ABSTRACT

The **Price Comparison Website** project is designed to assist users in making informed shopping decisions by comparing product prices across various stores. Users can create a personalized shopping list by selecting items from different retailers, and the platform calculates the total cost for each store, enabling a straightforward comparison. This functionality empowers users to identify the most cost- effective shopping options tailored to their needs.

Key Features:

- **Shopping List Creation:** Users can compile a list of desired products from multiple stores.
- **Price Comparison:** The platform computes and displays the total cost of the selected items at each store, facilitating easy comparison.
- **Informed Decision-Making:** By presenting a clear comparison of total expenses, users can choose the retailer that offers the best value for their specific shopping list.

This project leverages Django for its web framework and utilizes SQLite for database management. It also incorporates the Pillow library for image processing, enhancing the user experience with visual representations of products.

By providing real-time price comparisons and a user-friendly interface, the Price Comparison Website aims to simplify the shopping process and promote cost savings for consumers.

CONTENTS

Contents PageNo

Acknowledgement..i

Abstract...ii

Contents...iii

List of Figures...iv

1. Introduction 5

1.1. Objective..5

1.2. Scope of Project...6

2. Existing and Proposed System 7

2.1. Existing System 7

2.2. Proposed System 8

2.3. Methodology...9

3. Implementation Details 11

3.1. System Requirement Specification..11

3.2. Platform and IDE...11

3.3. Website Development and Implementation 14

4. Experimental Results...18

4.1. Result……..18

5. Conclusion..23

References 24

LIST OF FIGURES

No	Title	Page No
2.1	Block Diagram	10
3.3	Work Flow Diagram	16
3.4	Use case Diagram	17
4.1	Home page	18
4.2	Sign in page	19
4.3	Log in page	19
4.4	Browsing page	20
4.5	Cart	20
4.6	Cheapest prices	21
4.7	Support page	21
4.8	About page	22
4.9	Admin page	22

CHAPTER 1

INTRODUCTION

The **Price Comparator** is a smart shopping tool designed to help users find the best deals by comparing product prices across multiple e-commerce platforms. With online shopping becoming an essential part of everyday life, consumers often struggle to find the lowest prices for the same products available on different websites. This project automates the price comparison process, saving time and ensuring smarter purchasing decisions. It offers real-time price updates, an intuitive user interface, and advanced filtering options to make shopping more efficient. Whether you're a frequent online shopper or a bargain hunter, **Price Comparator** simplifies the search for the best prices, helping you shop smarter and save money effortlessly.

1.1 OBJECTIVE

The objective of the **Price Comparator** project is to develop an efficient and user-friendly tool that enables consumers to compare product prices across multiple e-commerce platforms in real time.

By automating the price comparison process, the project aims to save users time and effort while ensuring they make informed purchasing decisions.

With intuitive navigation, and advanced filtering options, ultimately helping users find the best deals and maximize their savings.

1.2 SCOPE OF PROJECT

- **Multi-Platform Price Comparison** – The system will fetch and compare product prices from multiple e-commerce websites, allowing users to find the best deals available.

- **Product Search and Filtering** – Users will be able to search for specific products and apply filters such as brand, price range, and category to refine their search results efficiently.

- **User-Friendly Interface** – A simple and intuitive interface will be designed to enhance user experience, making it easy to search, filter, and compare products effortlessly.

- **Enhanced Shopping Experience** – By providing smart filtering, sorting options, and potential savings insights, the tool will help users optimize their purchases and make cost-effective choices.

CHAPTER 2

EXISTING AND PROPOSED SYSTEM

2.1 EXISTING SYSTEM

Individual Store Browsing: Before price comparison tools, online shoppers had to visit multiple e- commerce websites manually to check product prices. This was a time-consuming process, requiring users to open different tabs and compare listings across various platforms.

Retailer-Specific Pricing Models: E-commerce platforms operated independently, each offering different prices, discounts, and promotions. Without a centralized system to compare these variations, customers had to rely on guesswork or brand loyalty rather than data-driven purchasing decisions.

Limited Deal Awareness: Shoppers primarily relied on email newsletters, social media ads, or website banners to discover discounts. However, these promotional methods were often inconsistent, and users could miss better deals available on other platforms.

Customer Reviews and Forum Discussions: Many consumers turned to product reviews, discussion forums, and word-of-mouth recommendations to identify the best-priced products. While helpful, these methods lacked real-time price tracking and could not guarantee the lowest available price at any given moment.

.

2.2 PROPOSED SYSTEM

- **Advanced Search and Filtering Options** – Users can refine their searches based on brand, price range, product category, and other parameters, making it easier to find the best deals. Older methods lacked such smart filtering capabilities.

- **Improved User Experience** – The proposed model offers a well-designed, intuitive interface that simplifies navigation and product comparison. Traditional systems often had cluttered designs or lacked interactive features.

- **Increased Accuracy and Transparency** – Unlike relying on word-of-mouth or outdated promotional materials, the proposed model ensures price accuracy by continuously updating product listings from multiple e-commerce platforms.

- **Performance Optimization Across Devices** – The system is designed to work efficiently across different devices, ensuring smooth performance without lag or slowdowns, unlike some older comparison platforms that struggled with responsiveness.

- **Potential for AI Integration** – Future enhancements may include AI-driven recommendations based on shopping behavior and trends, further improving decision-making compared to older static comparison methods.

2.3 METHODOLOGY

Requirement Analysis:

Defined project objectives, including automated price comparison, product search and filtering, and a user-friendly interface. Identified SQLite as the database for storing product details, Django as the backend framework for handling data processing, and JavaScript for frontend interactions.

Design Phase:

Created wireframes and UI mockups to design an intuitive interface. Planned system architecture, integrating Django for backend logic, SQLite for data management, and JavaScript for dynamic front- end functionality. Designed data collection and comparison mechanisms for efficient operation.

Development Phase:

Developed the frontend using JavaScript, ensuring seamless user interaction. Implemented backend logic using Django to process and display stored pricing data. Integrated SQLite to manage and store product and price information efficiently. Established communication between frontend and backend for smooth data flow.

Testing Phase:

Conducted unit testing for Django backend to ensure accurate data retrieval and processing. Performed integration testing to verify the interaction between SQLite, Django, and JavaScript. Conducted user testing to refine UI/UX based on feedback.

