Assignment:  
  
This morning a potential client, reached out to us in wanting to help build a prototype web application for them. They have wholesale product centers throughout the local community, where they offer many wholesale products ranging in categories such as: food, books, cleaning supplies, electronics, kitchen appliances**1**, etc; and they are looking to expand statewide.

Their customers can either go to the product center directly for a purchase on bulk items, or they can ask for delivery2.

This prototype web app should be able to show some content, so you will be responsible for coming up with fake data and interpret how their data structure should be represented;

you can use xml documents, json structures, any kind of free db engine (postgres or mysql), or any other data constructs. You should also think about what a possible competitor would have on their website and come up with any ideas that might be worth of value; feel free to implement those ideas/features, or you can write them down on a piece of paper. You are also responsible for the presentation of the web application, so try to make it look nice, clean, and fast.

Web application:

The prototype web app needs to be able to handle showing off information about inventory items and product centers3. Each inventory item has a name, description, category, # of stock left (total quantity), and the number of product centers that has that inventory item in stock.3

Each product center has a name, location, and a point of sale contact4 for buyers to call/email for making purchases.

There should be search criteria to help users filter what they want, with an option to remove those filters at any time 5.

Entity relationships:

An inventory item can be found in many product centers

A product center is allowed to carry more than one item

-----------------------------------------

Bonus!

Nice to have but not necessary:

-Site can be mobile ready4

-Images showing what an item looks like, with enlarging/slideshow options.6

-Show a history of viewed items from previous visits 7

Wonderful to have but not necessary

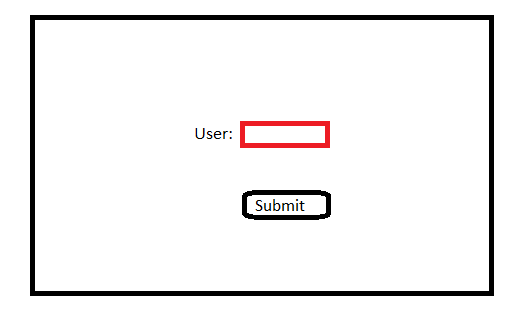
-When clicking the product to view a list of product centers, show a real time map displayed (through google maps or mapquest), which shows viewer location, and the product centers nearest the viewer, along with pins that show the product center's name & location 8

**Goal – High Level**  
build a prototype web application.

Requirement 1 – Log in screen

Some requirements will rely on being able to identify the person browsing the site. There will be a simple log in screen. For this demo we will allow the user to set a user name with no password. This will demonstrate the functionality of user to user data (history, cart, etc) without having to fully implement password security.

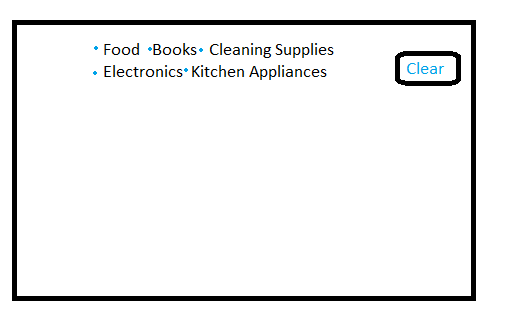
Functionality:  
When the user lands on the homepage, they will be able to enter a user name into a text field. There will be a button to click to set it.  
Once the button is clicked, they will be navigated away from the log in page, and enter the store page.



Requirement 2 – Store Landing Page 1, 5

When the user lands on this page they will be able to see all of the products that are available. The user will be able to click on an interface to allow them to sort by product type. The product types will be: food, books, cleaning supplies, electronics, kitchen appliances.

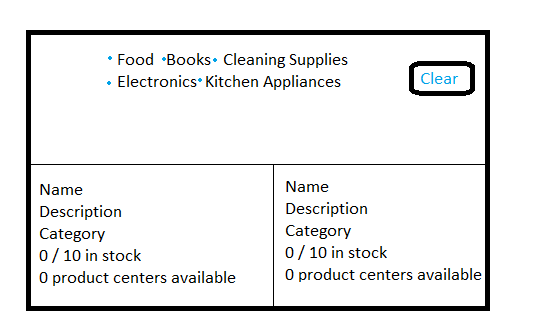
When the user clicks on a type, it will start to sort for only the matching items. The user can then click on more types, to add more to the filter.  
There will be some graphical indication as to what has been selected.  
Once the user clicks a first time, there will be a button that appears for the user to clear the filter.



Requirement 3 – Displaying Items 3, 5

When the user is on the store landing page, for each item available they will see the following information:  
Name, Description, Category, # of stock left / total quantity, # of product centers that have the item.

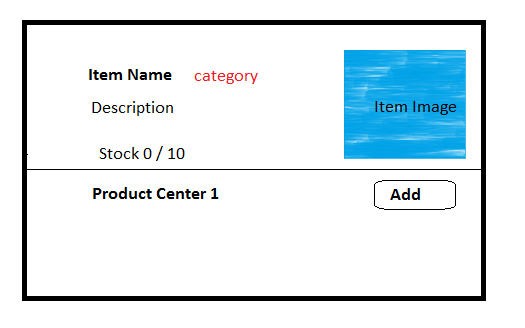
When the user clicks on the item, they will navigate to that items page to see more details.



\*The items should also display an image.

Requirement 4 – Item Page 4,6

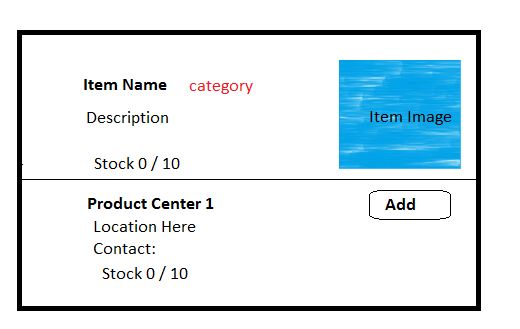
When the user clicks on the item, take them to the page that shows them more about the item.  
This page should show all of the product centers that carry the item, and all of the ones that have them in stock.  
This page will be where the user can add the item to the cart.



Requirement 5 – Display Product Information 4, 6

When the user is viewing the Item Page (Requirement 4) they should also be able to see the details of each product center.  
The information should include the Center name, location and point of sale contact.

If the stock is less than 0, do not allow them to add to cart.

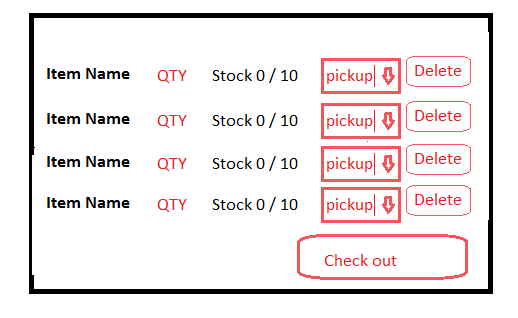


\*The page should change when viewing on mobile device (not implemented)

Requirement 6 – Cart Page

All items added to the cart should be visible on this page. You should be able to see the item name, link to the item, the quantity available, and the quantity you selected. You can click to remove the item and all of its quantities from the cart. For simplicity, the user will not be able to modify the quantity, or remove just 1 of 2 of the same item.

