OBJECT-ORIENTED DESIGN IMPLEMENTATION AND REFLECTION

Group 3

Tran Thanh Minh – 103809048

Cao Phuc Khai Hoan - 103804739

Bani-Hashemi Le Minh – 103846074

SWE30003 – Software Architectures and Design

Instructor Tuan Tran

Swinburne University of Technology

7 April 2024

Table of Contents

Detail Design

The design for the online shopping focuses on 4 essential functions to maintain the operating system, including:

- Login & sign up: customers must sign up for an account if they do not have one yet, or else if they already have the account, they need to log in to unlock all the necessary features to have a fully experience.
- Menu: this page displays all of the product of the healthy online shopping store.
- Shopping cart: a summary of desired products that the users already adds and adjusts products' quantity before proceed the checkout.
- Product detail: illustrate the information of the product more specific and more detailed, such as its origin, how it was produced, etc.

We have chosen to follow the Model-View-Controller design architecture to best align with the object-oriented design principles and efficiently carry out the overall web-based order and delivery application:

Model Layer

The Model layer represents the application's data and business logic. It directly manages the data, logic, and rules of the application.

Classes:

- OrderHistory: Stores the complete order transactions made by customers.
- SalesReceipt: Represents detailed proof of the transaction for each sale.
- Product: Contains information about products available for sale.
- InventoryManager: Manages the stock levels of products.
- ShoppingCart: Holds the selected items for customer purchase.
- DeliveryProcess: Manages the logistics and status of product delivery.
- AnalyticsEngine: Processes data to provide insights into business performance.
- Customer: Stores information and behaviors of customers.

- Payment: Handles the transactions and payment details.
- Staff: Represents the employees and their roles within the system.

View Layer

The View layer is responsible for displaying all or a portion of the data to the user. It is the interface that the user interacts with.

Views:

- OrderHistoryView: Displays the history of customer orders.
- SalesReceiptView: Shows the receipts of sales transactions.
- ProductView: Lists the products available for purchase.
- InventoryView: Provides an interface for inventory management.
- ShoppingCartView: Shows the items in the customer's shopping cart.
- DeliveryProcessView: Tracks the delivery status of orders.
- AnalyticsDashboard: Visualizes business analytics data.
- CustomerProfileView: Displays customer personal information and order history.
- PaymentView: Processes and confirms payment transactions.
- StaffManagementView: Manages staff information and roles.

Controller Layer

The Controller layer responds to the user input and performs interactions on the data model objects. It receives input, validates it, and then performs business operations that modify the state of the data model.

Controllers:

- OrderHistoryController: Manages retrieval and display of order history.
- SalesReceiptController: Handles the creation and display of sales receipts.
- ProductController: Manages product listings and details.
- InventoryController: Controls inventory management operations.

- ShoppingCartController: Handles the addition and removal of items in the shopping cart.
- DeliveryProcessController: Oversees the delivery operations and updates.
- AnalyticsController: Collects and processes data for analytics.
- CustomerController: Manages customer account information and interactions.
- PaymentController: Processes payment information and transactions.
- StaffController: Manages staff-related functionalities.

Relationships

- ✓ Customer interacts with ShoppingCart to select products.
- ✓ ShoppingCart sends information to OrderHistory to record transactions.
- ✓ OrderHistory is used by SalesReceipt to generate transaction proofs.
- ✓ Product information is managed by InventoryManager and displayed through ProductView.
- ✓ DeliveryProcess is updated by DeliveryProcessController based on customer orders.
- ✓ AnalyticsEngine analyzes data from SalesReceipt and OrderHistory.
- ✓ Payment details are processed in conjunction with SalesReceipt.
- ✓ Staff roles are defined and managed in relation to their interaction with other classes.

This proposed MVC structure ensures a clear separation of concerns, making the system more maintainable and scalable. Each layer has a distinct responsibility, and the classes within them interact with each other to perform the necessary operations of the online food store application. The controllers act as an intermediary, processing and responding to user input and updating the views and models accordingly.

Justification of Changes and Non-Changes

Our goal is slightly changed and now is focusing on the customer experience in online shopping of the All Your Healthy Foods store. The initial, detailed object-oriented designs are kept and we add some newer additional designs in order to fulfill, justify, and finalized the overall project.

Justification of Changes

We add some of the UML sequence diagrams to support the initial design, including:

- Log In and Sign Up: before proceeding the online shopping, logging in or signing up the account are the required actions for the user. For log in functionality, the page requests the login from the authenticator, which it ask the result of verification from the database, and the status got sent back, respectively, to the user. This diagram supports the Log In and Sign Up pages, which are demonstrated below section.

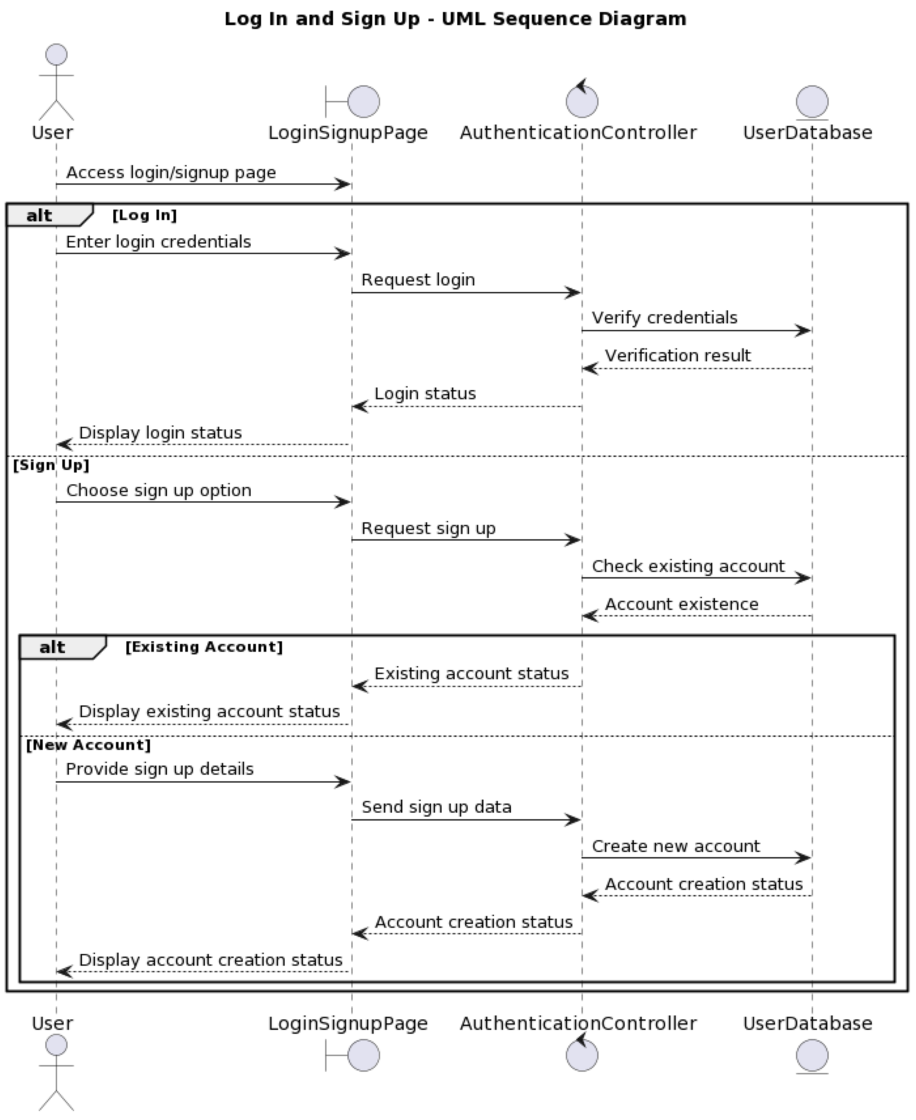


Figure 1: Log In and Sign Up UML Sequence Diagram

- Menu list and Product Detail Viewing: after logging in successfully, user can explore and browse the menu page, where it displays the available products to purchase. To acquire more information of a specific product, user needs to perform an interaction, by clicking on the image or name of the product, which leads to the page containing information of that item. The Menu and Product Detail page, which discussed below, are illustrated through this diagram.

