

Automated Price Comparison and Tracking System

Ilakiya R

Department of information
technology,
Rathinam College of Arts and
Science, Coimbatore
ilakiyar31@gmail.com

Baladyanesh JB

Department of information
technology,
Rathinam College of Arts and
Science, Coimbatore.

Balakrishnan S

Department of information
technology,
Rathinam College of Arts and
Science, Coimbatore.
krishnansbala13@gmail.com

Mentor details:

Mrs. Hemalatha

Assistant professor

Department of computer science
Rathinam College of Arts and
Science, Coimbatore.

Abstract -- In today's rapidly evolving e-commerce landscape, product prices fluctuate frequently due to various factors such as demand, supply, seller strategies, seasonal sales, and exclusive discounts. Consumers often struggle to track these frequent price changes manually, making it difficult to determine the best time to purchase a product. This project, "Flipkart Price Tracker Using Python," is designed to address this issue by automating the process of monitoring price variations of products listed on Flipkart.

The system utilizes web scraping techniques with BeautifulSoup and Requests to extract real-time price data from Flipkart's website. The collected data is then stored in a structured database to maintain historical records of price fluctuations. By analyzing the stored data, users can easily compare old and current prices, identify trends in pricing, and make informed purchasing decisions.

Additionally, this system eliminates the need for users to manually check prices repeatedly. Instead, it provides a user-friendly interface to display tracked prices efficiently. This document provides a comprehensive overview of the project, including the problem statement, objectives, proposed solution, system implementation, results, and

analysis. By leveraging Python for data extraction and database management, this project offers an effective solution for consumers to make better shopping decisions.

I. INTRODUCTION

The e-commerce industry has revolutionized the way people shop, offering convenience, variety, and competitive pricing. Platforms like Flipkart, Amazon, and other online marketplaces frequently update their product prices based on several factors such as market demand, stock availability, seller pricing strategies, and seasonal discounts. While this dynamic pricing strategy benefits businesses, it often creates challenges for consumers who wish to purchase products at the best possible price.

One of the biggest problems consumers face is the inability to track price fluctuations efficiently. Flipkart and similar platforms provide basic tools such as wishlists and notifications, but they do not offer a detailed historical comparison of price changes over time. Users are forced to manually check product prices multiple times to ensure they get the best deal. This process is not only time-consuming but also prone to

human error, as users may forget to check prices at the right time, ultimately missing out on cost-saving opportunities.

To address this issue, this project introduces a Python-based price-tracking system that automates the entire process. By utilizing web scraping techniques with BeautifulSoup and Requests, the system retrieves real-time price data from Flipkart and stores it in a structured database. The system records old and new prices, allowing users to view historical price changes and analyze trends. By doing so, users can make informed decisions about when to buy a product, ensuring they purchase at the lowest possible price.

Literature Survey

The concept of price tracking has gained significant attention in the field of e-commerce analytics and consumer decision-making. Several research studies and industry applications have explored various methodologies for tracking price fluctuations, data extraction, and trend analysis. While global platforms like Amazon have established third-party tools for price tracking, such solutions remain limited for Flipkart and similar region-specific e-commerce platforms.

Numerous researchers have examined web scraping techniques, database management strategies, and automation frameworks to extract and analyze e-commerce data efficiently. Below are some of the key studies and existing systems that form the foundation of this project:

1. Automated Price Tracking Systems

Several studies have focused on developing automated price monitoring tools using web crawlers and scrapers. These tools help in tracking price fluctuations over time by frequently fetching product price data from online marketplaces.

Web Crawlers for Price Monitoring: Researchers have investigated the efficiency of web crawlers and automated bots in retrieving price data at regular intervals. However, challenges such as anti-bot detection mechanisms, CAPTCHA verifications, and dynamically loaded content pose difficulties in effective implementation.

- **API-Based Tracking Solutions:** Some platforms offer public APIs for fetching e-

commerce data. However, Flipkart does not provide an easily accessible API for public price tracking, making web scraping a necessary alternative.

