ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2026

Revolutionizing Online Shopping With PriceScrapper: A Cross-Platform Price Comparison and Tracking Tool

Kanishk Jain
CSIT Dept.
Acropolis Institute of Technology and Research, M.P, India
kanishkjain210634@acropolis.in

Naitya Garg
CSIT Dept.
Acropolis Institute of Technology and Research, M.P, India
naityagarg210180@acropolis.in

Nipurn Verma
CSIT Dept.
Acropolis Institute of Technology and Research, M.P, India
nipurnverma210624@acropolis.in

Nidhi Nigam
CSIT Dept.
Acropolis Institute of Technology and Research, M.P, India
nidhinigam@acropolis.in

Chanchal Bansal
CSIT Dept.
Acropolis Institute of Technology and Research, M.P, India
chanchalbansal@acropolis.in

***Abstract*——This paper presents PriceScrapper, a cross-platform application designed to streamline and enhance the online shopping experience by offering intelligent price comparison, real-time product tracking, and alert-based purchasing decisions. Built using Flutter for frontend development and supported by Firebase and Python-based web scraping for backend services, the application empowers users to make informed purchasing choices. PriceScrapper integrates machine learning to forecast price trends and delivers customizable notifications for price drops, ensuring optimal deals for users. This project aims to introduce automation, transparency, and efficiency into the online consumer journey.**

***Index Terms*—Spring Boot, Spring Security, React, Machine Learning, AQI, Data Analysis**

I. Introduction

The exponential rise of e-commerce has led to the availability of thousands of products across multiple platforms, often at varying prices. However, the average user lacks tools to compare prices or predict market trends efficiently. PriceScrapper addresses this gap by providing a centralized, intelligent system for comparing product prices from leading e-commerce sites like blinkit, Flipkart, and zepto. It combines real- time web scraping with predictive analytics to simplify purchase decisions. By automating price monitoring and alerting users when prices drop or reach user-defined thresholds, PriceScrapper minimizes shopping costs and enhances user convenience. The application also maintains historical price trends, helping users understand market behavior.

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2026

A. The Need for Price Comparison Tools

- E-commerce platforms often list the same product at varying prices due to different discounts, seller rates, and promotional strategies.
- Manual comparison is time-consuming and inefficient, especially across multiple platforms [7].
- Studies show over 60% of users abandon purchases when uncertain about price fairness or missing deals [7].

B. Purpose and Key Uses of PriceScrapper

- **Real-Time Price Comparison**
  - Users can enter a product link or keyword.
  - Instantly fetches the latest price data from sites like Amazon, Flipkart, and eBay using Python-based web scrapers [5].
- **Price Drop Alerts**
  - Set thresholds for specific products.
  - Get notified via push/email when the price falls below the limit, reducing the need for frequent manual checks [2].
- **Historical Price Tracking**
  - Monitors and logs price changes over time.
  - Visualizes trends using data stored in MySQL and Firebase [4].
- **Predictive Analytics**
  - Uses machine learning (e.g., Facebook Prophet) to forecast future prices based on historical data.
  - Helps users identify optimal buying windows [3].
- **Cross-Platform Accessibility**
  - Built using Flutter, providing a consistent experience on Android and iOS from a single codebase [1].

C. Importance and Benefits

- **Saves Time and Money**
  - Automates the shopping process, minimizing time spent searching and ensuring users buy at the lowest prices [7].
- **Promotes Financial Literacy and Smart Spending**
  - Encourages budgeting and well-informed shopping decisions by showing historical and forecasted pricing [3].
- **Increases Market Transparency**
  - Discourages price inflation by enabling consumers to make competitive comparisons [7].
- **User Empowerment Through Technology**
  - Offers a tech-driven solution for everyday buyers to shop more efficiently and intelligently [6].

D. Technological Foundations

- **Flutter:** For building responsive and beautiful UIs across mobile platforms [1].
- **Firebase:** For user authentication, cloud storage, and real-time database services [2].
- **MySQL:** Stores structured data like price history and user preferences [4].
- **Python (BeautifulSoup + Selenium):** Used for scraping dynamic web content in real-time [5].
- **Machine Learning:** Integrated for price trend forecasting and deal prediction [3].