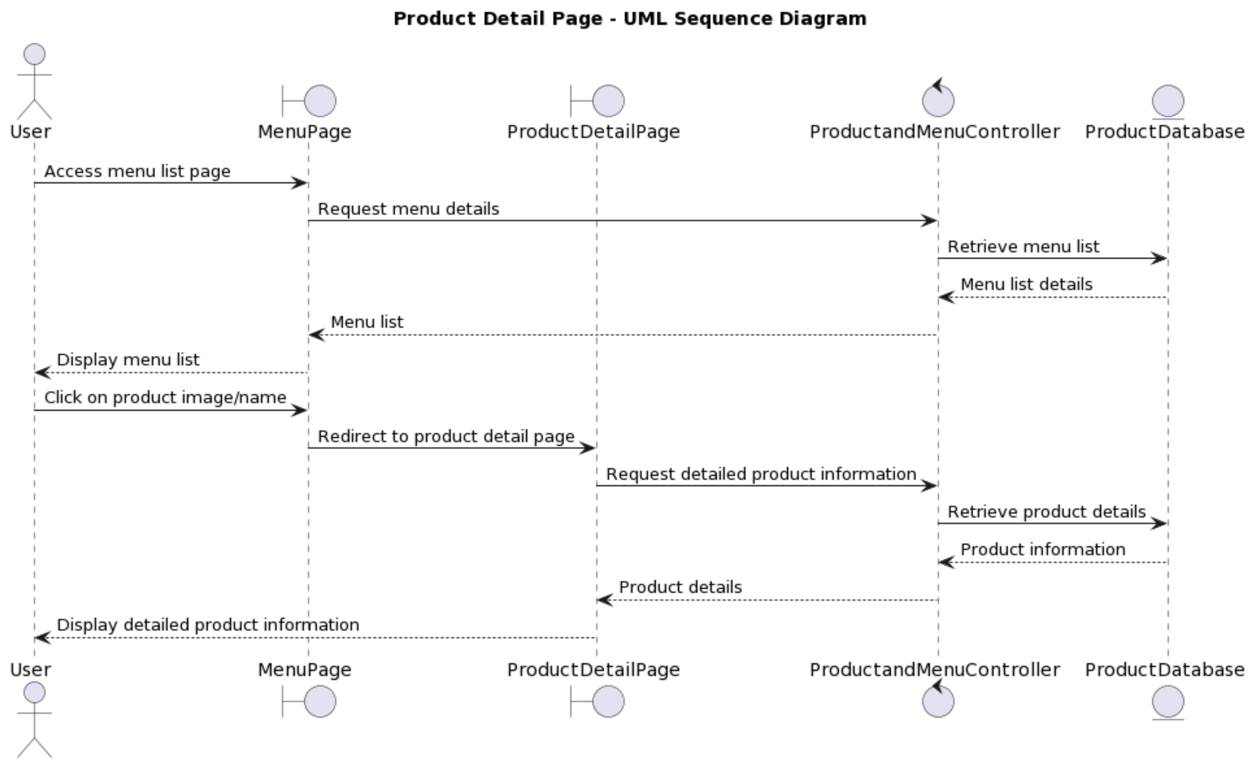


Figure 2: Menu and Product Detail UML Sequence Diagram

Non-Changes Details

The remaining of initial object-oriented designs from previous assignments are the same, including their:

- Class levels
- Responsiblities and Collaborators
- The dynamic aspects

Refinement and Adjustments:

Addressing Omissions and Errors:

- Security Considerations: We consider introducing UserAccount and Role classes to manage authentication and authorization. Implement standard security practices for user credentials.
- Persistence Layer: We try to PostgreSQL as the database system, with SQLAlchemy for object-relational mapping, ensuring efficient data management.
- Concurrency and Transactions: Utilize database transactions to manage concurrency and ensure data integrity.
- Error Handling: We are working on developing a comprehensive exception handling framework to manage unexpected conditions gracefully.

Discussion of Assignment 2

The initial design of “Add the product to the cart” fulfills our Menu, Product Detail, and Shopping Cart Page, where it supports necessary functions for user to interact with like adding, removing, removing entirely, and adjusting quantity of items.

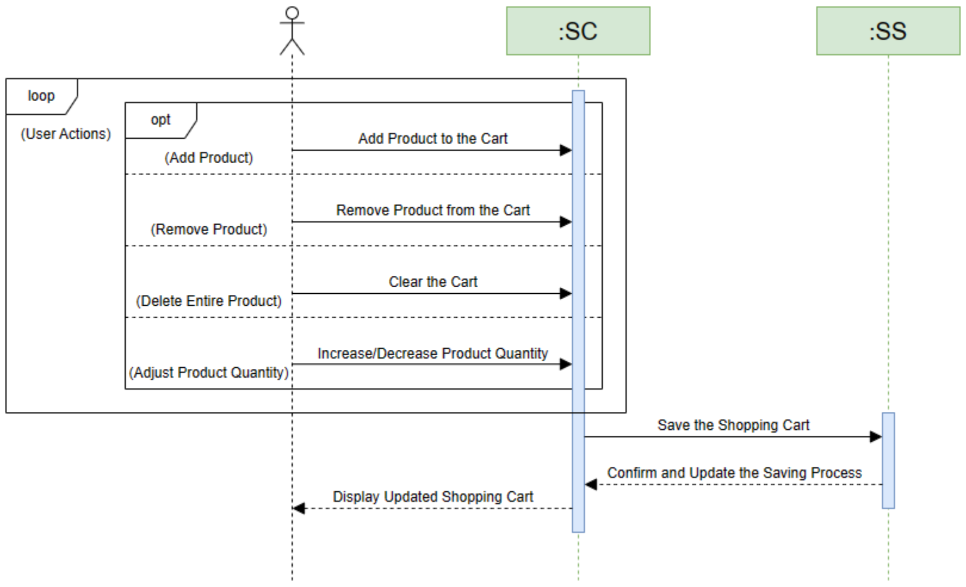

:SC
:SS
loop
(User Actions)
opt
(Add Product)
Add Product to the Cart
(Remove Product)
Remove Product from the Cart
(Delete Entire Product)
Clear the Cart
(Adjust Product Quantity)
Increase/Decrease Product Quantity
Save the Shopping Cart
Confirm and Update the Saving Process
Display Updated Shopping Cart

Figure 3: Assignment 2 implementation for Menu, Product Detail, and Shopping Cart page

- Missing aspect: the initial design falls short in detailing the implementation of functionalities within the MVC framework, particularly in areas like database integration, error handling, and security protocols. Additionally, it may not fully account for the breadth of user interactions and edge cases, which could result in overlooked functionality or usability issues.
- Flawed aspect: The original design exhibits flaws such as potential misalignment of classes with the MVC pattern, complexity in controller interaction due to poor management of user inputs and system responses, and a lack of foresight for scalability and extensibility, which could hinder future adaptations of the system.

Lesson Learnt

In crafting a comprehensive web-based application architecture, we have recognized one big imperative: to adopt a strategic approach that aligns closely with the needs and expectations of its users. That explains our later attempts in commencing with a thorough understanding of user stories first, which serves as the backbone for capturing requirements and steering design decisions afterwards. By delineating the specific functionalities and interactions that users expect from the application, user stories provide invaluable insights that inform the subsequent design process.

The iterative design approach is crucial for creating effective and meaningful object-oriented software architecture and design. It's all about continuously cycling through design, implementation, and review. This process allows our team to take in feedback, quickly adjust to new requirements, and steadily improve the user experience. With each iteration, the design gets better, and the team becomes more agile, ready to tackle the ever-changing demands of the market. It's not just about being flexible; it's about fostering a culture of constant innovation. This way, the architecture remains solid, focused on the user, and ready for whatever comes next.

In the realm of object-oriented design, the iterative approach plays a pivotal role in ensuring scalability and future-proofing the architecture. By adopting this methodology, we learned to create systems that are both robust and flexible, capable of expanding without becoming overly complex or unwieldy. It encourages the development of modular components that can be easily updated or replaced as requirements evolve. This foresight prevents the system from becoming a monolithic structure that is difficult to maintain or extend. Moreover, it allows for the anticipation of future needs, ensuring that the design can accommodate growth without significant rework, thus safeguarding the system's longevity and adaptability.

Throughout this whole process of creating and maintaining All Your Healthy Food’s web-based application software, we believe that collaborative design is essential for crafting a strong, user-focused architecture. When stakeholders and domain experts with varied experiences and viewpoints come together, it leads to a well-rounded grasp of the application's needs and limitations. This collective wisdom enables teams to devise solutions that are technically robust and in harmony with the project's broader aims. It's a co-creative process that ensures technical feasibility while staying true to the shared vision of all contributors.

By embracing a modular design which is at the heart of object-oriented principles, especially when our team collaborates on this web application for All Your Healthy Food store, we were able to see how this approach aligns perfectly with the object-oriented ethos of encapsulation and separation of concerns. By breaking down the system into well-defined, independent modules, each focused on a particular aspect of the store's functionality, the team can simplify complexity and enhance code reusability. Such a structure not only makes it easier to integrate new features and updates but also simplifies maintenance tasks like debugging. Ultimately, this leads to a more scalable and manageable codebase that can evolve alongside the business needs of the online food store.

Implementation

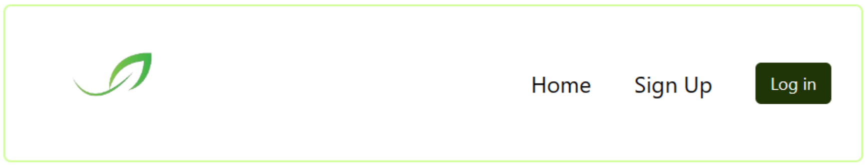
UI design for code implementation: SWE30003 – Group 3 – Assignment 3 – Figma

Link to our website: SWE30003 – Group 3 – Assignment 3 – Website

Source code on GitHub: SWE30003 – Group 3 – Assignment 3 – Github

4 areas of operation that we choose are demonstrated through scenarios in Execution and Operation section below.

