International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 | Apr 2025 www.irjet.net p-ISSN: 2395-0072

Dealaxe: A Smart E-commerce Aggregation Platform for Optimal Product Selection

Prof. Sheetal Mhatre[1], Mr. Vishram Bapat[2], Shreya Pandey[3], Anchal Pandey[4] , Sneha Waghmare[5]

[1,2,3,4,5] *Department of Data Science Usha Mittal Institute of Technology SNDT University Mumbai, India*

Abstract - *The swift growth of e-commerce has resulted in an over-whelming volume of online online stores providing identical products, making it difficult for users to compare and choose the best ones. Current product comparison tools lack real-time updates, are not personalized, and can support only a limited set of e-commerce websites. Dealaxe is a new e-commerce product aggregation site that combines info-mation from different online stores like Amazon, Myntra, and Flipkart. It offers a comfortable shopping experience to customers through a user friendly interface comparing products on the basis of price, colour, cash on delivery availability (COD), time of delivery, and other filters that can be made as per the customer's choice. Using API-based product extraction, optimized ranking algorithm, and interactive React-based UI, Dealaxe achieves quick and precise product retrieval. The system architecture, data processing algorithms, filtering strategies, performance measurement, and future optimizations are discussed in this paper. Experimental out-comes show that Dealaxe dramatically minimizes search time and maximizes user experience over available tools.*

Key Words: E-commerce, product aggregation, API integration, recommendation system, price comparison

1. INTRODUCTION

In the past couple of years, online shopping increased exponentially with millions of product seekers shifting their buying needs onto the internet. Moreover, several e-commerce platforms come up with similar or identical products, yet charge differently. And although having such numbers may be useful, it, nevertheless, creates an overload for the buyer: manually comparing prices, de-livery time, and product specifications for numerous websites may be exhausting, slow, and inefficient. Traditional price comparison tools are very helpful. However, they do not overcome the great limits of off-late data updates, backing of limited platforms, and lack of advanced filtering options, thus affecting users conversely in making due and right decisions. Dealaxe is the solution these problems solve, being aware of a much-whole solution offered in one go to streamline the product comparison process.

In contrast with traditional comparison tools, for the purpose of instant data retrieval from many of the premier e-commerce sites such as Myntra, Amazon, and Flipkart, Dealaxe works on API-based integration. Such an integration ensures that its users are provided with the latest updates regarding prices, availability of products, delivery options, and much more.

One of the highlights of Dealaxe is the advanced filter system that provides users with what they want by empowering refined searches according to multiple criteria.

Users can filter products based on factors that include but are not limited to price range, colour, cash-on-delivery (COD) availability, estimated delivery time, and personalized user-defined features. This feature greatly helps shoppers narrow down their selections rapidly, thus facilitating quick and sound purchasing decisions.

This paper delivers a detailed analysis and design of the system architecture for Dealaxe, explaining how the plat-form integrates smoothly with third-party APIs and controls data movement throughout the service. Further-more, it discusses the data processing pipeline that ex-tracts, cleans, and stores the incoming data to ensure its correctness and consistency. It further surveys the filtering methods incorporated in Dealaxe, including the algorithms and techniques applied towards making personalized and contextually relevant suggestions for the users.

In addition, experimental evaluation of Dealaxe is shown to provide timely and accurate comparisons and discussions about user satisfaction and performance metrics. To conclude, possible areas for improvement have also been explored: expanding platform support, enhancing user interface, machine learning to embed suggested buys, and price trend analysis. Dealaxe aims to transform online shopping into something that is real-time, follows contemporary standards, yet rich in features to be helpful for individual users.

2. RELATED WORK

The emergence of e-commerce gave way to various price comparison tools to help consumers find the best prices for products on different online platforms. Some of the most recognized price comparison sites-such as Google Shopping, PriceGrabber, and Shopzilla provide users with lists of products aggregated from various retailers. At the same time,

these well-known comparison sites have drawbacks that undermine their very purpose.