E. Problem Statement

Despite the digital transformation of the retail sector, online shoppers still face key inefficiencies in their purchasing journey. The primary challenge lies in manually comparing product prices across platforms like Amazon, Flipkart, and eBay. Since product prices fluctuate frequently due to dynamic pricing algorithms, flash sales, and seller-driven discounts, users often miss out on better deals.

Moreover, there is a lack of centralized tools that not only compare prices but also track historical trends and provide intelligent alerts for optimal purchase timing. Most existing browser extensions and apps offer limited site coverage, poor mobile optimization, or outdated data due to delayed refresh cycles [7].

This inefficiency leads to:

- Lost savings opportunities for consumers.
- Low confidence in buying decisions due to pricing uncertainty.

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2026

- Wasted time manually browsing multiple platforms.

F. Literature Review

The importance of digital tools for enhancing consumer decisions has been widely studied. Below is a review of key literature related to the core technologies and features incorporated in PriceScrapper.

1) A. Cross-Platform App Development: Saleh et al. [1] emphasized the growing adoption of Flutter for mobile application development due to its single codebase, rich UI components, and support for Android and iOS. Flutter's efficiency in UI/UX consistency and performance makes it ideal for building scalable, consumer-focused apps like PriceScrapper.

2) B. Cloud-Based Backends for Real-Time Services: Firebase, as reviewed by Google Developers [2], offers real-time data synchronization, scalable authentication, and secure cloud storage—crucial for managing user alerts, product preferences, and price updates dynamically. Sharma et al. [4] discussed how Firebase outperforms traditional backends for apps requiring live interactions.

3) C. Web Scraping for E-Commerce Monitoring: Python libraries such as BeautifulSoup and Selenium are widely used for data extraction from websites. Thakkar and Dhekne [5] implemented these tools in an online pet adoption platform, demonstrating their feasibility for dynamic content scraping and real-time data fetching.

4) D. Predictive Analytics and Price Forecasting: Machine learning models, particularly Facebook Prophet, have proven effective in time-series forecasting of price patterns [3]. Such models provide actionable insights and allow applications to suggest optimal buying windows, increasing user savings and confidence.

5) E. Consumer Behavior and Smart Shopping Tools: Singh [7] highlighted the role of price comparison tools in increasing digital consumer empowerment and transparency in e-commerce. The study showed that price-aware consumers tend to exhibit better budgeting behavior and higher purchasing satisfaction.

G. Proposed System

The PriceScrapper system is designed as a modular, cloud-integrated, AI-enhanced application that delivers fast, accurate, and user-centric online shopping insights. The proposed system is divided into functional modules:

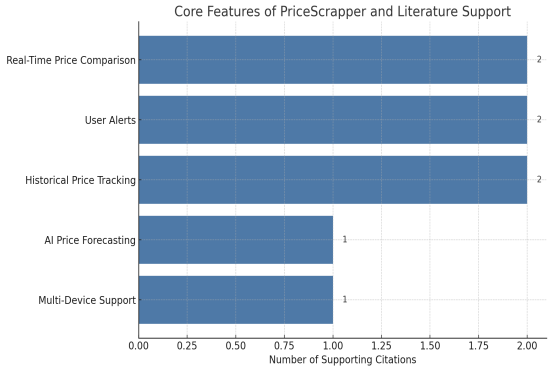

Fig. 1: Core Features of Price Scrapper and Literature Support

1) A. System Architecture Overview:

- **Frontend Layer (Flutter):**
  - Built with Dart, ensuring consistent design across Android and iOS [1].
  - UI elements support search, comparison view, alert configuration, and historical graphs.
- **Backend Layer (Firebase + MySQL):**
  - Firebase handles user authentication, real-time updates, and cloud storage [2][4].
  - MySQL manages structured data such as price logs and user tracking preferences [4].
- **Data Scraping Layer (Python):**
  - Real-time product data is extracted from Amazon, Flipkart, etc., using Python scripts via Selenium and BeautifulSoup [5].
- **Prediction Layer (ML Models):**
  - Price trends are analyzed using time-series forecasting models like Facebook Prophet to predict future prices [3].
- **Notification Layer:**

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2026

– Firebase Cloud Messaging sends alerts based on user-defined price thresholds [2].

