

Smart Price Comparison and Best Deal Prediction

Saleha Farha¹, Telluri Sushma², Lonavath Swapna³, Tala Swetha⁴

¹Assistant Professor; Department of Information Technology Bhoj Reddy Engineering College for Women
Hyderabad India

^{2,3,4}B.Tech Student's; Department of Information Technology Bhoj Reddy Engineering College for Women
Hyderabad India

Accepted 20-04-2026

Author(s) Retains the Copyrights of This Article

sushmatelluri@gmail.com, lonavathswapna@gmail.com, Swethatala7@gmail.com

Abstract

The rapid growth of online retail platforms has created a need for efficient tools that assist consumers in identifying the most economical purchasing options. This paper presents a Smart Price Comparison Assistant, an intelligent web-based system developed to compare product prices across multiple e-commerce platforms and support informed buying decisions. The proposed solution integrates a Python-based backend for data processing, React and Node.js for interactive frontend and server-side functionalities, and MongoDB for scalable data management. The system enables users to search for products and receive consolidated results from different online marketplaces through an intuitive interface. To improve comparison accuracy, machine learning techniques are employed to recognize equivalent or closely related products even when naming conventions, specifications, or listing formats differ across platforms. In addition, the system examines historical and real-time pricing patterns to estimate favorable purchasing periods, allowing users to optimize spending. Beyond price analysis, the platform incorporates product ratings, customer reviews, discount availability, and seller credibility indicators to promote transparency and trust. The architecture is designed to ensure responsiveness, scalability, and efficient retrieval of updated information. By minimizing the time and effort required for manual price comparison, the proposed assistant offers a practical and data-driven approach to modern online shopping. The Smart Price Comparison Assistant demonstrates how intelligent analytics, predictive modeling, and user-centered design can enhance digital commerce experiences by delivering cost-effective recommendations and simplifying purchasing decisions.

Keywords: Price Comparison, Machine Learning, E-commerce, Product Matching, Price Prediction, Consumer Analytics, Smart Shopping.

Introduction

The expansion of electronic commerce has significantly transformed consumer purchasing behavior, making online shopping a routine activity worldwide. Customers can now access a vast range of products through multiple digital marketplaces such as Amazon, Flipkart, and other retail platforms. Although this accessibility provides convenience and variety, it also creates challenges for consumers when identical or comparable products are listed at different prices, discounts, and seller conditions. As a result, identifying the most beneficial purchase option often requires considerable time and effort. Traditional shopping comparison methods typically involve manually visiting several websites, examining prices, reviewing product specifications, and evaluating seller credibility. This process is inefficient and may lead to inconsistent decisions, particularly when products are presented under different names, descriptions, or packaging formats. Existing comparison tools offer partial support, but many rely on direct keyword matching and limited rule-based filtering, reducing their effectiveness in real-world e-commerce scenarios. To overcome these limitations, this study proposes a Smart Price

Comparison and Best Deal Prediction System. The system is designed to automatically gather product information from multiple online platforms and compare available offers through intelligent data analysis. Unlike conventional tools, the proposed solution applies machine learning methods to identify equivalent or closely related products even when listings vary in naming conventions or specifications. This improves matching accuracy and provides users with more meaningful comparisons. In addition to current price evaluation, the system incorporates predictive analytics to study historical pricing behavior and estimate future trends. By forecasting price fluctuations, the platform can recommend whether a product should be purchased immediately or at a later time for better savings. The system also considers additional factors such as promotional discounts, customer reviews, product ratings, and seller trustworthiness to generate balanced recommendations. The proposed framework aims to simplify online shopping decisions by reducing manual effort, improving transparency, and helping users obtain maximum value for their money. Through the integration of intelligent comparison techniques and

predictive modeling, the system contributes to a smarter and more efficient digital shopping experience.

Purpose of the Project

The primary objective of the Smart Price Comparison and Prediction System is to assist consumers in identifying the most economical and reliable purchasing options across multiple e-commerce platforms. Since the same product is often available from different sellers at varying prices, users may find it difficult to determine the best available offer. This project addresses the problem by employing machine learning algorithms to compare products intelligently rather than relying solely on exact text matching. The system analyzes product similarity, pricing behavior, discounts, ratings, and reviews to generate more accurate recommendations. It also evaluates historical and real-time pricing information to predict favorable purchasing periods. By combining automation, analytics, and predictive intelligence, the project seeks to enhance consumer decision-making, reduce search time, and improve cost savings in online shopping environments.