- **Limitations of Existing Systems:** Popular price-tracking tools such as Keepa and Honey are primarily designed for Amazon, offering features like price history visualization, discount alerts, and comparison tools. However, these tools do not support Flipkart, highlighting a gap in the market for region-specific price monitoring solutions.

2. Data Extraction Using BeautifulSoup

Extracting meaningful data from dynamic web pages is a crucial aspect of price tracking. Researchers have extensively explored web scraping techniques using Python libraries like BeautifulSoup and Scrapy to retrieve data from structured HTML content.

Efficiency of BeautifulSoup: Multiple studies highlight the ease of extracting text-based data, links, and price tags using BeautifulSoup, making it a widely used tool for web scraping in research and industry applications.

Handling JavaScript-Rendered Content: Flipkart, like many modern e-commerce platforms, uses JavaScript-based dynamic loading, which makes traditional scraping techniques insufficient. Researchers have suggested the use of Selenium or headless browser automation to handle such cases.

Ethical Considerations: Several studies emphasize the importance of ethical web scraping practices, ensuring that scraping does not violate terms of service or disrupt website performance.

3. Database Management for Price History

Storing and managing historical price data is essential for effective trend analysis and user decision-making. Researchers have examined the role of relational databases (SQL-based) in structuring and retrieving price data efficiently.

- **SQL-Based Databases for Structured Storage:** Studies suggest that relational databases such as MySQL and PostgreSQL are well-suited for storing structured e-commerce data, allowing efficient retrieval, filtering, and comparison of product price histories.
- **Optimizing Queries for Fast Retrieval:** Researchers have proposed indexing

techniques and optimized SQL queries to reduce query response time when fetching price histories.

- **Data Redundancy Management:** Some studies highlight the challenge of storing large-scale e-commerce data, proposing techniques such as data compression, deduplication, and efficient indexing strategies.

4. Identified Research Gap & Project Significance

Despite advancements in automated price tracking, web scraping techniques, and database management, there remains a lack of a dedicated price tracker for Flipkart that effectively:

- **Extracts product prices in real-time using web scraping.**
- **Stores historical price data for easy comparison.**
- **Handles dynamic content loading efficiently.**
- **Provides a simple and scalable solution for users.**

Our project aims to bridge this gap by developing a Python-based Flipkart price tracker that integrates BeautifulSoup for data extraction, structured SQL databases for storage, and an intuitive interface for user-friendly analysis. This system will empower users to track price trends, compare old and new prices, and make informed purchase decisions without relying on manual price checks.

Problem Statement

In today's digital shopping era, e-commerce platforms like Flipkart have revolutionized how consumers purchase products. However, one of the biggest challenges consumers face is tracking frequent price fluctuations on such platforms. Flipkart dynamically adjusts prices based on various factors, including demand, supply, competitor pricing, promotional offers, and seasonal sales. While this benefits Flipkart's pricing strategy, it creates difficulties for consumers who wish to make cost-effective purchases but lack an efficient way to monitor price changes.

Challenges Faced by Consumers

1.Missed Opportunities for Cost Savings

Consumers often end up purchasing products at higher prices than necessary simply because they are unaware of previous price drops or potential future discounts.

Flipkart does not provide an inbuilt tool for tracking historical prices, making it difficult for users to assess pricing trends.

Many users wait for major sales, such as Big Billion Days or festive discounts, but there is no systematic way to determine if a product's current price is a good deal compared to previous rates.

2.Manual Tracking is Inefficient and Time-Consuming

The only way users can track price changes is by manually visiting product pages repeatedly and noting down the prices, which is a tedious and error-prone process.

Keeping track of multiple products at the same time is nearly impossible without an automated system.

Consumers need a centralized way to compare current and historical prices without spending hours browsing Flipkart manually.

3.No User-Friendly Price History Interface

While price-tracking tools exist for Amazon (e.g., Keepa, Honey), there is no dedicated price tracker specifically designed for Flipkart users.

