Final Project

Title

Storefront

Course

CSC-17B

Section

40502

Date

June 12th, 2020

Author

Brandon Sanchez

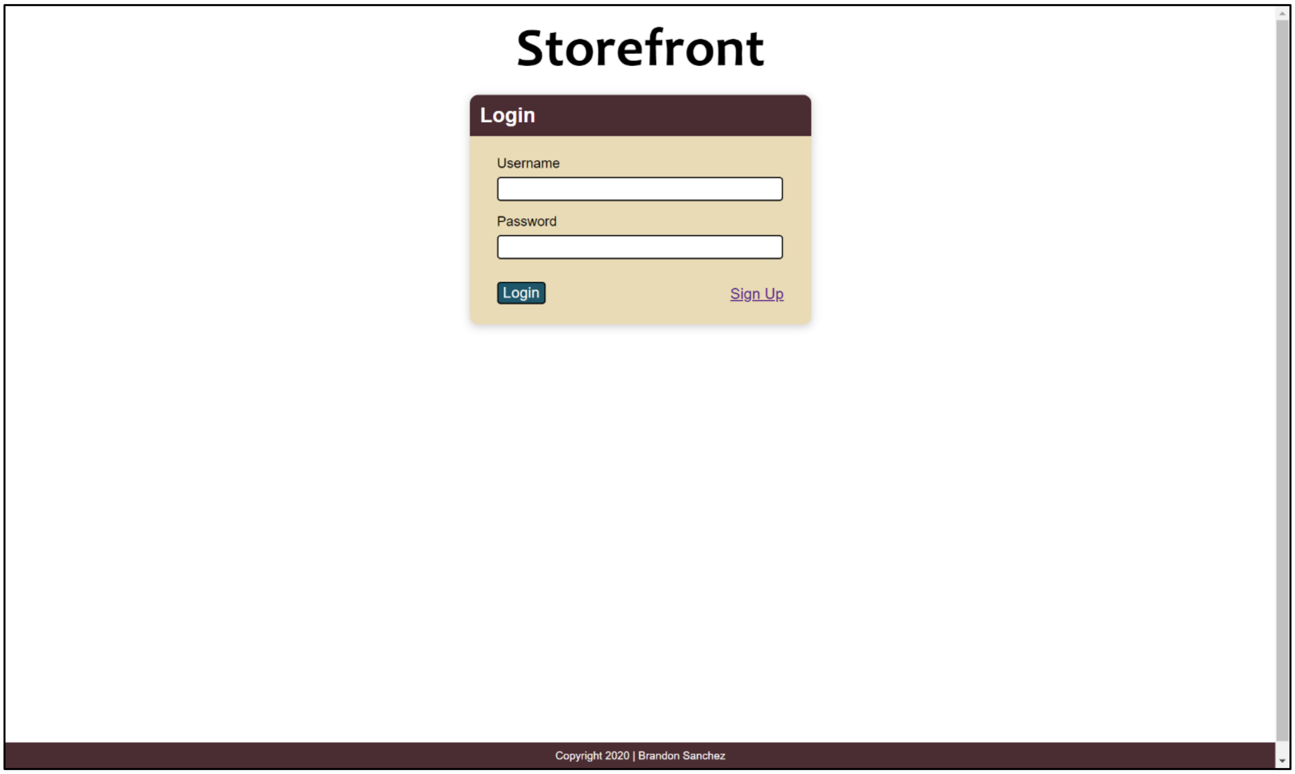
Introduction

This storefront web program allows store owners to upload their inventory to sell to other users who are looking to buy goods. Javascript is used to allow users to change static text to input forms so owners can edit their inventory. PHP/MySQL is utilized to store login information, user orders, and store inventory.