Existing System

In current practice, many users compare product prices manually by browsing several online shopping platforms individually. They must inspect product details, compare specifications, review customer feedback, and verify seller ratings before making a purchase. This approach is labor-intensive and inefficient. Several online comparison tools are available; however, many of them depend on exact product names or static filtering rules. Such systems often fail when similar products are listed with different titles, incomplete descriptions, or alternative model references. Furthermore, they usually focus only on present prices and do not provide future price trend analysis. Because of these limitations, users may overlook better offers, spend unnecessary time searching, or purchase products at non-optimal prices.

Proposed System

The proposed Smart Price Comparison and Prediction System introduces an intelligent framework for evaluating product offers across multiple e-commerce websites. The platform automatically retrieves product listings, compares available prices, and identifies equivalent items using machine learning-based similarity detection techniques. The system further analyzes historical and real-time pricing data to estimate future price movements and recommend the most suitable time to buy. In addition to price, the recommendation engine considers discounts, seller ratings, customer reviews, and product popularity to improve reliability. By integrating automated comparison, predictive analytics, and user-focused recommendations, the proposed system enables

faster, smarter, and more cost-effective purchasing decisions.

Related Work

Survey

To evaluate the practical demand for an intelligent price comparison platform, a structured survey was conducted among frequent online shoppers, including university students, salaried employees, and regular e-commerce users. These groups represent major segments of the target population for the proposed Smart Price Comparison and Prediction System. The objective of the survey was to understand consumer difficulties in comparing online prices, awareness of existing tools, and interest in predictive shopping assistance. The findings indicated that a large majority of respondents regularly purchase products through online marketplaces but encounter difficulty identifying the most economical option because prices, discounts, and seller conditions differ across platforms. Many participants stated that they manually browse websites such as Amazon, Flipkart, and similar marketplaces to compare offers. This process was widely described as time-consuming, repetitive, and confusing, particularly when the same product appears under different names or specifications. The survey further revealed limited awareness of currently available price comparison services. Among users who had experience with such platforms, many expressed dissatisfaction with their performance because most systems rely heavily on exact product-name matching and therefore fail to recognize equivalent products listed differently. A significant portion of respondents showed strong interest in a system capable of automatically detecting similar products and presenting accurate cross-platform comparisons. Another major observation was the demand for predictive purchasing support. Many participants reported that they would benefit from recommendations indicating the best time to buy a product based on previous price fluctuations and current market trends. Users also requested features such as live price updates, review summarization, seller trust indicators, discount alerts, and personalized recommendations. The survey results clearly demonstrate a strong market need for a machine learning-driven solution that simplifies online shopping decisions. Consumers increasingly expect tools that go beyond static comparisons and provide intelligent insights for saving time and reducing costs. These findings support the development of the proposed Smart Price Comparison and Prediction System.

Requirement Analysis

Requirement analysis defines the operational expectations, system capabilities, and technical resources necessary for implementing the proposed platform. The Smart Price Comparison and

Prediction System is intended to provide fast, reliable, and intelligent support for users seeking the best online purchase options across multiple e-commerce platforms.

Functional Requirements

The system must allow users to search for products using keywords, categories, or brand names and retrieve matching listings from multiple online marketplaces. It should compare prices, discounts, and delivery conditions in a consolidated interface. A machine learning-based matching module must identify equivalent or closely related products even when naming conventions differ across websites. The application should analyze pricing history and current trends to estimate future price movement and recommend whether the user should purchase immediately or wait for a better opportunity. Product details such as ratings, reviews, seller reputation, and availability must also be displayed to support balanced decision-making. The system should provide results quickly and maintain a user-friendly interaction process throughout all operations.