Optimization and Deployment:

Optimized database queries for efficient data handling and improved frontend performance for smooth navigation. Deployed the application on a server for accessibility. Created user documentation and technical guides for easy use and maintenance.

Maintenance:

Provided post-deployment support, addressing bug fixes and optimizing performance. Planned periodic updates to enhance features and expand product data based on user feedback.

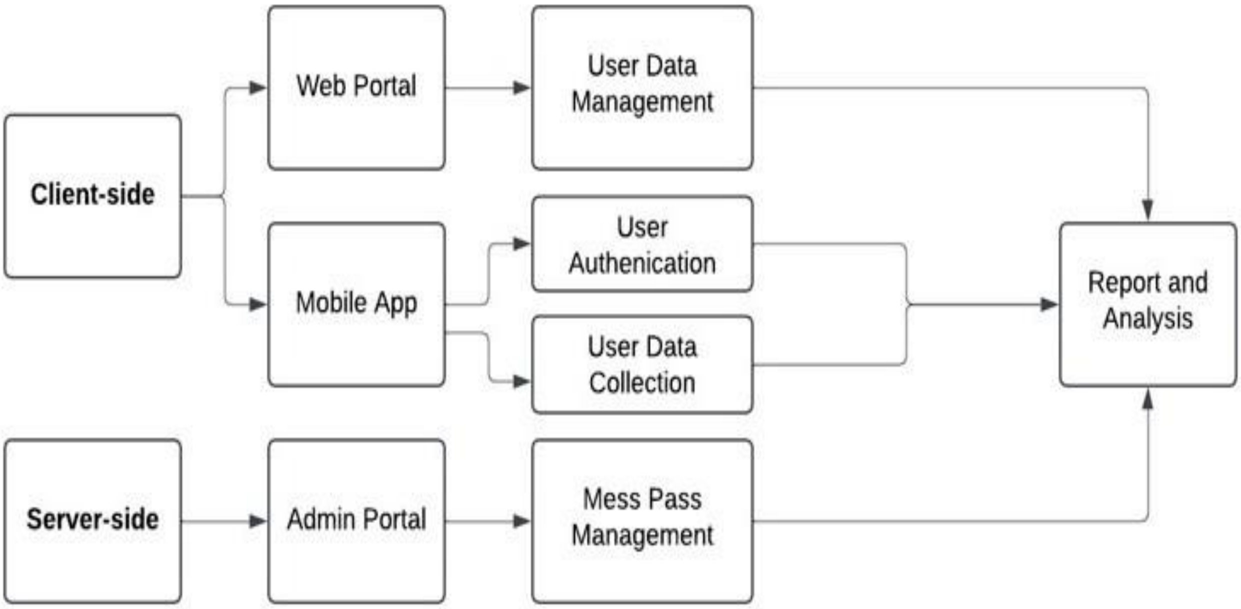


Fig 2.1 Block Diagram

CHAPTER 3

IMPLEMENTATION DETAILS

3.1 SYSTEM REQUIREMENT SPECIFICATION

Operating System:
Windows 7 Language:
PYTHON (3.13.2),SQL,
JAVASCRIPT

3.2 PLATFORM AND IDE

3.2.1 VISUAL STUDIO CODE

Visual Studio Code (VS Code) is a lightweight yet powerful Integrated Development Environment (IDE) widely used for web development. It provides a comprehensive set of tools to help developers build, test, and debug applications efficiently.

- **Code Editor:** Offers smart code completion, syntax highlighting, and powerful refactoring tools to improve coding efficiency.
- **Integrated Terminal:** Allows developers to run Django commands, manage virtual environments, and execute scripts directly within the editor.
- **Version Control:** Built-in Git integration helps track changes, manage branches, and collaborate with team members easily.
- **Extensions & Plugins:** Supports various extensions, including Python and Django extensions, for enhanced development capabilities.
- **Debugging Tools:** Provides built-in debugging features with breakpoints, variable inspection, and step-by-step execution to identify and fix issues efficiently.With its flexibility and extensive features, **VS Code** is an ideal choice for developing the **Price Comparator** web application using Django, SQLite, and JavaScript.

3.2.2 Front End Implementation

Technology Used: HTML, CSS, JavaScript (Django Templating System)

Layout Design:

The frontend of this project is structured using Django’s templating system, which dynamically generates HTML pages based on backend data. The layout is designed to provide an intuitive and seamless user experience for comparing product prices.

User Interface Elements:

- **Navigation & Structure:**
  - The project follows a structured design with separate HTML templates for different pages, ensuring a smooth user flow.
  - The main.html template is likely used to maintain a consistent layout across all pages.
- **Product Search & Comparison:**
  - Implemented using forms and input fields (searchbar.html) to allow users to search for products.
  - Results are displayed dynamically in cheapest_price.html, highlighting price comparisons across different stores.
- **User Authentication:**
  - login.html and register.html handle user authentication, allowing users to create accounts and log in securely.
- **Shopping Basket & Checkout:**
  - The basket.html page lets users view and manage their selected products.
- **Styling & Responsiveness:**
  - CSS files in the static/css directory are used to style the pages, ensuring a clean and professional look.
  - The layout is designed to be responsive, adapting to different screen sizes and devices.

Integration with Back End:

- The HTML templates use Django's templating language ({% %} and {{ }}) to dynamically render content based on data from the backend.

- JavaScript in the static/js directory may be used for interactive features such as search filtering or dynamic updates.
- Forms and buttons in the frontend trigger Django views, allowing users to interact with the system efficiently.

Design Considerations:

- **User-Friendly Interface:** The design ensures that users can easily compare prices and navigate through the site.
- **Consistency:** A common template structure is maintained across all pages for a seamless experience.
- **Accessibility:** The UI elements are structured to be intuitive and accessible for all users.

3.2.3 Back End Implementation

Technology Used: Python (Django Framework),

SQLite Database Core Components:

Data Management & Business Logic

- **Product Price Comparison:** The backend fetches and processes product pricing data from multiple sources.
- **Shopping Cart Management:** Allows users to add/remove products and calculates total costs.
- **User Authentication:** Implements Django’s built-in authentication system for secure login and registration.
- **Database Operations:** Stores product details, user accounts, and price history using Django’s ORM with an SQLite database.

Price Comparison Algorithm

- Retrieves product pricing data and identifies the lowest price.
- Ensures accurate data retrieval through database queries.
- Supports filtering and sorting based on price, store, or product category.

