Software Requirements Specification

For

MINOR PROJECT- SYNOPSIS

TITLE: BUDGET BASKET
(E-COMMERCE PLATFORM)

06th March 2025

Prepared by

Specialization	SAP ID	Name
CSE DATA SCIENCE(B3 Non-Hons)	500105016	Roshita Ailawadi
CSE DATA SCIENCE(B1 Hons)	500108000	Somya Yadav
CSE AI/ML(B-5 NonHons)	500109627	Manas Arora
CSE DATA SCIENCE(B3 Non-Hons)	500110143	Zaira Khan

School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

R. Dhinesh Kumar

(06/03/2025)

Panel Member's Signature **Mentor's Signature**

Table of Contents

Topic		Page No
Table of Content		2-5
1	Introduction	3
	Literature Review	2-3
	1.1 Target Beneficiary	3
	1.2 Project Scope	4
	1.3 References	4-5
2	Project Description	5-12
	2.1 Reference Algorithm	5
	2.2 Data/ Data structure	6
	2.3 SWOT Analysis	7
	2.4 Project Features	8
	2.5 User Classes and Characteristics	9
	2.6 Design and Implementation Constraints	9
	2.7 Design diagrams	10-11
	2.8 Assumption and Dependencies	12
3	System Requirements	12-14
	3.1 User Interface	12
	3.2 Software Interface	13
	3.3 Database Interface	13
	3.4 Protocols	14
4	Non-functional Requirements	14-15
	4.1 Performance requirements	14
	4.2 Security requirements	14
	4.3 Software Quality Attributes	15
5	Other Requirements	15
Appendix A: Glossary		16
Appendix B: Analysis Model		17
Appendix C: Issues List		17

1. Introduction

The Budget Basket project aims to provide consumers with a platform to compare grocery prices across multiple stores, ensuring they get the best deals. Thanks to fluctuations in products for food and increased demand for economically effective purchases, consumers often follow prices in real time. This platform helps users save time and money.
Actual price update. Monitoring product availability.
Personalized recommendations based on user preferences. Integrating scraping models, APIs and machine learning searches on the Internet, the platform predicts price trends and provides the optimal purchase period.

Total Population of India
(Data according to 2020)

Figure 1 Shoping Population of India(2020) [5]

LITERATURE REVIEW

A combined model of **online grocery buying intentions** was proposed in [1] , extending the **Technology Acceptance Model (TAM)** to include factors such as **physical effort, time constraints, entertainment value, product assortment, economic value, and website design**. The study concluded that these factors significantly influence user satisfaction with online grocery shopping platforms, providing insights into consumer preferences and e-grocery adoption.
The research in [2] explored the development of **online recommender systems** for traditional offline grocery stores. The study focused on leveraging **RFID technology and plan recognition** to track and predict consumer behavior in retail spaces. The proposed system aimed to balance the interests of both sellers and buyers by using data-driven insights to enhance the in-store shopping experience.
The role of **price comparison websites** in modern e-commerce was analyzed in [3] , highlighting how these platforms aggregate product information from multiple vendors, allowing consumers to compare prices efficiently. This enhances **market transparency and**

competitiveness, particularly benefiting time-conscious urban consumers seeking quick, well-informed purchasing decisions.
Further research in [4] examined how price comparison platforms enable consumers to make **informed purchasing decisions**. By centralizing product prices across multiple vendors, these platforms reduce the need to visit different websites or physical stores. The study emphasized the growing reliance on such tools, especially among **busy urban consumers** looking for the best deals.
Additionally, [5] investigated the impact of **price comparison websites** on consumer shopping experiences. The findings indicated that these platforms improve convenience by consolidating price information from various suppliers, helping shoppers identify the best value for their purchases. The study underscored the increasing role of price comparison tools in optimizing **consumer decision-making** and enhancing the competitiveness of online retailers.

1.1 Target Beneficiaries

The **main** beneficiaries of this project **are:**
Consumer buyers are looking for **budget transactions.**
Retailers **and store owners are companies** that want to **manage the prices of competitors.**
Platforms **and suppliers of electronic trade potential** partners **who** can **ensure the integration of** data **in real time.**
Administrative users - is responsible for **the management of** product data and **the supply of** system accuracy.