Non-Functional Requirements

The platform must be designed to ensure efficiency, security, scalability, and consistent usability. Response time should remain low even when processing large volumes of product data. Search requests, comparisons, and prediction outputs should be delivered with minimal delay to preserve user experience. Security is essential because user activity data, saved preferences, and system interactions must be protected from unauthorized access. Appropriate authentication mechanisms, encrypted communication, and secure database practices should be implemented. The architecture must support future growth in terms of users, product volume, and additional marketplace integrations. A modular design approach is recommended so that new features can be introduced without affecting core functionality. Usability is another critical factor. The

interface should be simple, intuitive, and understandable for both technical and non-technical users. Product comparisons, trend predictions, and recommendations should be clearly presented. Reliability requires the system to operate consistently under normal and peak usage conditions without crashes or data corruption. In the event of failures, the application should recover gracefully and present meaningful error notifications. Availability is necessary to ensure uninterrupted access. The system should remain operational for users with minimal downtime and should tolerate temporary network or server interruptions through robust service management.

Computational Resources

Software Requirements

The proposed system can be implemented using Python as the primary programming language due to its strong ecosystem for machine learning and web development. Libraries such as Scikit-learn, Pandas, and NumPy are suitable for data preprocessing, model training, and analytics. Frontend and server-side integration may be developed using React and Node.js. MongoDB or structured CSV datasets can be used for storing product records, historical prices, and user interaction data. Development environments such as Jupyter Notebook and Visual Studio Code are appropriate for experimentation and deployment.

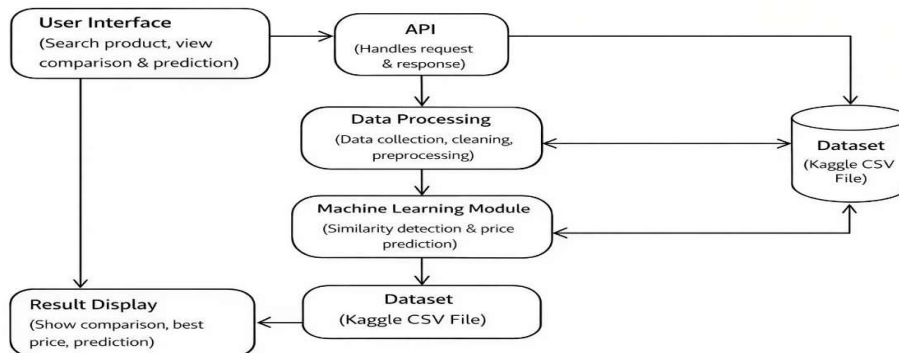
Hardware Requirements

A standard modern computer system is sufficient for development and moderate-scale deployment. Recommended specifications include an Intel i5 processor or equivalent, at least 8 GB RAM, 20 GB of available storage space, and a 64-bit operating system such as Windows, Linux, or macOS. Higher configurations may be required for large-scale real-time data processing and model training tasks.

Design

Architecture

System Architecture



System Architecture

System architecture defines the overall framework, major components, and operational workflow of the

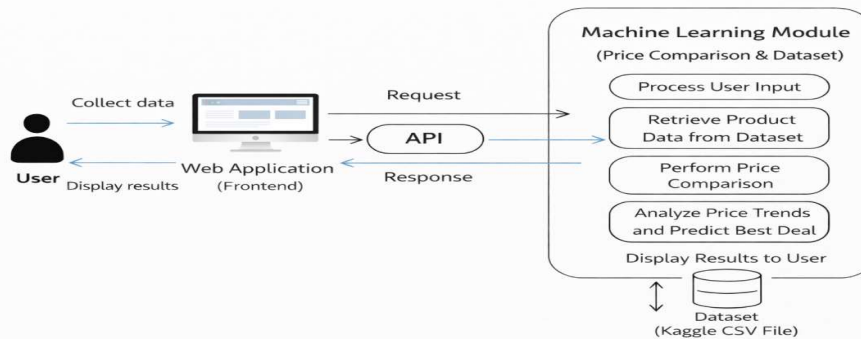
Smart Price Comparison and Prediction System. It explains how different modules interact to receive

