Software Requirements Specification

for

SuperPrice

Version 2.0 approved

Prepared by Darby, Laura, Lance, Lawrence, Saksham, Krisanahari

Group-P8-07

17/09/2023

Contents

1. Introduction

1.1 Product Scope

The product that will be covered in this document is SuperPrice; an application that is intended to enable an enhanced shopping experience for grocery shoppers. This document will cover the application's functional and non-functional requirements, the system architecture, the data model, the user interface design, assumptions and constraints, dependencies, as well as testing and acceptance criteria.

1.2 Purpose and Intended Audience

SuperPrice is intended to allow an easier and more efficient shopping experience for supermarket customers. Using this application, customers can find the best price option across multiple supermarkets and stores near them for the products that they intend to purchase. There is also an option for customers to have their items delivered to their doorstep. The project aims to develop a user-friendly platform to make it available for a wide range of users. This is done by prioritising a smooth and easy experience for everyone through an intuitive and straightforward user-interface. Furthermore, this document is drafted for the product's stakeholders; specifically - the software developers, the project manager, and the client.

2. Overall Description

2.1 Product Perspective

The SuperPrice - Price Matching and Delivery Application represents an innovative, standalone solution tailored to elevate the consumer shopping journey. Its primary objective is to refine the grocery shopping paradigm by offering a sophisticated platform where consumers can effortlessly compare prices from a myriad of retailers and coordinate deliveries. This application seamlessly integrates with a diverse range of local supermarkets and retail outlets, granting users access to a comprehensive repository of products and their respective prices. Such integration guarantees that consumers are furnished with precise and contemporaneous data regarding product prices and their availability. Architecturally, the SuperPrice application is web-centric, bifurcated into backend and frontend modules, and is designed to function in real-time, ensuring users receive the most recent data.

Central to the application's ethos is its commitment to economizing both time and financial resources for its users through an all-encompassing price-matching mechanism and streamlined delivery coordination. With an intuitive user interface at its core, SuperPrice is poised to serve a diverse demographic, from shopping novices to seasoned consumers. In addition, it offers a robust platform for users to impart their feedback via reviews and ratings for both supermarkets and individual products, thereby augmenting its credibility and user trust.

2.2 Product Functions:

- Product Search and Categorization: Allows users to find specific products or browse through different categories.
- Price Comparison: Enables users to compare prices across different supermarkets.
- Delivery Organization: Facilitates the organization of deliveries with multiple options.
- Notifications and Alerts: Provides timely notifications about price drops or special offers.
- User-Friendly Interface: Prioritizes simplicity and ease of use.
- User Reviews and Ratings: Allows users to leave reviews and ratings for supermarkets and products.

2.3 User Classes and Characteristics:

- Novice Shoppers: Those new to online shopping who prioritize a user-friendly interface.
- Experienced Shoppers: Users familiar with online shopping platforms and looking for advanced features.
- Bargain Hunters: Users specifically looking for the best deals and price drops.
- Reviewers: Users who rely on or provide reviews and ratings for products and supermarkets.

2.4 Operating Environment

- Web-based application with backend and frontend components.
- Specifications:
  - **Code Repository:** Github
  - **Task Management:** Github Project
  - **Group Communication:** Microsoft Teams
  - **Documentation:** Office 365 online
  - **CI/CD tool:** Github Action
  - **Java version:** 17
  - **Back-end framework:** Spring Boot
  - **Front-end framework:** Node.js
  - **Build tool:** Maven
  - **Data-base:** H2-console with Flyway Data migration
  - **Unit testing framework:** Junit4 for Java, Front-end manual testing framework .
  - Docker for configuration management

2.5 User Documentation

- Comprehensive user guide outlining the features and functionality of the SuperPrice application.
- Testing and quality assurance reports.
- Presentation and demonstration materials for stakeholders.

Pending Development…

2.6 Assumptions and Dependencies

2.6.1 Assumptions:

- The application assumes that the majority of users have access to a stable internet connection.
- The application assumes that all integrated supermarkets and stores provide accurate and timely data.
- It is assumed that users will primarily use the application for grocery shopping, rather than for other types of products.
- The application assumes that users will trust and rely on the reviews and ratings provided by other users.
- The application assumes that the majority of users will be using modern browsers and devices that support the latest web standards.

2.6.2 Dependencies:

- The application's functionality is dependent on the real-time data integration with various supermarkets and stores.
- Dependency on third-party payment gateways for processing transactions.
- The application might rely on third-party APIs for additional features like maps, notifications, etc.
- The success of the application is contingent on regular updates and maintenance to ensure accurate pricing and product availability.
- The application's performance might be influenced by the server's uptime and the efficiency of the hosting service.

----

3. User Stories

3.1 Product Search and Categorization – Saksham & Krisanahari

User Story 1: As a user, I want to be able to search for a specific grocery item.

Acceptance Criteria:

Scenario 1: Successfully search for a specific grocery item

Given: I am on the homepage of the online grocery website.

When: I enter the name of a grocery item into the search bar,

Then: The application should display a list of results that match the item name I entered. The results should include a photo of the product, the product name, and its price.

User Story 2: As a user, I want to be able to browse products by category.

Acceptance Criteria:

Scenario 1: Successfully browse products by category

Given: I am on the homepage of the online grocery website.

When: I click on a specific category, like 'Fruits' or 'Vegetables',

Then: The application should display a list of products that fall under the selected category. Each product listing should include a photo of the product, the product name, and its price.

User Story 3: As a user, I want to be able to checkout.

Acceptance Criteria:

Scenario 1: Successfully checkout

Given: I am on the cart of the online grocery website.

When: I click on checkout,

Then: The application should display the payment and delivery information screen.

(Laura)

User Story 4: As a user, I want to be able to view my liked items/ products.

Acceptance Criteria:

Scenario 1: I have successfully viewed my liked items/ products.

Given: I am on any page with view of the navigation bar,

When: I click on the heart icon in the top right corner of the navigation bar,

Then: I am taken to a page which displays for me all my liked items/ products, so next time I go to make a purchase, I can easily view my liked items and add them straight to my cart.

3.2 Delivery Organization - Lance

User Story 1: As a user, I want to be able to select the date and time of delivery of my items.

Acceptance Criteria:

Scenario 1: Successfully select the date and time of delivery

Given: I am on the delivery setup page.

When: The application displays the calendar,

Then: I can select a date that I prefer for my delivery

Then: It will display available timeslots on that day which I can choose from. After selecting these, a summary of the details will be displayed. That is, the date and time of the delivery.

User Story 2: As a user, I want to be able to select the delivery option that I prefer.

Acceptance Criteria:

Scenario 1: Successfully choose the preferred delivery option

Given: I am on the delivery setup page.

When: the application displays the available delivery options,

Then: it will proceed accordingly to ask for further details about my chosen method of delivery.

3.3 User Sign In Request

User Story 1

As a user, I want to sign-in to my account.

Acceptance Criteria:

Given: I am on the login page.

When: I enter my username and password,

Then: the application will verify my details and sign me in if it is correct.