Also, there will be a drop down to allow user to choose pick up, or shipping.



Requirement 7 – log out

There will be a top banner that will allow the user to log out, or go to their cart.

Requirement 8 – Item history

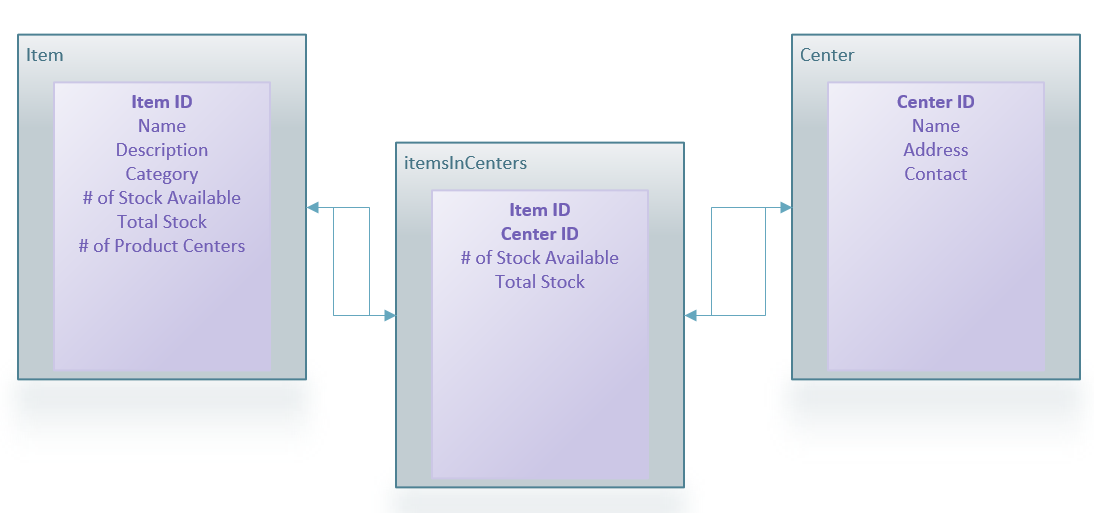
When the user clicks on an item, store it as history so that the user can view it later.

Requirement 9 – Google maps

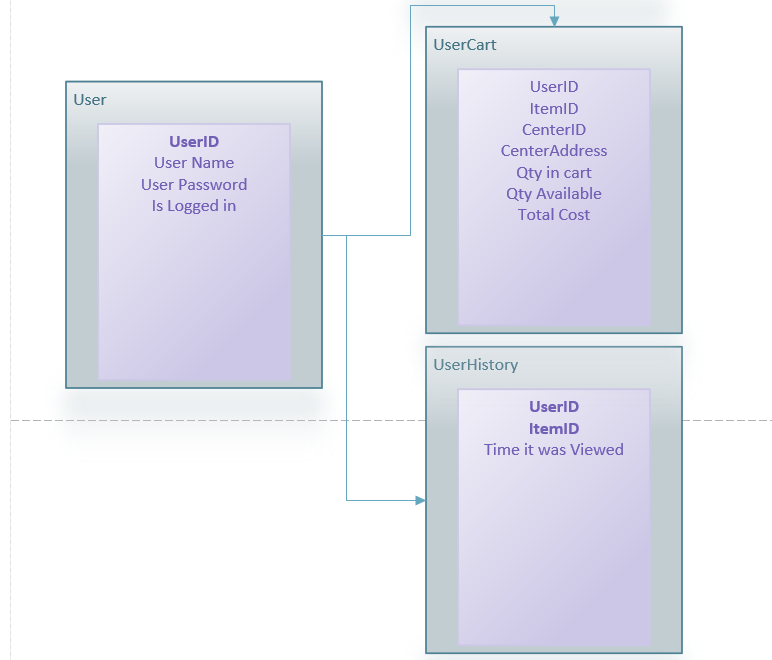
-When clicking the product to view a list of product centers, show a real time map displayed (through google maps or mapquest), which shows viewer location, and the product centers nearest the viewer, along with pins that show the product center's name & location

**Technical Details**

Data Relationships



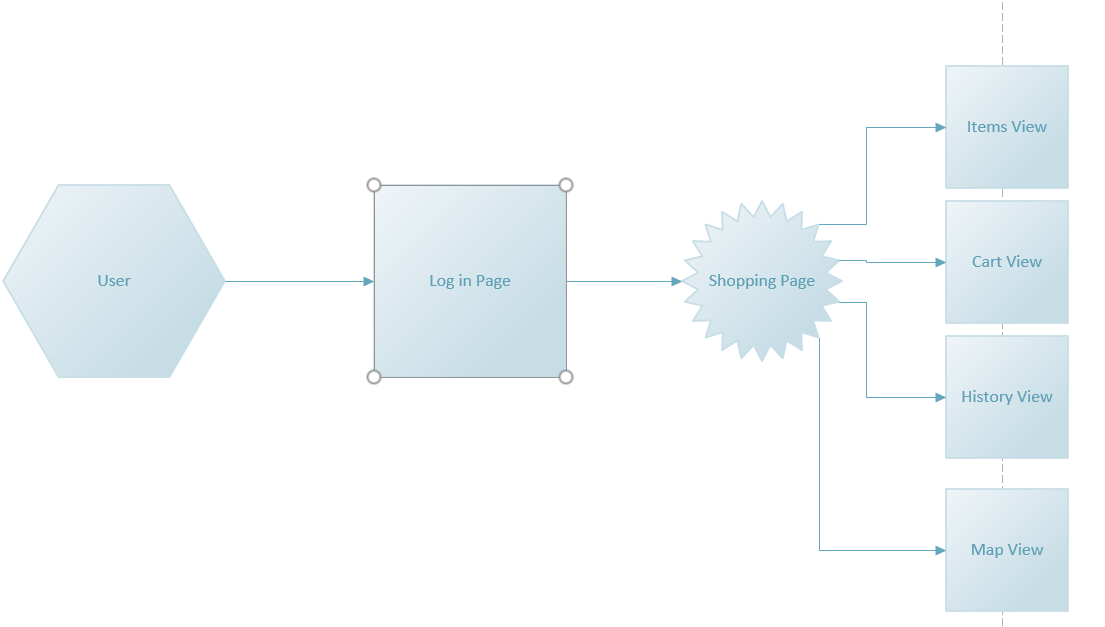
**Item**Item ID - Every item has a primary key of Item ID, this field is a unique identifier to represent that item.  
Name – The items display name  
Description – Text that describes the item  
Category – Could be food, books, cleaning supplies, electronics, or kitchen appliances  
# of stock available – This is the total count of stock of this item from all available centers. To get this number you would select from itemsInCenters passing the itemID and for each result, add the available stock from that center.  
Total Stock – This number is calculated from itemsInCenters where you add the total stock.  
# of product Centers – This is the number of centerIDs returned from itemsInCenters

**Center**  
Center ID – This is a unique identifier for each center  
Name – The display name of the center  
Address – The full address of the center.  
Contact – A text field that could contain a phone number or email string.  
  