Without a structured approach, users cannot determine whether they are getting the best price or not

Need for an Automated Solution

Given the above challenges, there is an urgent need for a dedicated Flipkart Price Tracking system that can:

- Automatically fetch product prices at regular intervals using Python-based web scraping.
- Store price data efficiently in a structured database for future analysis and easy retrieval.
- Provide clear comparisons of past and present prices, helping consumers decide the best time to make a purchase.

- Offer a simple, user-friendly interface to access historical price trends without technical expertise.
- Ensure scalability and efficiency, allowing tracking of multiple products simultaneously.

This project aims to develop an automated Flipkart Price Tracker using Python that will empower users with the necessary data to make informed shopping decisions. By eliminating the need for manual price checks and providing accurate historical pricing data, this system will enhance consumer savings and shopping efficiency.

Proposed Work

To address the challenges outlined in the problem statement, this project proposes a Flipkart Price Tracking System that automates price monitoring and provides a user-friendly interface for viewing price history. The system consists of three main components:

1. Web Scraping Module

The web scraping module is responsible for extracting real-time product price data from Flipkart using Python-based web scraping techniques. Since Flipkart does not provide an official API for price tracking, this module leverages BeautifulSoup and Requests to fetch relevant information.

Functions of the Web Scraping Module:

Extracts product prices dynamically from Flipkart product pages.

Retrieves key product information such as:

- Product Name
- Current Price
- Flipkart URL

Implements strategies to avoid detection, such as:

Rotating user agents to simulate different devices and browsers.

Introducing delays between requests to prevent triggering Flipkart's anti-bot mechanisms.

Handles dynamic content loading using libraries like Selenium if necessary.

2. Database Management

To efficiently store and retrieve product prices over time, the system uses a SQL-based relational database. This module ensures that users can view historical price trends and compare past and current prices

Functions of the Database

Management Module:

Stores product details and price history in an organized manner.

Maintains separate tables for:

Products Table – Stores product name, Flipkart URL, and unique product ID.

Price History Table – Maintains records of old and new prices along with the date of price retrieval.

Supports SQL queries to fetch price trends efficiently.

Enables automatic updates to the price history database at regular intervals.

3. User Interface (UI):

The system will include a simple and interactive user interface that allows users to manage their tracked products and view price history.

Functions of the User Interface Module:

Displays tracked products with their name, Flipkart URL, and latest price.

Provides historical price data in a structured format.

Allows users to compare past and current prices with a graphical representation.

Includes options to:

- Add new product URLs for tracking.
- Remove unwanted products from the tracking list.
- Refresh price updates manually.

Overall Workflow of the Proposed System:

1. The user adds a Flipkart product URL to track.
2. The web scraping module extracts the product's current price and updates the database.

3. Historical price data is stored in the SQL database for future comparison.
4. The UI displays the tracked products along with their past and current prices.
5. Users can analyze trends and make informed purchasing decisions.

By integrating automated web scraping, structured database management, and an intuitive user interface, this project provides an efficient and scalable solution for tracking Flipkart product prices.

System Architecture

The architecture of the Flipkart Price Tracker project is designed to ensure an efficient and scalable solution for monitoring product price fluctuations. The system follows a structured pipeline comprising four key components: Data Collection, Data Storage, Data Analysis, and User Interface.

- **Data Collection:** The first step involves collecting product data from Flipkart using web scraping techniques. The system utilizes the BeautifulSoup and Requests libraries to parse the HTML content of product pages and extract essential information such as product name, current price, and the URL. For dynamic content that requires JavaScript to load, the Selenium tool is used to automate the rendering of the page before data extraction. The system ensures that the web scraping process is optimized, using techniques such as rotating user agents and introducing delays between requests to mimic human behavior and avoid detection by Flipkart's anti-bot measures.
- **Data Storage:** Once the data is collected, it is stored in a relational database, such as MySQL or SQLite. The database structure consists of two primary tables:
 - **Products Table:** Stores the product's unique ID, name, and URL.
 - **Price History Table:** Stores historical price data for each product, including the old and current prices along with the timestamp of when the price was

last checked. The relational database ensures efficient querying and management of product price data over time, making it easy to track price fluctuations and retrieve historical data for comparison.