3.4 Price Comparison - Lawrence

Agile Epic: Price Comparison

User Story 1:

As a user, I want to compare prices across different supermarkets, so I can make informed decisions and find the best deals.

Acceptance Criteria:

Given that I am on the product details page,

When I select a product,

Then I should see a list of supermarkets in the local area offering that product and their corresponding prices.

And the list of supermarkets should be sorted in ascending order based on the product prices, with the lowest price displayed first.

User Story 2:

As a user, I want to compare prices of products in different categories across multiple supermarkets, so I can make informed purchasing decisions for my groceries.

Acceptance Criteria:

Given I am a registered user of the SuperPrice application.

When I browse through different product categories.

Then I should be able to see a list of products in each category.

And the list should display the prices of the products from various supermarkets.

And the products in each category should be sorted based on the lowest price available.

User Story 3.

As a user, I want to view price trends over time for a particular product, so I can make informed purchasing decisions and take advantage of cost-saving opportunities.

Acceptance Criteria:

Given The Super Price application is installed and accessible to the user.

When I search for a particular product using the application.

And I select the desired item from the search results.

Then the application should display the price trends over time for the selected product.

3.5 Database Implementation - Lawrence

Agile Epic: Design and Apply Database Schema for SuperPrice

User Story 1.

As a developer, I want to design and apply a database schema for SuperPrice so that the application can store, manage, and retrieve data efficiently.

Acceptance Criteria:

Given I have a clear understanding of the data requirements for SuperPrice,

When I design the database schema,

Then it should effectively represent all entities, relationships, and constraints necessary for the Super Price application.

Agile Epic: Flyway Used To Manage Database Migrations for H2 database

User Story 1.

As a database administrator, I want to use Flyway to manage database migrations so that I can ensure consistent and version-controlled changes to our H2 databases.

Acceptance Criteria:

Given I have an H2 database and a Flyway migration script,

When I execute Flyway commands,

Then the migration should be correctly applied to the H2 database

And I should be able to view the status of migrations using the Flyway info command.

Agile Epic: Implement H2-Console for Database Environment

<u>User Story 1.</u>

As a developer, I want to implement the H2-Console in the database environment so that I can easily interact with and monitor the H2 database via a web interface.

Acceptance Criteria:

Given I navigate to the H2-Console URL,

When I enter valid database credentials and attempt to login,

Then I should be able to access the H2 database and execute SQL statements.

Given I navigate to the H2-Console URL,

When I enter incorrect database credentials and attempt to login,

Then I should see an error message indicating a failed connection attempt.

Agile Epic: Input Sample Data into Database

<u>User Story 1.</u>

As a customer, I want to select products and place an order, So that I can purchase the items I need.
Acceptance Criteria:

Given a list of available products and my customer information,

When I select products with IDs 101 and 202, and place an order,

Then the database should record the order with the selected products, associate it with my customer information, and update the stock quantities of the purchased products.

3.6 Continuous Integration (CI/CD) - Lawrence

Agile Epic: Implement Continuous Integration

<u>User Story 1.</u>

As a developer, I want to have Continuous Integration (CI) implemented for our project so that my code is automatically tested and built whenever I push changes, ensuring that it integrates seamlessly with the existing codebase.

Acceptance Criteria:

Given that I have made code changes and pushed them to our version control repository,

When the changes are pushed,

Then the CI system should automatically trigger a build and test sequence for the latest changes.

3.7 Notifications and Alerts – Laura

User Story 1:

As a user, I would like to be notified of price drops and special deals on products.

Acceptance Criteria:

Scenario 1: Successfully notified of real-time price drops and special deals.

Given: I have agreed to receiving notification.

When: I agree to receiving notifications,

Then: The application must notify me anytime an item has dropped in price or when there are new special deals. The notification must include the item, its original price, and the price it has dropped to/ the new deal. It must also include (if applicable) the expiration date of these offers and deals.

User Story 2:

As a user, I want to unsubscribe to the notifications of price drops and special deals.

Acceptance Criteria:

Scenario 1: Successfully unsubscribed to the notifications and alerts.

Given: I have previously received notifications or am a new user to the application.

When: I have clicked the option ‘Account Notifications’ from the ‘Account’ drop down box on the navigation bar,

Then: I am prompted to either enable or disable the notifications feature. Upon clicking the ‘disable’ button, I will successfully unsubscribe from receiving notifications. I can also enable the notifications function again at any time if I choose to accept them again.

Scenario 2: Successfully unsubscribed to the notifications and alerts.

Given: I have previously received notifications or am a new user to the application.

When: I have clicked the bell icon from the navigation bar,

Then: I will have successfully disabled notifications. If I click it again, I will have enabled the notifications again.

3.8 User-Friendly Interface – Darby

“SuperPrice will boast a user-friendly and intuitive interface, making it accessible to a wide range of users. The application will prioritize simplicity and ease of use, ensuring a seamless experience for both novice and experienced shoppers.”

User Story 1: As an inexperienced user I want to be able to use SuperPrice without any training or prior experience.

Acceptance Criteria -
Scenario 1: Successfully navigate the website for the first time with no experience
Given: I have no prior experience with the website, or similar websites
When: I interact with the website, to execute core functions
Then: The interface must be simple and clean enough to not overwhelm a new user, or someone who has not used a competitor's website before. Subheadings must be clearly sectioned and provide a concise description of the content that it holds.
User Story 2: As a user living with vision impairment, I wish to engage with the product in as close of a way as possible to someone without a disability.

Acceptance Criteria-
Scenario 1: successfully navigate the website and complete all tasks and functions required.
Given: I have a vision impairment
When: I interact with the website
Then: The system must have clear alternate text for images, and clearly labelled and positioned heads and sub heads for users with screen readers so they can navigate the website and interact with/digest content the same as any other user.