1. Login and Sign Up



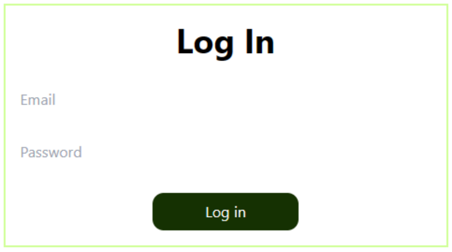


Figure 4: Log In Page

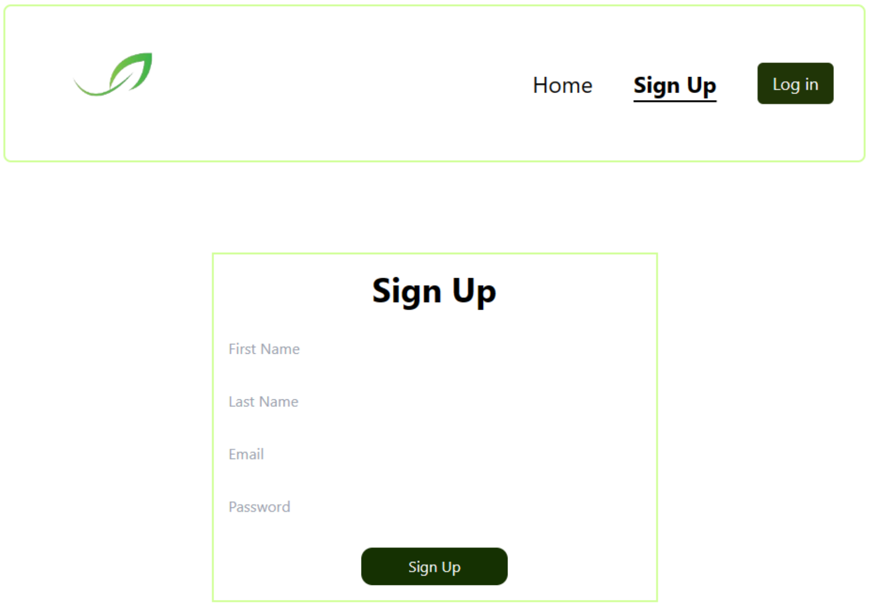


Figure 5: Sign Up Page

2. Menu Page

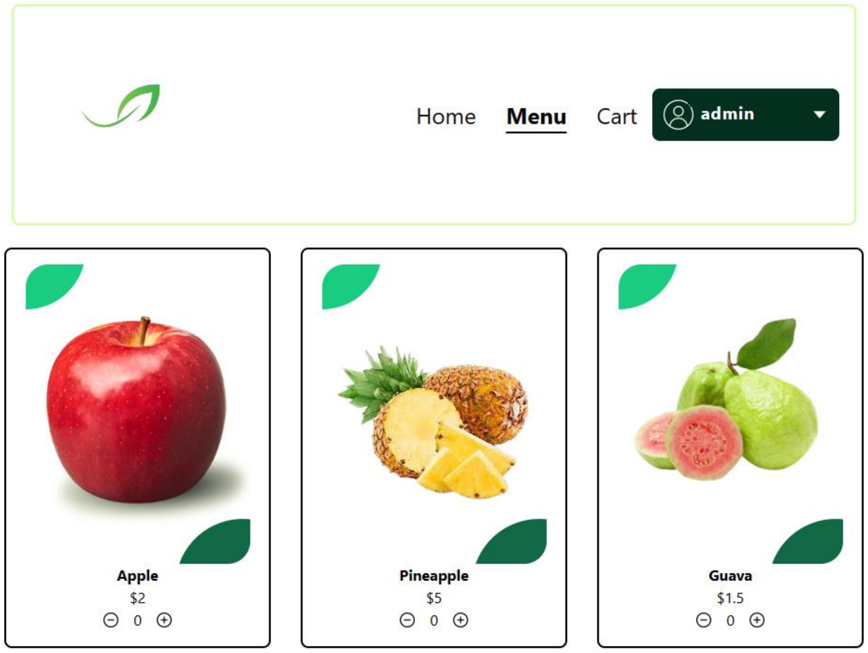
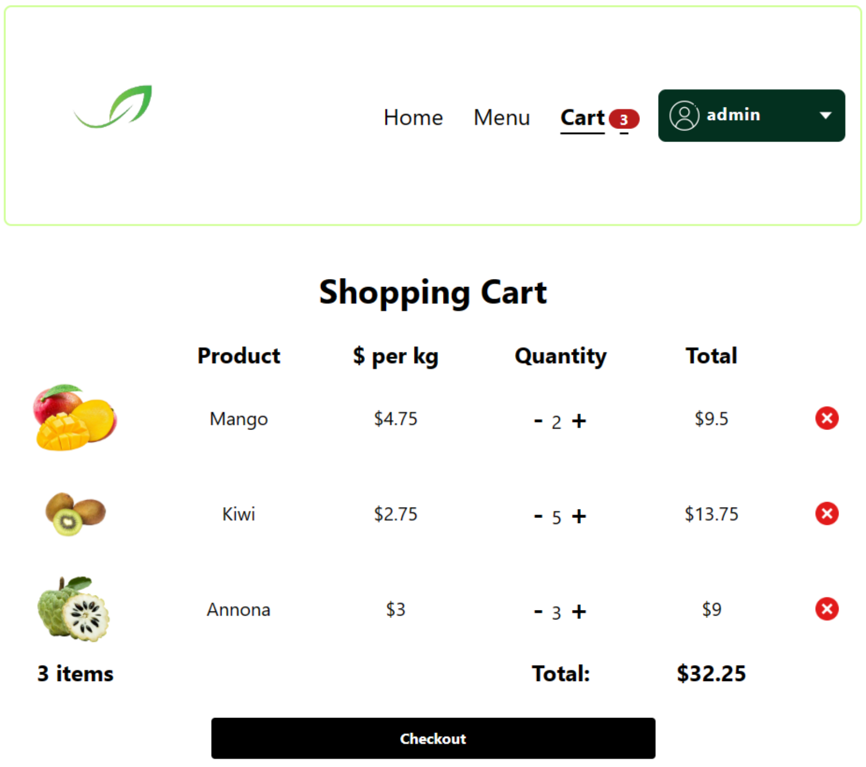


Figure 6: Menu Page

3. Shopping Cart

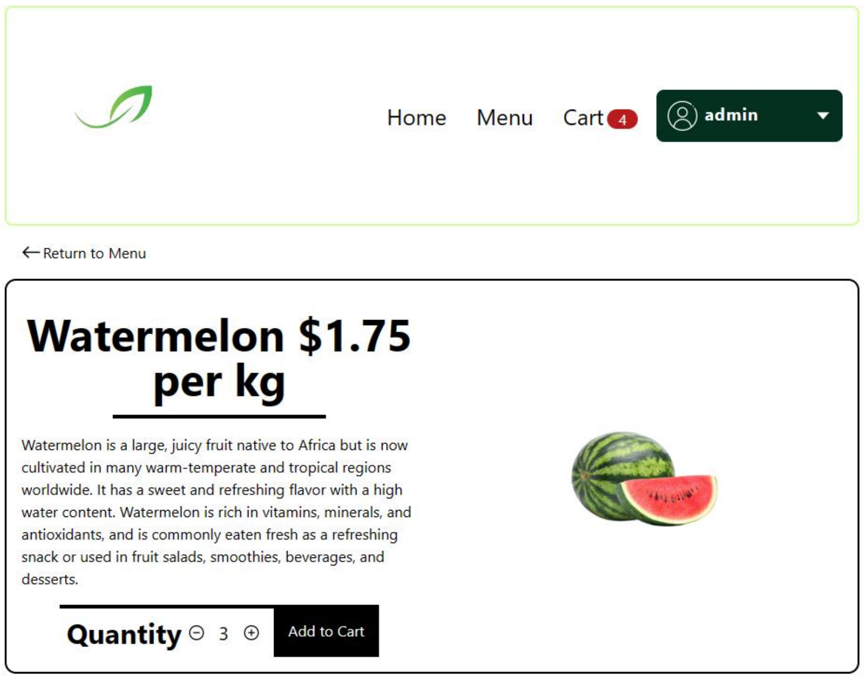


Shopping Cart

	Product	$ per kg	Quantity	Total	
	Mango	$4.75	- 2 +	$9.5	
	Kiwi	$2.75	- 5 +	$13.75	
	Annona	$3	- 3 +	$9	
3 items			Total:	$32.25	

Figure 7: Shopping Cart Page

4. Product Detail



Watermelon $1.75 per kg

Watermelon is a large, juicy fruit native to Africa but is now cultivated in many warm-temperate and tropical regions worldwide. It has a sweet and refreshing flavor with a high water content. Watermelon is rich in vitamins, minerals, and antioxidants, and is commonly eaten fresh as a refreshing snack or used in fruit salads, smoothies, beverages, and desserts.