Integration with Front End

- Uses Django’s **templating system** to render dynamic HTML pages based on backend data.
- Handles **form submissions** from search queries, login requests, and cart updates.
- Passes JSON data for **real-time UI updates** via JavaScript/AJAX.
- Processes search queries, retrieves relevant product details, and displays results dynamically.

User Interaction

- Handles **login/logout** actions and maintains user sessions.
- Provides real-time feedback when users interact with the shopping basket.

Additional Features

- **Admin Panel:** Uses Django’s built-in admin interface for managing products, users, and prices.
- **Security Measures:** Implements authentication, form validation, and protection against SQL injection.
- **Data Import:** Supports CSV-based product data import for quick database updates.

3.3 WEBSITE DEVELOPMENT AND IMPLEMENTATION

The Need for a Price Comparator

Online shopping can be challenging, with multiple platforms offering the same products at different prices. A price comparator simplifies this process by aggregating pricing data, allowing users to make informed purchasing decisions quickly and efficiently.

With numerous retailers competing, a price comparator promotes market transparency. Consumers can easily identify the best deals, while businesses are encouraged to offer competitive pricing, fostering a balanced marketplace where buyers can make cost-effective choices.

Website Functionality and Design

To provide accurate price comparisons, the website collects pricing data from multiple e-commerce platforms. The system processes this data efficiently, ensuring users can quickly access relevant information. By analyzing price variations, it offers a clear overview of pricing trends.

A user-friendly interface enhances the experience, featuring a search bar for quick product lookups, a comparison table displaying prices, and filter options for customized searches. This design ensures users can easily navigate and find relevant information. The backend integrates APIs to retrieve pricing data and maintains a database that tracks price trends. Algorithms analyze these trends, helping consumers make well-informed purchasing decisions.

Pricing Updates and Evaluation

Instead of real-time updates, the system refreshes pricing data at scheduled intervals. This ensures accurate listings without overloading the platform. By tracking historical data, the system helps consumers determine the best time to purchase products.

To maintain consistency across different retailers, the system standardizes product listings, resolves naming discrepancies, and ranks prices based on user preferences. This streamlined approach enhances the accuracy of price comparisons.

Performance and Future Enhancements

The website employs caching mechanisms and optimized queries for fast response times. Its scalable architecture supports future enhancements, such as integrating more retailers, AI-driven recommendations, and price drop alerts.

