

Price Contrast: Your personal shopping assistant

KAAVISH PROJECT PROPOSAL

By

Uzair Rafiq	(ur07589@st.habib.edu.pk)
Ahsan Raza Zardari	(az07517@st.habib.edu.pk)
M. Shahzar Khalique	(mk07166@st.habib.edu.pk)
Hamza Abdullah	(ha07194@st.habib.edu.pk)

October 22, 2024



In partial fulfillment of the requirement for
Bachelor of Science
Computer Science

DHANANI SCHOOL OF SCIENCE AND ENGINEERING

HABIB UNIVERSITY

FALL 2024

Copyright © 2024 Habib University

1 Abstract

In today's competitive online marketplace, consumers need tools to make informed purchasing decisions. Our project proposes the development of a price comparison website that centralizes product prices, features, and reviews from various e-commerce platforms. By leveraging web-scraping and LLM technologies, the platform will provide users with accurate price comparisons, personalized recommendations, and a seamless shopping experience.

Additionally, it'll feature a personalized shopping assistant which will allow the user to query data directly by asking the assistant which will enhance the online E-com experience for our users.

We also believe this project will enhance visibility for small retailers and improve customer satisfaction. The project's objective is to create a user-friendly, data-driven solution that benefits both consumers and retailers by bringing the entire shopping experience into a single place.

2 Problem definition

2.1 Background and Current Challenges

In the rapidly evolving digital marketplace, consumers face an increasingly complex landscape when making online purchases. As the abundance of choice continues to grow, it becomes increasingly challenging for consumers to make informed and cost-effective decisions.

One of the primary issues in the current online shopping ecosystem is the lack of centralized, reliable price comparison tools. Consumers often find themselves navigating multiple websites, comparing prices manually, and struggling to find the best deals. This process is not only time-consuming but also prone to errors and missed opportunities.

A study by Baymard Institute found that 67% of users abandon their online shopping carts due to unexpected costs or inaccurate pricing information, underscoring the need for a more transparent and informative shopping ecosystem. [1]

Moreover, the dynamic nature of online pricing adds another layer of complexity. According to Adobe's Digital Price Index, online prices for consumer goods can fluctuate significantly over time. Without access to historical price data and real-time updates, consumers may inadvertently purchase products at peak prices, missing out on potential savings.[2]

The issue of price disparity across different platforms further complicates the shopping experience. A study by McKinsey & Company found that consumers often encounter significant price differences for identical products across various e-commerce platforms. This discrepancy not only confuses consumers but also makes the buying process significantly slower. [3]

2.2 Problem Statement

The lack of comprehensive product information and reliable user reviews in one centralized location hinders consumers' ability to make well-informed decisions. Additionally, there's a huge space in Pakistan for a personalized shopping experience, a revolutionary way to do E-com where customers can ask directly for products and see price comparisons in real time. It's like roaming the mall, but much faster and accurate.

2.3 Proposed Solution

A comprehensive price comparison website, enhanced with a personalized chat bot that takes instruction in plain English from customers and queries the database to return products. This has the potential to revolutionize how consumers navigate the digital marketplace and make buying decisions.

3 Social relevance

The development of a comprehensive price comparison website with an integrated personalized chat-bot holds significant societal relevance in today's digital economy. This project addresses several key issues that impact consumers, businesses, and the broader economic landscape. Since the onset of the COVID-19 pandemic, consumer intent to purchase goods through e-commerce channels has increased by 40 to 60 percent compared with pre-pandemic levels across categories from everyday essentials to clothing and accessories [4].

Firstly, the platform aims to enhance consumer empowerment and financial well-being. In an era where income inequality continues to be a pressing issue, helping consumers make informed purchasing decisions and find the best deals can have a tangible impact on household budgets.

Moreover, this project has the potential to promote market transparency and fair competition. By providing clear, real-time price comparisons across multiple retailers, the platform can help expose price gouging and encourage more competitive pricing strategies. This aligns with the findings of a Harvard Business School study, which showed that increased price transparency can lead to increased purchase interest [5]. This allows consumers to trust online platforms and make more purchases.