1.2 Project Scope

The budget basket platform focuses on product comparisons for both local and online store products. This includes:

Benefits and important objective
Reduce the cost of purchasing products and identify the best prices. He monitors historic prices and predicts future prices.
Improves the user experience thanks to personalized recommendations and alerts.
Requirements & Deliverables
Data Collection: Web scraping and API integration for real-time pricing.
Comparison Algorithms: Dijkstra's algorithm for price optimization. Machine Training Model: Price Prediction Timeline (Arima, LSTM).
User Features: Survey, Filter, Notifications, Management Accounts. Safety and Compatibility: Follow RGPD and CCPA rules to protect your user data.

1.3 References

[1] A. Sreeram, A. Kesharwani, and S. Desai, "A combined model of online grocery buying intentions based on the Technology Acceptance Model," *Journal of Retailing and Technology*, vol. 12, pp. 45–58, 2017.

[2] D. C. Buser, "Development of online recommender systems for offline grocery stores using RFID technology and plan recognition," *Journal of Retailing Innovation*, vol. 6, pp. 121–134, 2007.

[3] H. Dharmik, P. Padmane, and others, "The role of price comparison websites in enhancing online shopping decisions," *Journal of E-Commerce Technology*, vol. 18, pp. 92–107, 2022.

[4] K. Varun, P. Rajesh, and others, "Websites that compare prices: Empowering customers to make informed purchase decisions," *International Journal of E-Commerce*, vol. 15, pp. 210–223, 2023.

[5] S. Bezalwar and others, "Evaluating price comparison websites and their impact on the consumer shopping experience," *Journal of Retailing Technology*, vol. 17, pp. 85–98, 2022.

2. Project Description

The **Budget Basket** project aims to create a grocery price comparison platform that helps consumers find the best prices for grocery items across multiple local and online stores. The platform will provide users with real-time price updates, product availability tracking, and personalized recommendations by leveraging data collection, price comparison algorithms, and predictive analytics.

2.1 Reference Algorithm

The platform will use a combination of the following algorithms for price comparison and prediction:

- **Web Scraping & API Integration Algorithm:**

Scrapes or retrieves product prices from multiple grocery store websites and APIs.
  - Ensures real-time or periodic price updates.
  - Uses data cleaning techniques to handle missing or inconsistent pricing data.
- **Price Comparison Algorithm (Dynamic Pricing Model):**
  - Implementing Dijkstra's Algorithm to determine the best price path in multi-store analysis by treating stores as nodes and prices as edge weights. This helps identify the cheapest possible route for a given product.
  - Using A/B testing and statistical models to validate pricing trends.
  - Providing users with the best price options for their selected grocery items.
  - Ranking stores based on affordability, quality considerations, and user preferences.
- **Price Prediction Algorithm (Time Series Analysis / Machine Learning):**

Using Time Series Forecasting (ARIMA, LSTM) to predict future price trends.
Implementing Machine Learning Regression Models (Linear Regression, Random Forest) to estimate upcoming price variations.
Providing users with insights into optimal purchase periods.

- **Data Visualization for Consumer Insights**

Interactive visual tools such as Matplotlib, Plotly, and D3.js will be integrated to graph price trends. These tools will provide insights into price fluctuations, seasonal trends, and budget optimization strategies, enhancing user decision-making.

2.2 Data / Data Structure

The project will involve collecting and managing large datasets related to grocery items, prices, store locations, and user preferences.
Data Sources:

- **Web Scraping / API Integration:** Extracts real-time prices from grocery store websites.
- **User Inputs:** User preferences, shopping history, and search behavior.
- **Historical Price Data:** Used for trend analysis and price prediction.

Data Structure:

- **Product Information Table** (Structured Data –Excel)
  - Product ID, Name, Brand, Category, Unit Price, Discount Price, Store ID, Availability Status.
- **Store Information Table** (Structured Data – SQL Database)
  - Store ID, Store Name, Location, API/Scraper Source, Rating.
- **Price History Table** (Time-Series Data – SQL/NoSQL)
  - Date, Product ID, Store ID, Price, Discount.
- **User Data Table** (User Profile – SQL/NoSQL)
  - User ID, Shopping Preferences, Search History, Wishlist.

About the Data

The dataset for the Budget Basket project was collected manually through a hands-on approach. Over the course of one month, we visited five local grocery superstores situated within a 7 km radius to capture real-time pricing data. Each day, we collected the after-discount prices for 35 daily essential products across these stores. The stores included in the dataset are: Arora Store, Bharti Store, Tiwari General Store, Dadi General Store, and Dhiman General Store. This process ensured a comprehensive and accurate representation of pricing trends, providing valuable insights into product price fluctuations and enabling informed comparisons for the platform.