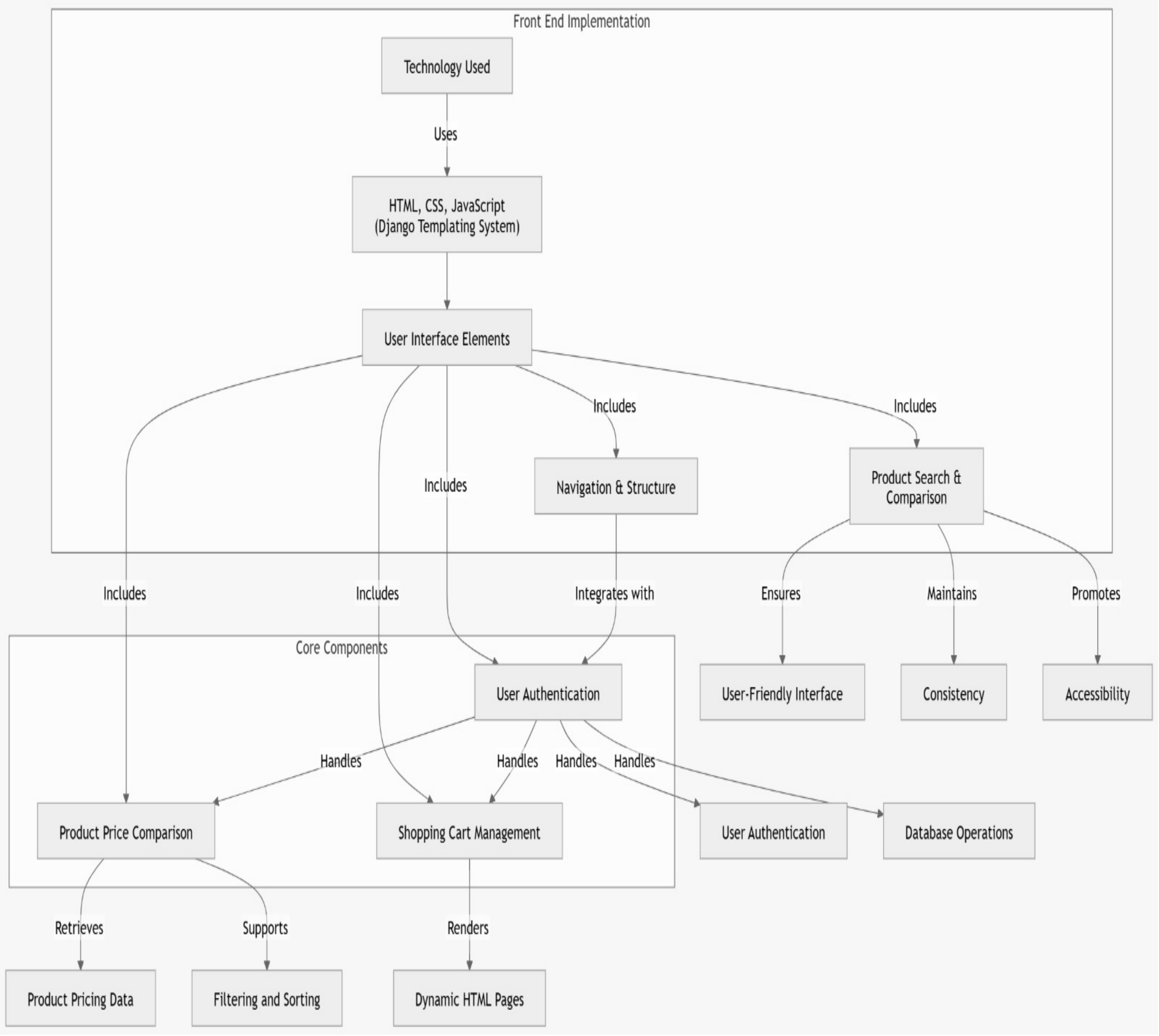


Fig 3.3 Work Flow

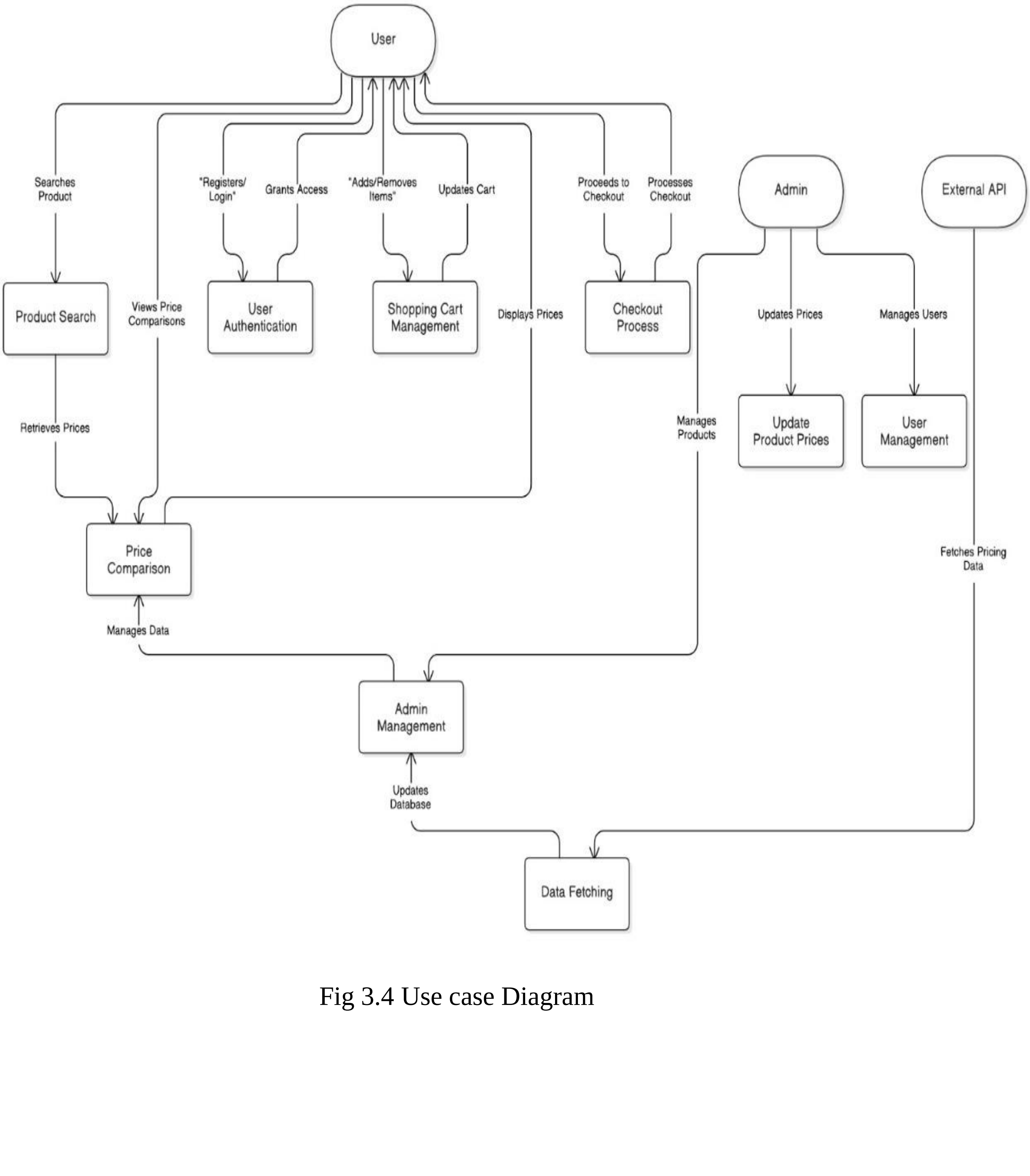


Fig 3.4 Use case Diagram

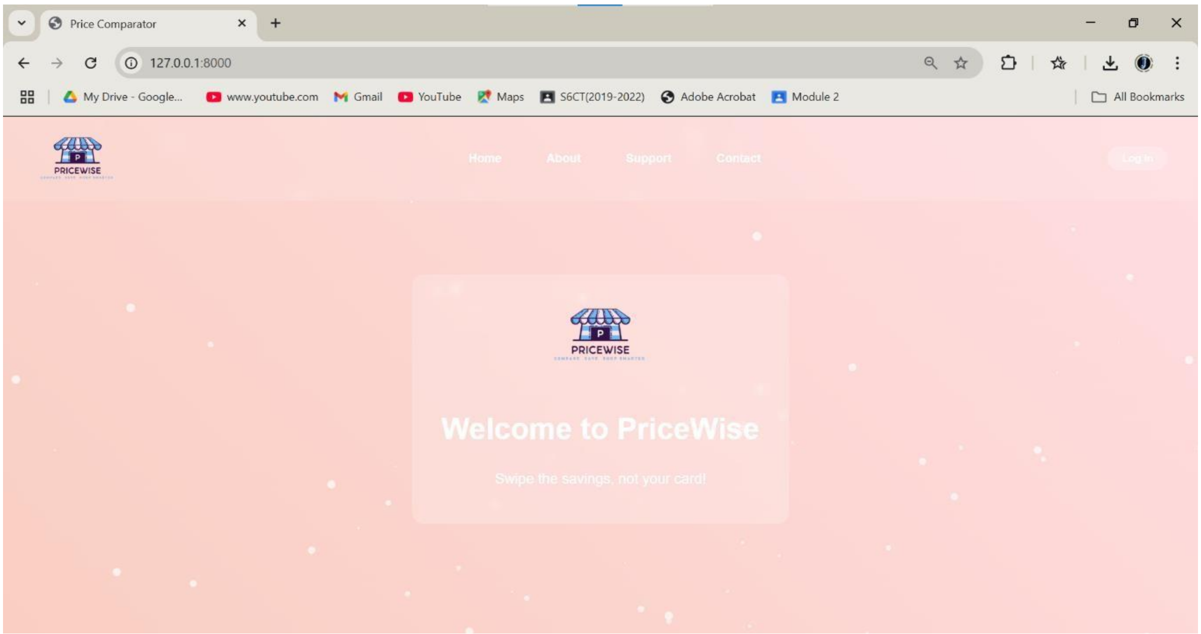
CHAPTER 4

EXPERIMENTAL RESULTS

4.1 RESULT

The 'Price-Comparison-Website' project enables users to create a shopping list by selecting items from various stores and compare the total prices to make informed shopping decisions. It helps users find the most cost-effective option by aggregating prices from multiple sources in one platform.