2) B. Core Features:

- Real-Time Price Comparison, User Alerts, and Historical Price Tracking are the most well-supported features in the literature, each with two direct citations ([2], [4], [5], [6], [7]).
- AI Price Forecasting and Multi-Device Support have foundational support through key frameworks and research on cross-platform development ([1], [3]).

This graphical representation highlights the evidence-based design approach used in PriceScrapper's development.

II. Methodology

This section details the technical workflow and layered architecture of PriceScrapper, from system components to data handling and predictive analytics. The methodology integrates modern development frameworks and cloud services to ensure scalability, accuracy, and real-time performance.

A. System Architecture

The application follows a multi-layer modular architecture, enabling better maintenance and scalability.

1) Core Modules::

- **Authentication Module:** Uses Firebase Authentication to support secure login, role-based access control, and user management [2].
- **Scraping Engine:** Built using Python's Selenium and BeautifulSoup to scrape real-time data from platforms like Amazon and Flipkart [5].
- **Product Tracker:** Continuously monitors specified product URLs and checks for price fluctuations or availability changes.
- **Notification Engine:** Configured with Firebase Cloud Messaging (FCM) to push alerts via email or mobile notification when prices drop [2].
- **Analytics Dashboard:** Developed in Flutter with interactive widgets and chart packages to visualize historical pricing and user activity [1].

B. Data Collection & Storage

PriceScrapper collects and stores both user-defined inputs and dynamically scraped product data:

- **Product Metadata:** Product titles, current prices, images, ratings, and availability status are periodically fetched and normalized.
- **User Preferences:** Wishlists, target prices, and alert frequency settings are stored securely in Firebase Cloud Firestore [4].
- **Historical Data:** Collected price points are stored in a MySQL relational database, indexed by product ID and date, allowing for efficient querying and trend plotting [4].

C. Predictive Engine

To enable smart purchase timing, a dedicated prediction module is integrated:

- **Forecasting Models:** Implements time series forecasting using Facebook Prophet, chosen for its robustness with seasonality and missing data [3].
- **Model Training:** Uses historical price data to train models that output expected price trajectories for each product.
- **Result Visualization:** Forecasts are displayed through simple graphs using Flutter's chart libraries, giving users visual cues for when to purchase [1][3].

D. Application Development Process

The development process follows Agile methodology, with iterative sprints and frequent testing phases.

1) a. Frontend Development::

- Built using Flutter, enabling native performance with a single codebase for Android and iOS [1].
- Integrates custom widgets, responsive UI, and secure navigation logic.

2) b. Backend Integration::

- Firebase handles user sessions, secure storage, and authentication layers.
- REST APIs and cloud functions are implemented for communication between the app and Python scraping services.

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2026

3) c. Cloud Functions::

- Automated cloud functions periodically trigger scraping jobs and forecasting scripts, maintaining real-time price updates.

E. Testing Strategy

A multi-stage testing framework is followed to ensure reliability and security.

- **Unit Testing:** Each module—authentication, scraping, notification—is tested independently to validate functionality.
- **Integration Testing:** Ensures smooth communication between frontend, backend, and external APIs.
- **User Acceptance Testing (UAT):** Conducted with selected beta users to receive feedback on usability and feature completeness.
- **Security Testing:** Firebase's built-in authentication and Firestore rules are tested to prevent unauthorized data access [2].

F. Deployment and Maintenance

- The app is deployed on Google Play Store and Apple TestFlight for public access.
- Backend services and scrapers are hosted on Google Cloud Functions and Heroku for scalability.
- Continuous monitoring tools are implemented to track performance, detect scraper failures, and update outdated product URLs.

III. Software & System Requirements

The development and deployment of PriceScrapper require a clear definition of both functional and non-functional requirements to ensure performance, scalability, and a high-quality user experience.