4. System Architecture

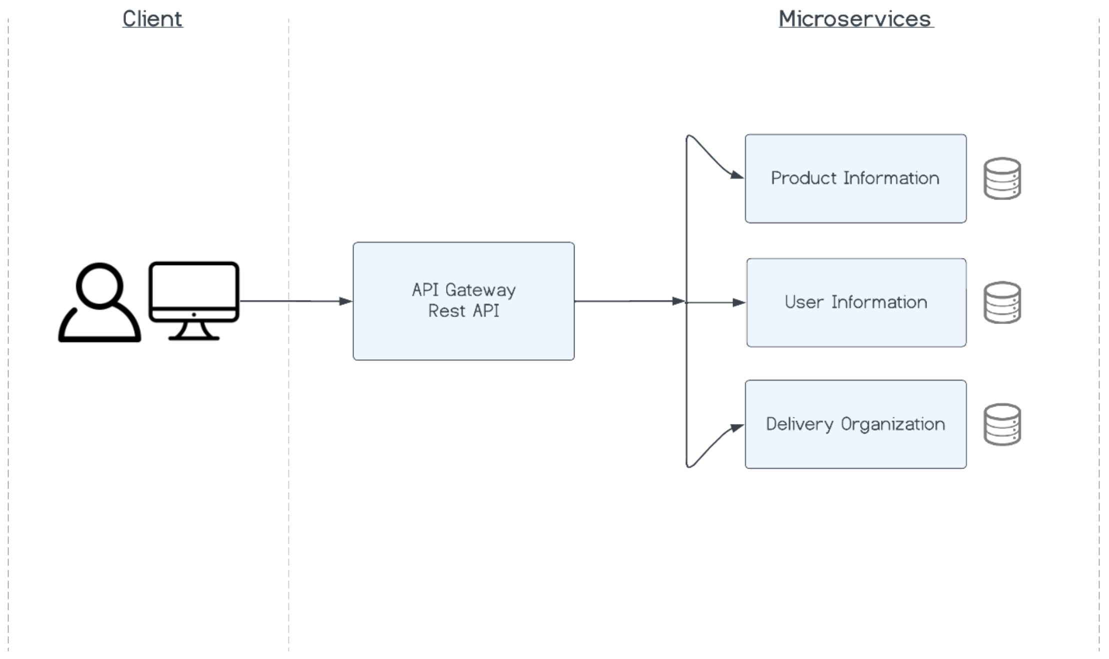
The system components include the Client, API Gateway (Rest API) and the services of the application.

The API Gateway being followed is REST API. This component serves as the gateway between the client and the server when the application's resources are required or requested.

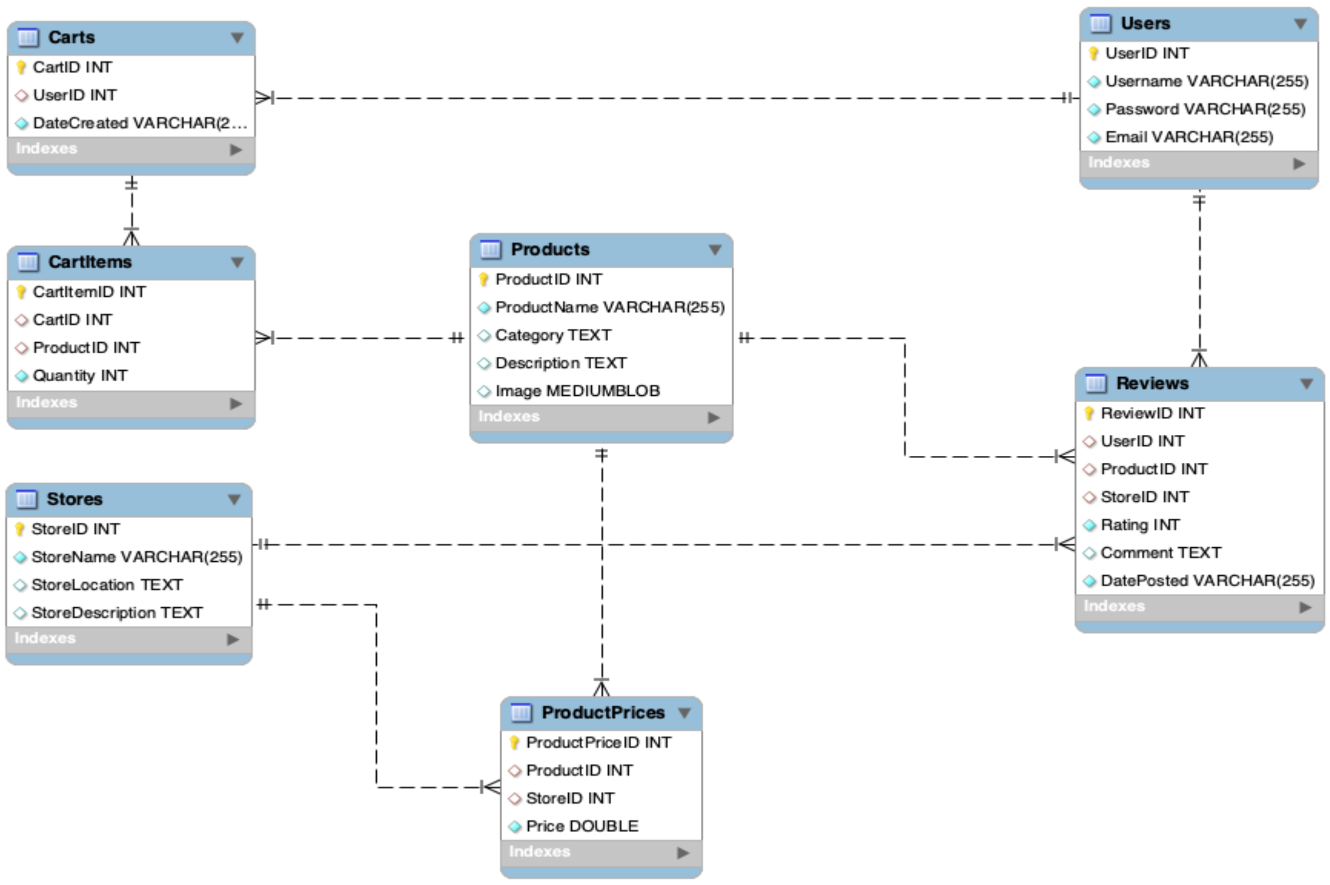
Next, are the services of the application, namely, Product Information, User Information, and Delivery Organization.

- The Product Information component involves any product-related functionality within the application. For example, the product search and the product prices.
- The User Information component is composed of user-related, such as their name, purchases, order delivery details, and notification preferences.
- The Delivery Organization component involves the application's delivery feature.

The interaction of these components starts with the client-side of the system. A request is sent to the server by the client which the API Gateway handles. As the request is made, the server validates this and proceeds to communicate with the system's services/resources and database. Finally, the system captures the result that will be conveyed to the client as a response to the request.



5. Database ERM Model



6. Other Nonfunctional Requirements

6.1 Performance Requirements

***Response Time**: The application should display search results within 2 seconds of the user initiating a search.*

***Load Time:** The web pages of the application should load within 3 seconds under normal network conditions.*

***Scalability:** The system should support up to 100,000 concurrent users without any degradation in performance.*

***Real-time Data:** Price updates and availability checks should be reflected in real-time to the end-users.*

6.2 Safety Requirements

Availability: *The system should achieve a minimum uptime of 99.9%.*

Backup: *All data should be backed up daily, and there should be a mechanism in place for quick data restoration in case of any failure.*

Disaster Recovery: *A disaster recovery plan should be in place to ensure the application's availability in case of any catastrophic events.*

Error Handling: *The application should provide clear error messages and should be able to handle unexpected failures gracefully, ensuring that users do not experience abrupt crashes.*

6.3 Security Requirements

Data Encryption: *All data transferred between the client and the server should be encrypted using industry-standard encryption protocols.*

User Authentication: *The application should implement multi-factor authentication for user accounts.*

Data Privacy: *User personal and financial information should be stored in encrypted formats and in compliance with GDPR or other relevant data protection regulations.*

Session Management: *User sessions should automatically expire after 15 minutes of inactivity.*