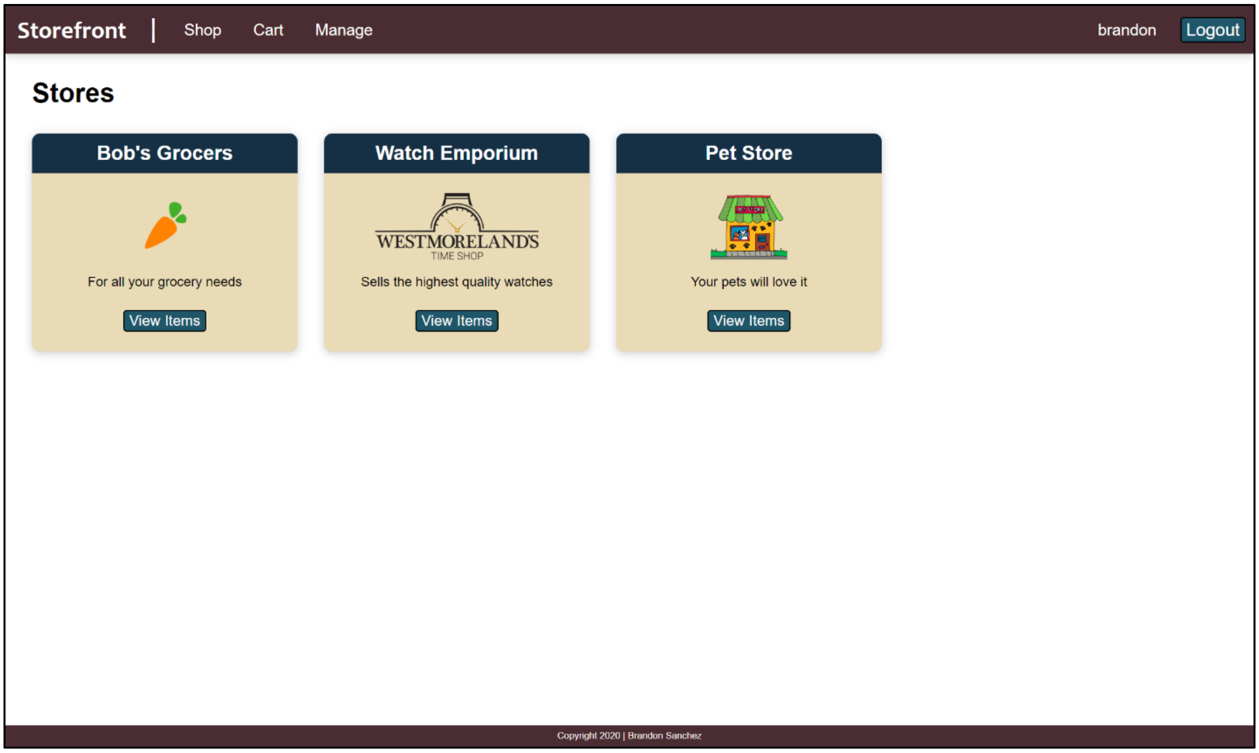
How to Run

If utilizing the serverside database, everything should run fine with the MySQL and Apache servers turned on in Xampp. If utilizing a local database, run the Storefront.sql statement in your GUI of choice and change the connection information in dbh_inc.php to the correct information.