- **Data Analysis:** The collected data is then analyzed to compare the current price with the historical price entries stored in the database. The system performs a comparison of price changes, highlighting price differences, and calculates the percentage change over time. Additionally, this component may generate visual representations, such as charts or graphs, to depict the price trends, providing the user with an insightful view of how the product's price has evolved over time. Alerts may also be set up to notify users when a price reaches a predefined threshold, offering them an opportunity to make timely purchase decisions.
- **User Interface (UI):** The final component is the User Interface, which presents the data to users in a clear and accessible format. The UI displays a list of all tracked products, showing essential details such as product name, current price, historical price data, and the corresponding Flipkart URL. Users can interact with the interface to compare the prices over time, add new products to track, or remove products that are no longer needed. The UI is designed to be user-friendly, ensuring that the system is easy to use, even for users with minimal technical knowledge.

Modules Used

The Flipkart Price Tracker system utilizes various Python libraries and tools to achieve its functionality. Below is a list of the key modules used in the project, each of which plays an essential role in data extraction, processing, storage, and presentation.

- **BeautifulSoup:**
 - **Role:** BeautifulSoup is a Python library used for web scraping purposes to parse HTML content. In this project, it is used to extract the product's name,

price, and other relevant details from Flipkart's product pages.

- **Usage:** The library simplifies the extraction of data from complex HTML structures. By providing methods to search and navigate the parse tree, BeautifulSoup helps isolate specific HTML elements such as product prices and names, which are then saved in the database.
- **Requests:**
 - **Role:** The Requests library is used to send HTTP requests and retrieve content from web pages.
 - **Usage:** This module is employed to fetch the raw HTML content of Flipkart product pages. It simplifies the process of making GET requests and handling responses, ensuring that the data needed for scraping is retrieved efficiently.
- **SQLite/MySQL:**
 - **Role:** These are relational database management systems (RDBMS) used to store the product and price details.
 - **Usage:**
 - **SQLite:** A lightweight, file-based database used for quick prototyping and small-scale applications. It is ideal for storing product details and historical price data locally.
 - **MySQL:** A more robust database solution that can be used for production environments where scalability and handling larger datasets are required.
 - In this project, these databases store the product name, product URL, current price, old price, and timestamp data. SQL queries are used to store, retrieve, and analyze price history.

- **Django:**

Role: Django are Python-based web frameworks used to create the user interface for the project.

Usage:

Django: A more feature-rich framework suitable for larger applications that require advanced functionalities such as authentication, form handling, and an admin interface. Django can be used if the project needs to scale to accommodate more complex features in the future.

Django allow the creation of web pages that present the tracked products, their current prices, and historical price trends in a user-friendly manner.

System Requirements

Hardware Requirements:

Minimum 4GB RAM (Recommended 8GB)

100GB Storage

Dual-core Processor or above

Software Requirements:

- Python 3.8+
- BeautifulSoup, Requests, Flask/Django
- MySQL/SQLite Database
- Web Browser

Result Analysis

The Flipkart Price Tracker system has been designed and tested to ensure that it efficiently tracks and displays product prices, providing valuable insights to users. The following observations were made during the testing phase:

Real-Time Price Updates:

The system is capable of fetching real-time price updates efficiently. Using web scraping techniques, it can retrieve the current price of any product listed on Flipkart and update the database immediately. This allows users to track the latest price fluctuations without delay, ensuring they are always informed about the current cost of the products they are interested in.

Accurate Database Storage:

The system successfully stores price changes in a structured database. Each price update is logged along with the date and time it was retrieved. This data is stored securely, ensuring that users can access historical price information whenever needed. The database provides an accurate historical record of price trends for each product, which is essential for comparing price differences and making informed purchase decisions.