Figure 8: Product Detail Page

Execution and Operation

Scenario 1 – Login and Sign Up

1. Start with the initial login screen where users can enter their credentials.

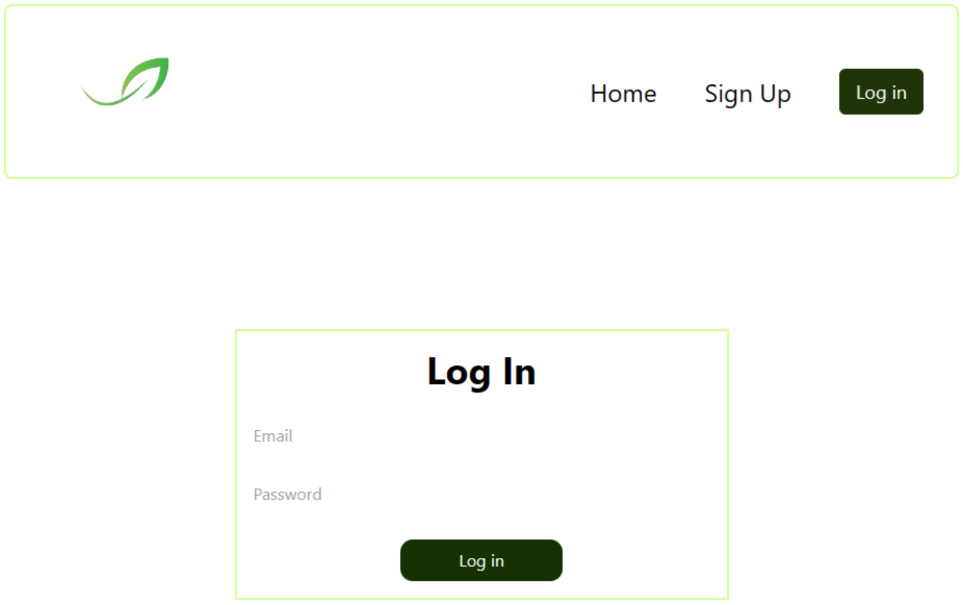


Figure 9: Overview of Login Screen

2. Provide options for users to sign in if they already have an account or sign up for a new one if they don't.

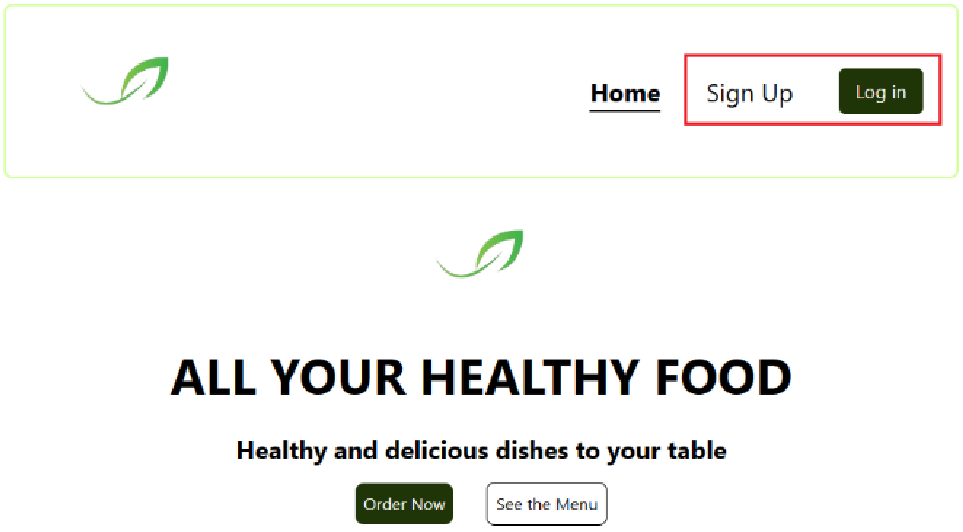


Figure 10: Provide users the login and sign up options

3. Illustrate successful login with correct credentials and display the menu screen.

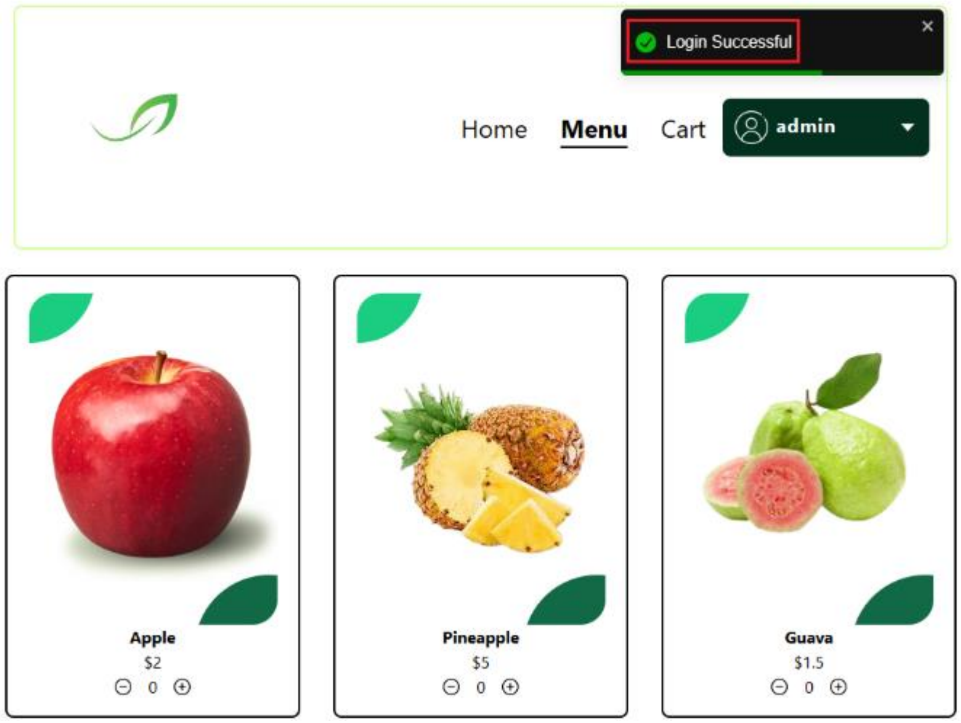


Figure 11: User is redirected to the Menu page after login successfully

4. Validate incorrect login details and display error messages accordingly.

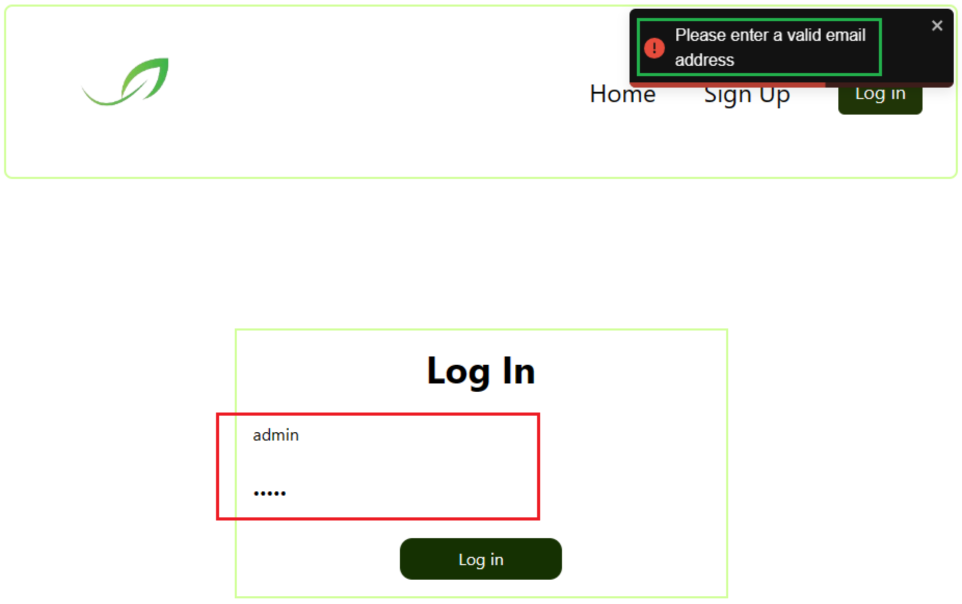
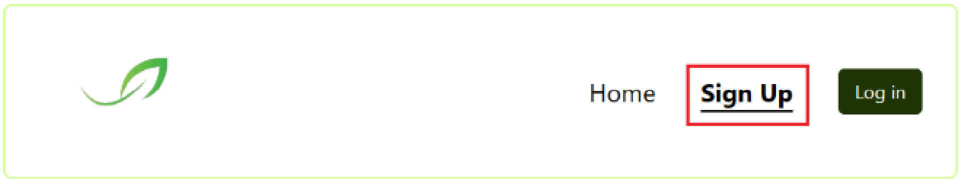
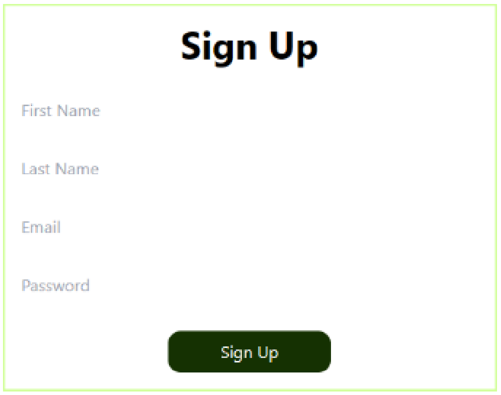


