

# Team Deliverable 3 - The Core 4

## Meeting Notes

This document contains updated Phase 1 requirements, UML design, user manual, compile and run instructions, reflection, and member contribution table. Content builds upon the previous Team Deliverable 2.

### **A) Requirements**

#### **Functional Requirements Implemented:**

- 1) The system must allow users to create an account by entering their name, email, and billing information.
  - a) Creates a unique profile that allows for future functionality with rewards, orders, and lists.
- 2) The system must display a product catalog showing grocery items with names, prices, and images.
  - a) This is an essential functionality because if users cannot access the catalog, no orders can be created.
- 3) The system must allow users to add, remove, and update quantities of products in their shopping cart.
  - a) This is a core functionality that enables users to manage their groceries and adjust quantities before checkout.
- 4) The system must calculate subtotal, total price, and discounts if applied in the shopping cart automatically.
  - a) Provides accurate order totals for users before checkout.
- 5) The system must allow users to enter coupon codes and apply the corresponding discount to the order total.
  - a) This incentivizes purchases and returning customers. Users may use “Welcome10” (10% off), and “Welcome5” (\$5 off \$25 orders) when using the program.
- 6) The system must award users reward points at the rate of one point per \$10 spent on groceries.
  - a) This functionality encourages users to use BasketBuddy and create a loyalty program for every time they return.
- 7) The system must update and display the reward points tracker on the home page after each successful checkout when logged in.
  - a) Registered users must be logged in, displaying their points tracker on the home page.
- 8) Users must be able to sort and filter items while looking through the catalog of goods.
  - a) Gives users more efficient feedback and sortability to navigate the catalog and find the desired groceries.
- 9) Users must be able to create, manage, or delete lists.
  - a) This gives users the ability to make note of important items or favorites, ensuring that no item is forgotten during checkout.
- 10) Users must be able to see past orders after they checkout.
  - a) This allows users to see a list of what they have already purchased in the past, which may be useful repeat orders.
- 11) Users must be able to edit their profile and delete their account.
  - a) Gives users the functionality to change their specific information and delete the contents if it needs to be modified.

#### **Functional Requirements Changed/Removed:**

- 1) The system was supposed to store all data in a SQLite database, but we decided to go with a local storage configuration.

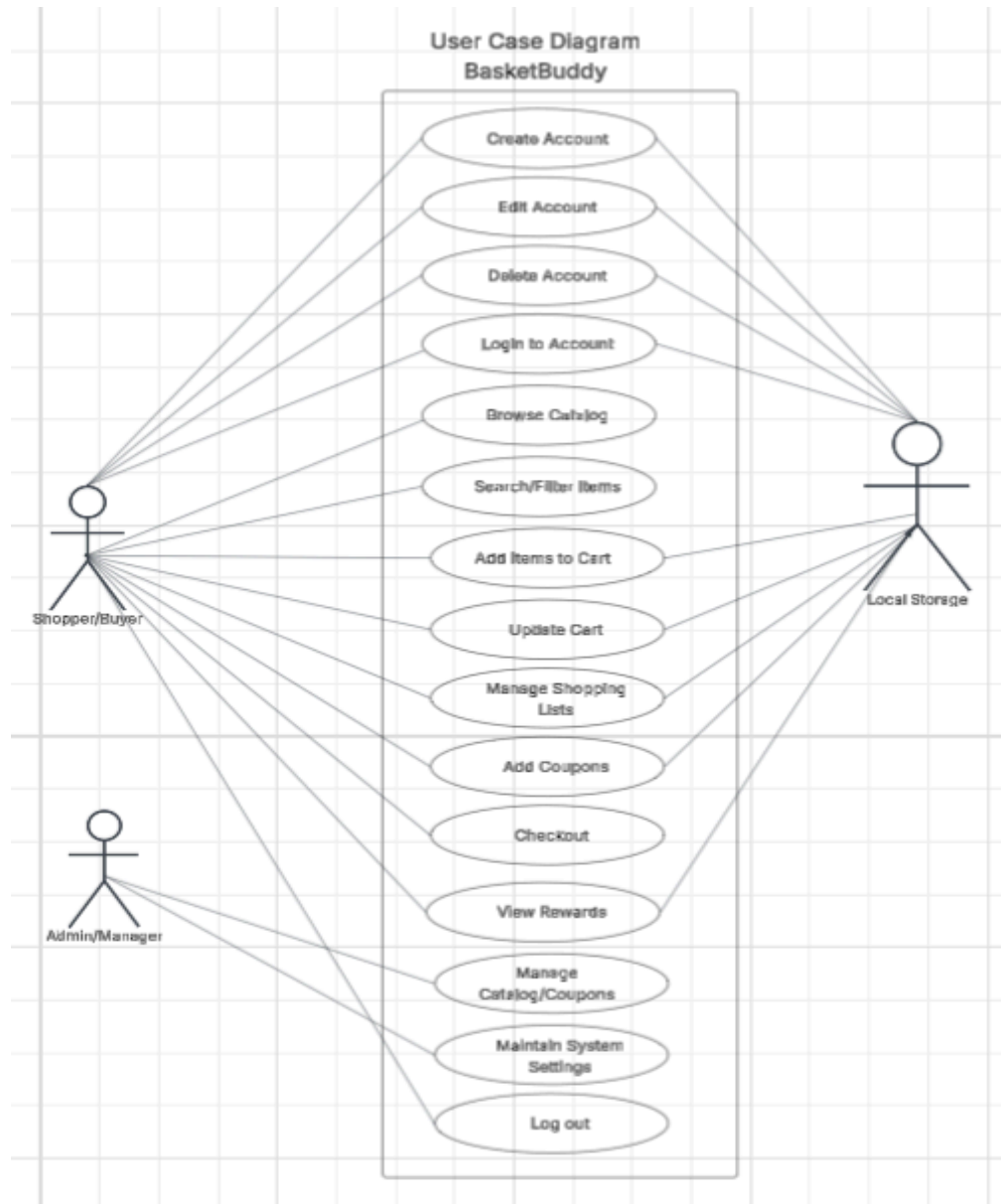
- a) We were planning on implementing a working off site database using SQLite, however, it seemed much more complicated to connect everything together. Using a local storage solution made it much easier to work and functioned with everything we needed for this deliverable.
- 2) The system was supposed to have an admin account to add or remove products.
  - a) We cut this feature to focus on the shopper's experience instead of the back end management side.