Audit Logs: *All user activities and system transactions should be logged and stored for at least 6 months.*

6.4 Software Quality Attributes

User Interface: *The UI should be intuitive and user-friendly, with a focus on ease of navigation.*

Mobile Responsiveness: *The application should be fully functional and responsive on various devices including desktops, tablets, and mobile phones.*

Accessibility: *The application should be accessible for users with disabilities, adhering to the Web Content Accessibility Guidelines (WCAG) 2.1 Level AA.*

Multilingual Support: *The application should support at least English and one other local language, with capabilities to easily add more languages in the future.*

API Integration: *The system should support seamless integration with various supermarket and store databases through well-defined APIs.*

Notifications: *The notification system should be capable of sending instant notifications to users without any delay.*

6.5 Business Rules

Automated Testing: *The system should support automated unit and integration tests to ensure the robustness of the application.*

Code Quality: *The source code should adhere to industry-standard coding practices and should be well-commented for easier maintainability.*

Updates & Patches: *The system should support seamless updates and patches without causing significant downtime.*

7. Use Cases

Use Case 1	Search for an item/product	
Actors	User/Customer	
Preconditions	The user must be on the homepage.	
Postconditions	A list of items that match the name of the product entered and the corresponding price.	
Flow of activities	**Actor**	**System**
	1. The user selects the search bar 2. The user enters the product that they want to purchase.	1. The search bar allows the user to enter a text. *2.* The application redirects the user to another page to show the result of their search.
Alternative Flows/ Exceptions	1. The product entered cannot be found (I.e., it is not in the system) - The application will then display a prompt to the user that this is the case.	

Use Case 2	Select the date and time of delivery	
Actors	User/Customer	
Preconditions	The user must be on the delivery setup page.	
Postconditions	The available timeslots on that selected day will be displayed and a summary of the delivery's details will be shown.	
Flow of activities	**Actor**	**System**
	1. The user first selects the date of delivery. 2. The user selects their preferred timeslot for their delivery. 3. The user selects proceed.	1. The application gets the available timeslots for the selected day and displays this. *2.* --- 3. The application displays a summary of the delivery details (date and time).

Alternative Flows/ Exceptions	1. The selected date does not have the user's preferred timeslot - The user selects the “back” button to go back to the calendar and choose another preferred date. - Then proceeds to number 2 of flow of activities.

Use Case 3	Compare prices across different supermarkets	
Actors	User/Customer	
Preconditions	The user must be on the product details page.	
Postconditions	A list of supermarkets in the local area offering the selected product will be displayed.	
Flow of activities	**Actor**	**System**
	1. The user selects a product.	*1.* The application fetches all the supermarkets that sell the product and displays a list.
Alternative Flows/ Exceptions	1. The product selected cannot be found in the local area. - The application will then display a prompt to the user that this is the case.	

Use Case 4	Allow notifications for price drops and special deals on products.	
Actors	User/Customer	
Preconditions	The user must be on the homepage.	
Postconditions	The user’s notification preference is allowed/enabled.	
Flow of activities	**Actor**	**System**
	1. The user selects “settings”. *2.* The user chooses “allow notifications”.	1. The application redirects the user to the settings page. 2. The application updates the user’s preference and displays a prompt that this has been successfully done so.
Alternative Flows/ Exceptions	2. The notification setting has already been allowed for the user.	

Use Case 5	Browse products by category	
Actors	User/Customer	
Preconditions	The user must be on the homepage.	
Postconditions	A list of products that fall under the selected category will be displayed. Each product will include its corresponding price.	
Flow of activities	**Actor**	**System**
	1. The user selects a specific category (for example, ‘Fruits and Vegetables”)	*1.* The application redirects the user to another page to show the result of their selection.
Alternative Flows/ Exceptions	1. The user cannot find the product that they want in their selected category. - Refer to use case 1 instead.	

Use Case 6	Design and Apply Database Schema for SuperPrice	
Actors	Developer	
Preconditions	Developer has access to the system where the application SuperPrice is being developed. Developer understands the requirements and data needs of the SuperPrice application.	
Postconditions	A database schema has been designed and applied. The application SuperPrice can efficiently store, manage, and retrieve data based on the schema.	
Flow of activities	**Actor**	**System**
	1. Analyzes the data requirements for SuperPrice. 3. Designs the database schema based on the analyzed requirements. 5. Applies the schema to the SuperPrice system.	2. Provides necessary data and interfaces for analysis. 4. Accepts the schema design. 6. Updates the database with the new schema. Confirms successful schema application.
Alternative Flows/ Exceptions	▪ The designed schema is found to be inefficient or not meeting the requirements. o The developer revises the schema and reapplies it to the system. ▪ Errors occur while applying the schema. o The developer debugs and corrects the schema before reapplying.	

Use Case 7	Implement Continuous Integration for the Project	
Actors	Developer	
Preconditions	Developer has access to the codebase of the project. The project has a version control system in place. Developer has knowledge of Continuous Integration (CI) tools and practices.	
Postconditions	Continuous Integration (CI) is set up and operational for the project. Code is automatically tested and built whenever changes are pushed, ensuring seamless integration.	
Flow of activities	**Actor**	**System**
	1. Sets up a CI server or uses a CI service. 3. Configures the CI server or service to watch the project's repository for changes. 5. Pushes changes to the codebase.	2. Registers and configures the CI setup. 4. Observes the repository for changes. 7. Triggers the CI process. Automatically tests and builds the project. - Generates and provides CI reports.
Alternative Flows/ Exceptions	▪ The CI process detects errors or failures in the code. o The developer reviews the error reports, fixes the issues, and pushes the corrected code. ▪ The CI process experiences issues or becomes non-operational.	

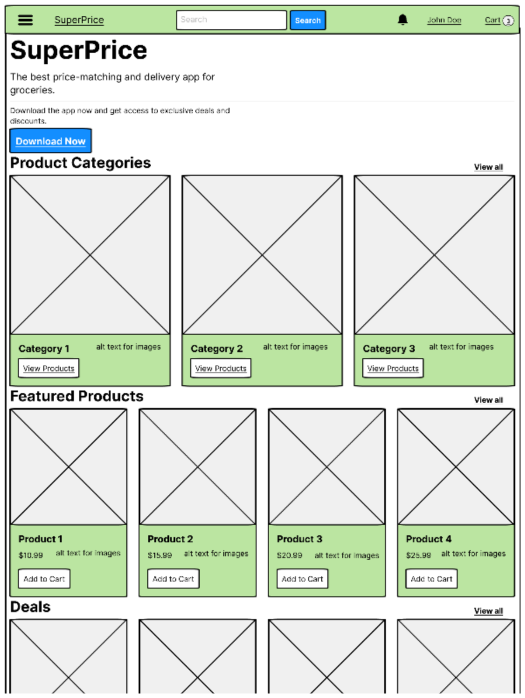
	○ The developer troubleshoots the CI setup and makes necessary adjustments.