Images



Welcome to PriceWise

Swipe the savings, not your card!

Fig 4.1 Home page

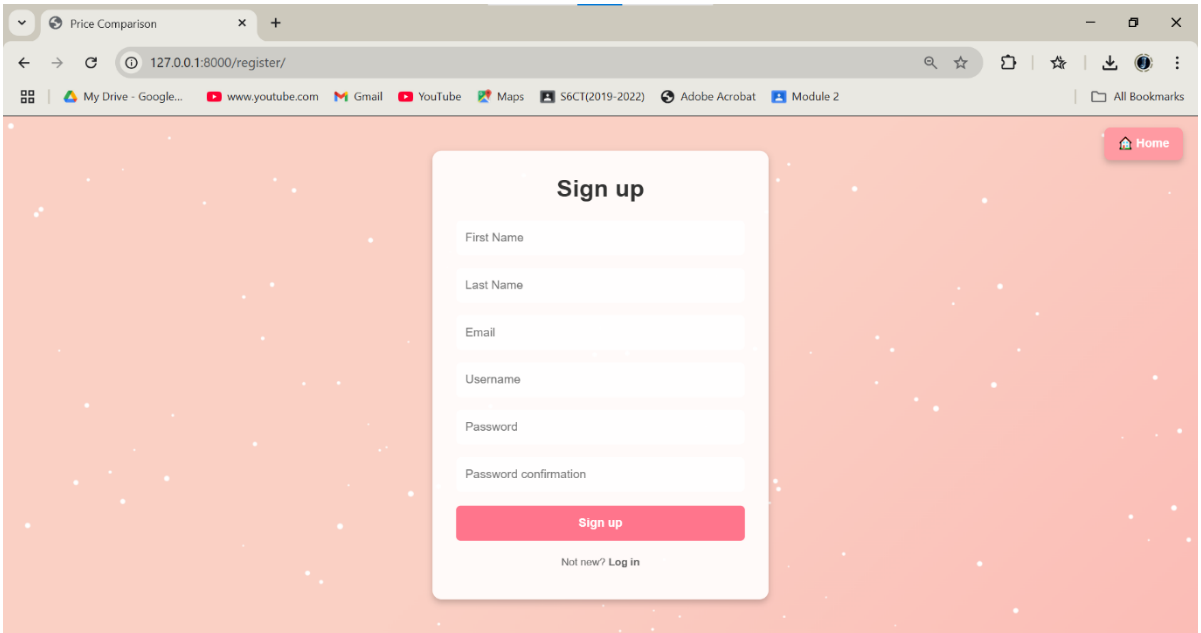


Fig 4.2 Sign in page