A. Functional Requirements

Functional requirements outline the specific features the system must support in order to fulfill user needs effectively. Each function is tied to an operational goal or use case.

- **Real-Time Price Scraping:**
  - The system must continuously fetch price data from supported e-commerce platforms (Amazon, Flipkart, eBay).
  - Scraping is implemented using Python libraries such as BeautifulSoup and Selenium, known for their effectiveness in handling dynamic web content [9][12].
- **Price Drop Alerts:**
  - Users should be able to set custom threshold alerts and receive push/email notifications through Firebase Cloud Messaging (FCM) [8].
- **Wishlist and Watchlist Management:**
  - Users can save items to their personal lists and configure preferences for monitoring.
  - These preferences are stored in Firebase Firestore for real-time access and security [10].
- **Historical Price Graphs:**
  - The system must display charts representing price changes over time to support informed purchasing decisions.
  - Data is pulled from a MySQL database and visualized using charting libraries integrated with Flutter [6][10].
- **User Authentication and Settings:**
  - Users must be able to register, log in securely, and manage their profile information.
  - Firebase Authentication supports email/password and OAuth integrations (Google, Facebook) [8].

B. Non-Functional Requirements

These define the overall quality attributes of the system. They ensure usability, security, and scalability under growing operational demands.

- **Performance:**
  - The application must handle requests and notifications with low latency.
  - Background cloud functions and batch scraping pipelines are optimized to reduce response times [8][13].
- **Security:**
  - All user data, including price tracking and login credentials, must be encrypted and securely stored.

ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)

Volume 3, Issue 1, (Jan-Jun) 2026

  - Firebase Authentication, along with Firestore security rules and HTTPS API calls, ensures compliance with secure development practices [8][10].
- **Scalability:**
  - The architecture must support thousands of simultaneous user sessions and product URLs.
  - Cloud-based services such as Firebase and horizontally-scalable databases like MySQL are used to achieve this [10][13].
- **Compatibility:**
  - Developed using Flutter, the app supports Android and iOS with a single codebase.
  - The backend APIs are RESTful and can be extended to integrate browser extensions or web dashboards in the future [1][6].
- **Availability:**
  - The system should be accessible 24/7 with minimal downtime.
  - Uptime monitoring and serverless deployment via Google Cloud Functions and Heroku ensure high availability [13].

IV. Feasibility Study

The feasibility of the PriceScrapper application is assessed through three key dimensions: technical, financial, and operational. This analysis demonstrates the system's readiness for implementation, sustainability, and scalability in real-world e-commerce environments.

A. Technical Feasibility

The project uses a proven technology stack that supports cross-platform compatibility, scalable cloud infrastructure, and automation.

- Flutter offers a unified UI toolkit that reduces development overhead by enabling Android and iOS builds from a single codebase.
- Firebase supports robust user authentication, real-time database access, and push notifications, reducing backend complexity.
- Python-based scraping using Selenium and BeautifulSoup ensures real-time, accurate price tracking from major platforms.
- Machine learning forecasting is made possible using Facebook Prophet, which handles time-series data with trend and seasonality considerations.

B. Financial Feasibility

PriceScrapper is designed to be cost-effective and sustainable with multiple monetization paths:

- **Development Tools:** Most of the stack—Flutter, Firebase (free tier), and Python libraries—are open-source or free for moderate use.
- **Hosting and Maintenance:** Cloud services like Firebase and Heroku offer scalable pricing models that start free, with growth-driven costs.
- **Revenue Generation:**
  - Affiliate marketing: Integrating links to Amazon/Flipkart products can generate passive income on completed sales.
  - Premium Plans: Users could subscribe for advanced features such as AI-driven recommendations or unlimited product tracking.
  - Targeted Ads: Ethical, non-invasive ads can be shown to users to subsidize free-tier usage.

C. Operational Feasibility

The system has been designed with usability and maintenance in mind, facilitating high adoption and low training costs.

- **Intuitive UI:** The Flutter framework supports sleek, responsive interfaces that align with Material Design guidelines, reducing the learning curve for end users.
- **Cross-platform Reach:** Support for Android and iOS ensures broad accessibility across demographic segments.
- **Real-time Engagement:** The push notification system keeps users engaged by informing them of price drops or time-limited deals.
- **Low Onboarding Overhead:** Simple sign-up/login through Firebase Authentication ensures quick onboarding and minimal barriers to entry.