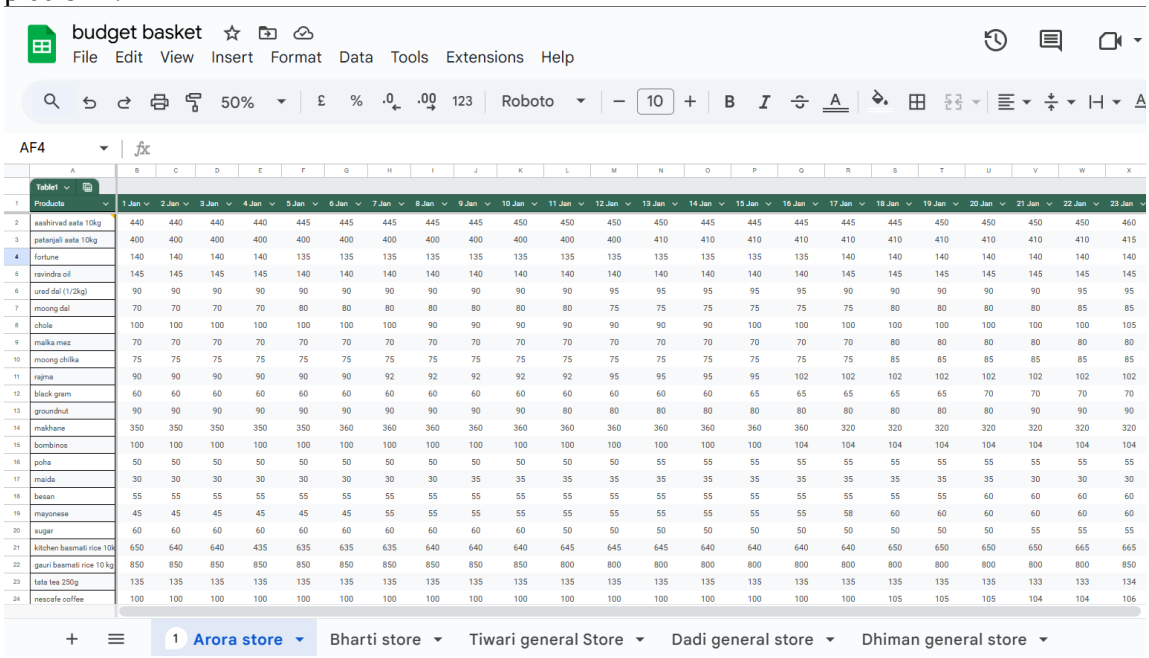

Figure2: Dataset

2.3 SWOT Analysis

Strengths:

- Provides real-time grocery price comparison across multiple stores.

- Saves time and money for consumers by identifying the best deals.
- Scalable to include more product categories beyond groceries.

Weaknesses:

- Dependency on third-party APIs and website scraping, which may be limited or blocked.
- Data inconsistency due to differences in product naming and packaging.
- Requires frequent updates to maintain accurate price listings.

Opportunities:

- Expansion to other product categories like electronics, clothing, and home essentials.
- Partnership with e-commerce platforms and local retailers for exclusive discounts.
- Integration of AI-based recommendation systems to improve user engagement.

Threats:

- Competition from existing price comparison platforms and grocery store apps.
- Changes in store APIs or website structures could affect data retrieval.
- Privacy concerns related to user data collection and analytics.