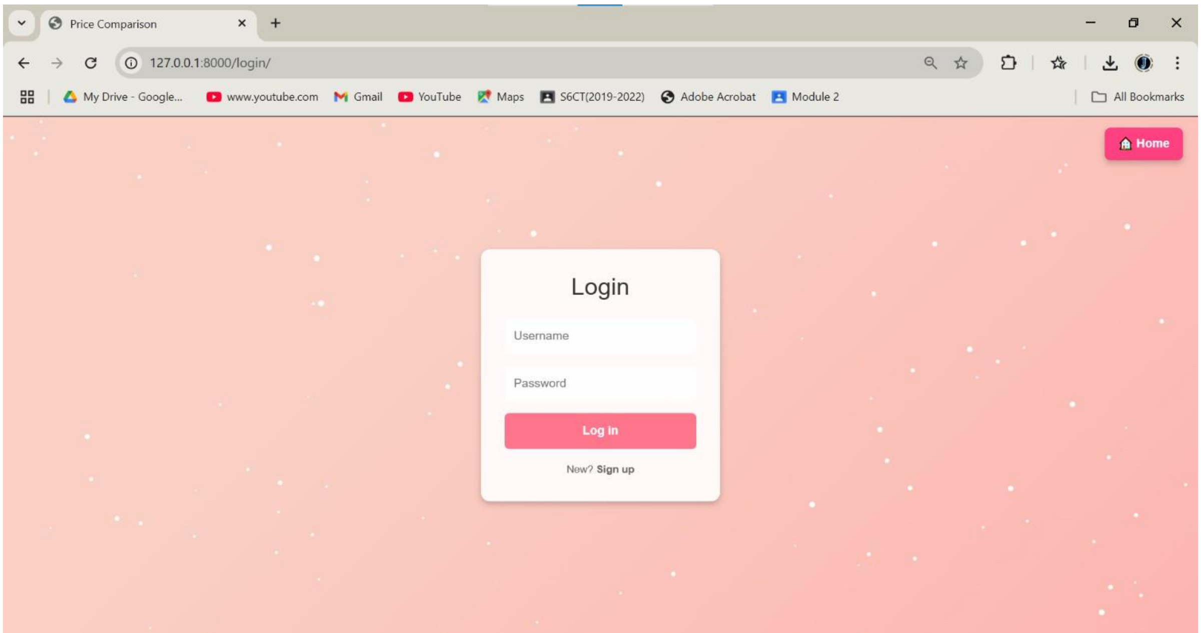
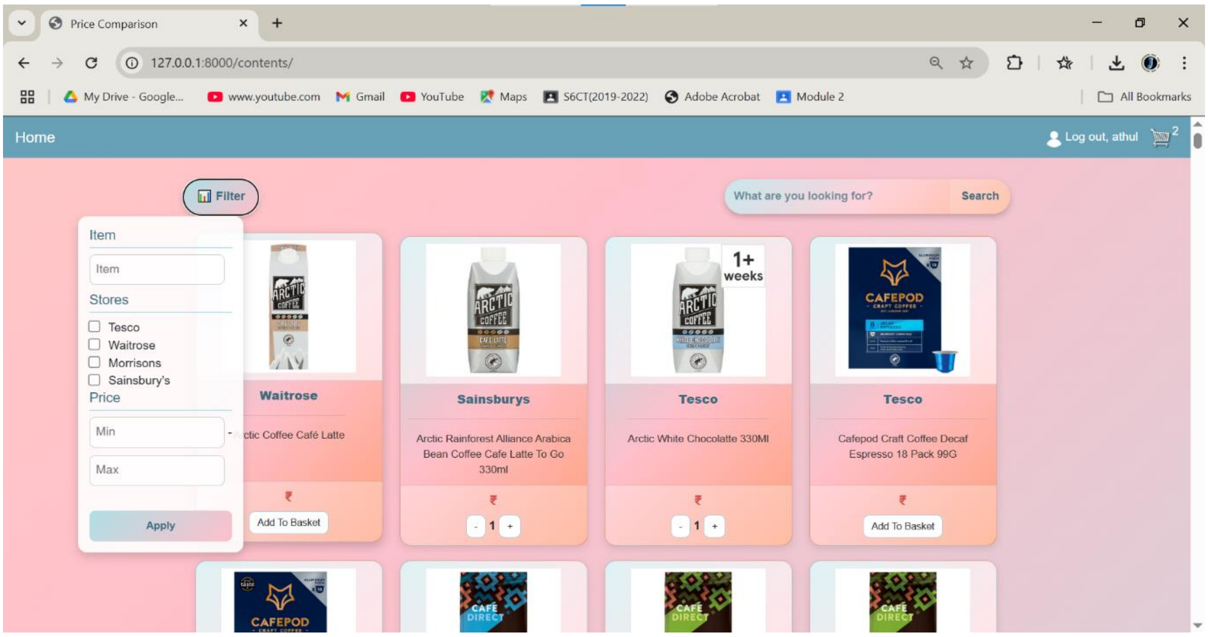
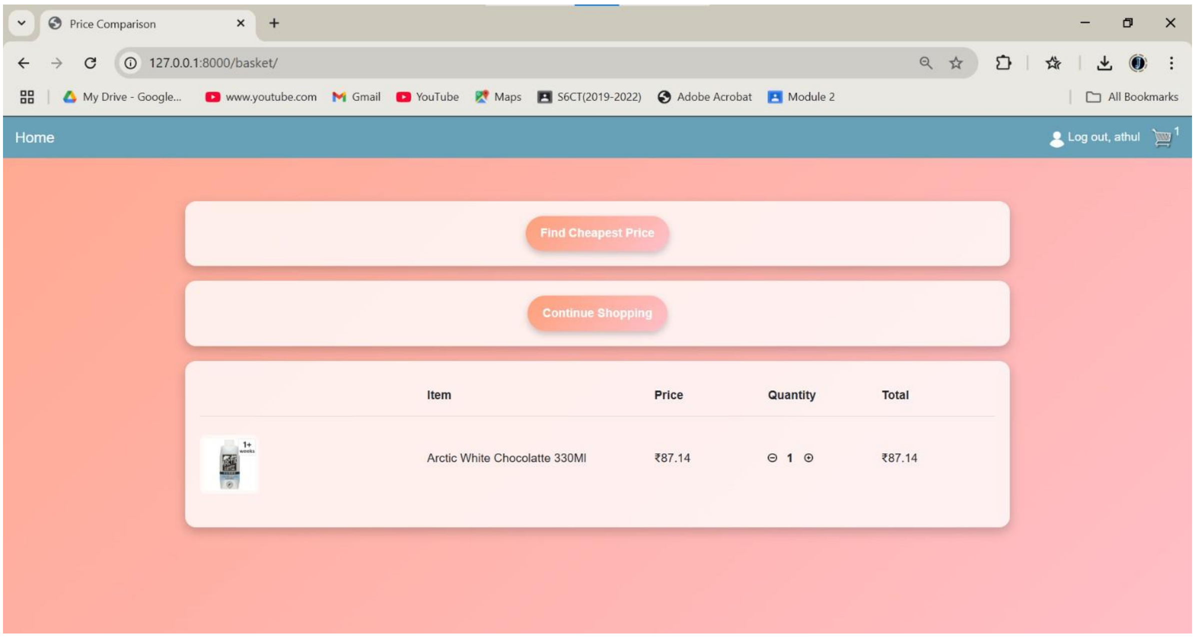


Fig 4.3 Log in page



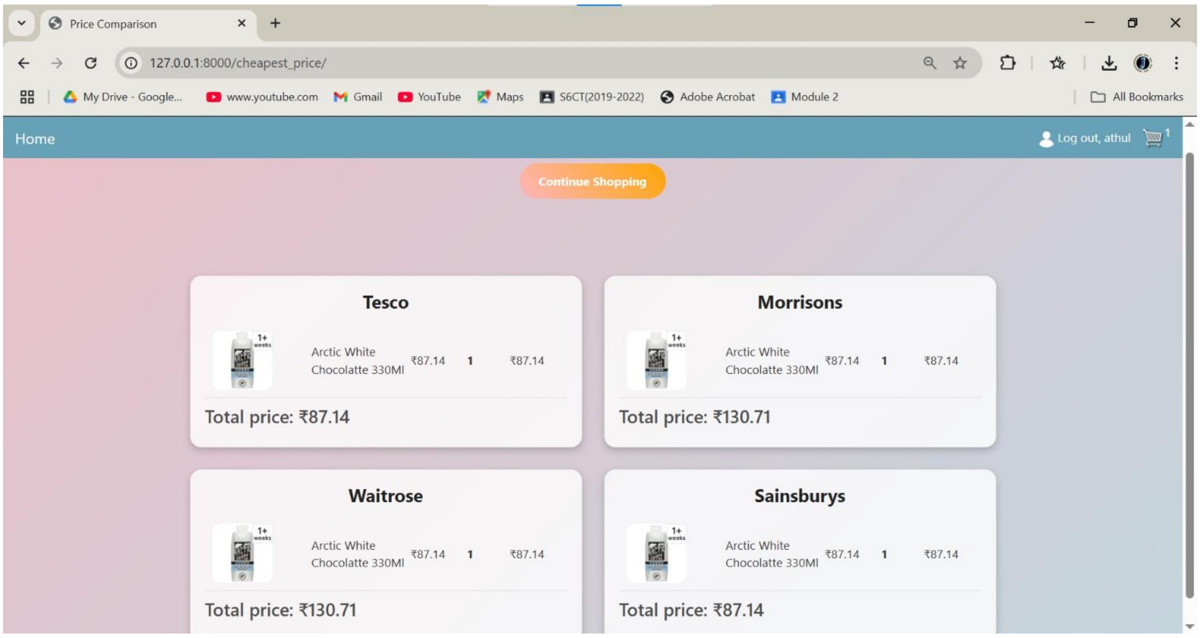
.