**itemsInCenters**  
ItemID – Used to match the Item ID and Center ID  
CenterID – Used to match CenterIDs and item IDs  
# of stock available – The number of stock in that center, for that item.  
Total stock – the max amount of stock the store could have for that item.  
  


**User**  
User ID – Unique Identifier for the user  
User Name – The account name of the user  
User Password – Encryted password string  
Is Logged In – Boolean for if the user is logged in or not  
  
**UserHistory**  
UserID – UserID from User Table  
ItemID –Item ID from item table  
Time it was viewed – the time and date of when the item was viewed by the user.

**UserCart**  
User ID – From user table  
Item ID – From Item Table  
Center ID – from Center table  
Center Address – From center table  
Qty in cart – How many of the item the user has.  
Qty Available - # of stock available from ItemsInCenter table

User Interface Page Relationship



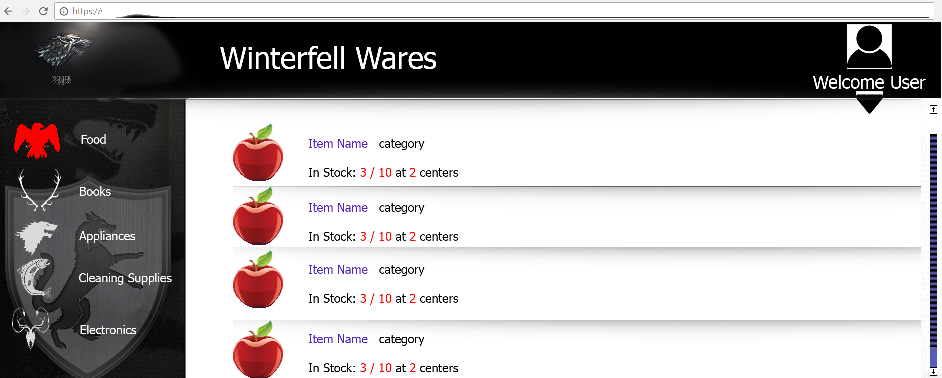
A user lands on the Login Page which handles logging in and redirects the user to the shopping page.  
The shopping page is where everything the user does on the site happens.  
Item View – Here the user can see all items available and filter them  
Cart View – Here the user can see all of the items in the cart  
Histtory View – Here the user can see the item history  
Map View – Here the user can see a google maps of all of the stores near by that match the stores name.

For mobile devices, another instance of shopping page will exist for mobile layout.

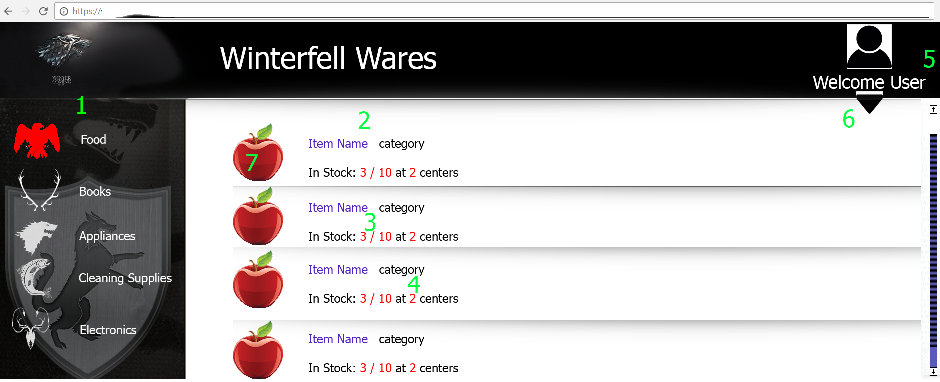
**Functional Details**

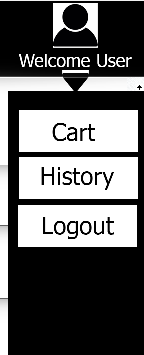
Item Shop Screen

This is what the user will see once they are logged in. (I took game of thrones art from google and do not own the rights to them, design/layout was done by me).

****

Here we mark the features and the functionality.



1. When the user hovers over the Food icon, it changes color. When the icon is selected it becomes red. This wll sort the results to all be food.
2. When the user clicks item name it will take them to the items page
3. List the number of available units and the total stock
4. List how many centers this can be purchased at
5. Show the users login name where you see ‘user’
6. When the user hovers over the drop down arrow, change color. When clicked, bring up a menu to show the log out button, the view history button and the cart button  
   
7. Hovering over an image should enlarge it for the user to see.

**‘Final Product’**

Here I will list all of the requirements again, but I will explain where to see the feature implemented

Requirement 1 – Log in screen

User can log in on this screen. Type in a name and click the button. This name will have a cart and history tied to it.



User.json contains the name of the user logged in.