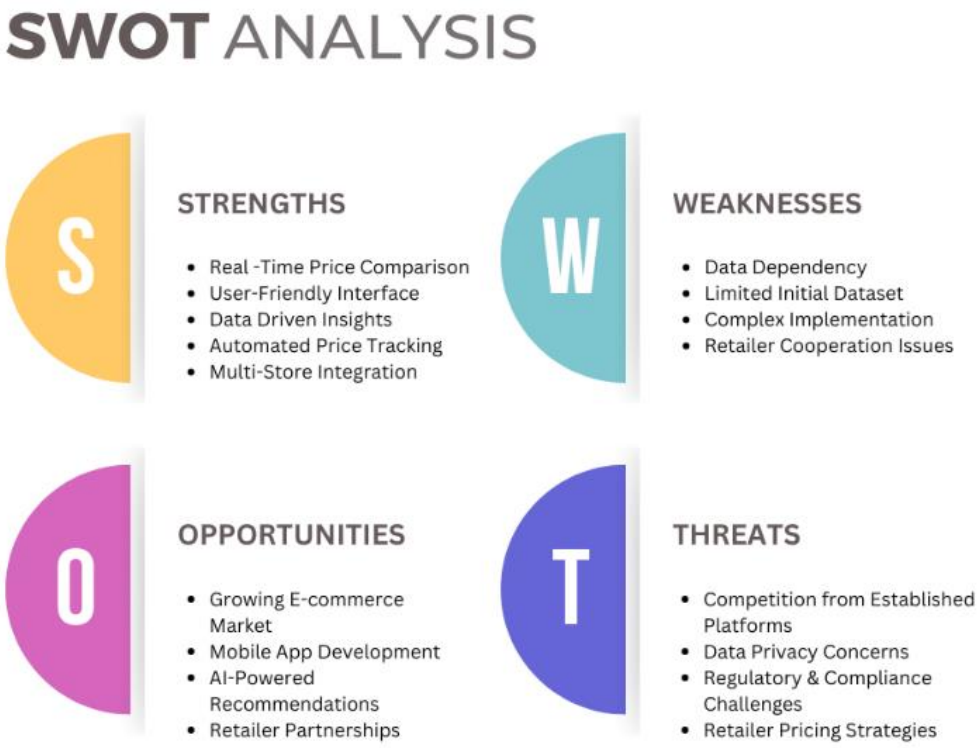

Figure 3: SWOT Analysis

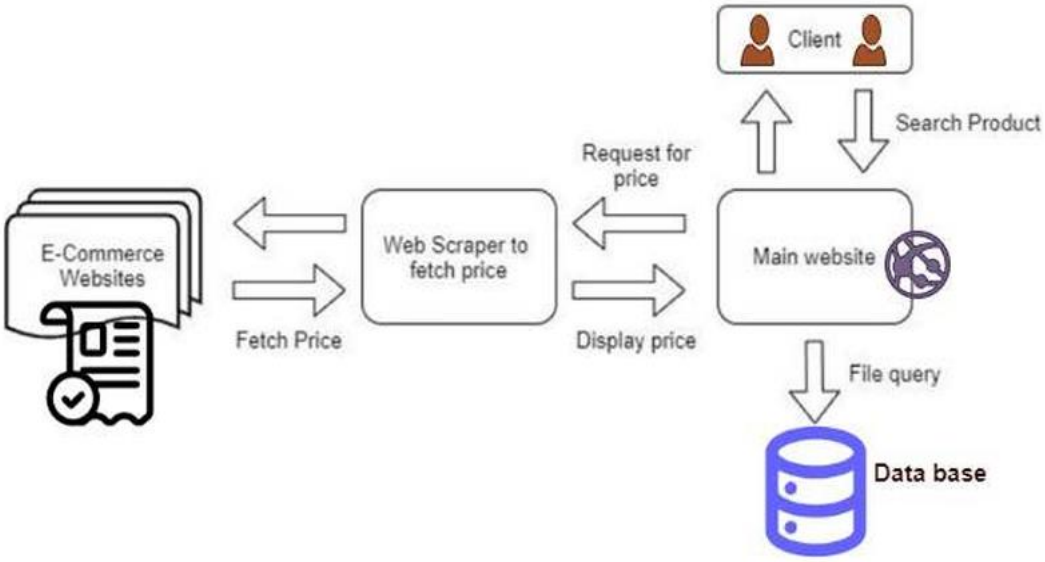

Figure 4: Framework/High-Level Architecture

2.4 Project Features

The **Budget Basket** platform will include the following key features:

- **Price Comparison:** Compare grocery prices across different stores.
- **Search and Filter Options:** Search by product name, category, and brand.
- **Product Availability Tracking:** Displays which stores have the item in stock.
- **Price Alerts & Notifications:** Alerts users when prices drop for specific products.
- **Personalized Recommendations:** Suggests products based on user preferences.
- **Historical Price Trends & Predictions:** Allows users to track price changes and forecast future pricing.
- **User Authentication:** Secure login via email or social media accounts.