Telluri Sushma *et. al.*, /International Journal of Engineering & Science Research

user requests, process product information, apply intelligent analysis, and generate meaningful outputs. The architecture is structured to support the complete lifecycle of a product search request, beginning with user input and ending with the presentation of ranked recommendations and price insights. The process starts when a user enters a product name, category, or keyword through the application interface. The request is forwarded to the backend, where the data acquisition module retrieves product information from multiple e-commerce platforms or previously stored datasets. Collected data may include product titles, prices, discounts, seller details, ratings, reviews, and availability status. After data retrieval, the preprocessing module cleans and standardizes the collected records. This stage removes duplicates, handles missing values, normalizes pricing formats, and converts inconsistent product descriptions into a structured representation suitable for analysis. Proper preprocessing improves the accuracy of subsequent machine learning tasks. The refined data is then passed to the intelligent analytics layer. In this module, machine learning techniques are used

to detect equivalent or closely related products listed under different names or descriptions. Similarity matching enables more reliable comparisons across different platforms. At the same time, predictive models analyze historical and current pricing patterns to estimate future price movement and identify favorable purchase opportunities. Once processing is completed, the recommendation engine ranks available offers based on price competitiveness, seller trustworthiness, user ratings, discount value, and predicted trends. The final results are displayed through the presentation layer in a clear and user-friendly format. The modular design of the architecture ensures maintainability, scalability, and operational efficiency. Individual modules can be upgraded independently without affecting the entire system. This structure also allows future integration of additional marketplaces, advanced recommendation models, or personalized shopping features. By maintaining organized communication among components, the system achieves dependable performance and fast response times.

Technical Architecture



Technical Architecture

Technical architecture focuses on the implementation technologies and software layers used to build the Smart Price Comparison and Prediction System. It explains how frontend services, backend processing, machine learning modules, and storage systems are combined to deliver a complete and efficient application. The frontend layer is responsible for user interaction and can be developed using modern web technologies such as React. This layer provides search functionality, product comparison views, filters, and dashboards for displaying recommendations and price forecasts. A responsive interface ensures compatibility across desktop and mobile devices. The backend layer manages business logic, request handling, and communication between system modules. Technologies such as Python and Node.js can be employed to support API services, workflow orchestration, and integration with external data sources. Backend services coordinate product retrieval, preprocessing pipelines,

recommendation generation, and prediction requests. The data processing layer performs extraction, cleaning, transformation, and feature engineering on collected marketplace data. Libraries such as Pandas and NumPy are suitable for handling structured datasets efficiently. This layer prepares information for machine learning tasks and ensures consistency in the stored records. The machine learning module forms the analytical core of the system. It may include classification, clustering, similarity matching, and regression models for product equivalence detection and price trend forecasting. Frameworks such as Scikit-learn can be used for model training, evaluation, and deployment. A database layer such as MongoDB stores product catalogs, historical price records, user preferences, search logs, and prediction outputs. NoSQL storage is beneficial because product information from multiple marketplaces may vary in structure and format. The technical architecture is designed to be scalable and flexible. As user demand

Telluri Sushma *et. al.*, /International Journal of Engineering & Science Research

increases, services can be distributed across multiple servers or cloud environments. New machine learning models, external APIs, or marketplace connectors can be integrated with minimal structural modification. This approach ensures long-term adaptability, faster processing, and sustained system reliability.

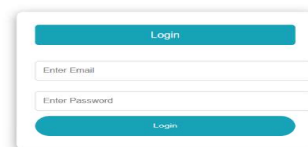
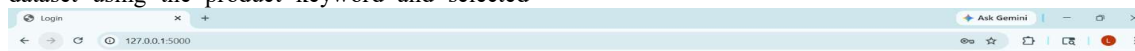
Implementation