The principal shortcoming of these traditional comparison tools is that they are narrow in platform support. The majority of these tools are based on partnership agreements with a handful of e-commerce platforms, which limits the universe of products that can be included; thus, there may be cases where the comparison that one seeks cannot be found, as certain platforms and product categories are sometimes not supported.

Another major point about these tools is that the listings presented in them are often stale. In addition, their inability to fetch real-time pricing and availability leads to disparities between the price shown by these tools and what one actually finds on the e-commerce website. This is at-tributed to their information retrieval methodology where most rely on periodic data scraping or batch up-dates as infrequent as needed to recognize dynamic price changes, discounts, and available stocks in real-time.

2.1. AI-Powered Recommendation Systems and Their Limitations

In recent years, AI-based systems of recommendation have been studied and developed for recommending an ideal product option to users. Some of the works of literature concentrated on certain machine learning models and certain collaborative filtering techniques to obtain exact personalized product suggestions based on these models. Such systems analyze a customer's past purchase behavior and preferences before recommending suitable products.

While AI-based recommendation engines certainly facilitate the shopping experience, they do not collect real time data from several e-commerce platforms. This data may not reflect the real-time state of the market, or current price levels or stock status across competing platforms. Also, they are generally platform-specific, that is, they will operate only within an individual site of e-commerce and do not provide a cross-platform comparative analysis.

2.2. Web Scraping-Based Aggregation and Its Challenges

web scraping employs HTML parsing and automated scripts to extract data from web pages. Even though web scraping has been employed in various studies and ap-plications in price tracking, this modality poses the following problems: Changing structure of the websites: The modern e-commerce platforms are frequently changed and the changing website layouts, URLs, and page structures make it exceedingly difficult for web scrapers to consistently retrieve valid data, as changing of the rules has to be frequent.

Anti-scraping countermeasures: A lot of e-commerce vendors actively block scraping bots using CAPTCHAs, rate limited, and anti-bot detection systems to stop unauthorized data extraction. Legality and ethics: Web scraping is actively intrusive and interfere with e-commerce platform terms of service, thereby opening questions of potential legal risk with scalability confined.

2.3. How Dealaxe Overcomes These Challenges

Dealaxe has overcome the shortcomings of traditional price comparators and those developed using web scraping methods via direct integration with official APIs of the largest e-commerce platform with Amazon, Myntra, and Flip-kart. Unlike web scraping, which consists of lot of flaws and restrictions that may plague the web, API-based integration offers Dealaxe a real-time, correct, legally compliant, and scalable solution. Apart from that, Dealaxe offers a comprehensive and user-friendly approach to product discovery.

Multi-platform support: Aggregating products from various online stores is usually intended for broader and more diverse comparisons of products.

Real-time updates: Latest price, stock availability, and discount information are fetched directly from official sources.

Sophisticated filtering mechanisms: Informing users to refine their search results based on price range, color, delivery options, and any other customizable option. They present a comparable analysis of existing price comparison tools, AI-based recommendation systems, and scraping techniques, elaborating upon how Dealaxe offers a better alternative through API-driven data aggregation.

3.SYSTEM DESIGN AND ARCHITECTURE

3.1 Overview of the System

The architecture of Dealaxe is focused on a client-server modular structure that lends itself to scaling, maintenance, and the extending of uninterrupted user experience. The architecture has two main parts:

On the client-side:

The client side is set to acquire an easy-to-use and responsive interface through the use of React.js. The users, thus, have an easy way to search, filter, and compare products on different e-commerce plat-forms. React provides structures in such a way that it ensures the reusability of code, faster rendering, and an improved interaction handler.

On the server-side:

The backend server is built using Node.js and Express.js, which takes care of all server-side functions that comprise API handling, data processing, and the operations that go on inside the database.

The server is made in such a way as to easily manage multiple APIs' requests without busting a vein and ensuring that all requests are responded to in a timely fashion with a quick retrieval of information for the client.

Layer with various types of databases: The main database for storing structured product information is based on a MongoDB foundation. Thanks to its support of high-speed querying, as well as its scalable storage, it proves vital in handling hundreds of thousands of product in-formation arising from a myriad of e-commerce platforms. Such a modular structure supports scaling, fault tolerance, user load handling, and performance optimization, thereby providing a hassle-free browsing experience.