2.5 User Classes and Characteristics

- **General Consumers:**
  - Primary users who compare grocery prices and make purchasing decisions.
  - Looking for budget-friendly deals and convenience in shopping.
- **Retailers & Store Owners:**
  - Can use the platform to monitor competitors' pricing and adjust their own strategies.
  - May offer exclusive discounts to attract more customers.
- **E-commerce Platforms & Suppliers:**
  - Potential integration partners that provide real-time product data.
  - Could collaborate to enhance the platform's reach and efficiency.
- **Admin Users:**
  - Responsible for managing store data, product listings, and price updates.
  - Monitors system performance and resolves any issues related to price inconsistencies.

2.6 Design and Implementation Constraints

Technical Constraints:

- **Real-time Data Retrieval:** Must handle API rate limits and website scraping challenges.
- **Data Storage & Scalability:** Database must be optimized for quick search and retrieval.
- **Algorithm Complexity:** Ensuring efficient execution of price comparison and prediction models.

Regulatory Constraints:

- **Compliance with Data Privacy Laws:** Must follow regulations like **GDPR** and **CCPA** for user data protection.
- **Retailer Agreements:** Need to obtain permissions for API integrations from grocery stores.

Performance Constraints:

- **Fast Response Time:** Users expect quick search results and real-time price updates.
- **Handling Large Data Volumes:** The system should scale to accommodate increasing users and products.