Use Case 8	Sign in	
Actors	User/Customer	
Preconditions	The user must be on the login page	
Postconditions	The user signs into their profile.	
Flow of activities	**Actor**	**System**
	1. The user first enters their username and password. 2. The user selects login.	1. The application handles the form changes. *2.* The application handles the username and password input and verifies if the user is in the database. *3.* The application finds the user in the database and proceeds to log them in.
Alternative Flows/ Exceptions	*1.* The user's credentials cannot be found in the database (he/she is not registered). • Proceed to sign up/register page.	

8. User Interface Design:

https://www.figma.com/file/iTvWwwcLNZOSkjWoyK4Ptc/SEPT?type=design&node-id=0%3A1&mode=design&t=LdbC65w11mw5y7L1-1

8.1 Home Page



SuperPrice

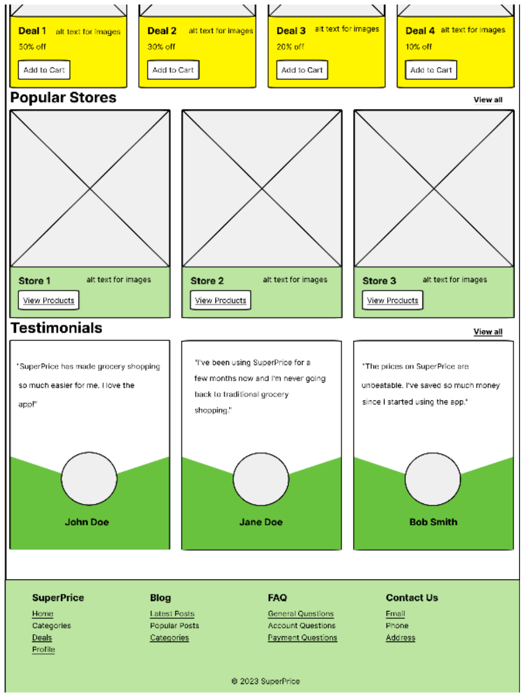
The best price-matching and delivery app for groceries.

Download the app now and get access to exclusive deals and discounts.

Product Categories

Featured Products

Deals



Popular Stores

Testimonials

"SuperPrice has made grocery shopping so much easier for me. I love the app!"

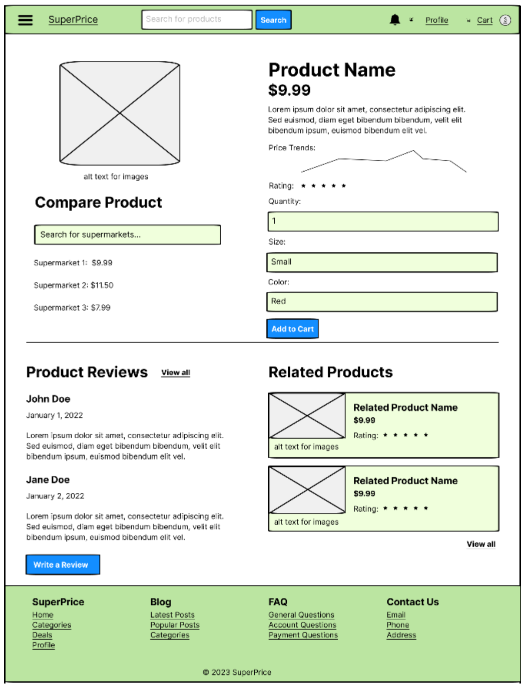
"I've been using SuperPrice for a few months now and I'm never going back to traditional grocery shopping."

"The prices on SuperPrice are unbeatable. I've saved so much money since I started using the app."

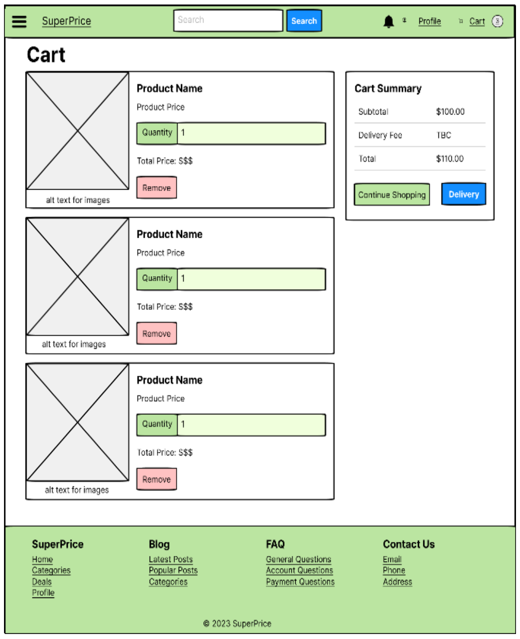
SuperPrice	Blog	FAQ	Contact Us
Home	Latest Posts	General Questions	Email
Categories	Popular Posts	Account Questions	Phone
Deals	Categories	Payment Questions	Address
Profile			

© 2023 SuperPrice

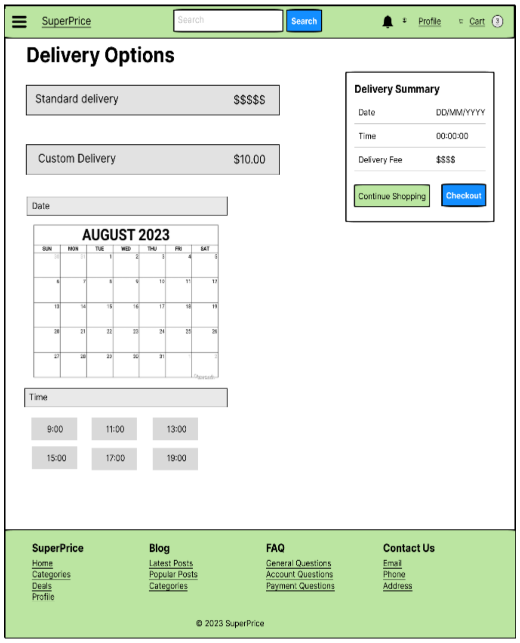
8.2 Product details page



8.3 Cart page



8.4 Delivery options page


SuperPrice | Search | Search | Profile | Cart

Delivery Options

Standard delivery $$$$$

Custom Delivery $10.00

Date

AUGUST 2023

Time

9:00 11:00 13:00

15:00 17:00 19:00

Delivery Summary

Date	DD/MM/YYYY
Time	00:00:00
Delivery Fee	$$$$

Continue Shopping | Checkout

SuperPrice: Home, Categories, Deals, Profile

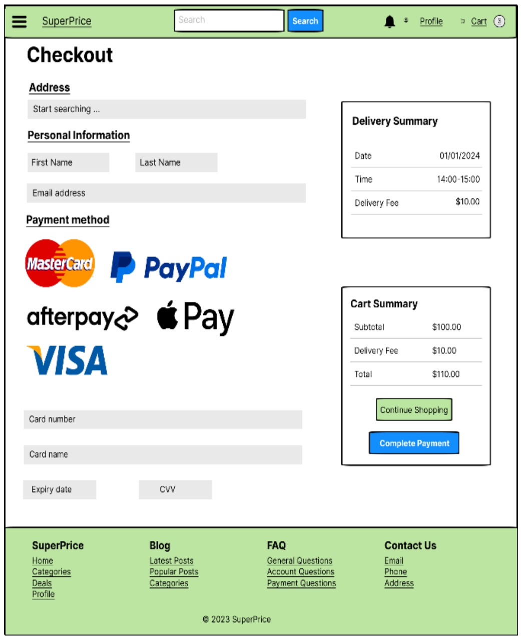
Blog: Latest Posts, Popular Posts, Categories

FAQ: General Questions, Account Questions, Payment Questions

Contact Us: Email, Phone, Address

© 2023 SuperPrice

8.5 Checkout page


SuperPrice | Search | Search | Profile | Cart

Checkout

Address

Start searching ...