**Non-Functional Requirements:**

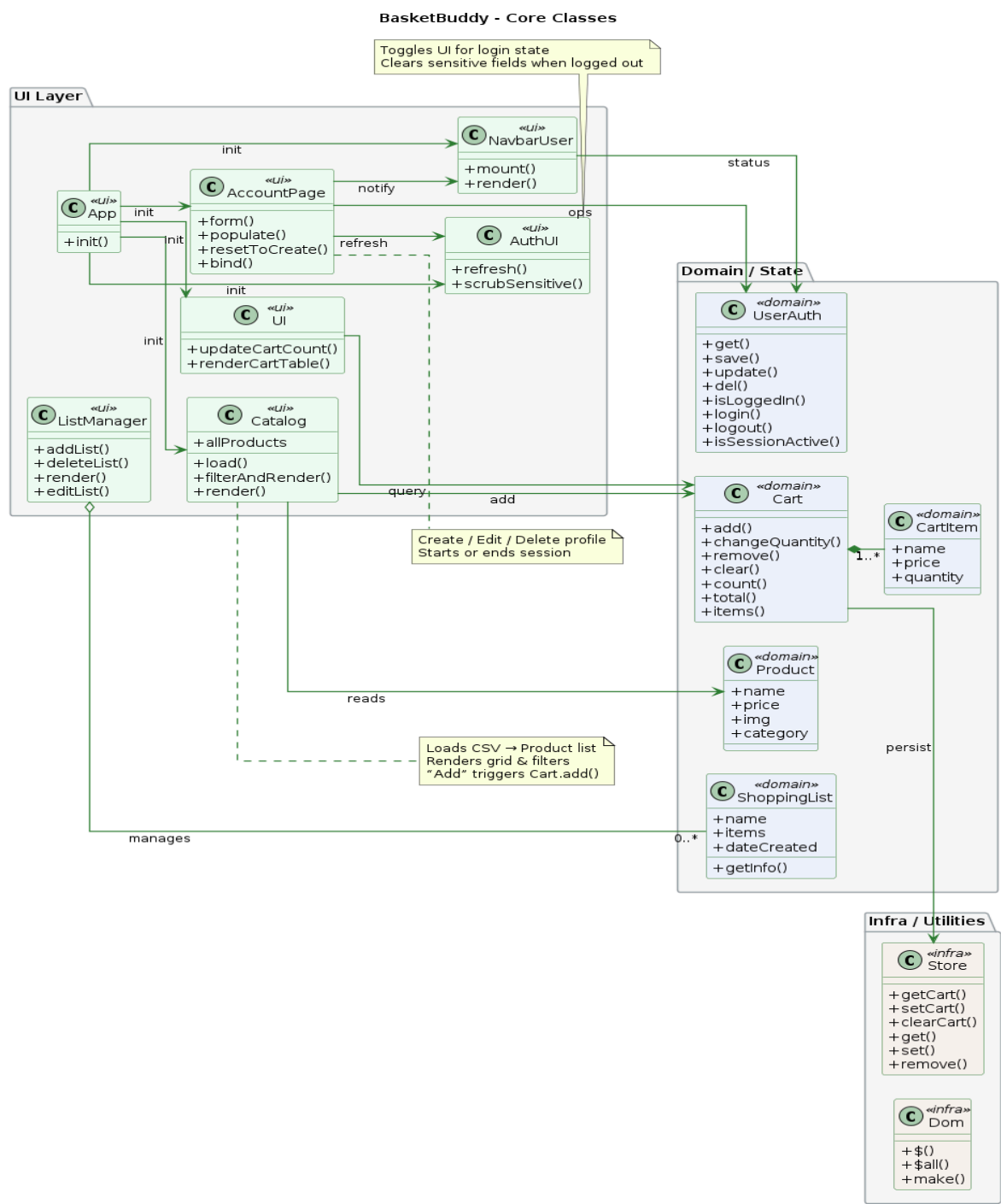
- 1) All pages must load within two seconds on a standard internet connection.
- 2) When the catalog is loaded, sorting and filtering must be completed in under two seconds after the user requests it.
- 3) When a user interacts with any button or menu, responses must occur within one second of the request.
- 4) When the cart page loads, updating quantities or removing an item must refresh the total within one second without reloading the page