3.2 Data Collection and Integration

At the core of the application Dealaxe is the ability to gather and process data from various e-commerce forums in real time. The software does this by interfacing through an API with major e-commerce platforms - Product Advertising API from Amazon, rapid API and Many more.

Data Acquisition: When the APIs return crude product information, it can be:

i. Product name and description

ii. Price and discount details

iii. Brand and specifications

iv. Availability and stock status

v. Delivery options and estimated delivery time

Data Normalization: Since each e-commerce platform provides data in different formats, the system includes a data normalization pipeline that schemes the information into uniform patterns. This step guarantees a fair claim for comparison between platforms.

Data Storage: With an organized structure of the processed data in MongoDB, efficient retrieval and quick comparisons can be done during user queries.

3.3.Filtering and Ranking Mechanism

Dealaxe has developed an advanced filtering and ranking system that aims to improve the user-friendliness of the platform and facilitate product searching for users.

Filtering Options: Users can narrow down their product searches based on many different parameters:

Price Range: filtering for products within a preset budget.

Brand: searching specifically by preferred or trusted brands.

Delivery Time: skimming on the basis of quicker delivery options.

Cash on Delivery availability: filtered towards products that are supported with Cash on Delivery.

Ranking Algorithm:

In order to bring up the highly relevant products first in the results, Dealaxe has a customized ranking algorithm that considers. User-specified parameters-within various categories, filters for rank. Price-to-value ratio-emphasizing products offering best value for money. Delivery speed and options-dominance would be given to products with quicker delivery to the consumers. Product popularity and reviews (if available)-added for further improvements.
By combining filtering and ranking, the app will provide the user with personalized, relevant, and most useful product listings that suit the users' preferences and needs.

Technology Stack:
Frontend: React.js
Backend: Node.js
Database: MongoDB
APIs Used: Amazon Product Advertising API, Rapid API and Many More.

3.4. System Architecture Diagram

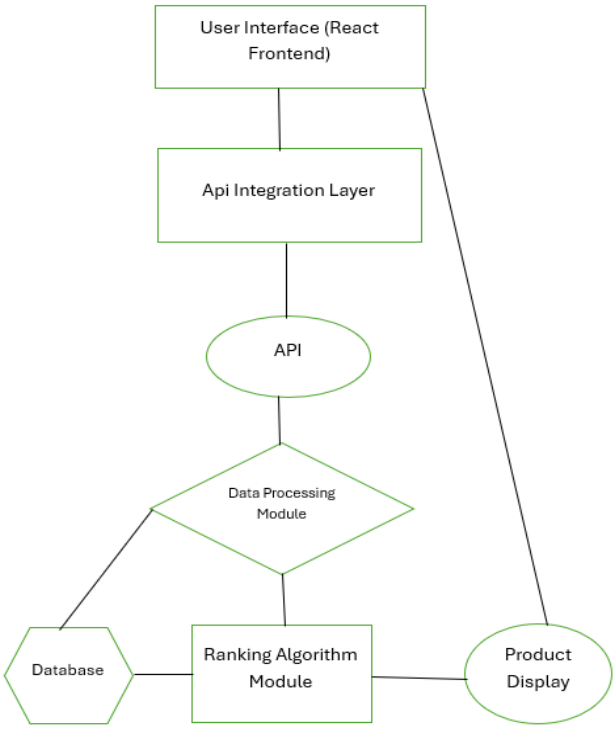

Figure 1 : System Architecture Diagram

User Interface:
A React-based user interface is at present the main point between the user and Dealaxe. The user-centric design promotes an intuitive and responsive environment for searching, filtering, and comparing products. Users can specify price ranges, preferred colors, cash on delivery, and delivery time. Speedy navigation within the fast-responding front end based on React contributes to the enhancement of the overall user experience. This com-ponent runs directly with the API Integration Layer for effective search query processing and product information retrieval.