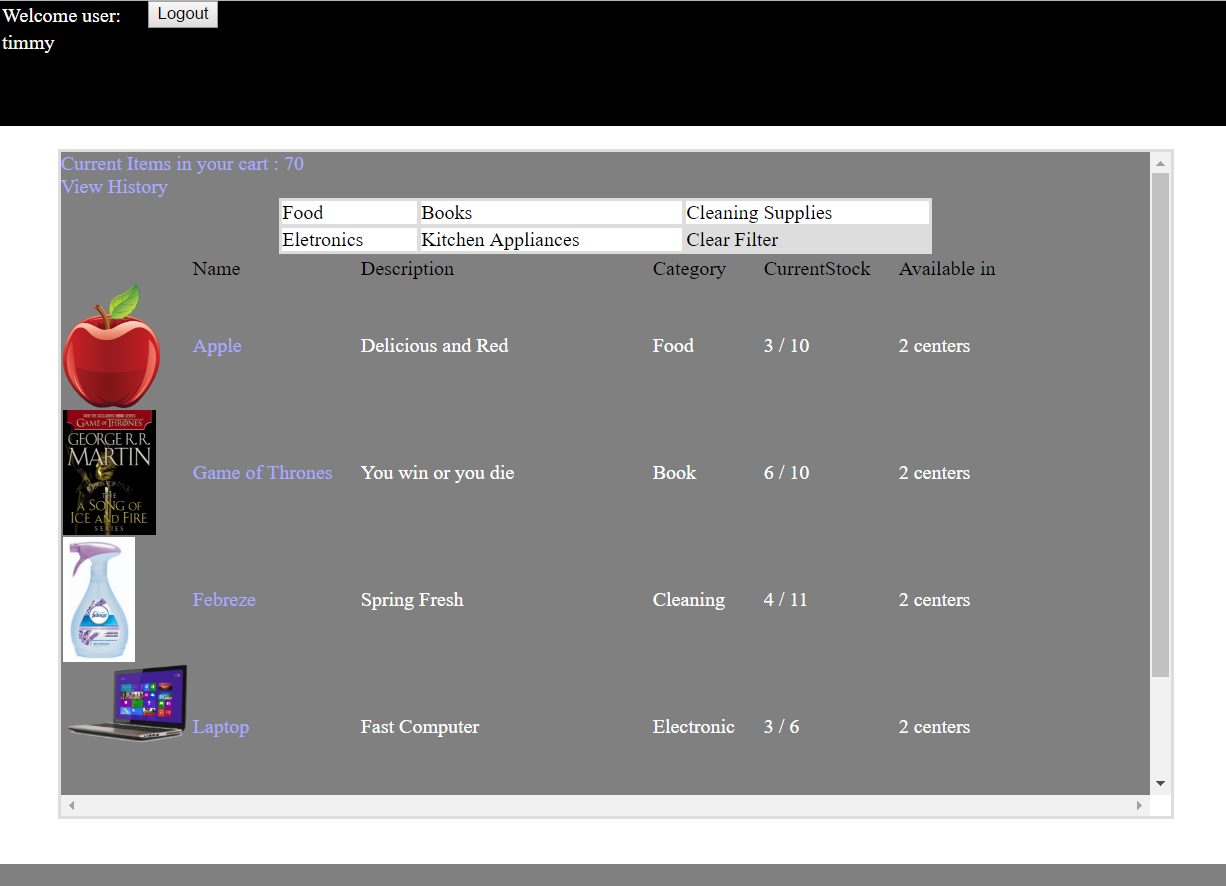


For this demo, only 1 user can exist logged in at a time, as it doesn’t use any cookies or sessions to determine if you are user A vs user B , just the name from the entry screen.

Requirement 2 – Store Landing Page 1, 5

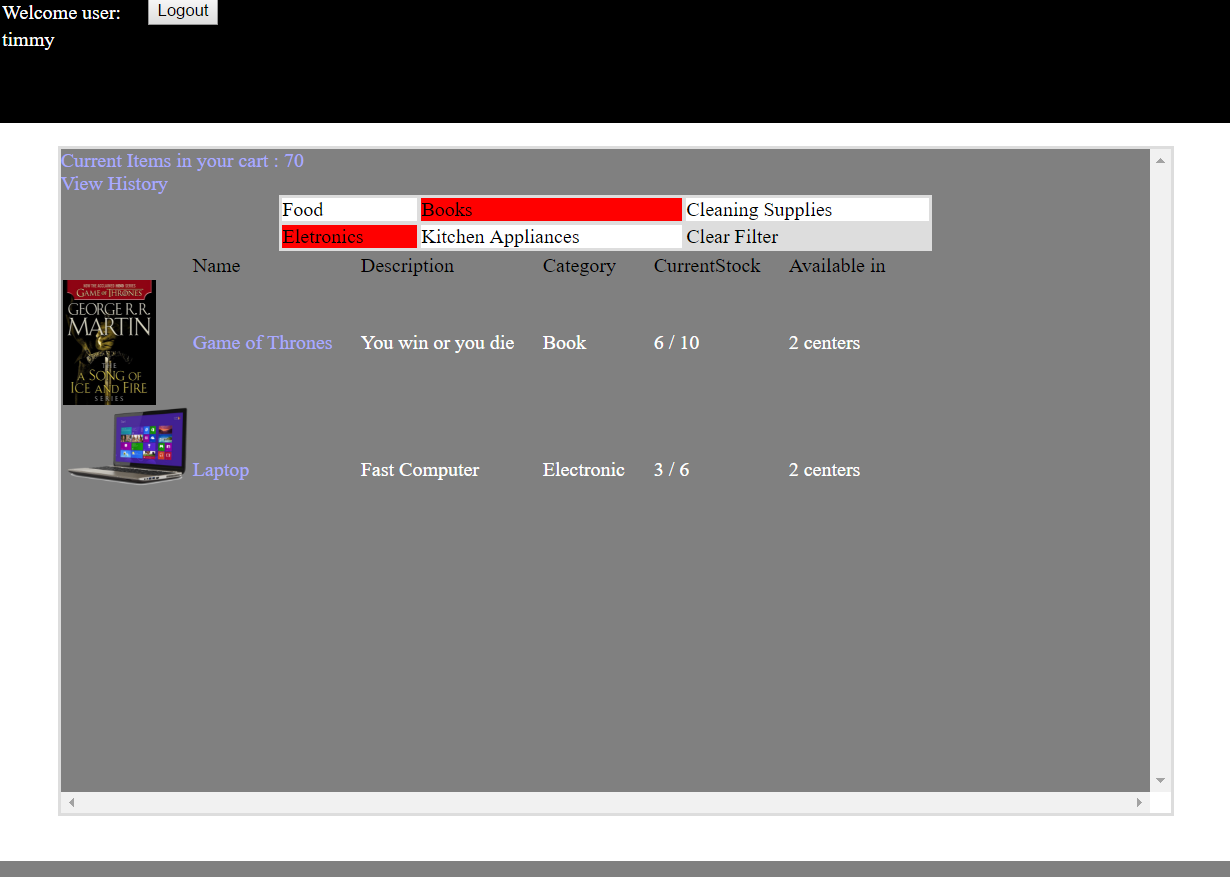
Please see requirement 3 below.

Requirement 3 – Displaying Items 3, 5



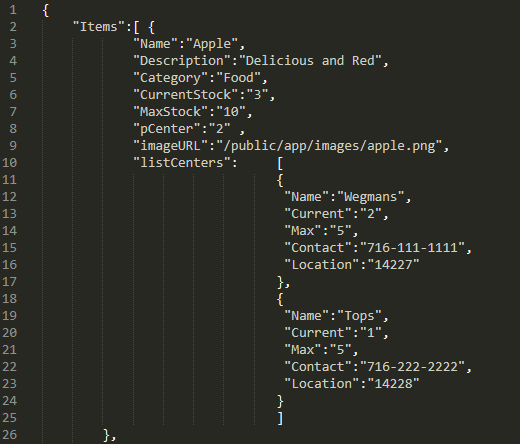
Here the user can see the items available. For example, apple and game of thrones.

When the user clicks to filter, they can click 1 or many options.



The user can click ‘clear filter’ any time to remove the filters. Or, clicking again on one of the boxes will un-select it.