Pseudo Code

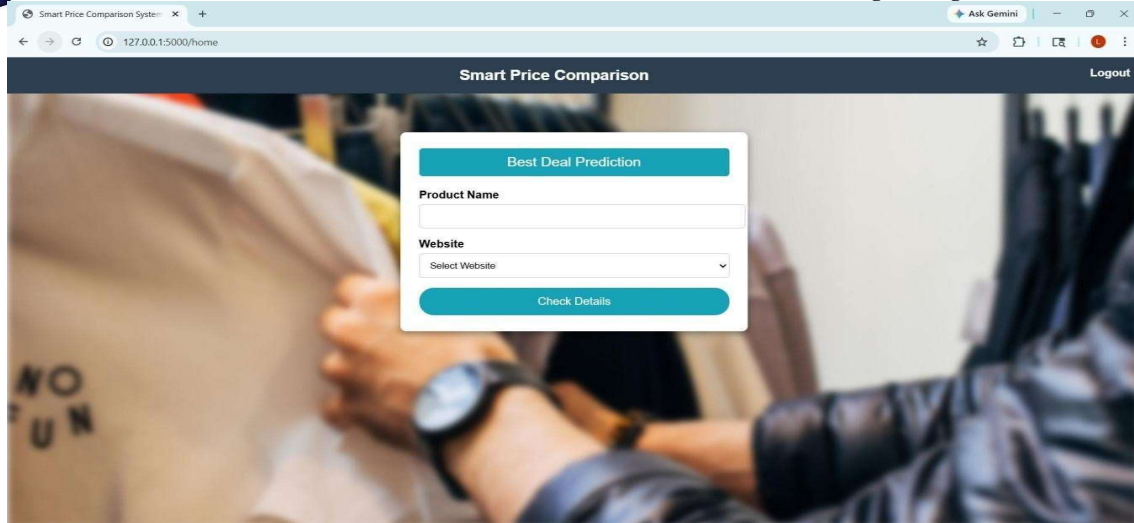
The Smart Price Comparison and Prediction System was implemented as a web-based application that combines frontend user interaction, backend data processing, dataset integration, and machine learning-based prediction. The implementation was organized into multiple modules to ensure maintainability, modularity, and scalability. The frontend interface was developed using HTML, CSS, and template rendering techniques. It consists of two major pages: a login page and a product comparison page. The login page authenticates users before granting access to the system. It includes input fields for email and password along with validation messages for incorrect credentials. After successful authentication, users are redirected to the main dashboard. The product comparison page serves as the primary user interface of the system. It allows users to enter a product name and select an e-commerce platform such as Amazon, Flipkart, or Ajio. Once the request is submitted, the system searches the integrated dataset and displays important product attributes including original price, discount percentage, discounted price, and customer rating. The interface is designed to present information clearly so that users can compare available offers with minimal effort. The backend was developed using the Flask framework in Python. Flask manages routing, session handling, form processing, and communication between the user interface and the analytical modules. A secure session mechanism ensures that only authenticated users can access system features. When a user submits a product query, the backend filters the dataset using the product keyword and selected

platform, then retrieves the most relevant product record. To support cross-platform comparison, the system temporarily stores prices retrieved from different websites during user interaction. Once product information from all supported platforms has been collected, the application identifies the lowest available price and determines the platform offering the best deal. This process automates manual comparison and reduces decision-making time for users. A rule-assisted prediction mechanism was integrated to estimate whether the current product should be purchased immediately or at a later time. The decision logic considers factors such as discount percentage, product rating, and current best price. Products with high discounts may be recommended for later purchase if a further reduction is likely, while highly rated products with moderate discounts may receive an immediate purchase recommendation. Dataset preparation was handled through a separate preprocessing module. Product records collected from Amazon, Flipkart, and Ajio were loaded individually and merged into a single consolidated dataset. A platform label was assigned to each source before combining all records into a unified file. This integrated dataset provides the foundation for search, comparison, and prediction operations. For predictive analytics, a machine learning training module was developed using Scikit-learn. The combined dataset was first cleaned and transformed, including conversion of percentage values into numeric form. Relevant input features such as actual price, discount percentage, rating, and rating count were selected. The target variable was the discounted selling price. The dataset was divided into training and testing subsets to evaluate generalization performance. A Linear Regression model was trained to learn relationships between pricing variables and final selling price. After successful training, the model was serialized and stored for future deployment within the application environment.