## **B) UML Design**

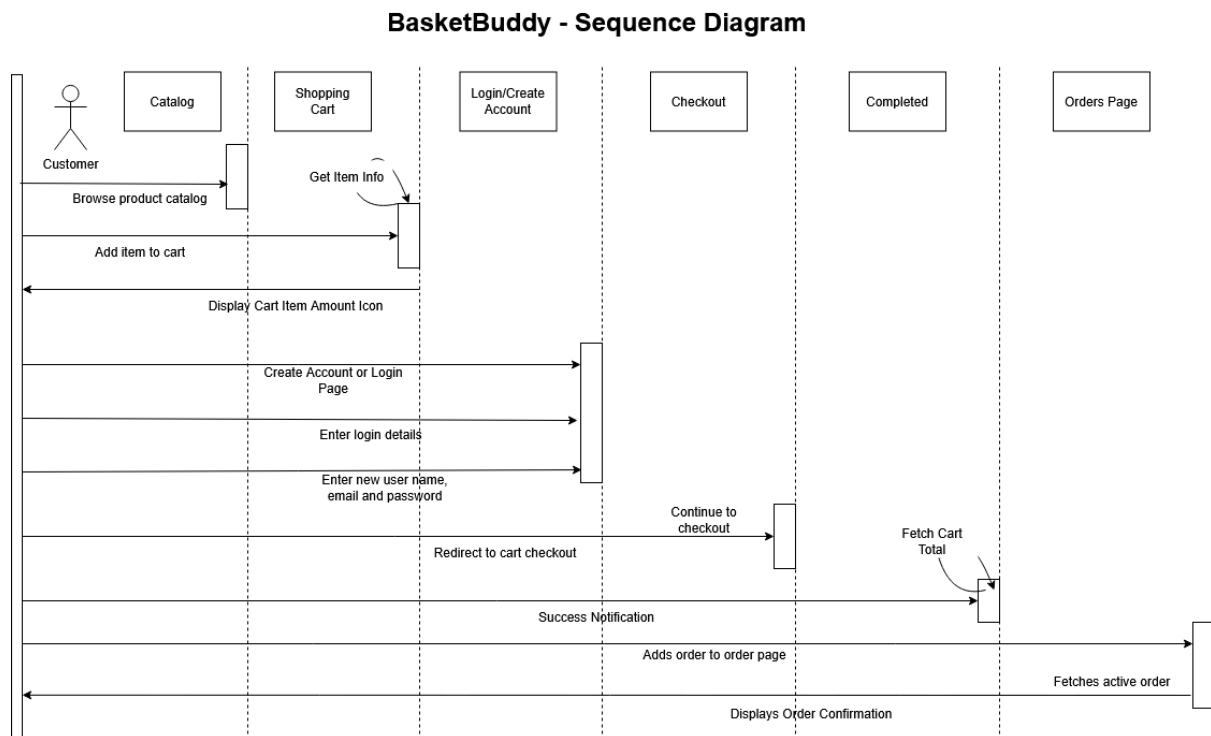
### **Use Case Diagram**



# Class Diagram



## Sequence Diagram



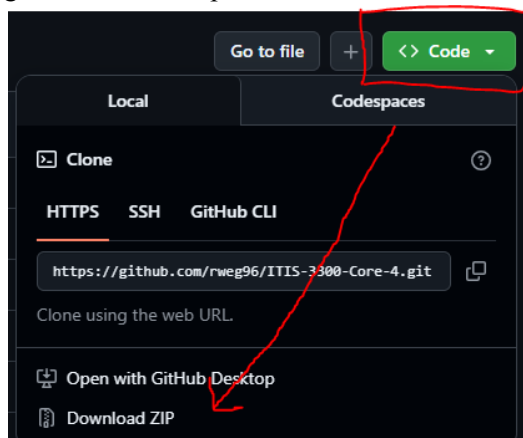
## C) User Manual - Install & Use

### Installation

To install BasketBuddy, there are two options to begin running the website. There are hyperlinks throughout this section to directly access specific relevant pages. Overall, this demonstrates the flow and guides a user that's unfamiliar with GitHub.

**Primary Method:** The website is successfully hosted utilizing GitHub Pages, built into GitHub, so that the user can access it on browsers such as Google Chrome, Edge, and Safari. There is no need to install anything, just access the link [HERE](#).

**Secondary Method:** If the primary link fails, you will need to navigate to the GitHub repository for our group. You can access the repository by clicking [HERE](#). Once on the page, you can click the green "Code" drop down button and download the ZIP file.