Figure 12: Error message will appear when user enters information incorrectly

5. Show the signup screen where users can enter their details to create a new account.




Sign Up

First Name

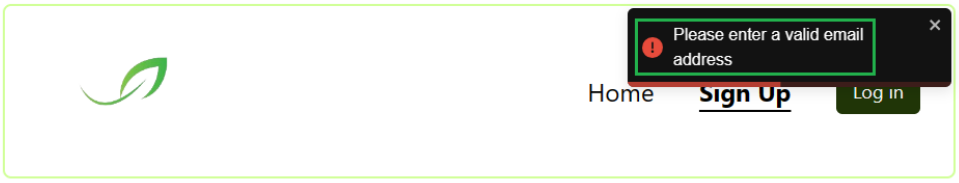
Last Name

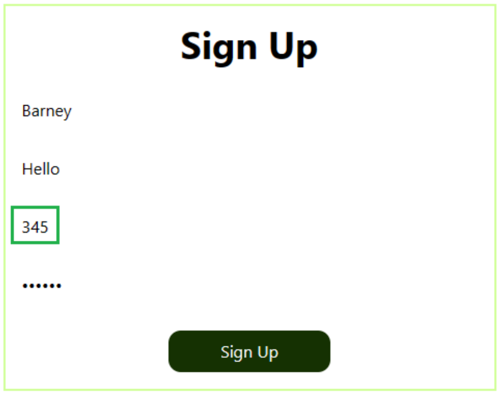
Email

Password

Figure 13: Overview of Sign Up page

6. Validate the input data during signup, including email format, etc.


Please enter a valid email address


Sign Up

Barney

Hello

345

••••••

Figure 14: User is asked if the requirements to Sign Up is incompleted

7. Illustrate successful signup and redirection to the menu screen.

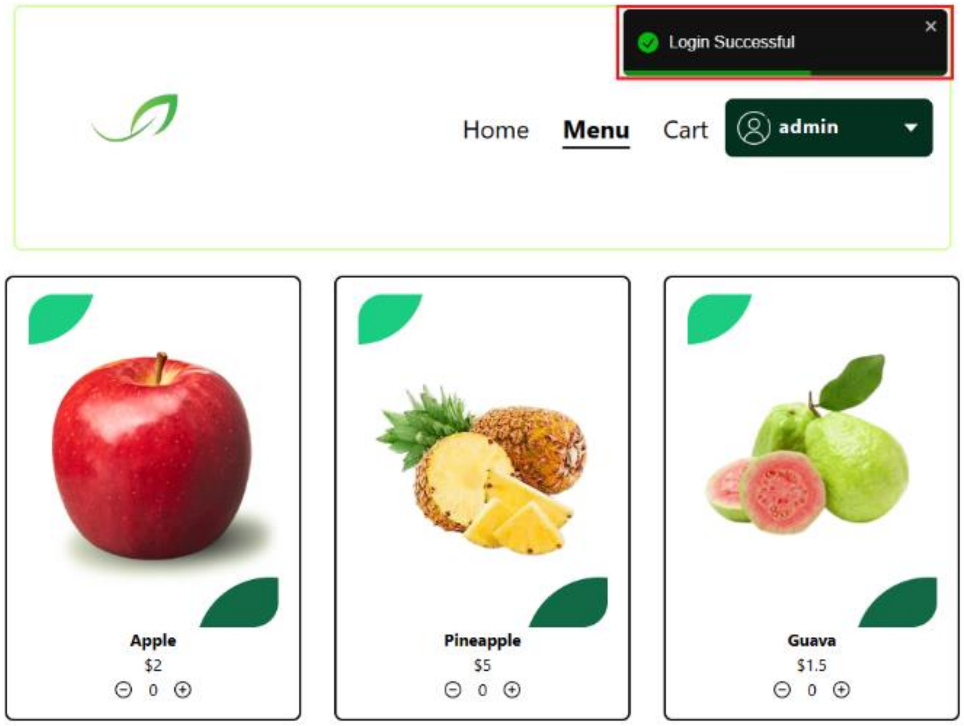


Figure 15: User is redirected to the Menu page after sign up successfully

Scenario 2 – Menu Page

1. Begin with the menu page displaying various products available for purchase.

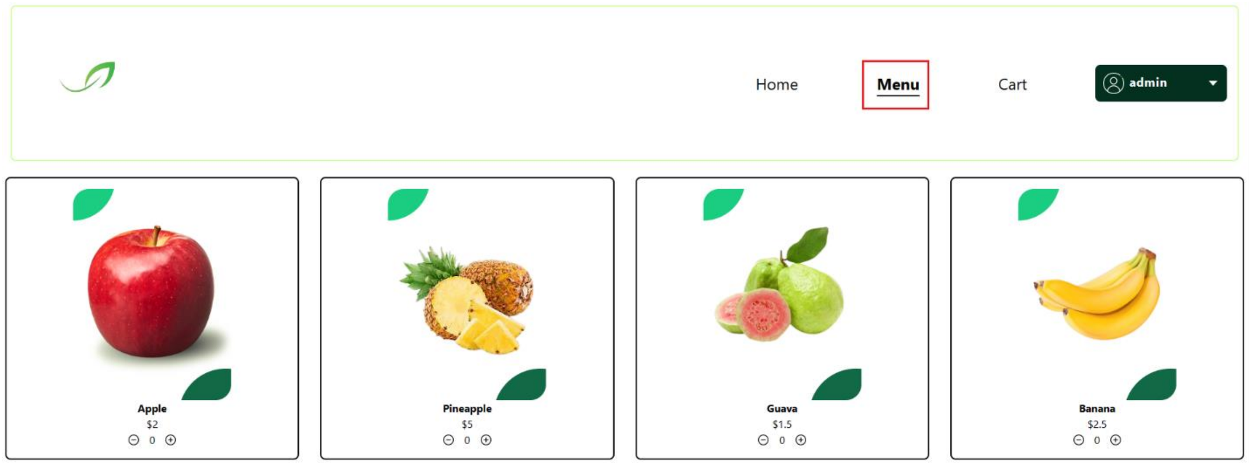


Figure 16: Overview of Menu page

2. Illustrate adding products to the cart by clicking on the "+" or remove by clicking the "-" button.

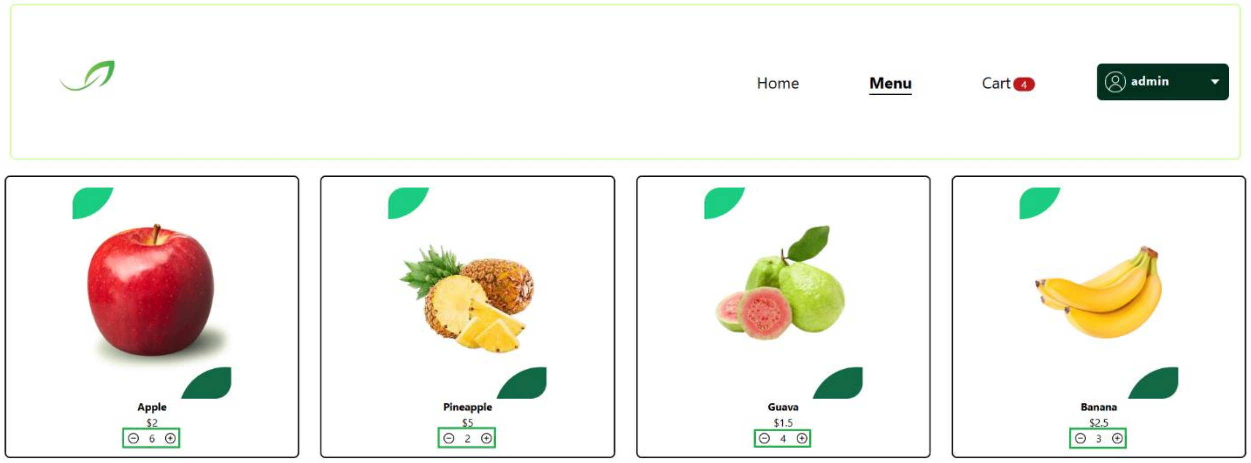


Figure 17: Adjusting product quantity directly below the images

3. Validate the addition of products to the cart when the quantity changes.

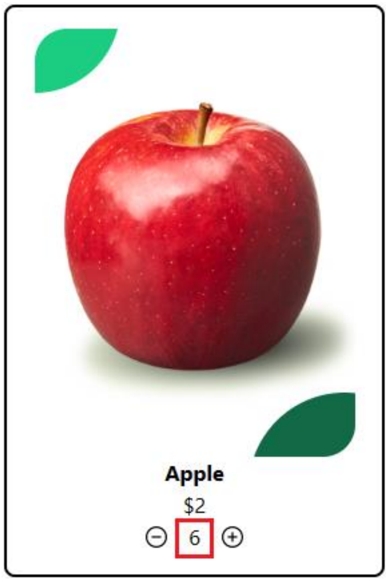


Figure 18: Quantity is visually changed when adjusting the number

4. Show the updated cart icon indicating the number of items added.

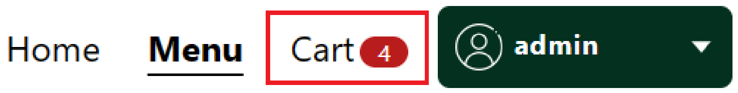


Figure 19: The Cart is updated with different number of items added

5. Provide options to view product details by clicking on the product name or image, as it will navigate to the product detail page.

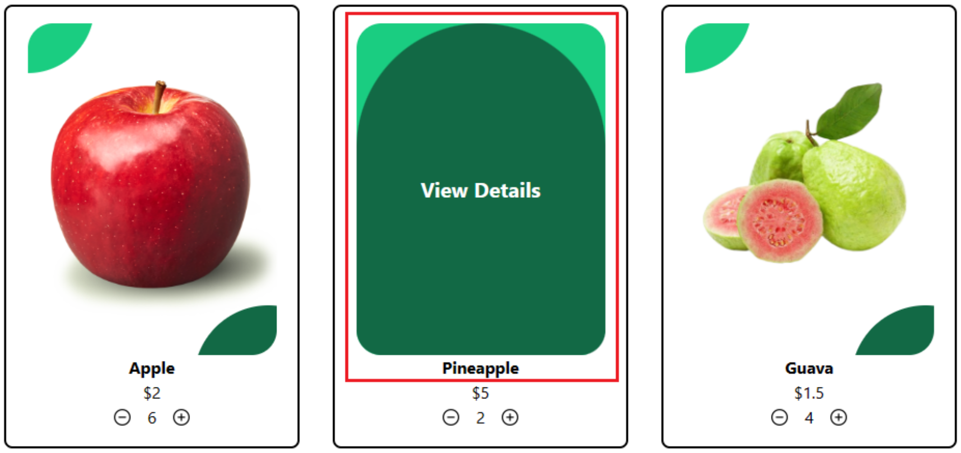
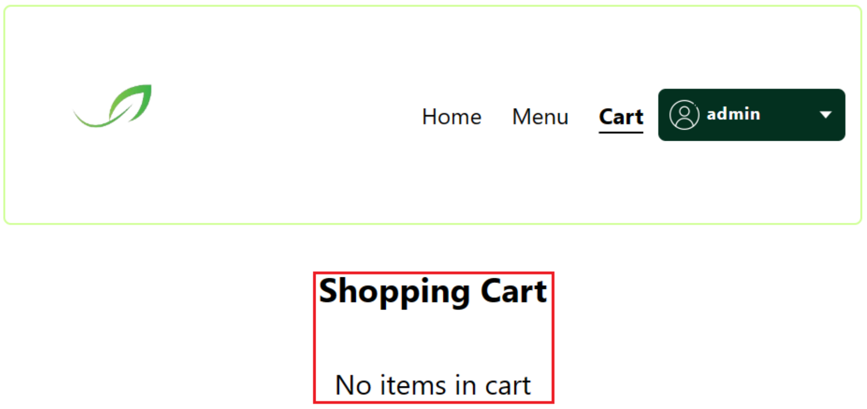