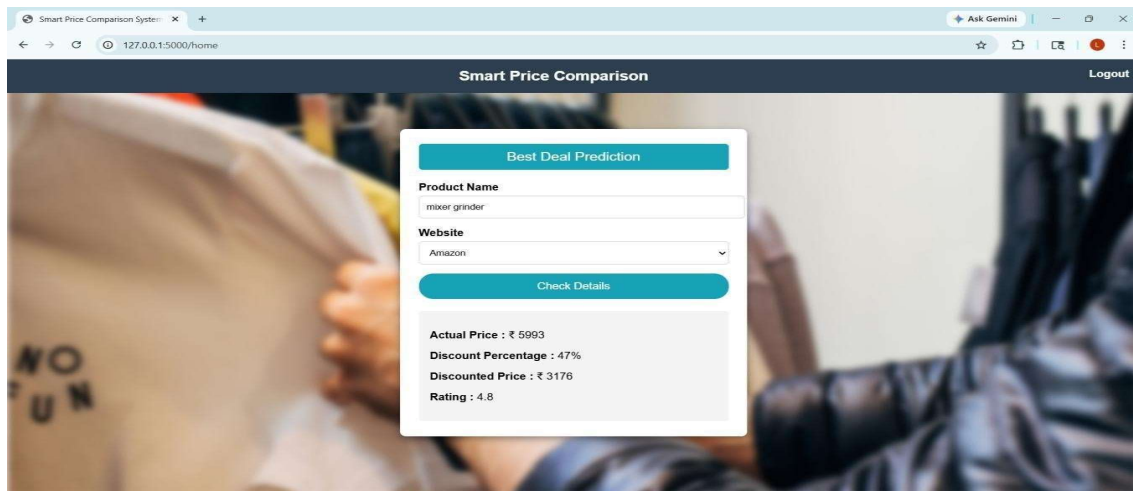
Screenshots



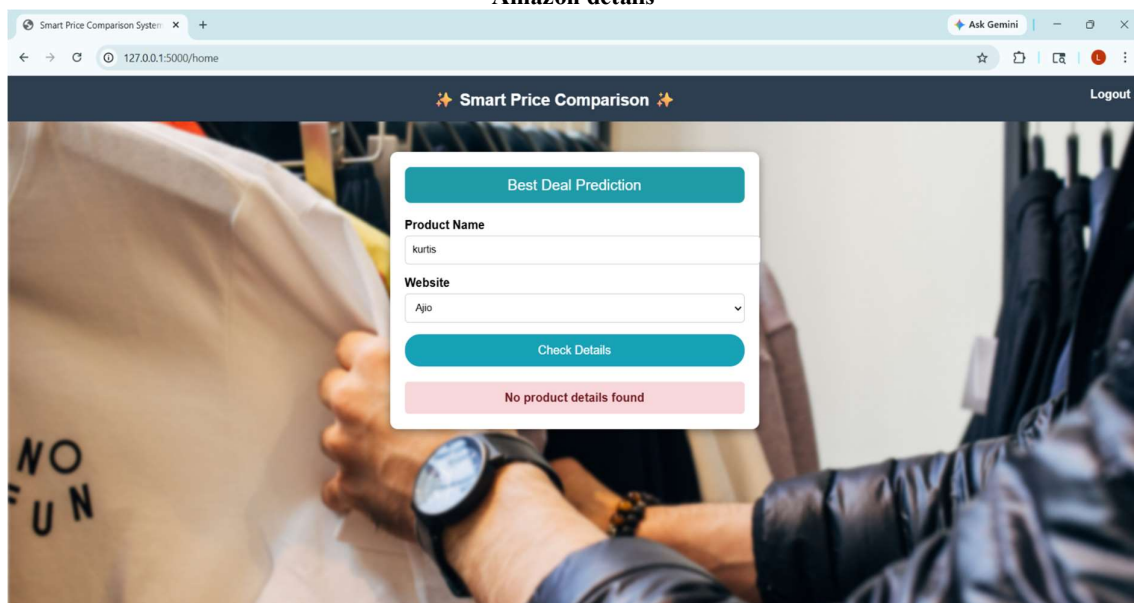
Login page



Home page



Amazon details



No details

Test Cases

System testing was conducted to verify the functionality, reliability, and performance of the Smart Price Comparison and Best Deal Prediction System. Different modules were evaluated using representative scenarios to ensure that the application performs as expected under normal and exceptional conditions. The product search module was tested with valid product names, and the system successfully displayed relevant product information retrieved from the dataset. When an empty search request was submitted, the application correctly prompted the user to enter a valid product name, confirming input validation support. The price comparison module was tested by selecting products available across multiple platforms. The system accurately displayed comparative pricing information and identified the most economical option. Similar product detection was also validated using products listed under different names, and the matching logic successfully recognized related items and presented them as comparable alternatives. The price prediction component was tested using available product pricing attributes. The module generated purchase recommendations indicating whether the product should be bought immediately or later, demonstrating correct execution of the decision logic. Performance testing was carried out using larger datasets to evaluate response time and scalability. The system maintained satisfactory speed and stable operation while processing increased data volume, indicating that the architecture can support moderate growth. One failure scenario was observed during testing of invalid or corrupted product records. In certain cases, incomplete data caused the application to terminate unexpectedly instead of displaying an appropriate error message. This issue highlights the need for stronger exception handling, input sanitization, and data validation mechanisms in future versions.

Conclusion

The Smart Price Comparison and Best Deal Prediction System presents an effective solution for assisting consumers in modern digital marketplaces where product prices frequently vary across platforms. By integrating multiple e-commerce sources into a single environment, the system enables users to search products efficiently, compare offers instantly, and identify the most economical purchasing option. The project reduces the effort traditionally required for manual comparison across several websites. Instead of individually reviewing different platforms, users receive consolidated product information, pricing details, ratings, and discount insights through a unified interface. This improves convenience, saves time, and enhances the overall shopping experience. An important contribution of the proposed work is the inclusion of