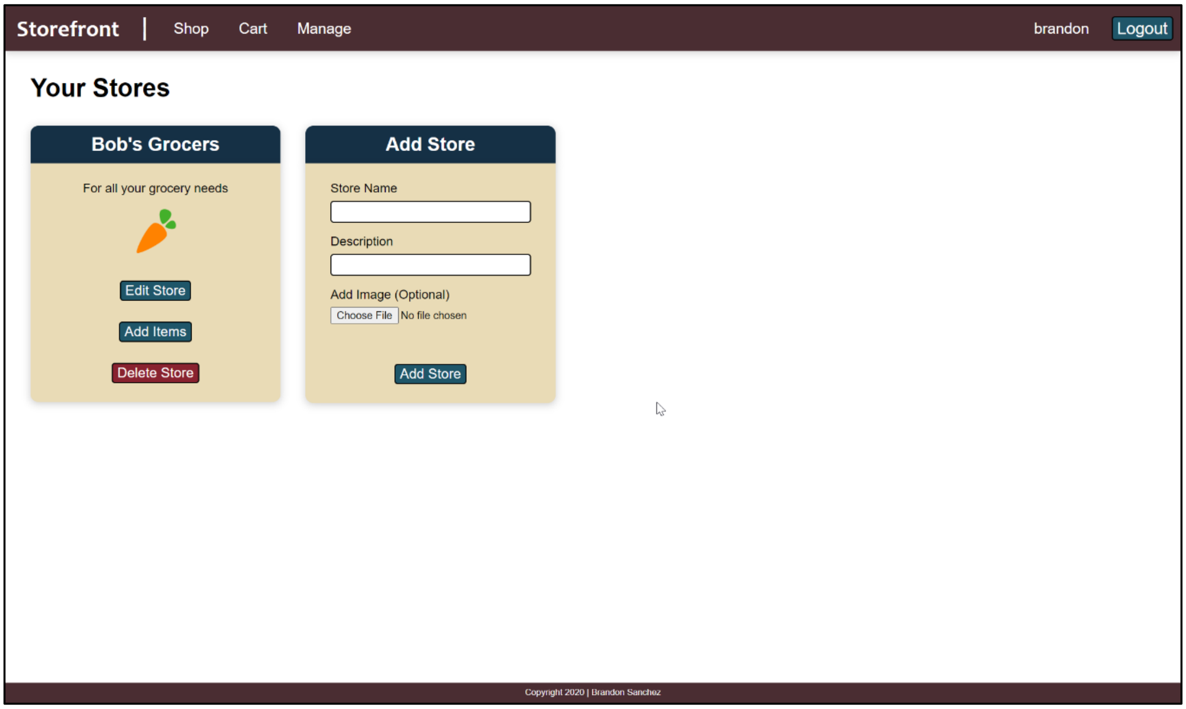
Sample IO



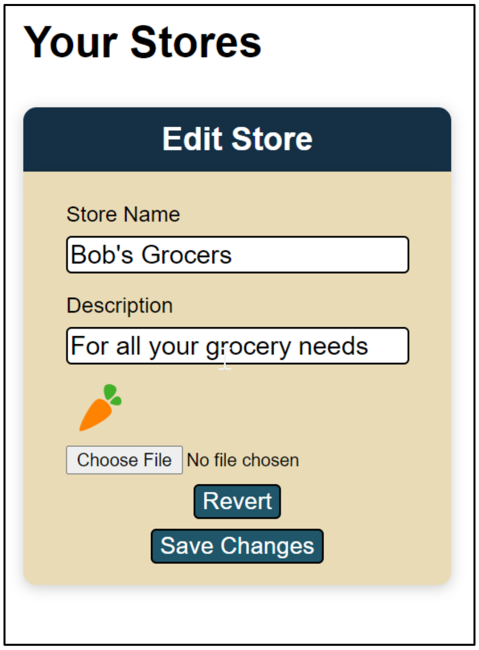
Upon entry to the website, the user is greeted with a login screen. If they have not made an account prior, they can click the sign-up button the create their account.



Once the user logs in, they are greeted with all the stores uploaded to the database by owners. From now on, the header is the same for all the pages. Their username and the logout button are displayed in the top right.



If the user selects manage at the top, they are taken to a list of their stores with the ability to add more stores or edit their existing stores.



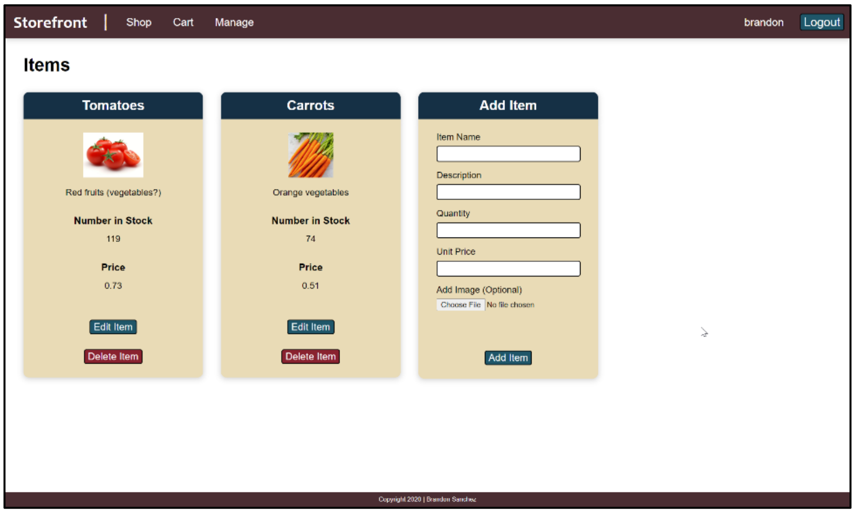
Your Stores

Edit Store

Store Name: Bob's Grocers

Description: For all your grocery needs

If the user presses edit store, then they can edit anything about their store entry.