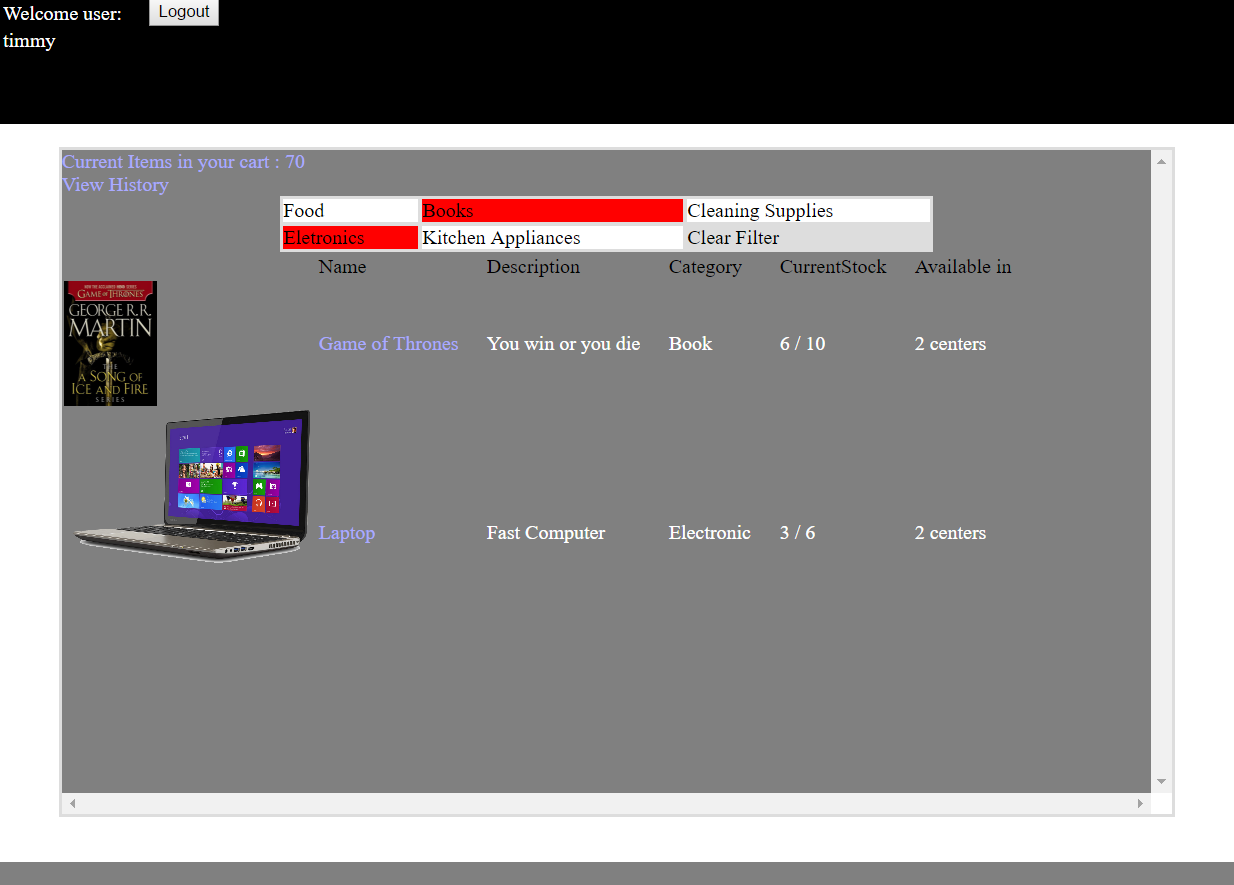
Item.json contains the data for each item. Every item has many stores.



If I spent more time I would have listed just the name under listCenters, and then had another .json source for all stores, with the name of each item in the store.

Since this demo only has about 1 item per store, I did this for the sake of time.

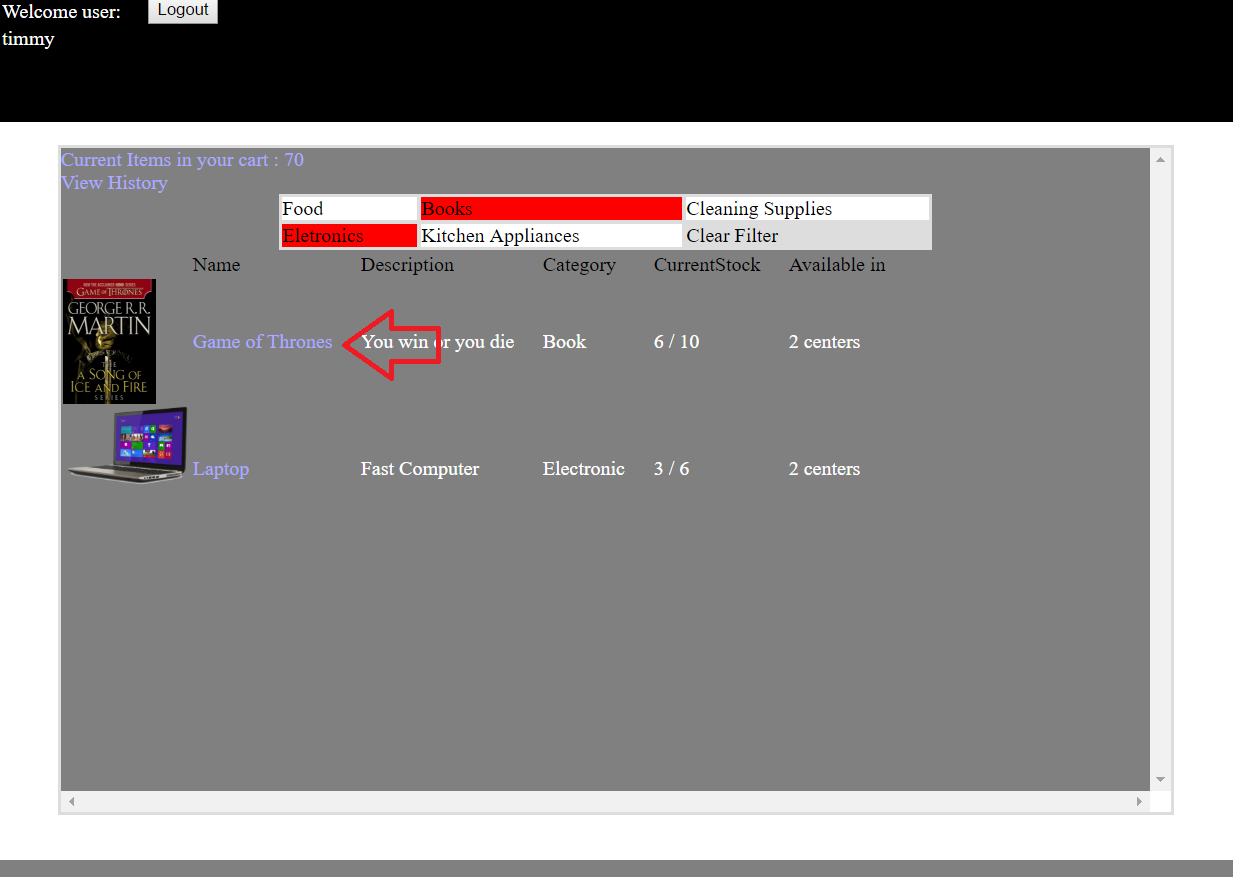
When a user mouses over an image, it gets bigger.



