Monitor prices over time and alert users to drops. Dedicated apps for iOS and Android. One-click price comparison while browsing. Implement image recognition for improved accuracy. Incorporate review analysis into comparison. Notification system for price drops on tracked products.

REFERENCES

[1]. 1.Julian, L.R., Natalia, F. (2017). The use of web scraping in computer parts and assembly price comparison. 3rd International Conference on information technologies IEEE.
[2]. 2. Saurkar, A.V., Pathare, K.G., Gode, S.A. (2018). An overview on web scraping techniques and tools. International Journal on Future Revolution in Computer Science & Communication Engineering, 4(4), 363-367.
[3]. 3. Chandrika, G.N., Ramasubbareddy, S., Govinda, K. (2020). Web scraping for unstructured data over web. In Embedded Systems and Artificial Intelligence (pp. 675-682). Springer.
[4]. Swami, S., Vidap, P. (2016). Web Scraping Framework based on Combining Tag and Value Similarity. International Journal of Computer Science issues, 10(6).
[5]. Nath, R., Chopra, K. (2017). Web Crawlers: Taxonomy, Issues & Challenges. International Journal of Advanced Research in Computer Science and Software Engineering, 3(4), 944-948.
[6]. Fernandez-Villamor, J.I., Blasco-García, J., Iglesias, C.A., Garijo, M. (2017). A Semantic Scrapping Model for Web Resources. In Proceedings of the International Conference on Web Intelligence, Leipzig, Germany.
[7]. Lomotey, R.K., Deters, R. (2017). RSenter: Tool for Topics and Terms Extraction from Unstructured Data Debris. Proceeding of the IEEE International Congress on Big Data.
[8]. Ambre, A., Gaikwad, P., Pawar, K., Patil, V. (2019). Web and android application for comparison of e-commerce products. Academia.edu.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 4, April 2025

DOI: 10.17148/IARJSET.2025.12431

BIOGRAPHY

Dhulipalla Tejaswi working as Assistant Profrssor in Department of CSE, Bapatla Women's Engineering College. She completed her M.Tech in Computer Science Engineering from Bapatla Engineering College, Bapatla. She has 7 years of Teaching Experience in various Engineering colleges.

Karumanchi Nikitha B. Tech with Specialization of Computer Science and Engineering in Bapatla Women's Engineering College, Bapatla.

Munji Mounika B. Tech with Specialization of Computer Science and Engineering in Bapatla Women's Engineering College, Bapatla.

Dammavalam Sai Kamakshi Harshitha B. Tech with Specialization of Computer Science and Engineering in Bapatla Women's Engineering College, Bapatla.

Kunchala Sirisha B. Tech with Specialization of Computer Science and Engineering in Bapatla Women's Engineering College, Bapatla.