Figure 20: View Details appears when hovering to product image or its name

Scenario 3 – Shopping Cart

1. Start with an empty shopping cart screen.

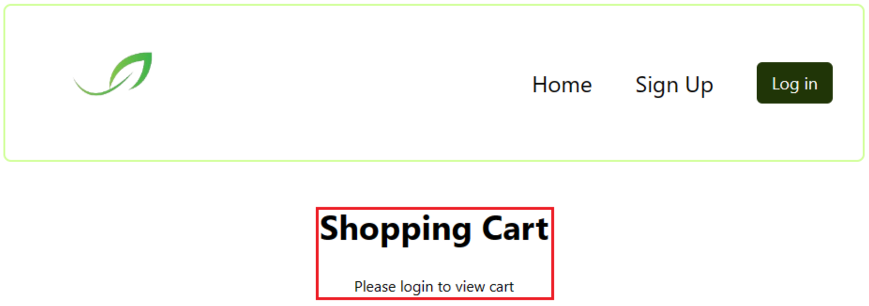


Shopping Cart

No items in cart

Figure 21: Overview of Shopping Cart page

2. If the user is not sign in, then it will display message require user to sign in



Shopping Cart

Please login to view cart

Figure 22: Shopping Cart page is only available when user alreay signs in

3. Illustrate adding products to the cart from the menu page.

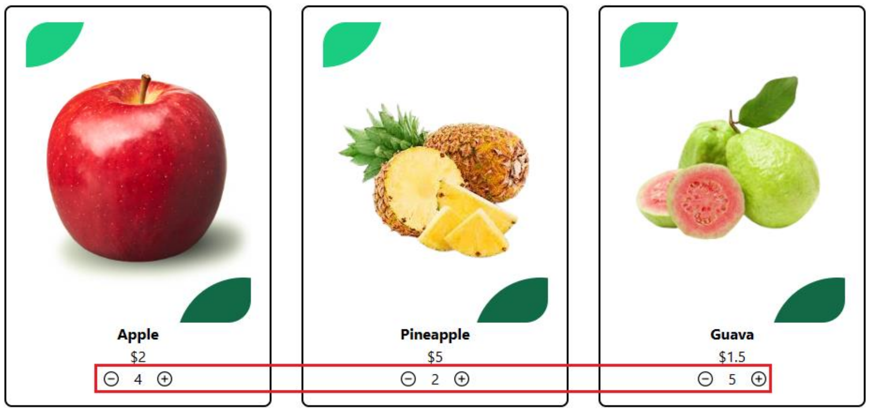
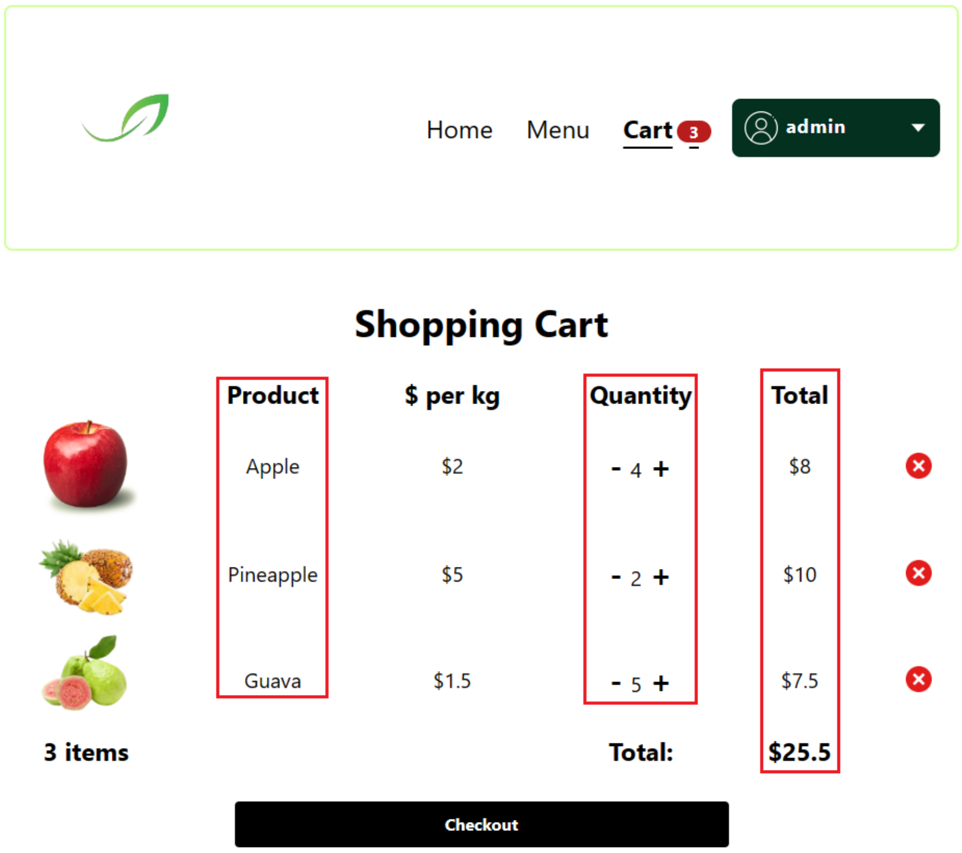


Figure 23: To check if shopping cart updates the record, adjusting various product quantities in the Menu page

4. Display the added products with their names, quantities, and prices in the cart.

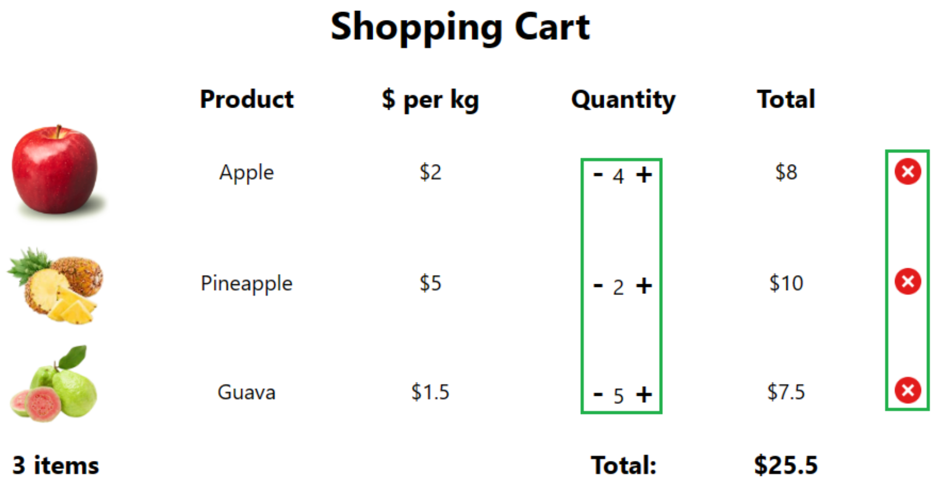


Shopping Cart

	Product	$ per kg	Quantity	Total	
	Apple	$2	- 4 +	$8	
	Pineapple	$5	- 2 +	$10	
	Guava	$1.5	- 5 +	$7.5	
3 items			Total:	$25.5	

Figure 24: In shopping cart page, products and their quantities are added and updated respectively from the menu page

5. Allow users to adjust the quantity or remove items from the cart.

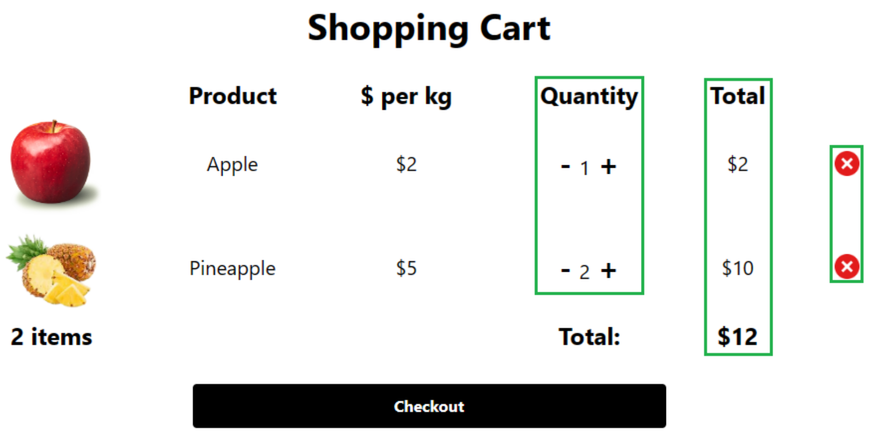


Shopping Cart

	Product	$ per kg	Quantity	Total	
	Apple	$2	- 4 +	$8	
	Pineapple	$5	- 2 +	$10	
	Guava	$1.5	- 5 +	$7.5	
3 items			Total:	$25.5	

Figure 25: Removing or adjusting quantity of the product's features also available in the shopping cart page

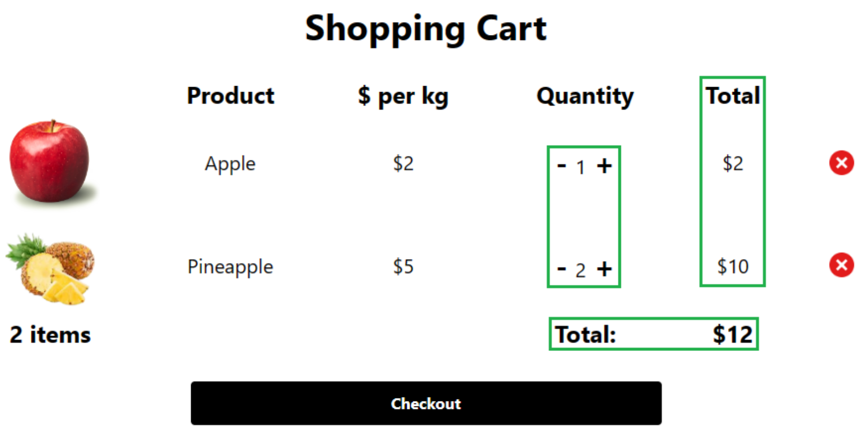
6. Illustrate reducing the quantity of products or removing them from the cart.