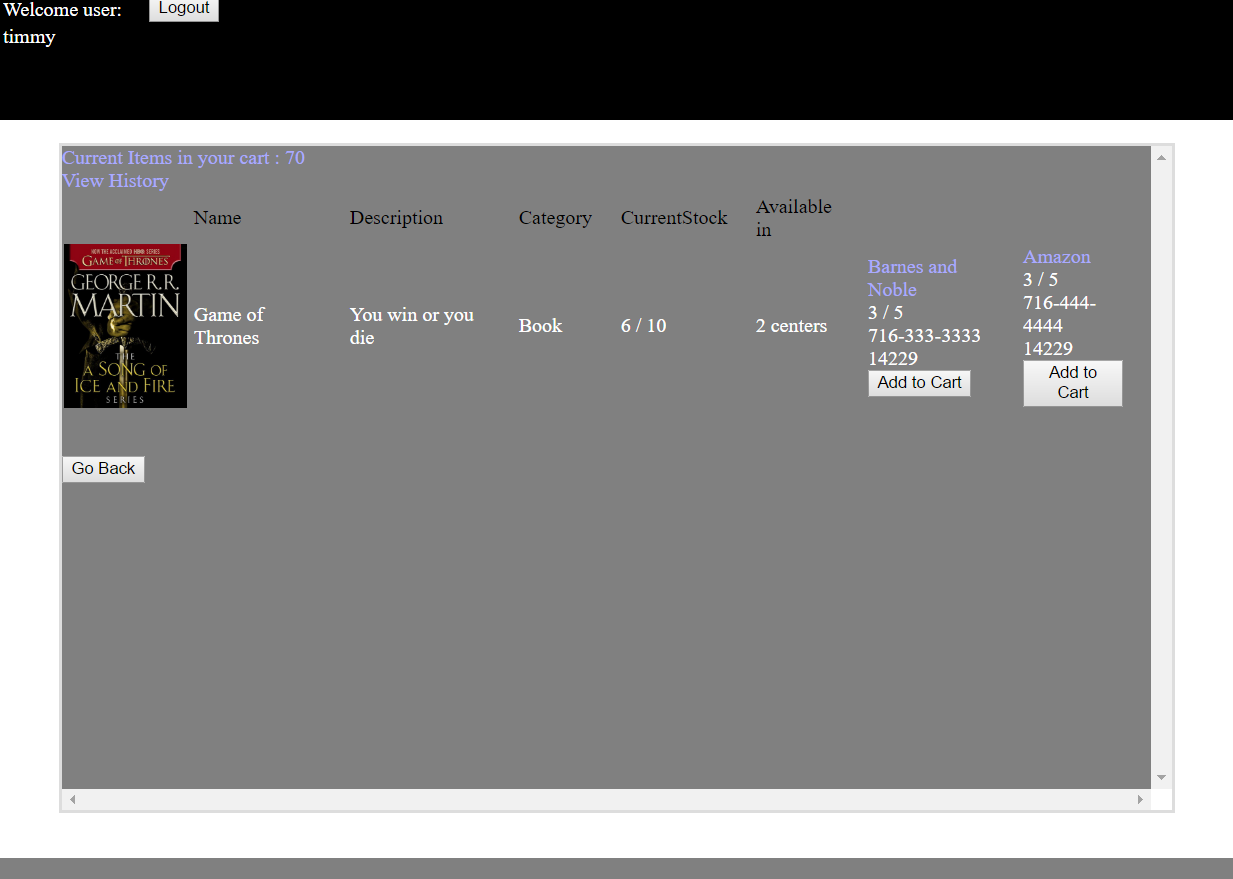
Requirement 4 – Item Page 4,6

Please see requirement 5 below.

Requirement 5 – Display Product Information 4, 6



Clicking on the blue name of the item you will navigate to the item screen.

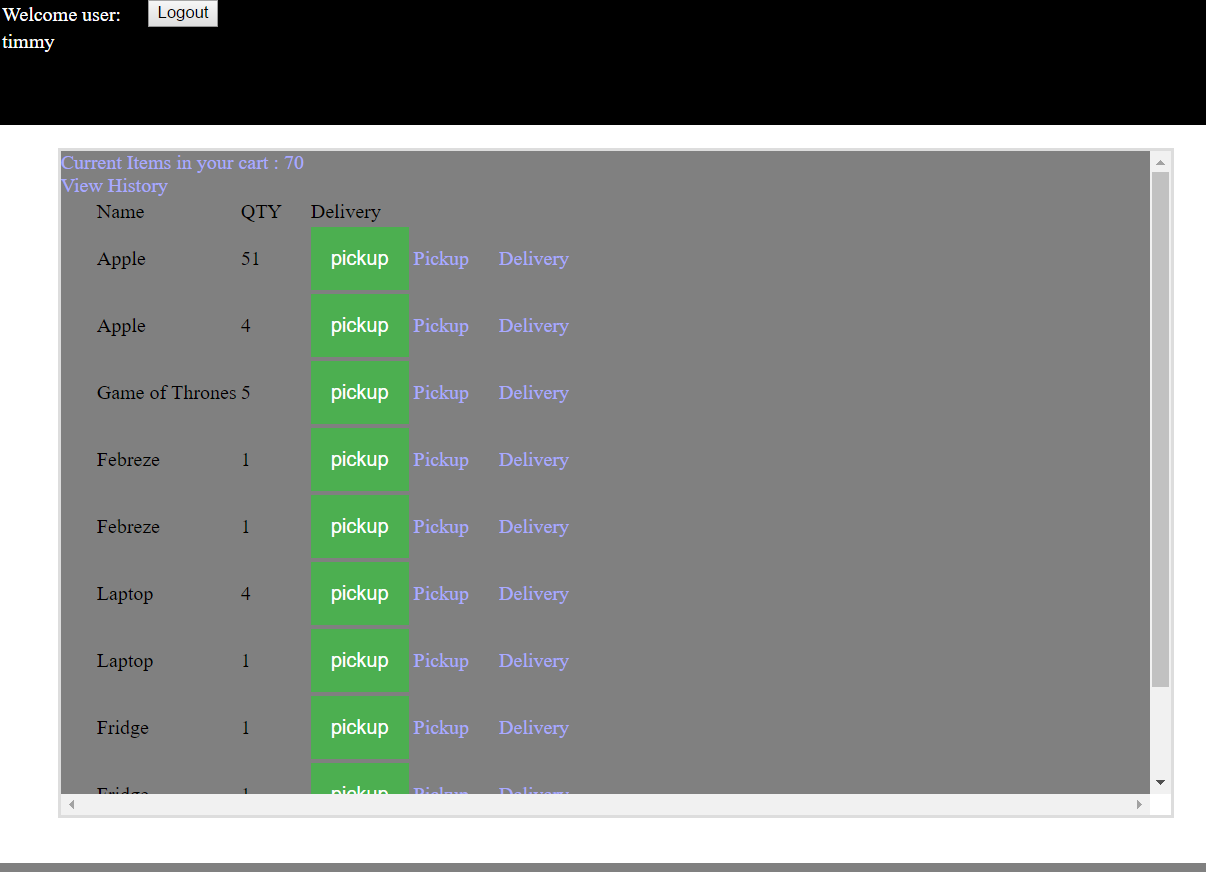


Here you can add an item to your cart. This will add the item and the location you chose. In this example, barnes and noble or amazon.

For an items contact I used a phone number. For location I used a zipcode. Both are fake.

Your cart will increase in quantity, but the page displayed quantity will not change. You can effectively add indefinitely.

Requirement 6 – Cart Page



Here we can see the item, and the QTY in your cart. IF you click ‘pickup’ or ‘delivery’ the green box will change. This screen has no check out page so this is the end of the road for transactions.

Apple appears twice because it was added to the cart from multiple locations. The location should be included on this page.