Once downloaded, you will see the folder named “**phase-1**” in the **src** directory. This is the folder that contains all of the files for the project and “**BasketBuddy**”

## D) Compile & Run Instructions

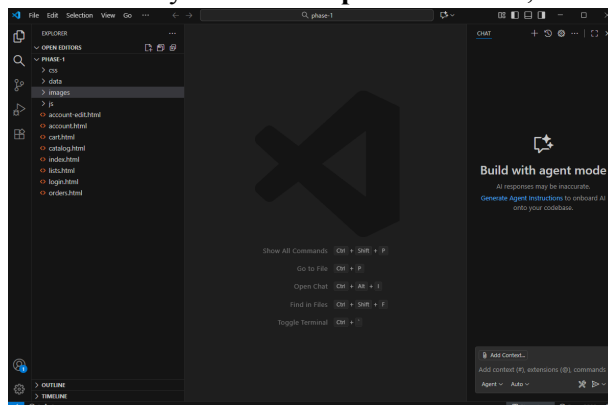
### Running the App

To run “BasketBuddy,” there are two options that coincide with whichever installation method you choose:

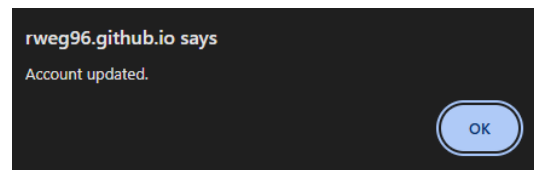
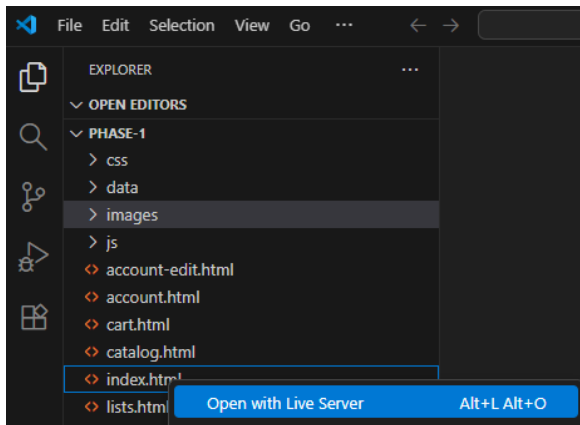
**Primary Method to Run Application:** As previously mentioned above, you may run the webpage without any sort of installation needed. Access it by clicking [HERE](#). This is the easiest and most efficient method of getting to the webpage.

**Secondary Method to Run Application:** If you want to download the ZIP file of our project files, there are just a few more steps involved.