Easy Comparison of Historical Prices:

The system provides a user-friendly interface that allows users to compare past and current prices of products. Through the interface, users can view the price history of any product, enabling them to identify trends and decide the optimal time to purchase based on price variations. This feature is highly valuable for consumers who are looking for the best deal and wish to take advantage of fluctuations in product prices.

Test Results with 50 Flipkart Products:

A comprehensive test run was performed with 50 different products from Flipkart. The system was able to accurately track and display price variations for all the products. The results from the test showed that the system recorded both the current and historical prices correctly, with no errors in data retrieval or storage. The user interface also displayed the price history effectively, allowing users to easily navigate through the price data.

Reliability and Efficiency:

The system proved to be both reliable and efficient during testing. It handled multiple product price checks simultaneously without any significant delays. The data was fetched in a timely manner, ensuring that the system remains efficient even as the number of products being tracked increases. Furthermore, the database showed no signs of corruption, and all historical prices were accurately saved, ensuring long-term reliability of the system.

Conclusion

The Flipkart Price Tracker project successfully delivers an automated solution to track price changes on Flipkart, addressing the challenges consumers face

when trying to monitor fluctuating product prices. By leveraging Python-based web scraping techniques, the system is able to retrieve, store, and display price data in a structured and user-friendly format. This eliminates the need for manual price checks and empowers users to make more informed purchasing decisions based on real-time data.

The system has proven to be both efficient and reliable in its operation, with successful data retrieval and accurate storage of price variations. The user interface provides a seamless experience for comparing historical and current prices, helping users identify the best times to make purchases and save money.

Despite the success of the project, there are several potential improvements that could be implemented in future versions of the system to enhance its functionality and broaden its scope. These include:

Email Alerts for Price Drops:

The system could be enhanced with email notifications to alert users whenever the price of a product drops below a set threshold. This feature would allow users to receive real-time updates on their tracked products, ensuring they never miss a potential deal.

Support for Additional E-Commerce Platforms:

In the future, the system could be expanded to support price tracking for other popular e-commerce platforms like Amazon, Myntra, or Snapdeal. This would allow users to monitor price variations across multiple websites from a single platform, improving the tool's versatility and usefulness.

Mobile-Friendly Interface:

A mobile-responsive version of the user interface could be developed to allow users to track price changes and view historical data on their smartphones or tablets. This would make the system more accessible and convenient, catering to a broader user base.

In conclusion, the Flipkart Price Tracker project effectively addresses the need for an automated price tracking solution and has the potential to be further improved and expanded in the future. By providing users with the tools to track and compare prices, the system empowers them to make better purchasing decisions, ultimately helping them save money and optimize their shopping experience.

References

1. **BeautifulSoup Documentation**
<https://www.crummy.com/software/BeautifulSoup/> - Python library for parsing HTML and XML documents.
2. **Requests Documentation**
<https://docs.python-requests.org/en/master/> - Simplifies HTTP requests in Python.
3. **Flask Documentation**
<https://flask.palletsprojects.com/en/2.1.x/> - Python micro web framework for building web applications.
4. **Django Documentation**
<https://www.djangoproject.com/> - High-level Python web framework for rapid development.
5. **SQLAlchemy Documentation**
<https://www.sqlalchemy.org/> - Python SQL toolkit and ORM library.
6. **Flipkart Website**
<https://www.flipkart.com/> - E-commerce platform for product price tracking.
7. **Web Scraping with Python**
<https://realpython.com/beautiful-soup-web-scraper-python/> - Tutorial on scraping data with Python.
8. **MySQL Documentation**
<https://dev.mysql.com/doc/> - Relational database management system used for storing data.
9. **SQLite Documentation**
<https://www.sqlite.org/> - Lightweight database engine used for data storage.
10. **Keepa - Price Tracker for Amazon**
<https://keepa.com/> - Price tracking tool for Amazon products.