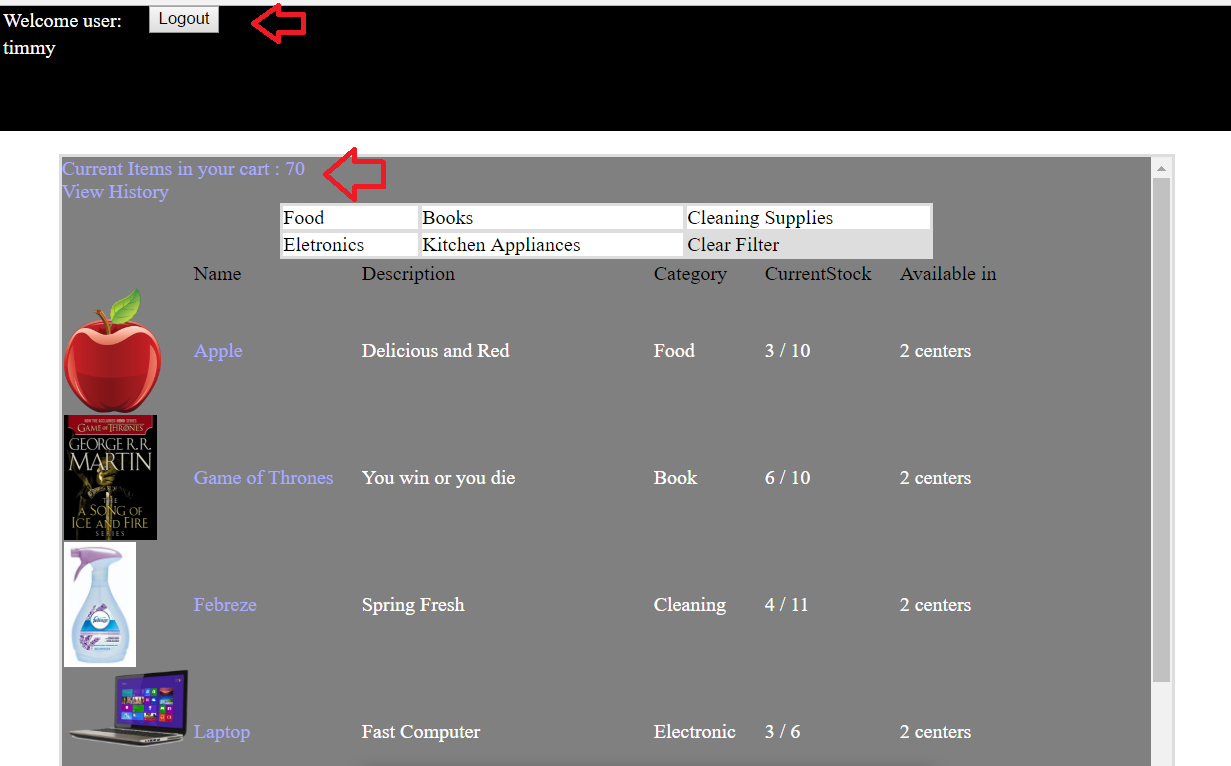
This page is missing the option to delete an item.  
It should also let you click on the item to navigate back to that items page but it does not (when viewing user history, it does do this).



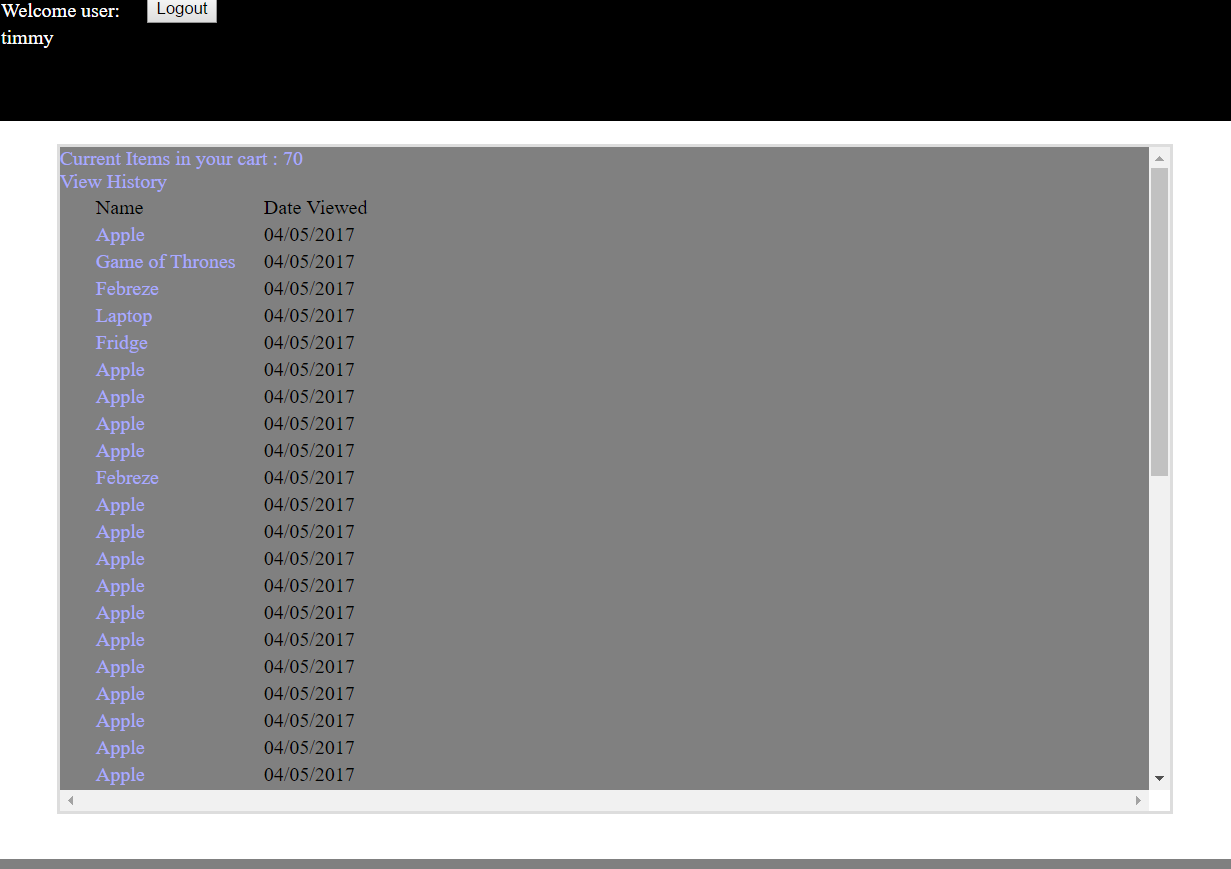
userCart.json uses uname to store the users name and track the history for every user.