- 1) Download the ZIP file and extract all the contents from the ZIP to a place on your computer you will remember.
- 2) After extracting the ZIP file, open any program that allows you to test code. We as a group recommend using **VS Code** because this is typically what students and faculty recommend in CCI classes at UNCC.
- 3) Once you have opened VS Code, add a folder by selecting in the top left **File** → “**Open Folder**”. Then, navigate to the place you extracted the ZIP file and navigate through the file paths to eventually get to our ZIP folder name “ITIS-3300-Core-4-main” → **src** → “**phase1**”. Once you click the “**phase-1**” folder, click open. Your screen should look like this.



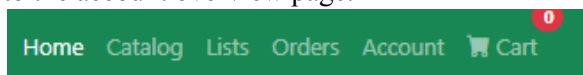
- 4) Now that you have the files you need to access “BasketBuddy”, right click on “**index.html**” and click “**Run With Live Server**”. You can now see our website, provided you have the Live Server extension downloaded in VS Code.



## Using the App

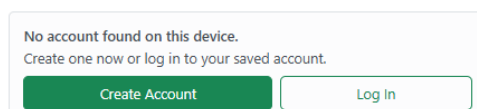
Once everything is set up, navigate to the website either locally or over the internet. Below are the steps of a user's flow as they navigate through BasketBuddy to complete a purchase.

- 1) Once on the home page, you can easily begin shopping. However, a rewards system gives registered users points after ordering, so you are going to want to create an account first. Navigate to the header in the top right and click on the “Account” button. This will direct you to the account overview page.



- 2) Next, click the “Create Account” button. Once you are a registered user, you would click “Log In” as a returning user instead.

### Account Overview



- 3) On the account creation screen, fill out all necessary fields with personal information. Once complete, there will be a pop-up that notifies the user that their account was created.

Edit Account

Full Name

Test1

Email Address

Test@test.com

Password

....

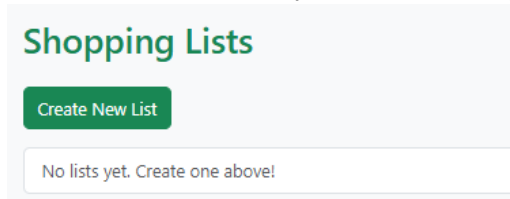
Address

123 Test Street

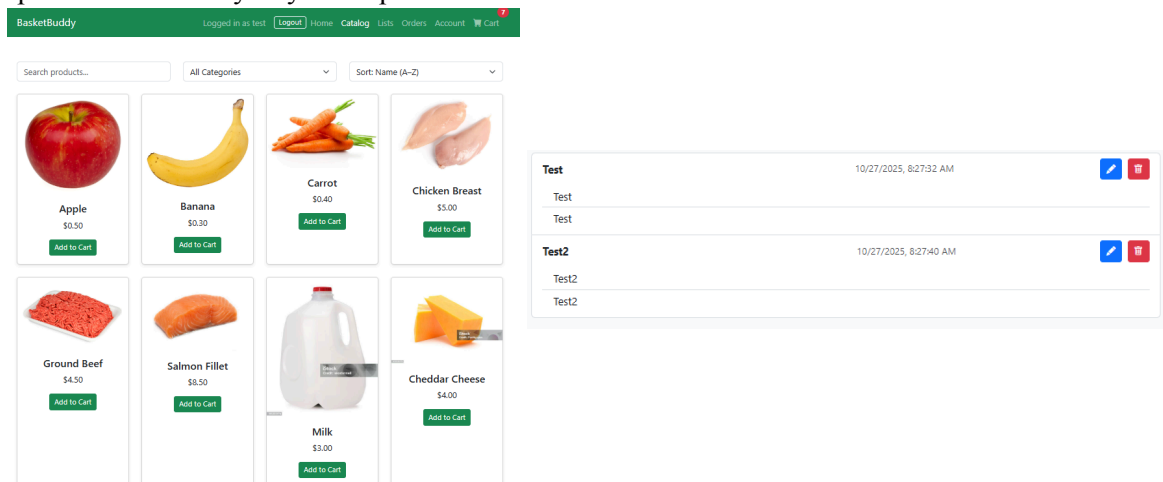
Card Number

1111 2222 3333 4444

- 4) Now that you have created an account, you may take advantage of all features within BasketBuddy. You can start by creating lists, containing your favorite items or weekly essentials. Lists are handy when referencing items that you order often from the catalog.



- 5) Next, you can order groceries! Navigate to the catalog page, displaying all available items for purchase. Simply click “Add to Cart” on the items you want in your order. You can also use the built in search or filtering function to make it easier to navigate the catalog for a particular item. The cart in the top right displays the total amount of items you have in your cart and will update automatically as you shop.