Shopping Cart

	Product	$ per kg	Quantity	Total	
	Apple	$2	- 1 +	$2	⊗
	Pineapple	$5	- 2 +	$10	⊗
2 items			Total:	$12	

Figure 26: Remove the Guava and reduce Apple quantity to 1 to see difference

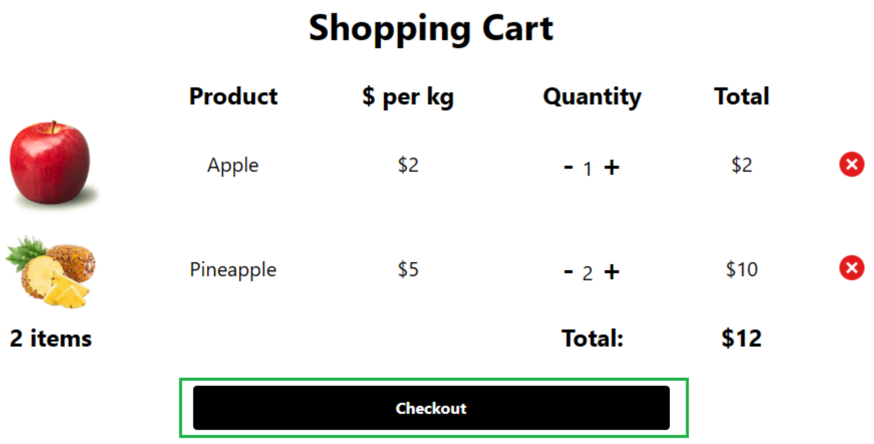
7. Validate the changes made to the cart and display updated totals.


Shopping Cart

	Product	$ per kg	Quantity	Total	
	Apple	$2	- 1 +	$2	⊗
	Pineapple	$5	- 2 +	$10	⊗
2 items			Total:	$12	

Figure 27: Displaying the changes of product quantity and total pricing

8. Provide options to proceed to checkout.

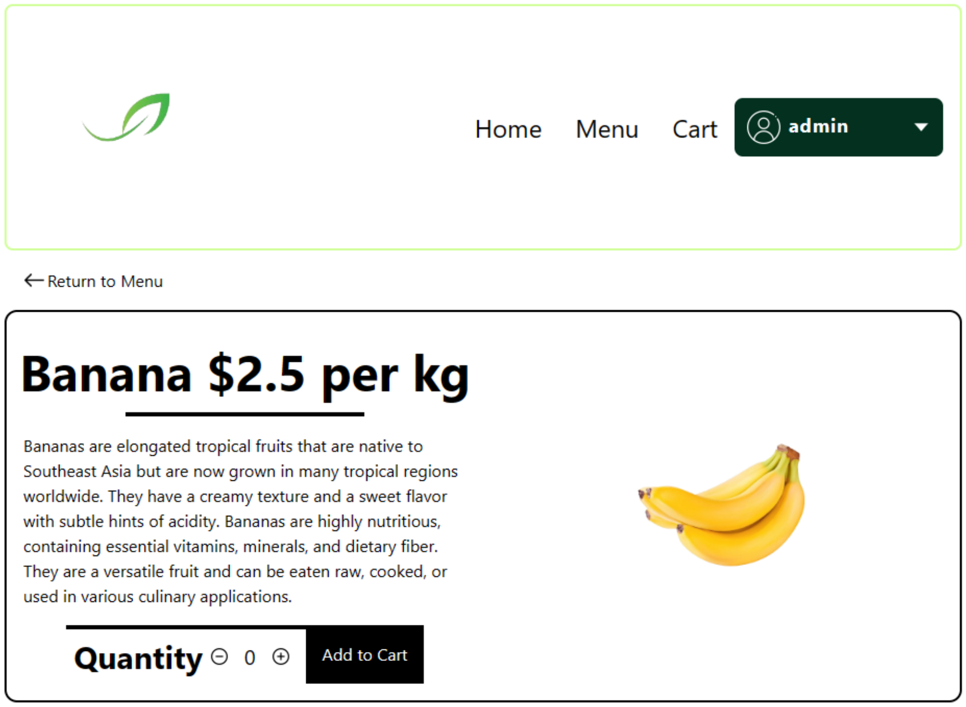

Shopping Cart

	Product	$ per kg	Quantity	Total	
	Apple	$2	- 1 +	$2	⊗
	Pineapple	$5	- 2 +	$10	⊗
2 items			Total:	$12	

Figure 28: Proceed to the payment section by clicking Checkout button

Scenario 4 – Product Detail

1. Begin with the product detail page displaying information about a specific product.

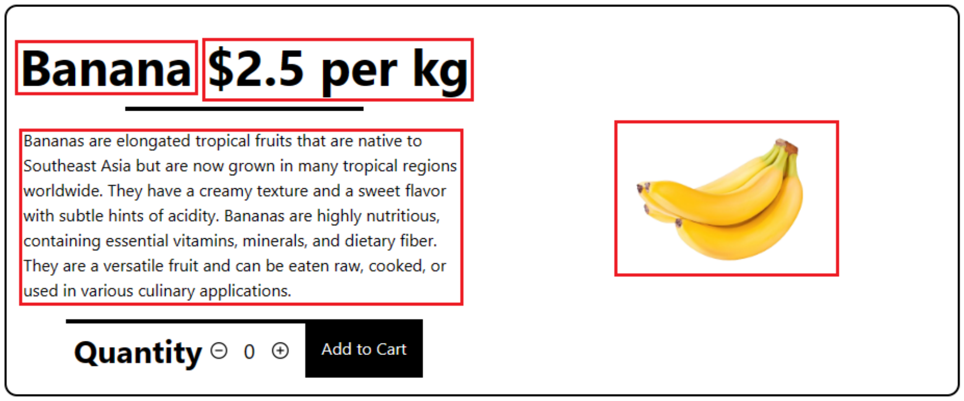


Banana $2.5 per kg

Bananas are elongated tropical fruits that are native to Southeast Asia but are now grown in many tropical regions worldwide. They have a creamy texture and a sweet flavor with subtle hints of acidity. Bananas are highly nutritious, containing essential vitamins, minerals, and dietary fiber. They are a versatile fruit and can be eaten raw, cooked, or used in various culinary applications.

Figure 29: Overview of Product Detail page

2. Show the product image, name, description, and price.

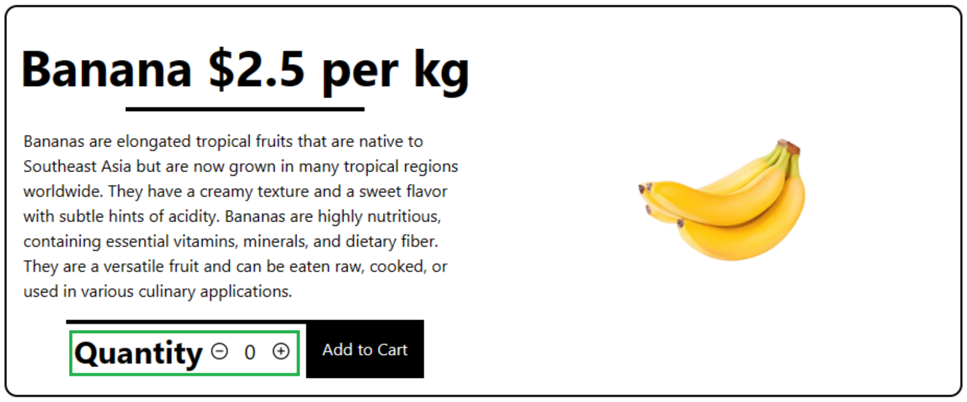


Banana $2.5 per kg

Bananas are elongated tropical fruits that are native to Southeast Asia but are now grown in many tropical regions worldwide. They have a creamy texture and a sweet flavor with subtle hints of acidity. Bananas are highly nutritious, containing essential vitamins, minerals, and dietary fiber. They are a versatile fruit and can be eaten raw, cooked, or used in various culinary applications.

Figure 30: Picture, name, description, and price are fully displayed

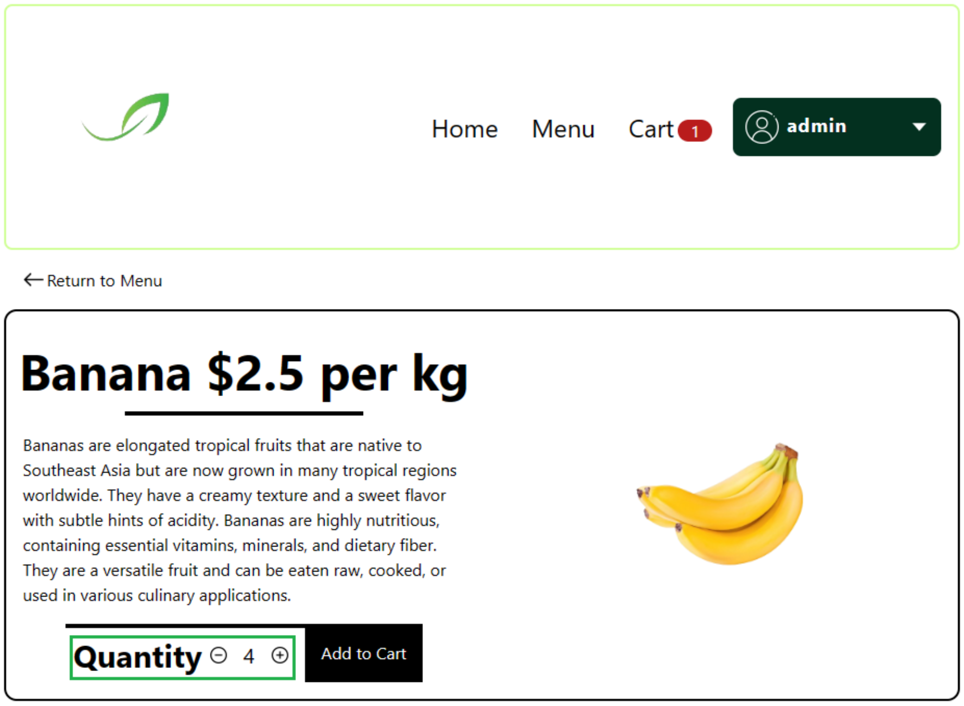
3. Display the current quantity of the product in the cart (0 if not added).