Items

Tomatoes

Red fruits (vegetables?)

Number in Stock: 119

Price: 0.73

Carrots

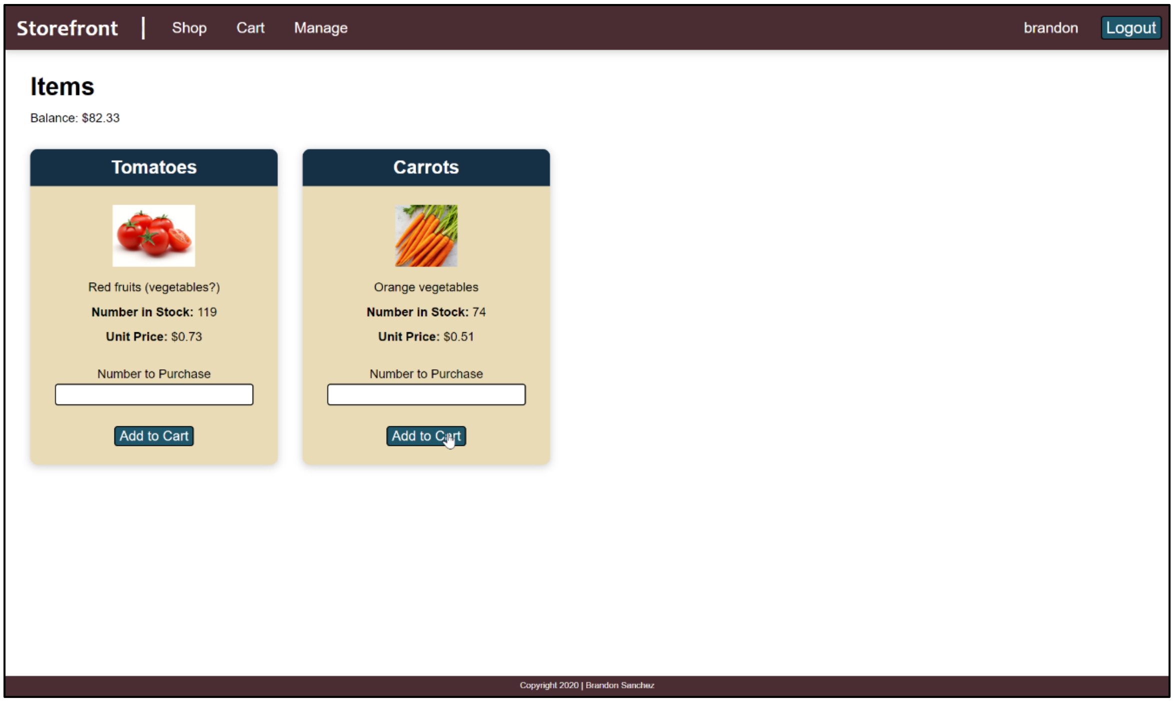
Orange vegetables

Number in Stock: 74

Price: 0.51

Copyright 2020 | Brandon Sanchez

If the user presses add items on their store card, then they can view all the items currently in their store, add items, and edit existing items.



Items

Balance: $82.33

Tomatoes

Red fruits (vegetables?)