- 6) If you have a cart full of groceries, it's time to check out. By clicking the cart icon in the top right corner, you can see all the items you have added along with a total price. If you have a coupon code, add it! It will automatically apply and adjust the total cost of your order.

#### Your Shopping Cart

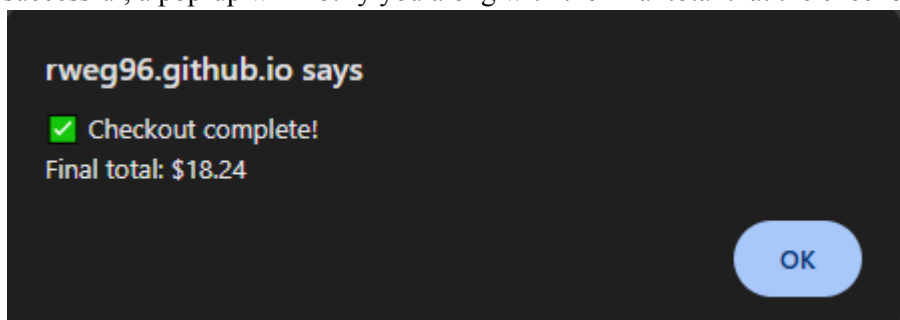
Item	Quantity	Price	Total	Actions
Carrot	3	\$0.40	\$1.20	Remove
Chicken Breast	2	\$5.00	\$10.00	Remove
Cheddar Cheese	2	\$4.00	\$8.00	Remove

Have a coupon? WELCOMES

☒ Coupon "WELCOMES" applied successfully!

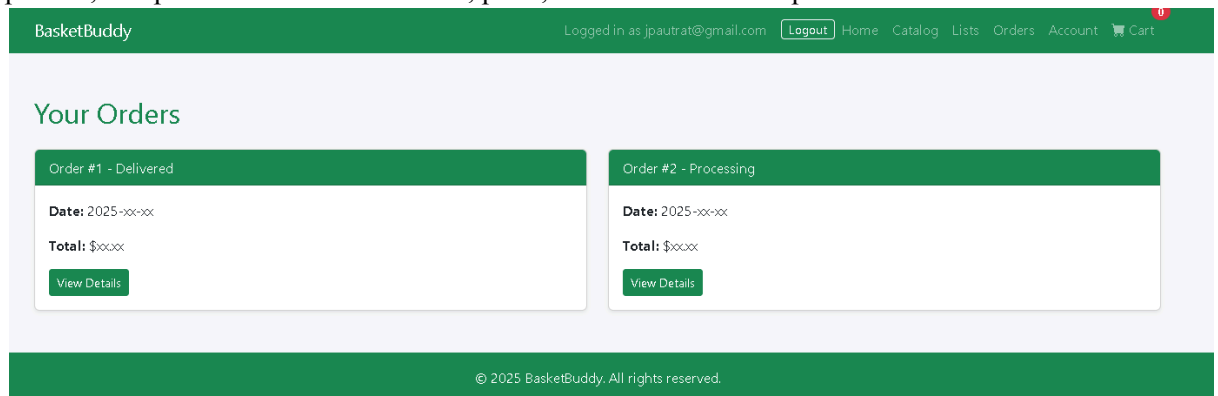
Subtotal: \$0.00  
Discount: -\$0.00  
**Total: \$18.24**

- 7) With payment information already filled out, you may click the “Checkout” button. If successful, a pop-up will notify you along with the final total that the checkout is completed.





- 8) Navigate to the “Orders” page along the top header. This will reflect your orders, past or present, and provides details about date, price, and items that were purchased.



## Test Cases

#	Test Description	Steps	Expected Result	Pass/Fail
1	A user is able to create an account on the website.	The user navigates to account → create account, and fills out the required fields and clicks create account to make it.	Confirmation appears on screen saying “Account Created”	✓
2	Order History Displayed	Add items to the cart, proceed to checkout while logged in, then open the orders tab.	Completed purchases will properly show up under the orders tab.	✗
3	Add Item to Cart	When a user clicks the “add to cart button” the total amount of items is properly displayed on the cart icon.	The cart count increases and the item appears in the Cart page.	✓
4	Update Cart Quantity	On the cart page, clicking the + or - besides the item changes the amount	Quantity changes immediately; subtotal and total refresh automatically.	✓
5	Apply Coupon	On the cart page, when ready to checkout, apply a coupon (SAVE10), and click Apply.	Message: “Coupon ‘Save10’ applied successfully!, and total updates to reflect discount	✓
6	Invalid Coupons	Enter an invalid coupon, example (test1)	There should be a message that says “Invalid Coupon” therefore the total doesn't change	✓
7	Checkout Completed	When you are logged in, create an order and then checkout.	Alert shows final total. Cart clears, rewards points increase, and totals reset to \$0.00.	✓
8	Rewards display for user on homepage	When a user is logged in, go back to the home page and after an order you should see rewards points. (1 point = \$10)	The rewards section is visible with specific point totals based on items purchased	✓