The inclusion of a personalized chat bot addresses the growing need for accessible, on-demand customer support in the digital age. According to a report by Accenture, nearly 40% of consumers have left a website because they were overwhelmed by too many options [6]. By offering tailored assistance, the project can help bridge the digital divide, making online shopping more accessible to those who may struggle with technology or have limited time for research.

4 Originality/Novelty

4.1 Existing Solutions

While there have been advancements in product catalogue management and image recognition technologies, our research indicates an absence of an exact solution that provides price comparisons currently in Pakistan.

Existing solutions like `idealo.de` cater to a German audience. Similarly some sites like `price.com` and Google Shopping also deal primarily with foreign markets and do not quite provide the personalized experience that we intend to build.

4.2 Our Solution

We propose an innovative price comparison website with an integrated personalized chat-bot. Our approach will include the following key features and technologies:

- **Data Aggregation and Normalization:** Implement a web scraping engine that runs multiple scripts based on schedule to scrape data from a list of websites and store it in a database. It also links to API's whenever possible. Data is tagged cleaned and stored in an efficient manner for quick retrieval.
- **Natural Language Processing (NLP) for Chatbot Intelligence:** Implement state-of-the-art NLP models like BERT or GPT-3 to power our personalized chat bot. This will allow the chat bot to receive complex user queries and call functions based on them to query our database and retrieve data.
- **Dynamic Visual Comparison Tools:** Create an interactive dashboard using D3.js or Chart.js to display price histories, compare product features and reviews and highlight the best deals.
- **Real-Time Price Alerts:** Implement a notification system using WebSockets or server-sent events to provide users with instant alerts when prices drop for the product they wish to monitor.

5 CS contribution

5.1 Courses

- **Database Management Systems (CS 355):** Equips us with the knowledge to design and optimize databases for efficiently storing and retrieving vast amounts of product data.
- **Web & Mobile Development (CS 370):** Enables us to create a responsive, user-friendly interface for our price comparison platform.
- **Software Security (CS 371):** Ensures we implement robust security measures to protect user data and prevent vulnerabilities.

- **Data Science (CS 457):** Provides the skills to analyze large datasets for accurate price comparisons and recommendations.
- **Artificial Intelligence (CS 351):** Gives us the expertise to build an AI-driven shopping assistant capable of making personalized recommendations and predicting price trends.
- **Data Structures II:** Understand how to efficiently organize data and optimize algorithms for time. This will be useful when we create scripts in our engine and run them through algorithms.

5.2 Methodology

The development of a price comparison website with an integrated AI assistant involves several stages, each addressing a critical aspect of the system's functionality. The methodology is divided into five key components: data acquisition, data processing, AI-driven shopping assistant, user interface, and system deployment.

Data Acquisition

The first step in the project is to focus on data acquisition, which involves both web scraping and API integration. We will implement web scraping techniques using Python libraries such as BeautifulSoup and Scrapy to collect real-time pricing data. Additionally, we will integrate with APIs of online retailers wherever available. To ensure efficient data retrieval, we will set up regular data-fetching cycles using cron jobs. The scraper will be optimized using several techniques, including tagging data, storing cache, and categorizing sites. Our niche focus will be on clothing websites operating in Pakistan.

Data Processing

Data processing is critical for ensuring the quality and consistency of the information we collect. We will develop algorithms using **Celery** and **Scrapy** to perform large-scale scraping, ensuring we obtain clean data. It is essential to maintain consistency in price formats and unify product descriptions across different sources. Additionally, we will build a price comparison engine that operates based on a cron job, updating the database and displaying real-time price data from multiple retailers.

AI-Driven Shopping Assistant

An integral part of our project is the AI-driven shopping assistant. This includes developing an AI-powered chatbot using the OpenAI API, which will allow users to ask queries in natural language (e.g., "Find the cheapest white polos") and receive tailored suggestions. The shopping assistant will also offer interactive guidance, recommending alternative products with better reviews or lower prices, and assist users with filtering and sorting products based on their preferences eg lower price, newer release data.

User Interface (UI/UX)

For the user interface, we will utilize JavaScript frameworks to develop a responsive design that works seamlessly across desktop and mobile devices. The interface will display product comparison tables, price history graphs, and detailed product information. Users will have the ability to interact with the AI chatbot directly through a chat interface for personalized shopping assistance. We will implement dynamic charts using D3.js or Chart.js to visualize price trends and product comparisons.