intelligent prediction mechanisms. By analyzing price-related factors and historical trends, the system can recommend whether users should purchase immediately or wait for potentially better pricing opportunities. Such predictive support transforms the platform from a simple comparison tool into a decision-assistance system. From a technical perspective, the solution combines a responsive frontend, efficient backend processing, structured data management, and machine learning techniques to ensure accuracy, scalability, and dependable performance. Core software quality factors such as usability, reliability, and efficiency were considered throughout development. In summary, the Smart Price Comparison and Best Deal Prediction System successfully achieves its objective of helping users maximize value during online purchases. It also establishes a strong foundation for future intelligent commerce applications involving automation, recommendation systems, and personalized shopping assistance.

Future Scope

The proposed system offers multiple opportunities for future enhancement. One important direction is the integration of advanced artificial intelligence models capable of producing more accurate price forecasts using larger historical datasets, seasonal trends, and market demand indicators. Personalized recommendation features can also be introduced by analyzing user browsing behavior, purchase interests, and preferred brands. This would allow the system to generate customized deal suggestions tailored to individual shopping habits. Real-time data synchronization with additional e-commerce platforms would improve coverage and ensure continuously updated comparisons. Expanding support to international marketplaces could further increase system usefulness. Future versions may also include voice-based search, multilingual interfaces, dedicated mobile applications, and browser extensions that automatically compare prices while users browse online stores. Automated notifications for discounts or price drops could significantly improve user engagement. Trust-oriented features such as fake review detection, seller credibility scoring, and return-policy comparison may further strengthen decision-making confidence. With these enhancements, the platform can evolve into a comprehensive smart shopping assistant powered by machine learning, predictive analytics, and real-time commerce intelligence.

References

1. Varian, H. R. *Intermediate Microeconomics: A Modern Approach*. W.W. Norton & Company, 2019.
2. Laudon, K. C., and Traver, C. G. *E-Commerce: Business, Technology, Society*. Pearson, 2020.

3. Aggarwal, C. C. *Recommender Systems: The Textbook*. Springer, 2016.
4. Han, J., Kamber, M., and Pei, J. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011.
5. Kotu, V., and Deshpande, B. *Predictive Analytics and Data Mining*. Morgan Kaufmann, 2014.
6. Tan, P. N., Steinbach, M., and Kumar, V. *Introduction to Data Mining*. Pearson, 2018.
7. Schneider, G. *Electronic Commerce*. Cengage Learning, 2017.
8. Ricci, F., Rokach, L., and Shapira, B. *Recommender Systems Handbook*. Springer, 2015.
9. Amazon India. Online Shopping Platform.
10. Flipkart. E-commerce Marketplace.
11. Google Shopping. Product Price Comparison Platform.
12. Razorpay. Secure Online Payment Gateway.
13. Beautiful Soup Documentation. Python Web Scraping Library.
14. GitHub Contributors. "Price Comparison and Web Scraping Projects," 2023.