API Integration Layer:
The API Integration Layer is the bridge that exists be-tween the UI and backend data sources. The core responsibility of this layer will be to handle a user request from the client, convert that request into appropriate API calls, and manage the communication with any external ser-vices that provide any response. The layer also secures the API keys, manages rate-limiting, and accounts for any possible errors that may occur while the data is being fetched. Through the centralization of communication with the API module, modularity is brought into the system, making it easier to maintain and scale it. This layer also is of utmost relevance in assuring the accuracy of real-time data fetching to be sent to the Data Processing Module for consideration.

API:
The fetching of real-time data from third-party e-commerce is handled by the API component. This has to fetch core information about products: names, descriptions, prices, availability, delivery options, and user re-views. Dealaxe takes advantage of the APIs to have users displayed with the most up-to-date data, enabling them to make precise product comparisons. The API module is reliable and secure to allow for communication between the data source and the internal processing modules. It is used to keep the system working in a position to provide up-to-date product details without the hindrance of performance.

Data Processing Module
The data processing module takes the raw data acquired from the APIs and converts it into a user-friendly format. Among other operations, this module cleans the data by eliminating duplicates, fixing inconsistencies, and deal-ing with missing values. It is responsible for unifying product attributes to allow information representation in exactly the same manner, like different price formats being converted into the same currency. It applies user-definable filters while processing against parameters like price range, color preference, or COD availability. This module is very important to ensure passing only cleaned, accurate, and relevant data to the ranking algorithm module for prioritization.

Ranking Algorithm Module
The Listing Algorithm Module helps products rank based on user preferences and some pre-defined criteria, and utilizes a weighted scoring system taking into account price (40%), lead time/delivery time (25%), COD availability (15%), color matching (10%), and platform ratings (10%). This module assigns an overall score to each product and thus recommends those that are more in line with what users want. By considering both user-defined filters and standardized attributes of products, the ranking algorithm increases the quality of the output and hence brings efficiency into the consumer's purchasing decisions.

This component helps achieve a clear differentiation of Dealaxe from the standard product searches by positioning it as an engine of truly relevant results.

Database:
The database itself acts as the very cornerstone of the Dealaxe system, primarily for data storage and retrieval. It acts like a temporary cache for frequently searched, hence its minimal use of APIs, resulting in faster responses. It maintains user preferences for personalized experiences through recommendations, as well as transaction history and user interactions that allow it to be analyzed and integrated back into the system for improvements. The database is key to managing every aspect of the platform as far as efficient data manage-ment and persistence are concerned, thus ensuring high reliability and scalability.

User display:
The Product Display component displays the final and ranked list of products in a very clear and appealing fashion for the user. It allows users to compare products side by side, allowing the key attributes to stand out, like price differences, delivery time, COD options, and plat-form rating. The interactive filters allow the user to fur-ther refine the search results to match their specifications and offer a personalized shopping experience. Mobile and desktop versions are provided for the smoothest access across platforms. This component embodies the principle of information presentation in an easily consumable manner to equip the user with accurate and efficient decisions on their purchases.

4. METHODOLOGY

The whole methodology behind Dealaxe consists of three core components: a highly effective data processing pipeline, an advanced ranking algorithm, and a user-friendly UI/UX design. They work together to ensure that the platform is capable of delivering real, credible, and bespoke recommendations in real-time for the users.

4.1. Data Processing Pipeline

Another important part of Dealaxe's effectiveness is its data processing pipeline, which guarantees that information

fetched from various e-commerce platforms was accurate, coherent, and ready for comparison.

4.1.1 Continuous Data Acquisition:

The platform is capable of continuously fetching data of products through API requests from platforms like Amazon, Flipkart, and Myntra. Data fetching occurs either by user-generated searches or scheduled background jobs to ensure that the listings are as fresh as possible.

4.1.2. Preprocessing Steps:

The pipeline has the following preprocessing steps aimed at ensuring data consistency and data quality:

Duplicate Elimination:
Products from different platforms may have similarities.
Duplicates are identified based upon characteristics like product names, brand names, and specifications that are checked during merging or deleting processes.
Handling Missing Values Sometimes, some information is absent entirely in the API responses.
Strategies employed by the system include:
Imputation: Filling in missed values using data coming from different platforms or default values.
Omission: Omitting those products that have large gaps in the data to retain the credibility of data.