Requirement 7 – log out / view cart buttons

Both options are presented in this screen shot



Requirement 8 – Item history 8

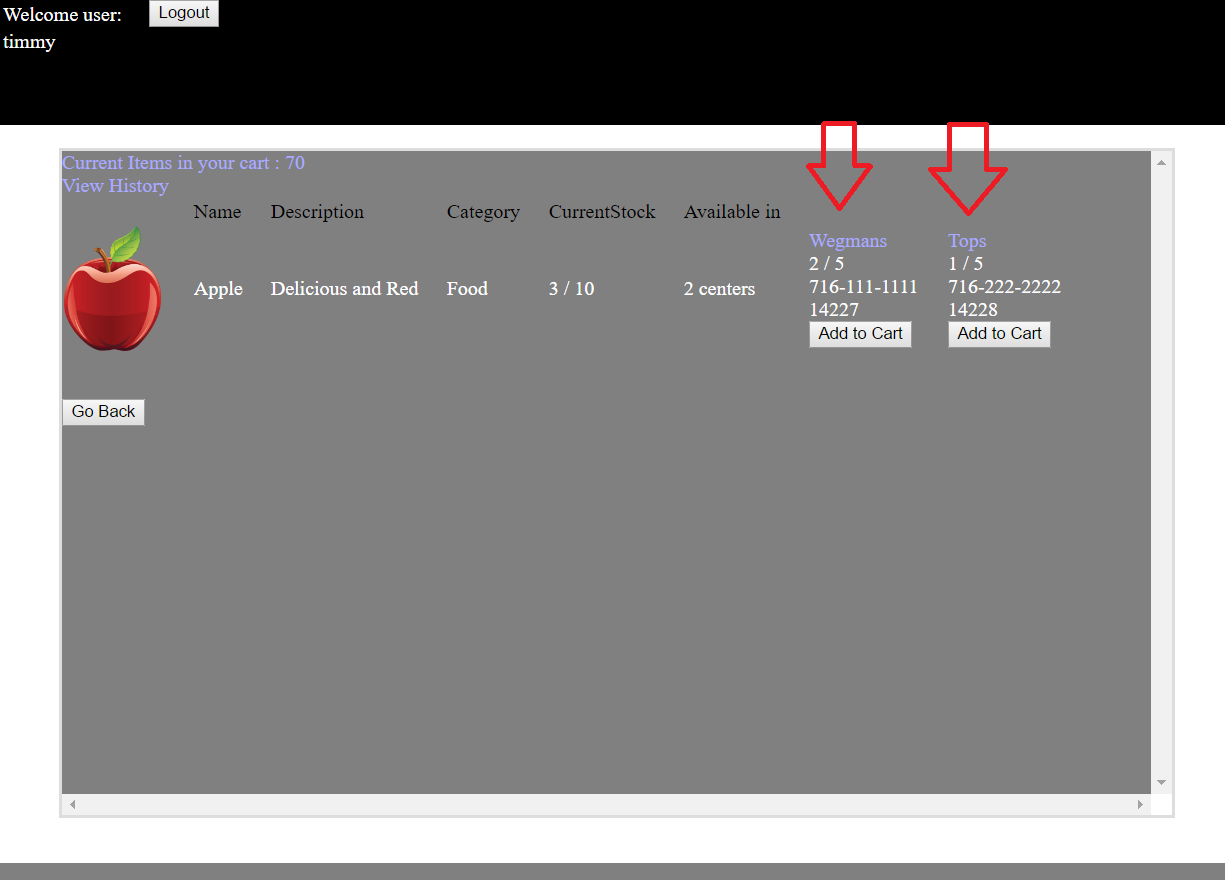


Every time a user clicks an item in the shop, it logs it here. If you click on the name it takes you to that item page. When you navigate to the item page from history , it does not record that as a page visit.

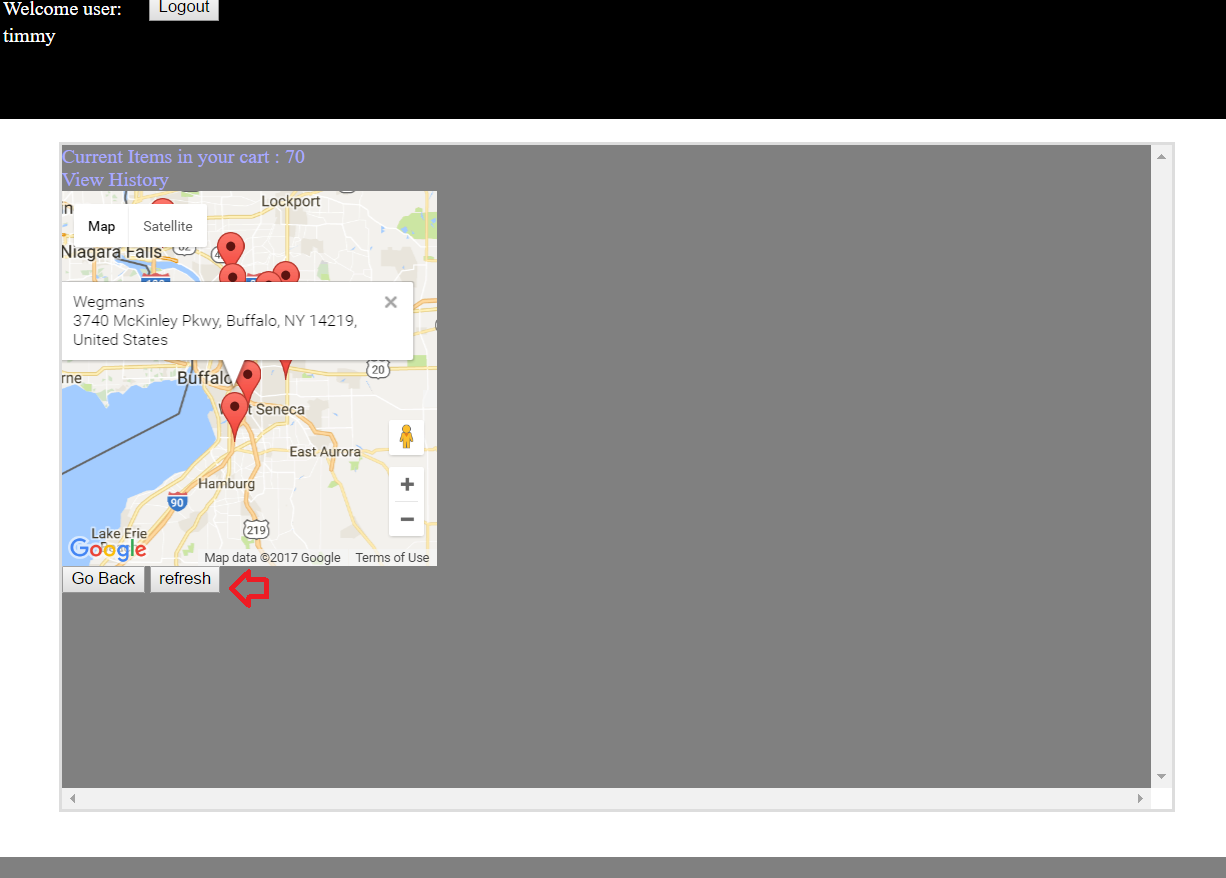


userHistory.json ties the history data to the uname value.

Requirement 9 – Google maps



When you click on a location from the item view page, you will be taken to the map view of that place nearby.



Unfortunately, you need to click Refresh to display the map properly.  
It shows all the pins near the center of buffalo and has no option to set your location.

Missing Features

* There is no mobile CSS done for the site. 4
* When a user clicks to add something to their cart, it doesn’t subtract the items stock
* When the user views their cart page, they cannot delete an item there.
* When the user views a location on google maps, they cannot enter their location
* When viewing the cart screen it doesn’t show the items location, or let the user click the item to view its page from there.