Number in Stock: 119

Unit Price: $0.73

Number to Purchase

Carrots

Orange vegetables

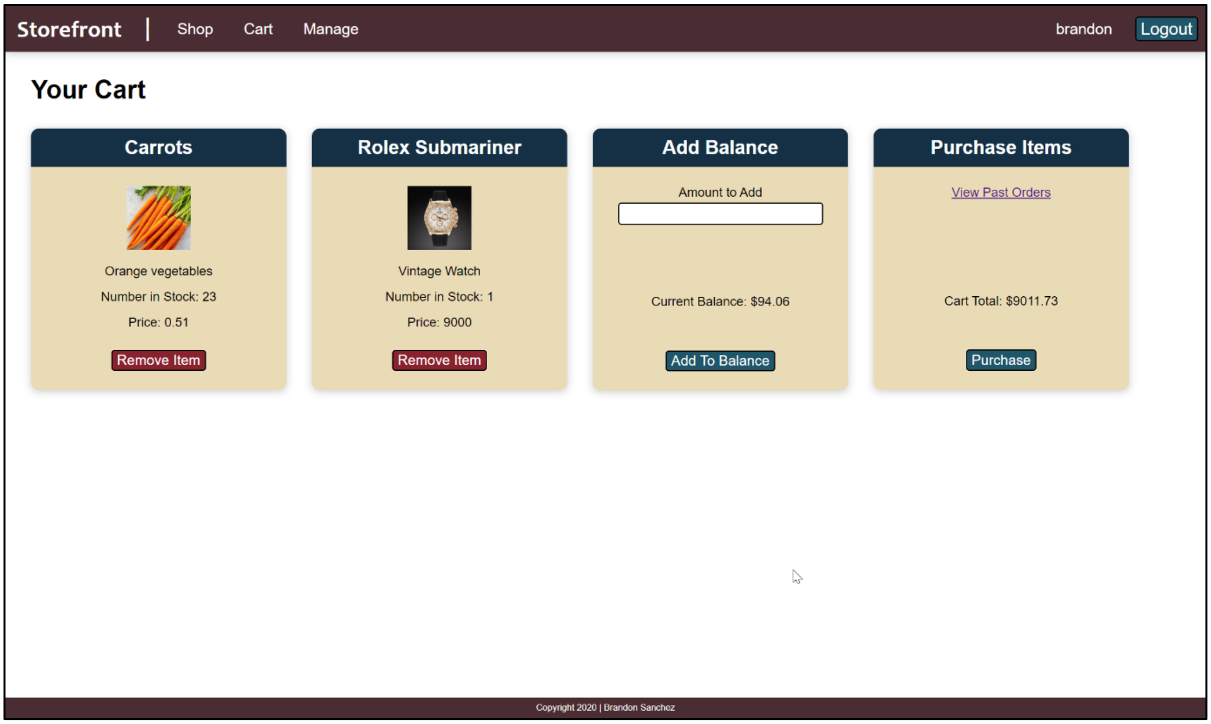
Number in Stock: 74

Unit Price: $0.51

Number to Purchase

Copyright 2020 | Brandon Sanchez

Upon viewing the items in a store, the user can type the number of items to add to their cart. Also, it shows their balance in the top left.