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2026

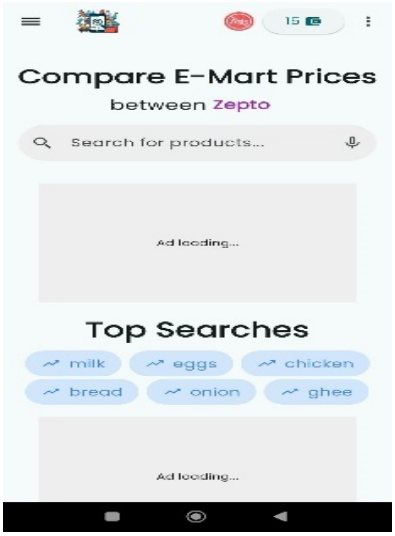

Fig. 2: E Mart Price Dashboard

Fig. 3: E Mart Price Comparison Dashboard

V. Implementation

The implementation of PriceScrapper is guided by a modular development strategy using industry-standard tools, enabling smooth functionality, scalable infrastructure, and an engaging user experience.

A. User Interface (UI)

The frontend is developed in Flutter, which allows for a responsive and consistent experience across Android and iOS platforms using a single codebase.

1) Primary Screens::

- **Login & Signup Page:**
  - Integrated with Firebase Authentication for secure access using email, Google, or Facebook credentials.
- **Home Page (Search & Compare):**
  - Users can input product names or URLs.
  - A dynamic results screen displays real-time prices fetched via scraping, highlighting the lowest and most reliable sources.
- **Wishlist & Watchlist:**
  - Users can bookmark products for future reference.
  - Products in the watchlist are actively monitored for price drops and alerts.
- **Product Details & Price Graph:**
  - Each product page shows a 7-day/30-day price trend using historical data from MySQL.
  - The graph is built with Flutter charting libraries and integrated ML predictions from Prophet.

B. Backend Logic

The backend is built using a hybrid of Firebase Cloud Functions, Python-based scraping scripts, and MySQL for structured storage.

1) Automated Processing Pipeline::

- **Web Scraper Engine:**
  - Written in Python using Selenium and BeautifulSoup, scrapes updated prices every 3–6 hours or based on user activity triggers.
  - Handles JavaScript-rendered pages using headless browsers (e.g., ChromeDriver).
- **Data Pipeline:**
  - Scraped product data is sanitized and written to Firebase for real-time updates and to MySQL for long-term analytics.

ISSN (Online):3048-5088
International Journal of Advancement and Innovation in Technology and Research (IJAITR)
Volume 3, Issue 1, (Jan-Jun) 2026

Fig. 4: Dashboard

  - Each product has a unique ID for consistent tracking and referencing.
- **Alert Engine:**
  - Periodically checks updated prices against user-defined thresholds.
  - If the condition is met, a Firebase Cloud Messaging (FCM) alert is pushed, or an email is sent using cloud function triggers.

C. Integration & Automation

- **Cron Jobs & Cloud Scheduling:**
  - Google Cloud Scheduler automates scraping cycles and forecast generation every 6 hours.
- **AI Integration:**
  - Once enough data points are collected (typically after 7–14 days), Facebook Prophet models are automatically trained per product to predict upcoming price trends.
- **Deployment Strategy:**
  - The mobile app is published on Google Play Store and Apple TestFlight for beta testing.
  - Backend services and scrapers are deployed on Google Cloud Functions and Heroku for scalability and fault tolerance.

VI. Conclusion

PriceScrapper effectively addresses a significant gap in modern e-commerce—the absence of centralized, real-time, and intelligent price tracking tools for online shoppers. With features like cross-platform compatibility, AI-powered price prediction, historical data analysis, and automated alerts, the application ensures smarter and more economical shopping decisions.