## **E) Reflection**

**What We Did:** For this project, our group was able to successfully implement many of the features we intended to. These features were identified in previous steps and group planning to ensure that we gave the end users a working online grocery shopping solution. We were able to create the following:

- 1) A cart that displays all of the items that the user has added to it.
- 2) Items in the cart are automatically totaled for a total order price, allowing for an easier purchase and checkout process.
- 3) Implemented a rewards system that applies points after a registered user makes a purchase.
- 4) Created an item catalog that displays all of the grocery items for sale.
- 5) Created a search filtering functionality that allows users to search by name (A-Z or Z-A) or price (Low-High or High-Low)
- 6) Created a login/sign up account creation process.

### **What Went Well:**

Overall, our group accomplished exactly what we set out to do from the start of our planning back in August. We envisioned improving the lives or schedules of users who face difficulties with time management to shop for groceries. Our website functions smoothly and uses the local storage to transfer specific data between pages as the user navigates. Our login and account creation process enabled us to implement features that were associated with the account the user created. We were able to keep uniformity between pages when using our UI/UX layout, color scheme, and format of webpages that gave users a better visual experience. Group communication overall was good, and most members took on clear responsibilities and were able to find solutions to issues we were facing during the development phase.

### **Future Improvements:**

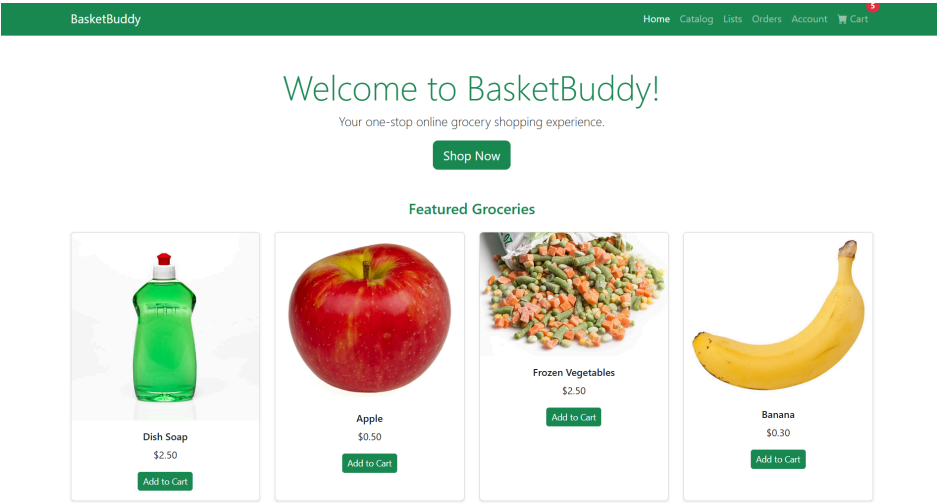
- 1) In the future, we need to ensure that some of the smaller details are changed to make the website better overall.
  - a) Adding more items that encompass the catalog of an entire a grocery store
  - b) Switch from a local storage solution to implementing a database (likely SQL) that can be used to store all the user accounts, groceries, payment history, etc.
  - c) Ensuring that when users are logged in that all their previous orders show up once they have checked out.
  - d) Adding a tracker feature that shows the user the current status of their orders.
  - e) Ensure discounts are properly shown on the checkout screen after applying.
  - f) Adding rewards that people can redeem with the points that they earn by buying groceries.
  - g) Populate the active order on the home page after it is successfully submitted.