2.7 Design Diagrams

1. Pert Chart

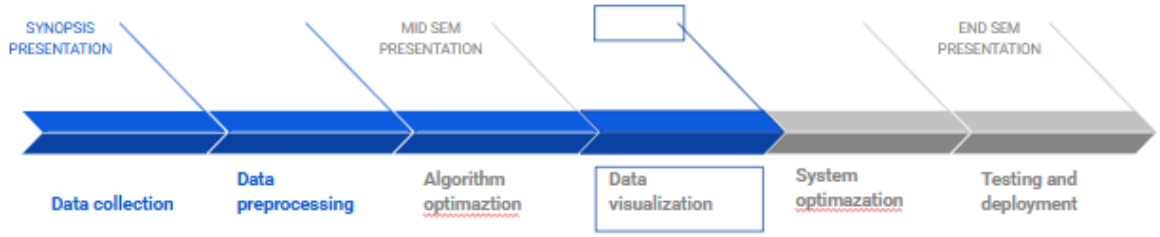

Figure 5. Pert Chart

2. Class Diagram

Figure 6: Class Diagram

3. Use-case diagram

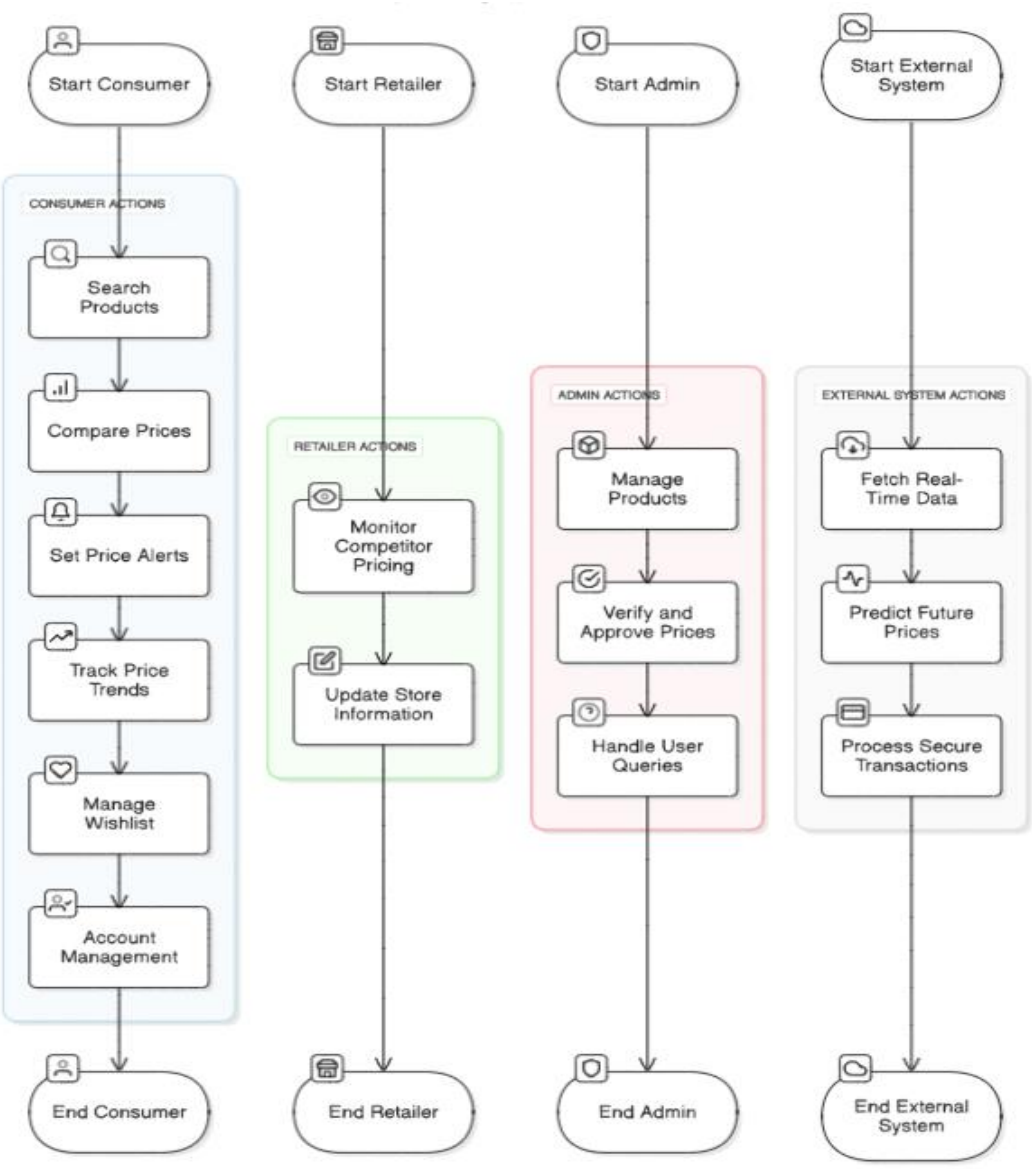

Figure 7: Use-case diagram

4. Data-flow Diagram

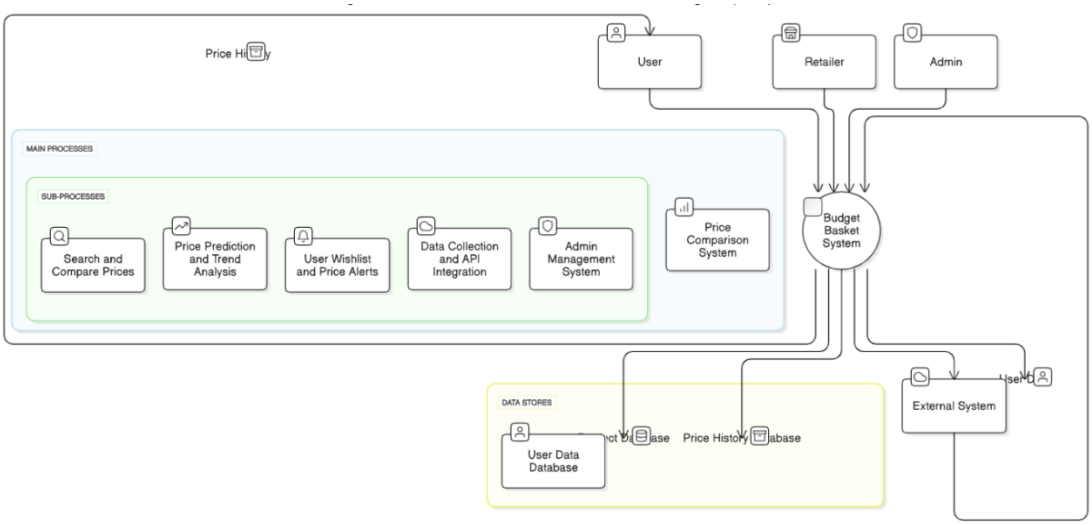

Figure 8: Data Flow Diagram

2.8 Assumptions and Dependencies

Assumptions:

- Grocery stores will allow API access or scraping for real-time price updates.
- Users will actively engage with the platform for price comparisons and alerts.
- Product information across different stores can be matched based on unique identifiers (e.g., barcode, SKU).

Dependencies:

- **Third-Party APIs & Data Sources:** The platform relies on store APIs or web scraping to retrieve product details and pricing.
- **Cloud Infrastructure:** Requires a robust hosting solution for database management and algorithm processing.
- **Machine Learning Models:** Price prediction accuracy depends on historical data availability and quality.
- **User Engagement:** Success depends on acquiring and retaining a large user base for data-driven insights.