Your Cart

Carrots

Orange vegetables

Number in Stock: 23

Price: 0.51

Rolex Submariner

Vintage Watch

Number in Stock: 1

Price: 9000

Add Balance

Amount to Add

Current Balance: $94.06

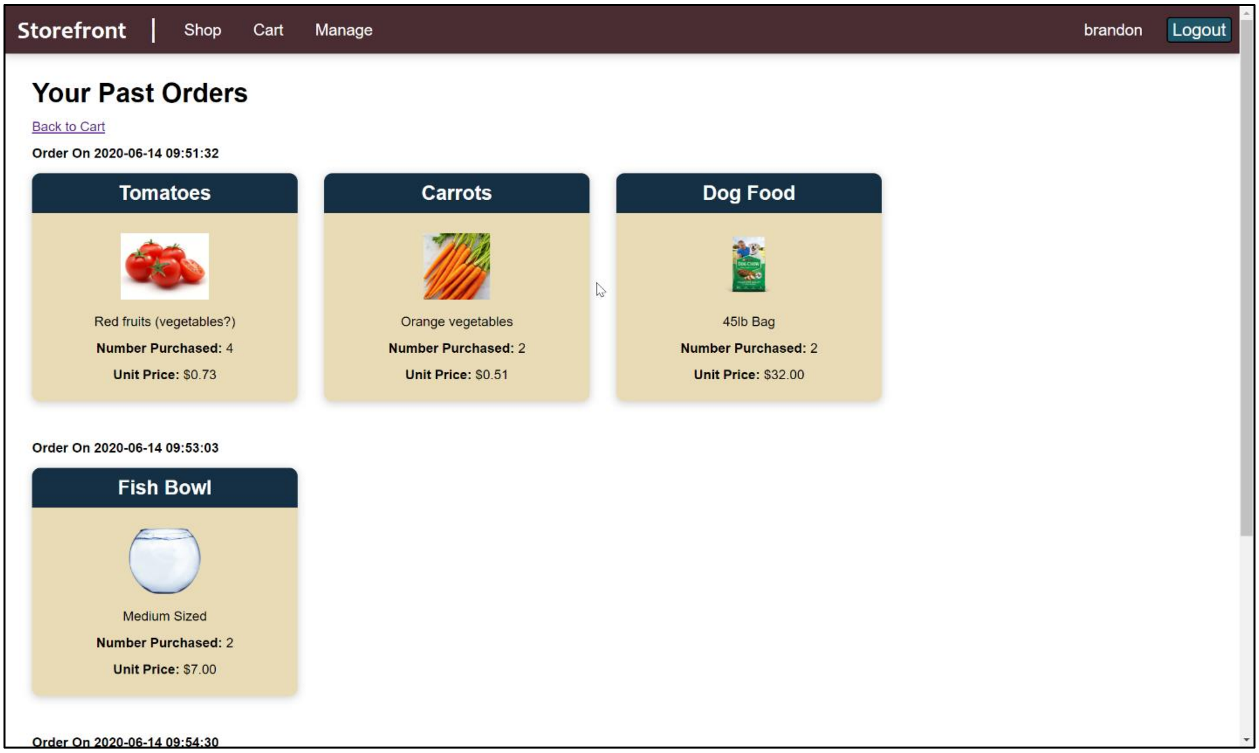
Purchase Items

View Past Orders

Cart Total: $9011.73

Copyright 2020 | Brandon Sanchez

Once the user is done shopping, they can view their cart but clicking the cart link at the top. Here users can remove items from their cart, add money to their balance, and purchase the items in their cart.



Your Past Orders

Back to Cart

Order On 2020-06-14 09:51:32

Tomatoes

Red fruits (vegetables?)

Number Purchased: 4

Unit Price: $0.73

Carrots

Orange vegetables

Number Purchased: 2

Unit Price: $0.51

Dog Food

45lb Bag

Number Purchased: 2

Unit Price: $32.00

Order On 2020-06-14 09:53:03

Fish Bowl

Medium Sized

Number Purchased: 2

Unit Price: $7.00

Order On 2020-06-14 09:54:30

Clicking on view past orders shows all the user’s past orders and the dates they placed them on. Users can place this order whenever they want to.

Checklist

1. **MVC**
   The controller in this case is all the include .php files that manipulate the database and send it to the web pages. The view is all the normal .php files that show things on the screen. The model is the MySQL databases.
2. **Objects**
   Each include.php is stored in their own file to help compartmentalize the code through objects.
3. **Storage**
   When a user uploads an image of their store or items, it is stored on the server in a file called uploads and the directory name is stored in the database. The code for this is found in addstore_inc.php and additem_inc.php.
4. **SQL**
   Almost all the code in the includes file uses php to connect to the database. The relational chart for the database is at the bottom of this document.
5. **Form Validation**
   The login page uses form validation so users can only enter letters and numbers in for their username. The regular expression can be found at signup_inc.php line 16.
6. **Login**