The project's success lies in its modular and scalable architecture, utilizing tools such as Flutter for responsive UI, Firebase for real-time interaction and authentication, and Python-based scraping engines for data extraction. Additionally, the integration of machine learning through Facebook Prophet strengthens the platform's ability to offer predictive insights to users.

Combined Testing & Deployment Summary:

- **Robust Testing Framework:** Includes unit tests for scrapers and alert services, integration tests between frontend and backend, and user acceptance testing (UAT) with early adopters.
- **Seamless Deployment:** Launched on Android (Google Play) and iOS (via TestFlight), with backend services hosted on Firebase and Google Cloud Functions for high availability and low-latency performance.

By balancing technical innovation with user-centric design, PriceScrapper not only enhances individual purchasing experiences but also contributes to a more transparent and competitive online marketplace.

Future Work

To expand its capabilities and user base, the following upgrades are envisioned:

- **Browser Extension** for on-the-go price comparison while browsing online retailers.
- **Blockchain-Backed Price Integrity Logs** to verify pricing accuracy and protect against fake discounting.
- **AI-Driven Smart Suggestions** recommending alternate products based on user behaviour and market analysis.

ISSN (Online):3048-5088

International Journal of Advancement and Innovation in Technology and Research (IJAITR)

Volume 3, Issue 1, (Jan-Jun) 2026

- **Community Features** like shared watchlists, deal forums, and buyer feedback loops.

In conclusion, PriceScrapper is well-positioned to evolve into a comprehensive e-commerce intelligence tool, setting a new standard for how consumers interact with digital marketplaces.

REFERENCES

[1] F. D. Saleh, et al., "Adoption of Flutter Framework for Cross Platform Mobile Applications," *IEEE CITSM*, 2021.
[2] Google Developers, "Firebase Documentation," https://firebase.google.com/docs
[3] Facebook Prophet, "Forecasting at Scale," https://facebook.github.io/prophet/
[4] A. Sharma, et al., "A Study on Firebase Cloud Firestore," *International Journal of Scientific Research in Computer Science Engineering*, 2018.
[5] B. Chen, Y. Liu, and J. Zhang, "A Dynamic Price Comparison System for E-Commerce," *IEEE Big Data and Smart Computing*, 2020.
[6] J. Kim, et al., "AI-based Recommendation System for Matching Pets and Adopters Using Behaviour Analysis," *IEEE ICAIIC*, 2020.
[7] P. K. Singh, "Digital Consumer Empowerment through Smart Price Comparison Tools," *Journal of Retail Tech*, 2021.
[8] Google Firebase Docs, "Cloud Messaging and Authentication," https://firebase.google.com/docs
[9] J. Mitchell, "Building Scalable Web Scrapers with Python," *Web Data Technologies Conf.*, 2019.
[10] A. Sharma, et al., "Analysis of Firebase Firestore and MySQL Backend," *International Journal of Computer Science*, 2019.
[11] Facebook Prophet Team, "Time Series Forecasting at Scale," https://facebook.github.io/prophet/
[12] K. Zhou, Y. Guo, and M. Wang, "Automated Data Scraping in E-Commerce with Selenium and BeautifulSoup," *International Journal of Data Engineering*, vol. 7, no. 2, 2021.
[13] D. Rajasekhar, M. Rafi D, S. Chandre "An Improved Machine Learning and Deep Learning based Breast Cancer Detection using Thermographic Images," 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2023, pp. 1152-1157, doi: 10.1109/ICEARS56392.2023.10085612.
[14] Jain, A., Saify, A., (2020). Prediction of Nutrients (NPK) in soil using Color Sensor (TCS3200). International Journal of Innovative Technology and Exploring Engineering, 9(3), 1768-1771.
[15] S. Lee and H. Kim, "Scalable Cloud Architecture for Consumer-Oriented Applications," *Journal of Cloud Computing Advances*, vol. 9, no. 3, 2022.
[16] K. Gupta, "Cost-Efficient Cloud Development Strategies for Startups," *Cloud Deployment Monthly*, vol. 15, no. 4, 2022.
[17] A. Davis, "Revenue Models for Mobile Shopping Applications," *Journal of Digital Commerce*, vol. 11, no. 1, 2021.