3. System Requirements

The Budget Basket platform requires a robust and scalable system architecture to ensure smooth functionality, seamless user experience, and secure transactions. This section outlines the key system requirements, including the user interface, software interfaces, database interactions, and communication protocols.

3.1 User Interface

The user interface (UI) plays a crucial role in enhancing user experience, ensuring ease of navigation, and providing intuitive functionality.

Features of the User Interface:

- **Responsive Design:** The platform should be compatible with various devices (desktop, tablet, and mobile).

- **Easy Navigation:** Users should be able to search for grocery items, compare prices, and filter results by store, brand, and price range.
- **Interactive Dashboard:** A user-friendly dashboard should be available for users to manage their shopping lists, saved items, and price alerts.
- **Checkout Redirection:** Once users select a product, they should be redirected to the respective grocery store's website for checkout.
- **Personalized Recommendations:** The platform should suggest products based on user preferences and past purchase history.
- **User Authentication:** Secure login and registration options should be available using email, social media, or third-party authentication services.

3.2 Software Interface

- **Frontend Technologies:**
  - The UI will be developed using **HTML, CSS, JavaScript**, **Bootstrap, and wordpress** to ensure a responsive and visually appealing design.
  - JavaScript frameworks like **React.js or node.js** may be used to enhance performance and maintainability.
- **Backend Technologies:**
  - The back end will use **Node.js with Express** or **Python with Django/Flask** to manage user requests and process data.
  - RESTful APIs will be created to fetch product data from multiple sources in real time.
- **Third-Party Integrations:**
  - Integration with e-commerce platforms and grocery stores' APIs to retrieve updated price information.
  - Secure payment gateway support (if needed in the future).
  - Support for third-party login authentication (Google, Facebook, etc.).
- **Admin Panel:**
  - An admin dashboard will be developed to manage store data, monitor system performance, and address user concerns.
  - The panel should also allow manual price updates for unavailable API data cases.

3.3 Database Interface

The database is a critical component of the Budget Basket platform as it stores and manages product information, user data, and transaction history.

Database Requirements:

- **Relational Database:**
  - **MySQL or Excel** will be used for structured data storage, including product details, pricing, store information, and user profiles.
  - Tables should be designed efficiently with indexing to optimize query performance.
- **Data Update Mechanism:**
  - APIs will be scheduled to **fetch and update prices at regular intervals** to maintain accurate price comparisons.
  - The system will include an **error-handling mechanism** in case of failed API requests or missing product data.
- **Scalability & Security:**
  - The database structure should support an increasing number of products and users without performance degradation.

- **Data encryption** should be implemented to protect user credentials and payment-related information.

3.4 Protocols

Protocols define the rules for secure communication and data exchange between different components of the system.

Communication & Security Protocols:

- **HTTPS (Hypertext Transfer Protocol Secure):**
  - Ensures encrypted communication between users and the platform to prevent unauthorized access.
- **RESTful APIs:**
  - APIs will be designed using RESTful architecture to standardize data exchange between the front end, back end, and external data sources.
- **User Authentication:**
  - Used for secure user authentication and authorization when integrating third-party login services.
- **SSL/TLS Encryption:**
  - Implements **Secure Sockets Layer (SSL)** or **Transport Layer Security (TLS)** encryption for secure data transmission.

4.Non-Functional Requirements

4.1 Performance requirements

The system must respond to user requests within 3 seconds at normal load and within 5 seconds at cutting edge.
API calls to get price updates should average over 5 seconds. The platform should support at least 10,000 parallel users without slowing down performance.
The system must update the API product price every 15 minutes to maintain accuracy. The mobile version must be loaded within 2 seconds on a 4G network.
Database queries should be run within 100ms for optimal performance.
The platform should handle at least 1 million product entries efficiently

4.2 Security requirements

Authentication and authorization: Users must safely enter the system using Oauth 2.0, E-mail / password or authentication in social networks.
Data encryption: All user data, including passwords, must be encrypted using AES-256 encryption. Secure Transactions: If payment integration is added, you must use SSL/TLS encryption.
Access Control: Various user roles (admins, users, API consumers) must have access restricted based on privileges. Correspondence of data privacy: the platform must comply with GDPR, CCPA or any relevant data protection rules.
API safety: The final points of the API should be protected from injections of SQL, CSRF and XSS. Regular Safety Audit: The system must undergo quarterly security tests to identify vulnerabilities.
Booking and emergency recovery: For emergency recovery, you need to make daily backups and make sure you keep them.