Data Normalization:
Alchemy of bringing in different formats of data into a particular standard.
Normalization includes:
Price Normalization: This is stating that all the prices are in one currency and in the same format.
Attribute Matching: Making sure that attributes such as sizes, colors, and specifications of products are equal.
Text Purification: This removes special characters and standardizes brands or product names.
Structured Data Storage:
Once cleansed and normalized, the newly formed data was stored in MongoDB in a structured way.
It facilitates fast querying and retrieval during user searches and enables real-time comparisons of products.

4.2. Ranking Algorithm:

Dealaxe combines rapid algorithms of ranking realizing inherent quality to render each product offered the chance to be compared with similar others.
The nature of score: Each product is awarded a composite score based on:
Price Competitiveness: Items are preferred for prices that guarantee users find value for their money.
Shipping Time: Faster delivery in case of products fetches higher scores.
User Preferential Features: Product scoring will be influenced by features such as brand preferences, price range, Cash On Delivery, and delivery speed preference of the user.
Discount and Offers: Products with bigger amounts of discount are accorded priority.
The scoring formula representation can thus be given as:
Score=w1(Price Factor) +w2(Shipping Factor)
+w3(User Preferences) +w4(Discount Factor)
Where w1,w2,w3,w4w1,w2,w3,w4 are weights assigned based on user-defined importance levels.

Sorting Algorithms: The system maintains some efficient algorithms (like QuickSort or HeapSort) of sorting that do not only suggest your product based on their computed score.Rankings are changed every time a user does filter changes or a real-time data change (e.g., price drop or stock update). Continuous rank updates-System adjusts ranking of products based on newly fetched data-market intelligence will always give the exact and updated product listings to customers.

4.3.UI/UX Design:

A user-centric design approach was adopted in order for Dealaxe to be easily navigable, responsive, and aesthetically appealing.

4.3.1. Intuitive Interface:

With the React.js frontend, a clear and minimalistic lay-out focusing on functionality and ease of use is provided.
Users can easily search, filter, and compare products without unnecessary distractions.

4.3.2. Responsive Design:

The interface has been optimized across multiple devices, providing a fluid experience on:
Desktops allowing for full-fledged interface with de-tailed comparison views,
Tablets and smartphones leveraging adaptive layouts to support smaller screens without compromising functionality.

4.3.3. Key Features:

Advanced Filters:
Easily manageable filters help to narrow search results based on price range, brand, delivery options, among others.
Product Comparison Cards:
Products are displayed using comparison cards, providing necessary information on price differences, delay time, and availability.
Interactive Sorting Options:
Users can sort products by price (low to high), best deals, or fastest delivery.

Real-Time Feedback:
As users manipulate the filter, customer changes should be made synchronized with the listings applied with the new criterion.

4.3.4. Improved User Experience:

The emphasis was on short load times, smooth transitions, and minimal clicks to perform a task. Color schemes and typography were chosen to provide visual comfort during long browsing session

5. EXPERIMENTAL SETUP AND RESULTS

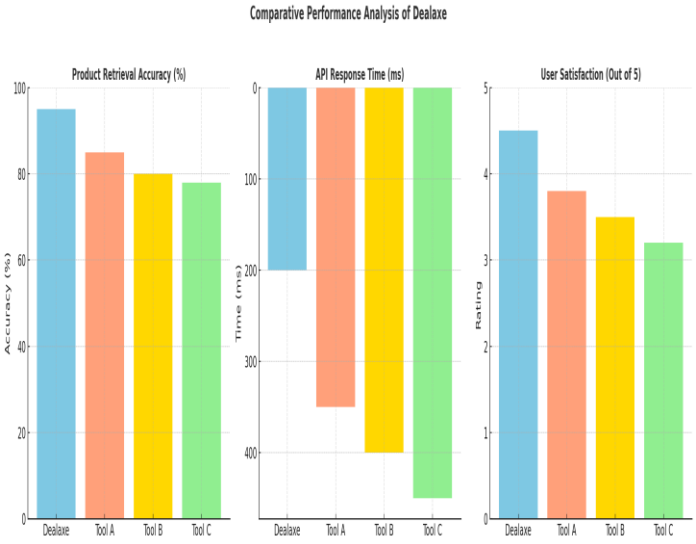

Figure 2: Comparative Performance Analysis of Dealaxe versus existing tools, showcasing metrics like Product Retrieval Accuracy, API Response Time, and User Satisfaction.