Banana $2.5 per kg

Bananas are elongated tropical fruits that are native to Southeast Asia but are now grown in many tropical regions worldwide. They have a creamy texture and a sweet flavor with subtle hints of acidity. Bananas are highly nutritious, containing essential vitamins, minerals, and dietary fiber. They are a versatile fruit and can be eaten raw, cooked, or used in various culinary applications.

Figure 31: Quantity of the product is shown

4. Illustrate adding the product to the cart or adjusting its quantity.

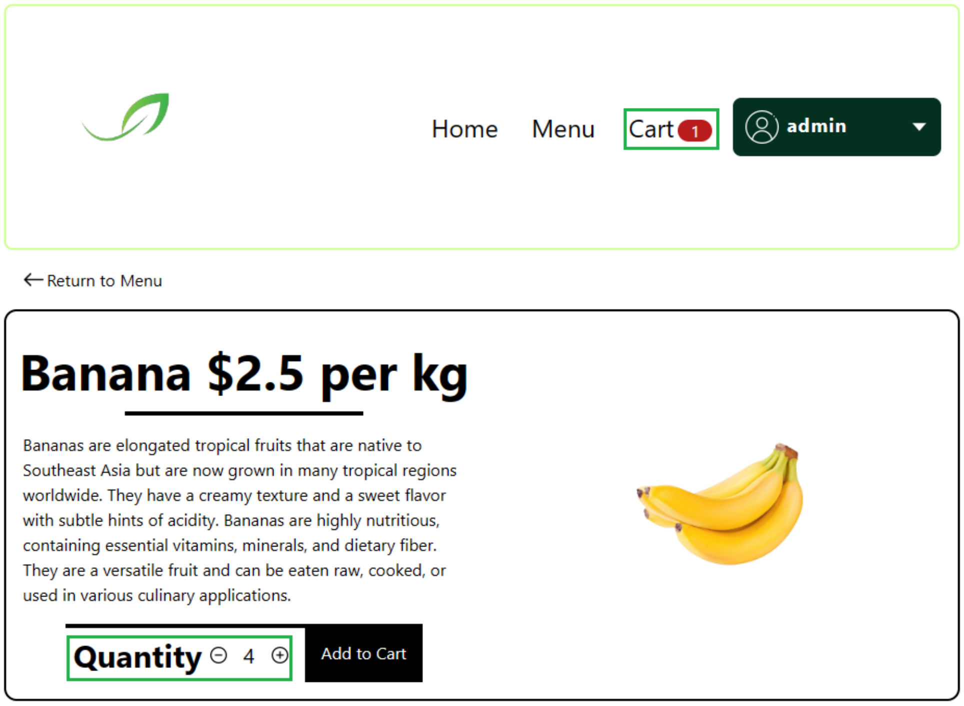


Banana $2.5 per kg

Bananas are elongated tropical fruits that are native to Southeast Asia but are now grown in many tropical regions worldwide. They have a creamy texture and a sweet flavor with subtle hints of acidity. Bananas are highly nutritious, containing essential vitamins, minerals, and dietary fiber. They are a versatile fruit and can be eaten raw, cooked, or used in various culinary applications.

Figure 32: Product quantity can be adjusted from product detail page

5. Validate the addition or modification of the product in the cart.

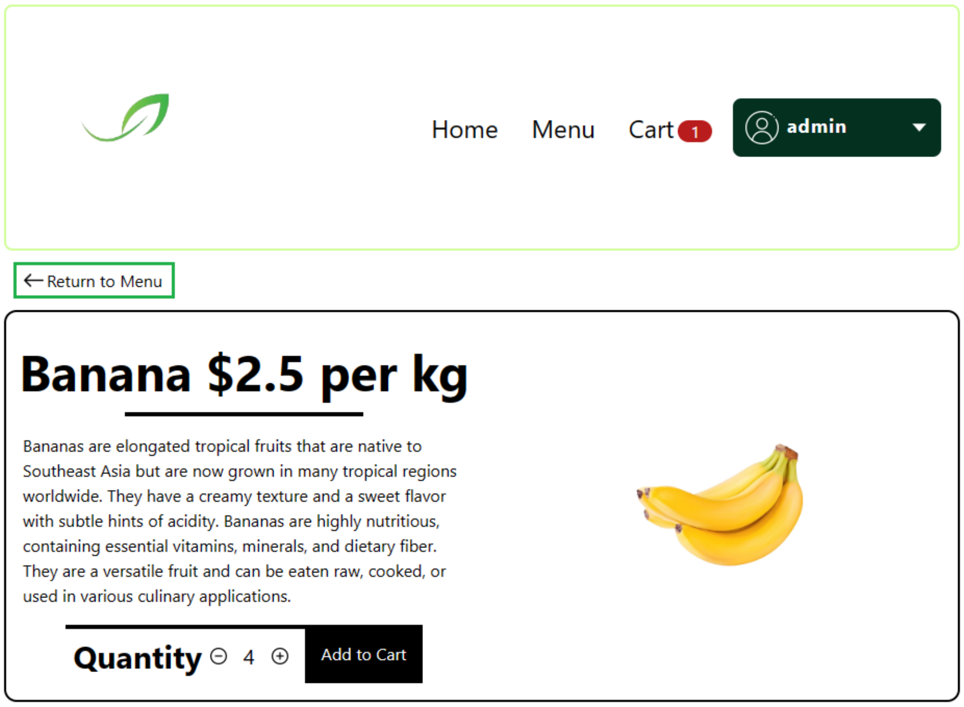


Banana $2.5 per kg

Bananas are elongated tropical fruits that are native to Southeast Asia but are now grown in many tropical regions worldwide. They have a creamy texture and a sweet flavor with subtle hints of acidity. Bananas are highly nutritious, containing essential vitamins, minerals, and dietary fiber. They are a versatile fruit and can be eaten raw, cooked, or used in various culinary applications.

Figure 33: Adding new item by adjusting quantity will illustrate the number of item besides shopping cart link in navigation bar

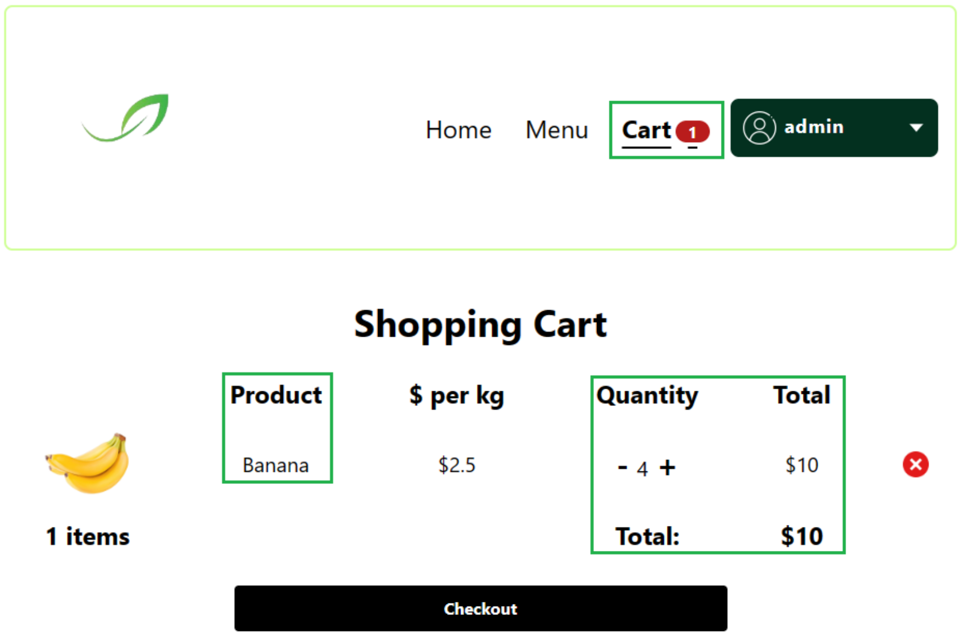
6. Provide options to return to the menu page or continue shopping.



Banana $2.5 per kg

Bananas are elongated tropical fruits that are native to Southeast Asia but are now grown in many tropical regions worldwide. They have a creamy texture and a sweet flavor with subtle hints of acidity. Bananas are highly nutritious, containing essential vitamins, minerals, and dietary fiber. They are a versatile fruit and can be eaten raw, cooked, or used in various culinary applications.

7. Show successful addition of the product to the cart.



Shopping Cart

	Product	$ per kg	Quantity	Total
	Banana	$2.5	- 4 +	$10
1 items			Total:	$10

Deploy and Run

Version:

- IDE: Visual Studio Code
- Operation: Window 11
- react: 18.2.0
- react-dom: 18.2.0
- react-icons: 5.0.1
- react-scripts: 5.0.1
- react-toastify: 10.0.5
- tailwindcss: 3.4.3

Steps for deployment and run:

1. Clone the repository from the provided URL.
    - Using *git clone provided URL* or source code zip file can be extracted.
2. Install the necessary dependencies using npm or yarn.
    - Using *npm i*
3. Build the application for production using the appropriate command.

    - Using *npm build*

4. Deploy the built files to the hosting platform or server.
5. Access the deployed application through the provided URL.
    - I have deployed using Vercel: https://swe-30003-software-architectures-and-design.vercel.app/
6. Test the functionality of each scenario to ensure proper operation.
7. Monitor logs and error messages for any issues during deployment and runtime.