Personal Information

First Name | Last Name

Email address

Payment method

MasterCard | PayPal | afterpay | Pay | VISA

Card number

Card name

Expiry date | CVV

Delivery Summary

Date	01/01/2024
Time	14:00-15:00
Delivery Fee	$10.00

Cart Summary

Subtotal	$100.00
Delivery Fee	$10.00
Total	$110.00

Continue Shopping

Complete Payment

SuperPrice: Home, Categories, Deals, Profile

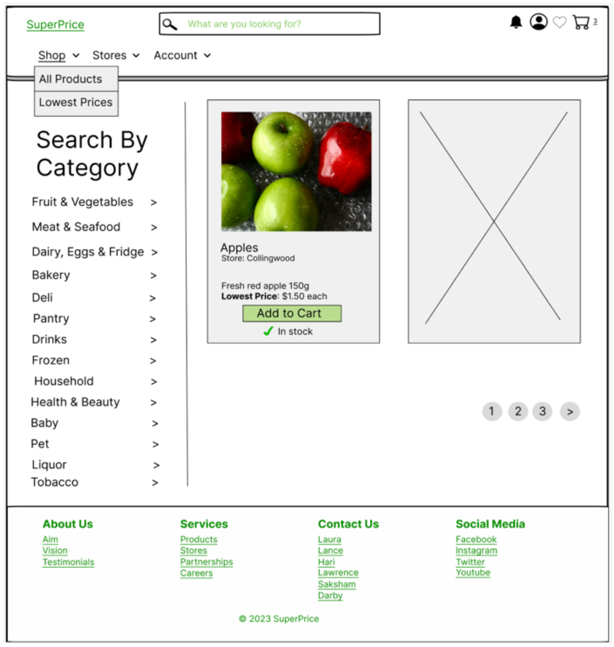
Blog: Latest Posts, Popular Posts, Categories

FAQ: General Questions, Account Questions, Payment Questions

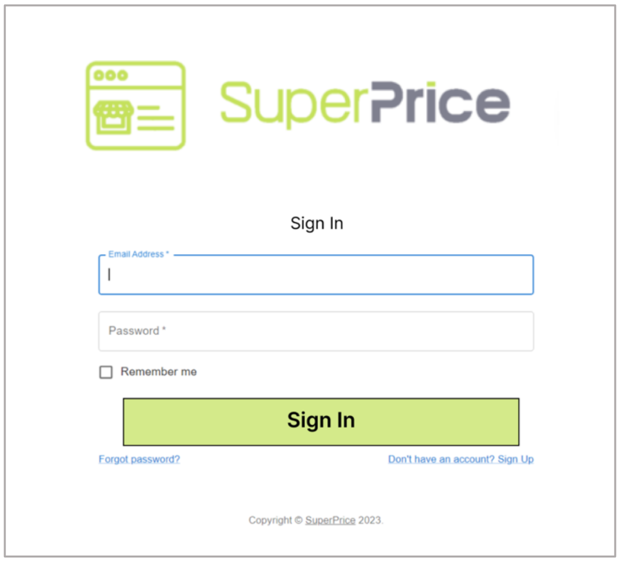
Contact Us: Email, Phone, Address

© 2023 SuperPrice

8.6 Product List



8.7 Sign in Page



9. Testing and Acceptance Criteria

Epic	User Story	Acceptance criteria	Testing
Product Search and Categorization	As a user, I want to be able to search for a specific grocery item	Successfully search for a specific grocery item	There is a search bar at the top of the page that will always be accessible no matter what page you are on. Upon clicking, the user is able to search for the item they want, and once they hit enter, a page will display their item or alternatives/ similar items if their desired item is unavailable.
Product Search and Categorization	As a user, I want to be able to browse products by category	Successfully browse products by category	On the home page, there is a section of the page that displays the most popular categories. The user is also able to select "view all" which will expand the page to show all categories. Alternatively, the user can also click the hamburger icon in the top left corner of the page which will also display a range of hyperlinks, including product categories. Upon selection, the user will be taken to a page where they can browse products under their desired category.
Product Search and Categorization	As a user, I want to be able to checkout	Successfully checked out	In the top corner of every page, there is a cart icon that users can click to view their cart. Once clicked and on the cart page,

			users can review their items and finalise their transaction by clicking on the “checkout” button which will take the user to another page where they can safely and successfully pay for their items.
Delivery Organization	As a user, I want to be able to select the date and time of delivery of my items	Successfully select the date and time of delivery	Once the user has clicked on the cart page/ once the user has reviewed their cart, they can continue to the delivery page. This will show options for delivery and also gives the user the ability to choose which date and time they would like the order to be delivered. The delivery summary will be updated and displayed to the user once they have clicked on their desired options, as well as any additional fees.
Delivery Organization	As a user, I want to be able to select the delivery option that I prefer	Successfully choose the preferred delivery option	Given the user is on the delivery page, they will be given the option to choose standard delivery, which would be the delivery option which delivers the users items to them as soon as possible. The other option would be the custom delivery option, where the user can select a particular

			date and time which may include a fee.
Price Comparison	As a user, I want to compare prices across different supermarkets, so I can make informed decisions and find the best deals	Successfully compare the prices across different supermarkets	When the user has selected on a product, the page will come up with product information. There will also be a section on the page that automatically shows different prices across different stores, but also allows users to search for a particular supermarket. Once searched, the page will update and show the price of the product at that store.
Price Comparison	As a user, I want to compare prices of products in different categories across multiple supermarkets, so I can make informed purchasing decisions for my groceries	Successfully compare prices of products in different categories across multiple supermarkets	The user can go onto the home page and find the categories section where they can select one of the featured categories, or click view all to see the rest. Once selected, the user can find products in that category and and compare prices of that product between supermarkets once they've clicked on a specific product.
Price Comparison	As a user, I want to view price trends over time for a particular product, so I can make informed purchasing decisions and take advantage of cost-saving opportunities	Successfully viewed price trends over time for a particular product	Users can click on a particular product, and upon viewing the product details page, there will be a section of the page which shows the trends that the product price has undergone recently to give the user the best idea of when to buy the product.