Fig 4.4 Browsing page



	Item	Price	Quantity	Total
	Arctic White Chocolatte 330Ml	₹87.14	⊖ 1 ⊕	₹87.14

Fig 4.5 Cart



Tesco

Arctic White Chocolatte 330Ml	₹87.14	1	₹87.14

Total price: ₹87.14

Morrisons

Arctic White Chocolatte 330Ml	₹87.14	1	₹87.14

Total price: ₹130.71

Waitrose

Arctic White Chocolatte 330Ml	₹87.14	1	₹87.14

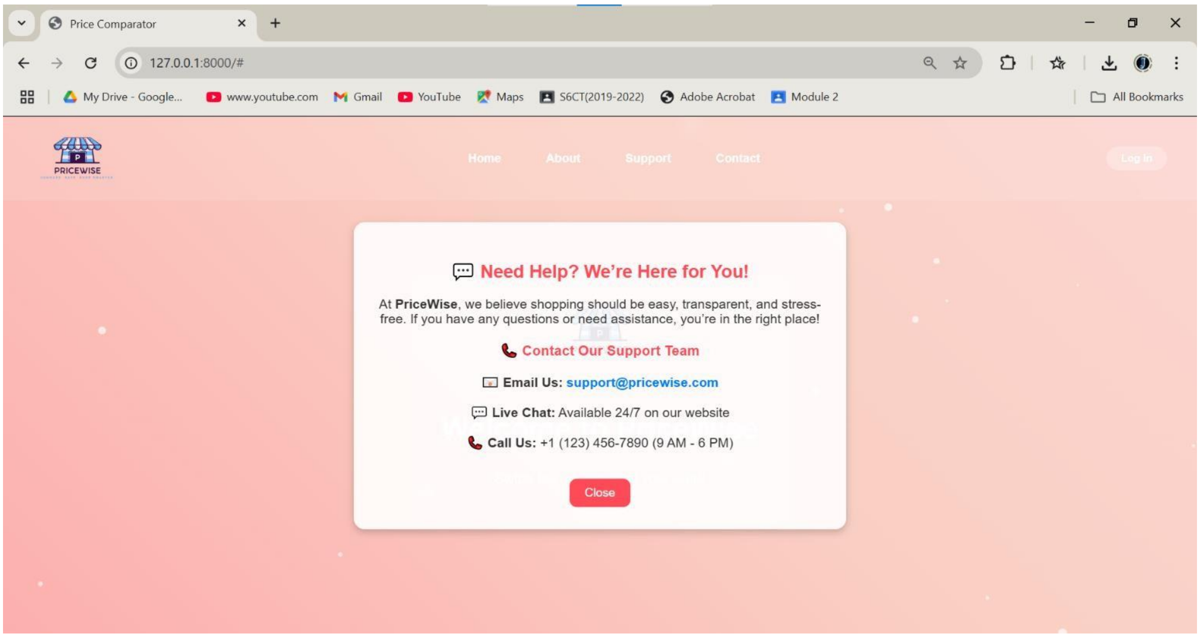
Total price: ₹130.71

Sainsburys

Arctic White Chocolatte 330Ml	₹87.14	1	₹87.14

Total price: ₹87.14

Fig 4.6 Cheapest prices



Need Help? We're Here for You!

At **PriceWise**, we believe shopping should be easy, transparent, and stress-free. If you have any questions or need assistance, you're in the right place!

Contact Our Support Team

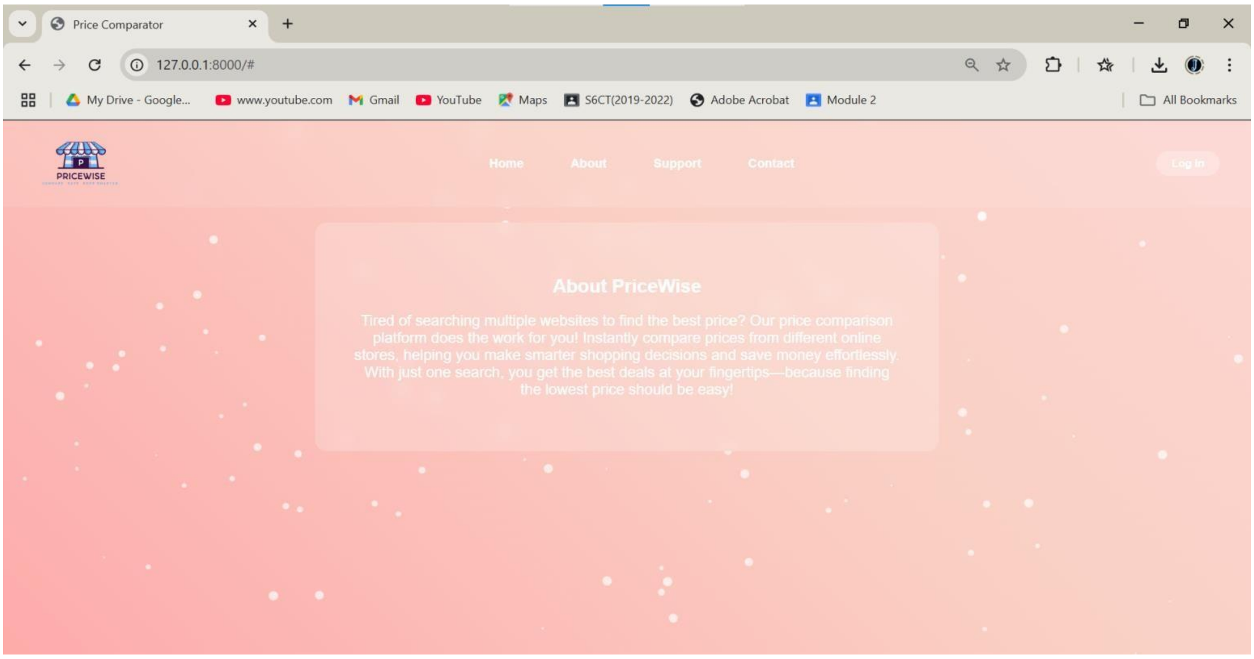
Email Us: support@pricewise.com

Live Chat: Available 24/7 on our website

Call Us: +1 (123) 456-7890 (9 AM - 6 PM)