System Deployment & Performance

The backend development will involve using Python or Node.js for the server that handles API requests, manages databases, and processes business logic. We will create a robust database schema using PostgreSQL or MongoDB to store product data, user information, and pricing history, implementing indexing to speed up search and comparison functionalities.

Real-time price alerts will be implemented through a notification system that utilizes WebSockets or server-sent events, providing instant alerts to users when prices drop on their watched items. Security measures will include implementing encryption and secure authentication protocols (OAuth2) to protect user data.

6 Scope and Deliverables

6.1 Scope

The scope of the "Price Contrast: Your Personal Shopping Assistant" project is both ambitious yet doable within a year-long timeline and a small development team.

We are focusing on developing a personal shopping assistant for the clothing retail sector within a one-year timeframe by a team of four. The primary skill-sets required are an understanding of backend and frontend development, web scraping experience and a know-how of build LLM models and chatbots. Due to the Variety of different products available in the market, We will specify the clothing side for our target area to narrow it down.

The project's scope is ambitious but achievable over one year. It involves complex tasks such as web scraping, AI model training, and developing a user-friendly interface. The focused approach on the clothing sector helps keep the workload manageable while still delivering a valuable product.

Here's a list of key struggles and obstacles:

- Web scraping challenges: Overcoming anti-scraping measures and handling diverse website structures.
- Data management: Storing and updating large volumes of clothing data efficiently.
- Natural language processing: Accurately interpreting diverse user inputs and fashion terminology.
- System integration: Seamlessly connecting scraping, AI, and UI components.
- AI model development: Ensuring accuracy in understanding user preferences and queries.
- Performance optimization: Ensuring fast response times and efficient data processing.

6.2 Deliverables

Given this structure, the following deliverables are anticipated throughout the project life-cycle:

- **Web Application:** A fully functional price comparison website with a responsive design for desktop and mobile devices.
- **Chatbot Integration:** An AI-powered chatbot seamlessly integrated into the website interface, capable of handling user queries and providing personalized assistance.
- **Admin Dashboard:** A backend interface for managing product data, user accounts, and system analytics.
- **API Documentation:** Comprehensive documentation for potential integration with third-party services or mobile applications.
- **Technical Report:** A comprehensive report detailing the architecture, algorithms, and technologies used in the development of the price comparison platform.

7 Feasibility

To complete this project, we need to achieve the following tasks:

- Create an efficient scraping script for each retail clothing website online
- An engine that periodically runs script and updates data in the database
- Efficient scraping techniques that gets past restrictions and uses alternate routes to get data when possible

Building such an engine is possible and we've run a test scraping on the Ecom Website Daraz to retrieve mobile phone data using BeautifulSoup and Selenium. We extracted data in the following format:

```
data = {  
    "name": product_name,  
    "price": product_price,  
    "image": product_img,  
    "url": product_url,  
    "store": product_store  
}
```

Here's a few more details about the scraper's approach.

Functionality

1. **Generate Product URLs:** The `generateList()` function gathers product URLs from the first eight pages of Daraz.pk.

2. **Scrape Product Information:** The scraper iterates through the retrieved URLs, focusing on script tags (`<script>`) as Daraz potentially embeds product data within JavaScript code.
3. **Identify Relevant Script:** Through analysis, the second-to-last script tag is chosen for its high likelihood of containing product details.
4. **Extract Data:** The script content is parsed using string manipulation techniques to identify specific keywords and patterns indicating product name, price, and image URL. Examples include `"salePrice"` for price, `"pdt_name"` for name, and `"mainImage"` for image URL.
5. **Data Volume:** The scraper retrieves information for 400 products (40 entries per page across 10 pages).

Our approach is to create similar tailored scripts for each clothing website in our list and merge it all into an engine. The main objectives is to tie all that data together in a web app and make the database as interact-able as possible so we can query it and add functionality to our chat bot and web app.