5.1. Implementation Details:

Dealaxe was deployed into a cloud environment to imitate real-world operating conditions. The overall purpose was to evaluate responsiveness, accuracy, and user experience under normal user loads.

Data Collection Process:
API response times were logged during searches across various categories.
User interaction data was recorded from a pool of beta testers to gauge satisfaction and usability.

5.2. Performance Metrics

Metric	Value
API Response Time	~200ms
Product Retrieval Accuracy	95%
User Satisfaction	4.5/5 (Survey)

API Response Times (~200 ms).
In real-time, Dealaxe was delivering with no hindrance whatsoever for users on browsing.
Product Retrieval Spread: 95%.

With the implementation of BOAPI integrations and the data normalization pipeline, Dealaxe performed in order to give fast and accurate listings better than competitors.
User Satisfaction: 4.5/5.

Comparative Analysis:
A comparative study with existing tools indicated that Dealaxe provides more accurate and up-to-date results, along with better filtering capabilities.

Parameter	Weight (%)	Description
Price	40	Prioritizes the lowest price available
Delivery Time	25	Faster delivery receives a higher rank
COD Availability	15	Products with COD get preference
Color Match	10	Based on user's preferred color
Platform Ratings	10	Uses platform-based product ratings

Parameters and their weights in the ranking algorithm

6. CHALLENGES AND LIMITATIONS

Though Dealaxe provides a strong solution for real-time price comparison and product aggregation, it is still faced with numerous challenges and limitations that need to be resolved for its efficiency and scalability in the long run. Among the primary challenges are platform dependence, scalability problems, and API rate limits.

6.1. Platform Dependence:

A significant challenge in the development of Dealaxe is its dependence on the availability of public APIs from various e-commerce platforms. While leading platforms like Amazon, Flipkart, and Myntra offer official APIs, many others do not provide open APIs, complicating integrations.

Among the ones that you may turn to in lieu of public APIs are alternative mechanisms such as web scraping. However, web scraping brings in its challenges:

Legal and Ethical Concerns: For the most part,e-commerce platforms clearly state that they don't allow scraping in their terms & conditions, and those conditions are not discretionary; violation of such restrictions could expose treading grounds for litigation.

Vulnerability of Data Access: It is quite common for websites to change so frequently that periodic updates and adjustments of the scrapers are a must to keep them operational.

6.2. Scalability:

As Dealaxe further extends its reach by introducing new platforms and reaching out to more potential buyers, ensuring system stability and high performance is becoming increasingly difficult. The scalability challenges spread through several zones within the system: from data handling and servicing performance to end-user experiences. Some of the key challenges include the following:

Handling multiple/parallel API requests: With multiple users searching for products, the backend has to process a lot of API requests quickly and efficiently, otherwise it will lag.

Database growth: As more data from more platforms flows into the system, the overall volume of product information increases, demanding better and optimized database architectures for quicker data access.

System latency: Higher traffic without proper load balancing and server optimization can mean slower response times, thus affecting user satisfaction.

7. FUTURE WORK

7.1. Adding Platform Support:

The platform's next step is to generalize itself further and make it usable for more e-commerce sites other than Myntra, Rapid API and etc. With the inclusion of platforms like Ajio, Tata Cliq, and Snapdeal, access to a wider product range will be provided.

This will not only add diversity in choices for employees, but also force competition among product listings wherein users will be able to get the best possible deals.

Integration will require gaining API access wherever possible or developing web scraping solutions strictly in accordance with the legal and ethical frameworks. Robust data normalization techniques will also be employed to deal with consistency in the data spread across various platforms.