Database Implementation	As a developer, I want to design and apply a database schema for SuperPrice so that the application can store, manage, and retrieve data efficiently.	1. A comprehensive database schema is designed which covers all the necessary entities and relationships. 2. The schema is successfully applied to the database. 3. The application can efficiently store, manage, and retrieve data without errors.	1. Test the database for any normalization errors. 2. Run CRUD (Create, Read, Update, Delete) operations on the database to ensure data integrity. 3. Benchmark the database for performance during high load scenarios.
Implement Continuous Integration	As a developer, I want to have Continuous Integration (CI) implemented for our project so that my code is automatically tested and built whenever I push changes, ensuring that it integrates seamlessly with the existing codebase.	1. A CI pipeline is set up that automatically triggers on code pushes. 2. Automated tests run successfully on every code push without any manual intervention. 3. Developers receive feedback on the results of the build and test phases promptly.	1. Push a sample code change to trigger the CI pipeline. 2. Verify that the code is automatically built and tests are executed. 3. Ensure that the feedback mechanism (e.g., notifications) works as intended and informs the developer of the CI process outcomes. Is there anything else you'd like to add or modify?
Notifications and alerts	As a user, I would like to be notified of price drops and special deals on products	Successfully notified of real-time price drops and special deals	At the top of the page, there will be a bell icon which users can click. Upon clicking, the bell will have a little tick next to it to show that the user is now subscribed and will continue to be notified of special drops and deals.
Notifications and alerts	As a user, I want to unsubscribe to the notifications of price drops and special deals	Successfully unsubscribed to the notifications and alerts	At the top of the page, there will be a bell icon which users can click. If the icon has already been clicked,

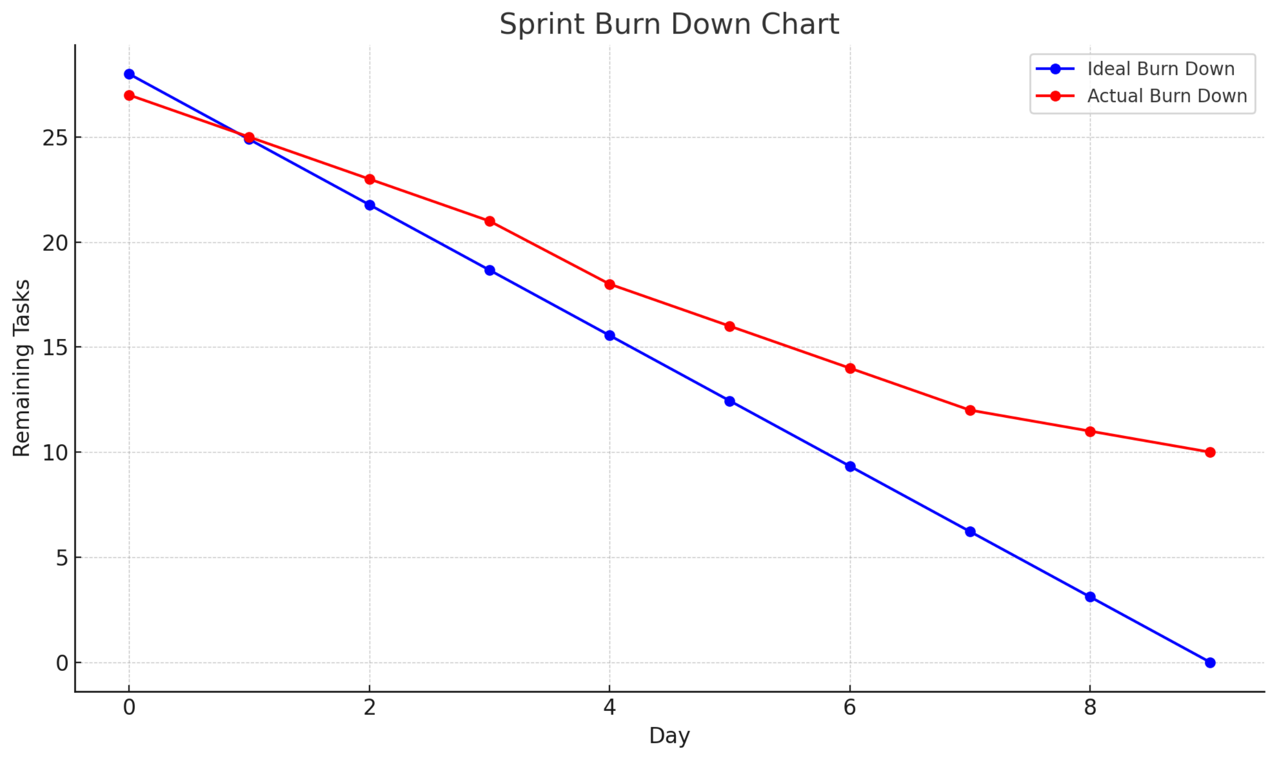
			there will be a tick next to the bell. If the user no longer wants to be notified, they can re click the bell which will remove the tick, and unsubscribe users from receiving notifications.
User-Friendly interface	As an inexperienced user I want to be able to use SuperPrice without any training or prior experience	Successfully navigate the website for the first time with no experience	The website is very self-explanatory, with large fonts for important headings and subheadings. There are big blue buttons on the website to help users be drawn to where to click, for example “checkout” is in a big blue box to help users see where to click.
User-Friendly interface	As a user living with vision impairment, I wish to engage with the product in as close of a way as possible to someone without a disability	successfully navigate the website and complete all tasks and functions required	Throughout the website, there’s big headings for everything, with clear text. There is also alternate text for images as well.

10. Product Backlog

ID	ITEM	OWNER	PRIORITY	EFFORT	STATUS	DEFINITION OF DONE	NOTES	EPIC
1	Frontend-Backend Integration	Backend	High	13	In Progress	Frontend and backend communicate smoothly	Requires thorough testing and debugging	Application CI/CD
2	Delivery Management System	Backend	High	13	Not Started	Users can schedule and track deliveries	Requires integration with third-party	Product Delivery
3	Price Comparison Algorithm	Backend	High	13	In Progress	Application compares prices across stores	Requires complex algorithm development	Product Search & Comparison
4	Product Search Functionality	Backend	High	8	Completed	Users can search for products	Implement search algorithms and filters	Product Search & Comparison
5	Product Database Setup	Backend	High	8	Completed	Database stores product information	Integrate with chosen database (e.g., MySQL)	Product Search & Comparison
6	User Registration and Login	Backend	High	5	In Progress	User can register and log in securely	Requires encryption and validation	User Management
7	User Interface Design	Frontend	High	8	In Progress	Application has a user-friendly interface	Collaborate with UI/UX designers	UX/UI Design
8	Category Browsing	Frontend	High	5	Completed	Users can browse products by category	Integrate with backend for category data	Product Search & Comparison
9	User Reviews and Ratings	Frontend	Medium	8	Not Started	Users can leave reviews and ratings	Integrate with backend for review storage	User Management
10	User Profile Management	Backend	Medium	5	In Progress	Users can edit profile settings	Allow users to update personal information	User Management
11	Notifications and Alerts	Backend	Medium	8	Not Started	Users receive price drop notifications	Implement notification system	Alerting & Notifications
12	CI/CD Implementation	DevOps	Medium	8	Completed	Automatic testing and deployment set up	Integrate with GitHub Actions	Application CI/CD
13	Performance Optimization	DevOps	Medium	8	Not Started	Application is optimized for performance	Fine-tuning of backend and frontend	Application