4.3 Software Quality Attributes

Attribute Description

The adaptability platform must work on several devices (desks, mobiles, tablets).

The availability system must be 99.9% available, ensuring continuous access. System modifications must ensure an accurate comparison of the price and product parts.

Flexibility architectures need to maintain simple integration of new stores and features. Interoperability The system must be integrated transparently with several APIs from different stores.

Maintained the code base must follow the modular architecture to allow easy updates and fixes. Portable platforms must be located in the cloud and deployed to AWS, GCP, or Azure.

The reliability of the system must be carefully treated for failures and recover within 10 seconds of downtime. Commonly used components such as authentication, API handling, and database interactions must be reused.

System reliability must detect and handle incorrect responses, absent data, and user errors in the API. Testability systems require automated unit testing and integration testing for key features.

The UI must be intuitive, user-friendly, and accessible to all users.

5. Other Requirements

5.1 Performance Requirements

- Fast Response- Price comparisons in **2-3 seconds**, search in **1 second**.
- Scalability- Supports growing users, stores, and products efficiently.
- Optimized Data Handling- Efficient retrieval and background processing of large datasets.

5.2 Security Requirements

- Secure Login- MFA and encrypted user data.
- Data Protection- GDPR & CCPA compliance
- Anti-Web Scraping- Rate limiting

5.3 Compliance Requirements

- Institutional Agreements- Adhere to store API terms and affiliate programs.
- Regulatory Compliance- GDPR, CCPA, and fair pricing policies.

5.4 Usability & Accessibility

- User-Friendly UI- Responsive Wordpress design for all devices.
- Multi-Language Support- Option to switch languages.
- Accessibility- WCAG compliance, voice search.

5.5 Integration & Interoperability

- API Integrations- Real-time price updates from store APIs.
- Cross-Platform Compatibility- Works on web & mobile (Android/iOS).
- Social Media Sharing- Users can share deals and view product reviews.

5.6 Backup & Disaster Recovery

- Daily Automated Backups- Secure cloud storage with version control.

Appendix A: Glossary

This section defines key terms, acronyms, and abbreviations used in the **Software Requirements Specification (SRS)** for **Budget Basket** to ensure clarity and proper interpretation.

Term	Definition
SRS	Software Requirements Specification – A document outlining the system's functional and non-functional requirements.
API	Application Programming Interface – Allows the system to interact with external services, such as grocery store price data.
GDPR	General Data Protection Regulation – European data privacy law ensuring user data protection.
CCPA	California Consumer Privacy Act – U.S. data privacy law similar to GDPR.
MFA	Multi-Factor Authentication – A security mechanism requiring multiple authentication steps.
UI/UX	User Interface / User Experience – The design and usability aspects of the platform.
Price Comparison Algorithm	A system that collects, analyzes, and compares product prices from multiple stores.
Data Scraping	Extracting product information from e-commerce websites or APIs.
Cloud Backup	Storing system data in a secure online location for disaster recovery.

Table 1: Glossary

Appendix B: Analysis Model

This section includes relevant models that help in understanding the system's structure and behavior.

1. Use Case Diagram

- Represents how users interact with the **Budget Basket** platform, including searching for products, comparing prices, and viewing deals.

2. Data Flow Diagram (DFD)

- Shows how data moves through the system, from data collection and preprocessing to price comparison and visualization.

3. SWOT Analysis

- The **SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis** helps in evaluating the potential of the **Budget Basket** platform in the market..

4. System Architecture Diagram

- Illustrates the components of the platform, including user interface, backend, database, and third-party APIs.

5. Class Diagram:

- It represents the key entities, their attributes, and relationships in the system.

6. Pert chart:

- A **PERT (Program Evaluation and Review Technique) chart** helps visualize the sequence of tasks, dependencies, and estimated completion times for the **Budget Basket** project.

Appendix C: Issues Lists

- Finalizing the price comparison algorithm
- Ensuring compliance with GDPR & CCPA
- Selecting the most effective machine learning model for price prediction
- API integration with local grocery stores

- Implementing accessibility features
- Optimizing system performance for large-scale data