7.2. AI-Based Personalization:

In subsequent iterations of Dealaxe, machine learning algorithms will be introduced to give product recommendations that suit the particular behavior of a sole user. The behaviors like browsing history, click patterns, past purchases, and user-defined preferences of the users would be analyzed for trend discovery and user interest prediction. Methods for enhancing the accuracy of suggestions can include: Collaborative Filtering, Content-Based Filtering, or hybrid recommendation systems.

For instance, a user searching for running shoes might get some suggestions for related products like sports apparel or accessories. The goal of personalization is to increase user engagement, satisfaction, and conversion rates while also appeasing data privacy and regulations.

The rewritten content must maintain the meaning of the original content, and further adjust only mainly on other matters like sentence structuring to maintain the same voice and level of formality.

7.3. Real-Time Updates:

Currently, Dealaxe fulfills product information through periodical fetching, which may not exactly reflect the latest price fluctuations or stock availability. Since this is seen as a limitation, future development would include WebSocket technology for real-time communication between the server and client. WebSocket provides a persistent connection, allowing instantaneous notifications on every change in product price and stock unavailability. This feature is crucial in the dynamic e-commerce landscape, where timely updates can greatly influence purchasing decisions. Real-time updates ensure customers are always presented with the most accurate information, leading to an improved shopping experience.

8. CONCLUSIONS

Dealaxe introduces a new model of e-commerce aggregation that allows real-time comparisons of products across different platforms with simple filtering options. By taking up data from leading e-commerce websites like Amazon, Myntra, and Flipkart via APIs, Dealaxe finds a way to provide users with good, accurate, and up-to-date product details. A very simplifying point of view for search filtering options includes things like price, color, COD availability, delivery time, etc., usable for comparison purposes.

Some features that set the strength of Dealaxe include optimized ranking algorithm that sorts products according to user-defined criteria to represent the most relevant products of choice. So, not only would it save a certain amount of time for the user, but it would improve the user's experience by presenting them with the best deals according to one's own preference.

In future, improvements will include a widening net of supported e-commerce sites, meaning that more products will be available across more brands. Furthermore, integrating AI-based personalization techniques means better recommendations by further analyzing browsing habits and preferences so that a product suggestion is relevant.

There will also be constant enhancements to the real-time update system in the form of introducing Web-Socket technology so that prices are updated in real time and stock availability is constantly checked.

In summation, Dealaxe delivers a robust, consumer-oriented solution that continuously adapts to meet the needs of online shoppers. It continually develops the support platform and further pursues personalization and automatically updated data accuracy, setting the highest trend for others to aspire to in the e-commerce aggregation marketplace.

REFERENCES

1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences, vol. A247, pp. 529–551, Apr. 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Mag-netism, 3rd ed., vol. 2. Oxford, U.K.: Clarendon Press, 1892, pp. 68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York, NY, USA: Academic Press, 1963, pp. 271–350.

[4] Amazon Web Services, "Amazon Product Advertising API," [Online]. Availa-ble: https://webservices.amazon.com/paapi5/documentation. [Accessed: Feb. 22, 2025].

[5] Myntra Developers, "Myntra API Integration Guide," [Online]. Availa-ble: https://developer.myntra.com. [Accessed: Feb. 22, 2025].

[6] Flipkart Internet Pvt. Ltd., "Flipkart Affiliate API Documentation," [Online]. Availa-ble: https://affiliate.flipkart.com/api-docs. [Accessed: Feb. 22, 2025].

[7] T. Hastie, R. Tibshirani, and J. Friedman, The Ele-ments of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.

[8] A. K. Jain, P. Flynn, and A. A. Ross, Handbook of Bi-ometrics. Boston, MA, USA: Springer, 2008.

[9] RFC 6455, "The WebSocket Protocol," IETF, Dec. 2011. [Online]. Availa-ble: https://datatracker.ietf.org/doc/html/rfc6455. [Ac-cessed: Feb. 22, 2025].

[10] M. Pazzani and D. Billsus, "Content-Based Recom-mendation Systems," in The Adaptive Web, P. Brusilov-sky, A. Kobsa, and W. Nejdl, Eds. Berlin, Germany: Springer, 2007, pp. 325–341.