								Enhancement
14	Delivery Options Setup	Backend	Medium	5	Not Started	Users can choose delivery options	Implement flexible time slot selection	Product Delivery
15	Deployment and Hosting	DevOps	Medium	5	Not Started	Application deployed to recommended env	Utilize recommended hosting environment	Application CI/CD
16	User Documentation	Documentation	Low	5	Not Started	Comprehensive user guide and documentation	Essential for user onboarding	User Management

11. Sprint 2 Burndown Chart



12. Sprint 3 Planning (Estimate)

Project Sprint Planning Notes

Team: 7

Sprint: Sprint 3

Date: 18/09/2023 – 15/10/2023

Attended: (TBC)

Scrum Master: Lawrence Dawood

Product Owner: Prakash Adhikari

Development team: Darby, Laura, Lance, Saksham, Lawrence, Krisanahari

1. Goal

To enhance the frontend user interface by adding features like user reviews and ratings, and to further optimize the price comparison functionality to include real-time data and promotions.

2. Duration of the sprint

4 weeks

3. What is the team's vision for this sprint?

For this sprint, we envision:

- Expanding the frontend UI to display user reviews and ratings for products.
- Improving the price comparison functionality by incorporating real-time data and promotional prices.
- Enhancing the overall user experience by introducing features like sorting, filtering, and a more intuitive layout. By the end of the sprint, users should be able to not only compare prices but also gauge product quality through reviews, and take advantage of any current promotions.

4. Estimation in story points

- **Frontend design for user reviews and ratings**: 10 points.
    - **Justification**: Incorporating user reviews and ratings requires both frontend design changes and backend data integration. The team must ensure that the reviews are displayed in a readable manner and that users can easily submit their reviews.
- **Optimization of price comparison feature**: 15 points.
    - **Justification**: Incorporating real-time data and promotional prices adds a layer of complexity to the price comparison feature. It requires real-time API calls, error handling, and ensuring that promotional prices are clearly flagged to users.
- **Enhanced user experience features**: 7 points.

- **Justification**: While features like sorting and filtering are standard, their seamless integration into the existing UI requires careful design and testing.

13. Sprint 3 Backlog (Estimate)

PBI ID	PBI NAME	TASK ID	TASK	ASSIGNED TO	NOT STARTED	IN PROGRESS	COMPLETED	NOTES
PBI-1	User Interface Enhancements	#28	Sign-In Page UI	Saksham/Laura		✓		
PBI-1	User Interface Enhancements	#29	Product Details Webpage	Saksham/Laura		✓		
PBI-1	User Interface Enhancements	#36	Show the entire list of products webpage	Saksham/Laura		✓		
PBI-1	User Interface Enhancements	#37	Search for a specific Item	Saksham/Laura	✓			
PBI-2	Frontend and Backend Integration	#32	Integrating Frontend with the Backend	Hari/Lance		✓		
PBI-3	User Navigation	#33	Breadcrumbs Menu for easy navigation	Saksham/Laura		✓		
PBI-3	User Navigation	#34	FilterBar to search through list of products	Saksham/Laura		✓		
PBI-3	User Navigation	#48	Create links on nav bar to different pages	Saksham/Laura	✓			
PBI-4	User Reviews & Ratings	#49	Frontend design for user reviews and ratings	Saksham/Laura	✓			
PBI-4	User Reviews & Ratings	#50	Backend API for user reviews and ratings	Lawrence	✓			
PBI-5	Price Comparison	#53	Optimization of price comparison	Lawrence	✓			

			for real-time data					
PBI-5	Price Comparison	#54	Integrate promotional prices into price comparison	Lawrence	✓			
PBI-6	Improved User Experience	#56	Enhanced sorting feature for product list	Saksham/Laura	✓			
PBI-6	Improved User Experience	#57	Enhanced filtering feature for product categories	Saksham/Laura	✓			
PBI-6	Improved User Experience	#58	Improve overall site layout for intuitive navigation	Saksham/Laura		✓		
PBI-7	Backend Database	#59	Database schema refinement	Lawrence		✓		
PBI-7	Backend Database	#60	Database connection pooling	Lawrence	✓			
PBI-7	Backend Database	#61	Data backup automation	Lawrence		✓		
PBI-8	Backend Optimization	#62	API response caching	Hari	✓			
PBI-8	Backend Optimization	#63	Backend load balancing	Hari	✓			
PBI-9	Backend Security	#64	Implement JWT token authentication	Lance	✓			
PBI-9	Backend Security	#65	Encrypt sensitive user data	Lance	✓			

14. Sprint Retro

Project Sprint Planning Notes
Team: 7
Sprint: 0
Date: 20/08/23 + 17/09/23
Attended: Darby, Laura, Lance, Saksham, Lawrence, Krisanahari
Scrum Master: Lawrence Dawood
Product Owner: Prakash
Development team: Darby, Laura, Lance, Saksham, Lawrence, Krisanahari

1. Things That Went Well

What went well? What the team is happy about?

- We had good communication throughout the sprint.
- The team had clear distinction of the tasks allocated to each member.
- Everyone was concerned about each other's progress and were helping each other when needed.
- The allocation of tasks for frontend, backend, and database was clearly agreed upon.
- The end goal for the sprint was clear and straightforward.

2. Things That Could Have Gone Better

What could have gone better? What the team could improve?

- Everyone having a good and clear understanding of how the application is built (I.e., how everything is connected) could have brought out a better outcome for this sprint.
- The team could revise more on each of their shortcomings in terms of technical skills and communication skills.
- Although we already have good communication, it could still be improved by maintaining a constant/consistent schedule or routine.

3. Things That Surprised Us

What wasn't expected?

- We were surprised by the amount of preparation to setup the base code before actually implementing the application.
- We were also surprised by the amount of trial and errors before actually having everything running smoothly and properly.

4. Lessons Learned

What have you learned from the above points?

- We could have saved more time by fully understanding how the application works and how everything is connected first before diving straight into implementing it.
- We could have also planned out the implementation of the application better by agreeing on important aspects to avoid spending too much time on fixing errors that could have been easily avoided.

5. Final Thoughts

Things to Keep

- Meetings and constant communication are necessary to keep for the team to achieve the goal.
- Constant use of the project board and communication tools to update everyone on their progress.

Things to Change

- Inconsistent routine or schedule for meetings and communication between team members.
- Better planning of the tasks to be done and allocation of participation in the project.

Summary

Overall, the sprint went well as the goal was achieved. Good communication was practised throughout but could be improved by having a consistent routine or schedule and response time between team members to avoid blockage of tasks. Planning the activities needed to be done beforehand could have also been improved to save time that could have been spent on more important things which could have led to a better outcome for this sprint.