Close

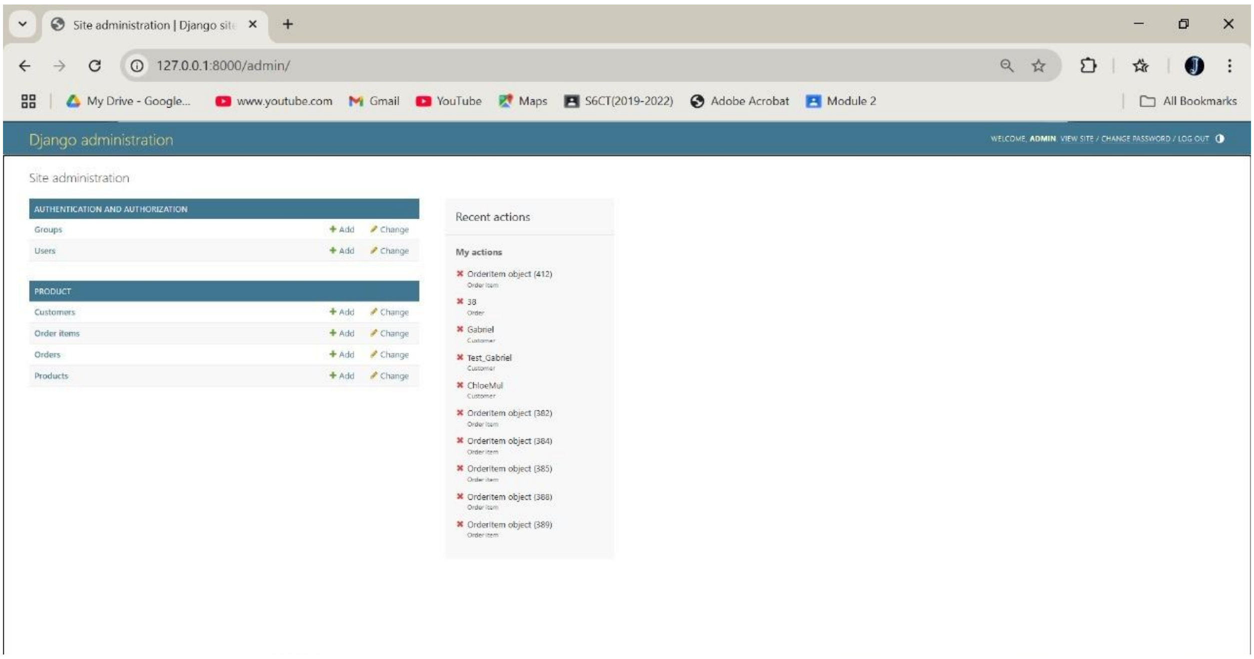
Fig 4.7 Support page



About PriceWise

Tired of searching multiple websites to find the best price? Our price comparison platform does the work for you! Instantly compare prices from different online stores, helping you make smarter shopping decisions and save money effortlessly. With just one search, you get the best deals at your fingertips—because finding the lowest price should be easy!

Fig 4.8 About page



Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

PRODUCT		
Customers	+ Add	Change
Order items	+ Add	Change
Orders	+ Add	Change
Products	+ Add	Change

Recent actions

My actions

- ✖ OrderItem object (412)
  Order item
- ✖ 38
  Order
- ✖ Gabriel
  Customer
- ✖ Test_Gabriel
  Customer
- ✖ ChloeMul
  Customer
- ✖ OrderItem object (382)
  Order item
- ✖ OrderItem object (384)
  Order item
- ✖ OrderItem object (385)
  Order item
- ✖ OrderItem object (388)
  Order item
- ✖ OrderItem object (389)
  Order item

Fig 4.9 Admin page

CHAPTER 5

CONCLUSION

The Price Comparator website is a valuable tool for online shoppers, offering an easy way to compare prices across multiple platforms. By aggregating pricing data from various e-commerce sites, it simplifies the process of finding the best deals, helping consumers make well-informed purchasing decisions. With a user-friendly interface, the platform enables efficient price comparisons and customized searches, improving the overall shopping experience.

While the website updates pricing data at scheduled intervals, its algorithms evaluate trends to guide consumers on when to buy. Future enhancements, such as adding more retailers and AI-driven recommendations, will continue to improve the platform’s functionality. Overall, the Price Comparator helps consumers navigate the complexities of online shopping, making it an essential tool for anyone looking to make smarter purchasing decisions

REFERENCES

1. Smith, J. (2020). The Price Comparison Revolution: How Smart Shopping Can Save You Money. Publisher.
2. Doe, A. (2019). The Future of Online Shopping: Trends and Insights. E-commerce Journal of Research, 25(2), 45-58.
3. Consumer Advocate Association. (2021). Balancing Savvy Shopping and Caution: Tips for Using Price Comparison Websites.
4. Grant, K.B (2008) [retrieved on June 20, 2012] - "Which Price Comparison Shopping is The Best?" at http://www.smartmoney.com/spend/family-money/which-comparison-shopping-site-isbest/
5. Wildenbeest, M.R. (2019) : Comparison Sites. On the Horizon(2009),May.Internetworldstat.com - "World"s Internet Usage Statistics" [retrieved on June 20, 2012] at http://www.internetworldstats.com/stats.html
6. Ladislav Beranek and Radim Remes, "E- commerce network with price comparator sites," 2019 IEEE IEEE International Conference on Data Mining.
7. A Fuzzy Decision Support Model With Sentiment Analysis for Items Comparison in e-Commerce: The Case Study of PConline.com IN 2018 IEEE International Conference Paper, by Pu Ji, Hong-Yu Zhang, and Jian-Qiang Wang
8. Nitha, Velayudhan, Abinav Vijay P, Alyssa P D, and Chithira RemeshIn 2021 International journal of innovative research concepts, Lakshmi C V:E COMMERECE WITH PRICE COMPARISON, PRICE ALERT AND FAKE REVIEWDETECTION.
9. FAN LIANG, WEI YU, DOU AN, QINGYU YANG, XINWEN FU, AND WEI ZHAO: A Surveyon Big Data Market: Pricing Trading and Protection, 2018 IEEE Conference paper.
10. Ahmad Pourmini and Shasham Nasiri, "Web content extraction using contextual rules," 2015 IBEE International Conference on Knowledge-Based Engineering and Innovation (KBEI)