## **F.) Member Contribution Table**

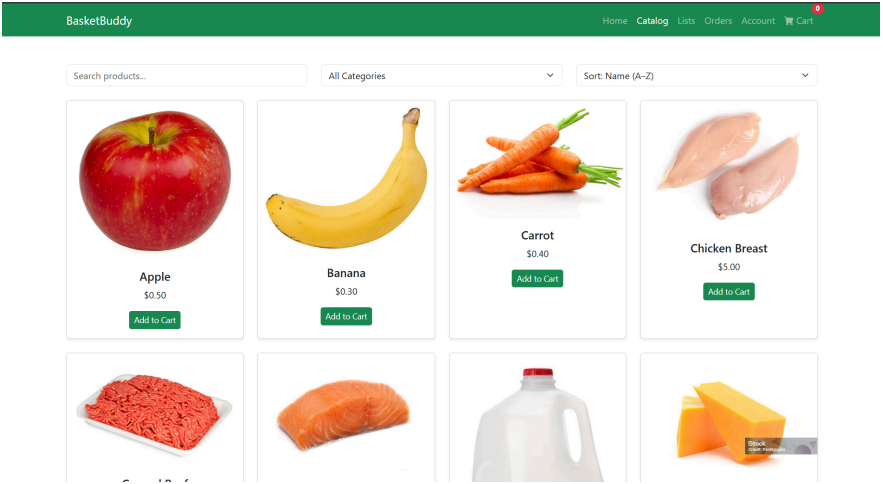
Member Name	Contribution Description	Overall Contribution	Note if applicable
Tyler Niemonen	Created catalog usability, helped build index.html, built the flow chart of overall process flow, worked on updating/changing our functional &NFR requirements helped organize the Deliverable 3 Turn in, installation and how to run portion, completed section C&E, updated meeting notes, created test cases	30%	N/A
Jeff Pautrat	Initial github repo testing for website deployment options as well as sources and developed the database for the app as well as portions of uml design	15%	N/A
Jakia Gary	Implemented the coupons and rewards feature, Contributed to UI interactions involving user reward tracking and discount application	15%	N/A
Carson Carmody	Created the base/initial HTML pages and implemented the shopping list functionality. Created the sequence diagram. Captured UI preview screenshots and formatted deliverable.	20%	N/A
Ryan Wegener	Implemented catalog search, sorting, and filtering functionality, developed account session management, and produced the class diagram with core code snippets for documentation.	20%	N/A

# Page Overview Screens

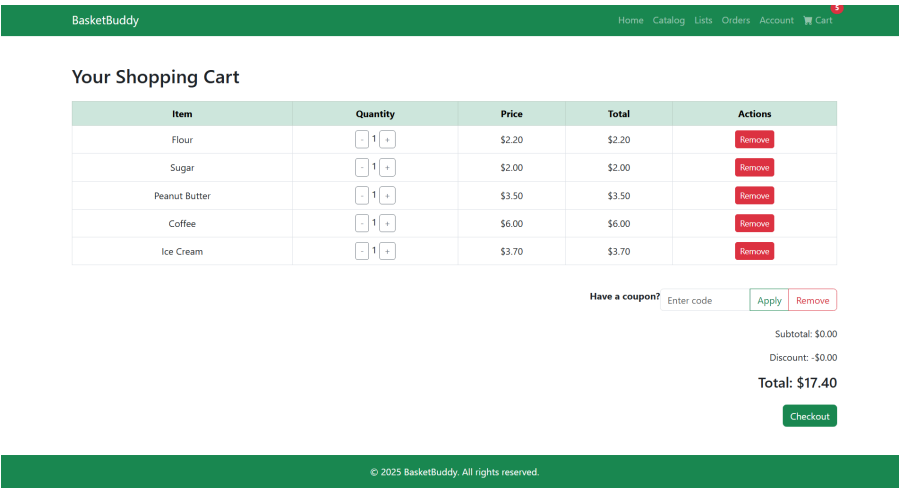
## Home



## Catalog



## Cart



# Login

BasketBuddy

HomeCatalogListsOrdersAccountCart5

Login to BasketBuddy

Email Address

you@example.com

Password

Enter password

Login

Don't have an account? [Create one](#)

© 2025 BasketBuddy. All rights reserved.

# Account Overview

BasketBuddy

Logged in as carson@example.comHomeCatalogListsOrdersAccountCart5

Account Overview

Edit Account

Full Name

Carson Carmody

Email

carson@example.com

Address

9201 University City Blvd, Charlotte, NC 28223

Billing

\*\*\*\* \* 7659

Edit

Delete Account

Deleting your account removes your saved profile from this device.

© 2025 BasketBuddy. All rights reserved.

# Orders

BasketBuddy

HomeCatalogListsOrdersAccountCart5

Your Orders

Order #1 - Delivered

Date: 2025-xx-xx

Total: \$xx.xx

View Details

Order #2 - Processing

Date: 2025-xx-xx

Total: \$xx.xx

View Details

© 2025 BasketBuddy. All rights reserved.