8 Team dynamics

Team Member	Responsibility
Backend Developer (M. Shahzar Khalique)	<ul style="list-style-type: none">• Implement web scraping or integrate with APIs to collect data from multiple e-commerce websites.• Develop data cleaning and normalization processes to ensure consistency across different sources.• Design and implement the server-side logic and database schema for storing product data, user information, and price alerts.• Create and manage API endpoints for product search, filtering, price history, and user management.• Implement role-based access control for administrators and users.• Ensure data security by implementing encryption and secure user authentication.• Optimize database queries and server response times for handling large datasets.
Frontend Developer (Hamza Abdullah)	<ul style="list-style-type: none">• Develop the user interface for the website, including product search, comparison tables, and user dashboards.• Implement responsive design to ensure the site works well on various devices (desktop, tablet, mobile).• Integrate with backend APIs to display real-time product data, user profiles, and alerts.• Implement interactive elements such as filters, sorting options, and search bar functionality.• Set up role-based navigation to distinguish between users (e.g., regular users, administrators).• Ensure smooth navigation between different sections, such as product categories, price history, and user settings.

Team Member	Responsibility
Frontend Developer (Ah-san Raza Zardari)	<ul style="list-style-type: none">• Design a user-friendly interface for product search, comparison, and price tracking.• Create wireframes and high-fidelity prototypes for key screens like product details, comparison tables, and user profile.• Conduct user research to understand the needs and pain points of potential users.• Run usability tests to gather feedback and refine the design based on user interactions and preferences.• Design interactive elements such as graphs for price history, alert notifications, and quick filters.• Develop a cohesive visual style, including color schemes, typography, and icons while also aligning design to the brand of the website.
Machine Learning/Backend Engineer (Uzair Rafiq)	<ul style="list-style-type: none">• Build and train NLP models for chatbot functionality, allowing the chatbot to understand user queries, provide product recommendations, and perform price comparisons.• Conduct data preprocessing and feature engineering to optimize the accuracy and efficiency of machine learning models.• Implement sentiment analysis on user reviews to summarize product feedback and assist users in making informed decisions.• Continuously monitor and improve the performance of machine learning models through A/B testing and feedback loops.• Collaborate with the backend team to integrate machine learning models with the overall system architecture, ensuring smooth real-time data flow.

9 Tech stack

- Scraping: Scrapy, which is Python-based, for large-scale scraping.
- BeautifulSoup or Selenium can handle JavaScript-heavy pages.
- Scheduler: Celery with a task queue for periodic scraping, or a cron job for simpler scheduling.

- Storage: A PostgreSQL or MongoDB database to store product listings, prices, and comparisons.
- Web Interface: A React or Django web app to allow users to input keywords (e.g., "white polo") and see the comparisons.
- Comparison Logic: Python for data comparison and handling price history.
- Notifications: Email API like SendGrid to Slack/Telegram for updates.
- Frontend: Nextjs/React

References

- [1] Baymard Institute. (2023). Check out the optimization report. Retrieved from <https://baymard.com/research/checkout-usability>
- [2] Adobe. (2023). Digital price index. Retrieved from <https://business.adobe.com/resources/digital-price-index.html>
- [3] McKinsey & Company. (2023). The future of retail. Retrieved from <https://www.mckinsey.com/industries/retail/our-insights>
- [4] McKinsey & Company. (2021, December 30). Solving the paradox of growth and profitability in e-commerce. Retrieved from <https://www.mckinsey.com/industries/retail/our-insights/solving-the-paradox-of-growth-and-profitability-in-e-commerce>
- [5] Harvard Business School. (2020). Lifting the veil: The benefits of cost transparency.
- [6] Accenture. (2021). Personalized marketing services & solutions.

Undertaking of Kaavish advisement as an External Supervisor

I hereby affirm that I have read the project details as described on the preceding pages and agree to undertake advisement of this Kaavish project as an External Supervisor. I understand that this role entails the following.

Meeting Meeting the project team regularly, at least once every two weeks, for the entire duration of the Kaavish. The meetings may be held remotely if required.

Advisement Providing supervision and advice to the team in order to ensure steady progress of the project toward its goals.

Liaison Liaising with the Internal Supervisor as required, e.g. to provide feedback or engage in grading.

Other Any other task, depending on availability and suitability, relevant to the Kaavish as communicated by the Internal Supervisor or Kaavish Working Group.

Name: _____

Email: _____

Phone: _____

Designation: _____

Affiliation: _____

Signature: _____