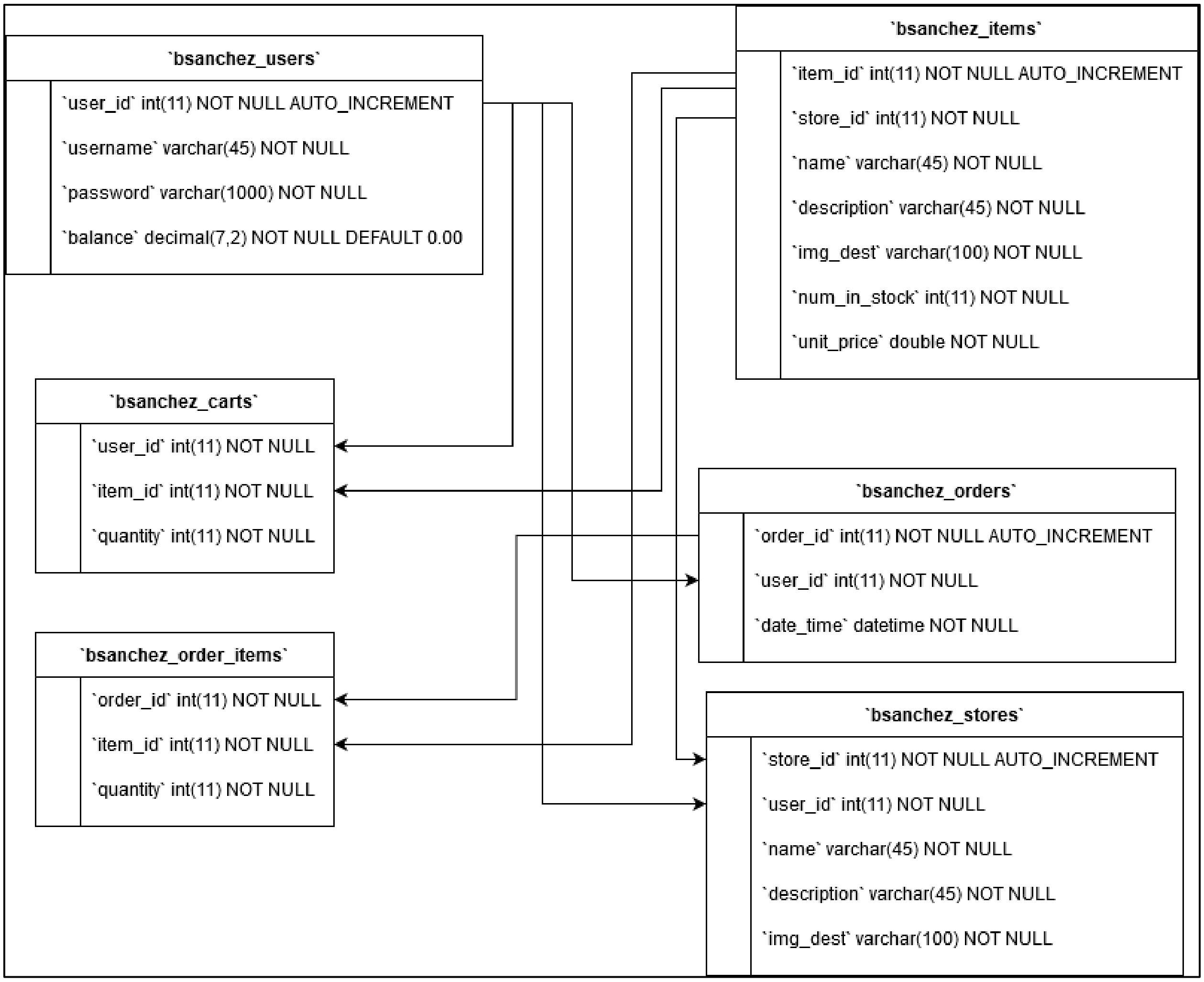
The login system code can be found in index.php and signup.php for the actual visual web pages. For working with the database, it can be found in login_inc.php, logout_inc.php, and signup_inc.php.

7. **Sessions**
   Sessions are started in header.php which is attached to the beginning of every page. This stores the user id of the user logged in, their balance, their cart total, and many more things.

Things I Learned

This was my first big web application I built so there were many lessons I learned. One is to separate your web files even more than you think you should. I wish I had more folders to separate web files but changing that now would require me to change links in every single file. Another is to always push your code to GitHub because I lost almost a day's worth of progress and since I wasn't pushing to GitHub consistently, I wasn't able to recover it. Developing my stylesheets for mobile first was something that I started but quickly forgot and it was also too much work to change.

Entity Relationship Diagram



`bsanchez_users`

`user_id` int(11) NOT NULL AUTO_INCREMENT
`username` varchar(45) NOT NULL
`password` varchar(1000) NOT NULL
`balance` decimal(7,2) NOT NULL DEFAULT 0.00

`bsanchez_items`

`item_id` int(11) NOT NULL AUTO_INCREMENT
`store_id` int(11) NOT NULL
`name` varchar(45) NOT NULL
`description` varchar(45) NOT NULL
`img_dest` varchar(100) NOT NULL
`num_in_stock` int(11) NOT NULL
`unit_price` double NOT NULL

`bsanchez_carts`

`user_id` int(11) NOT NULL
`item_id` int(11) NOT NULL
`quantity` int(11) NOT NULL

`bsanchez_orders`

`order_id` int(11) NOT NULL AUTO_INCREMENT
`user_id` int(11) NOT NULL
`date_time` datetime NOT NULL

`bsanchez_order_items`

`order_id` int(11) NOT NULL
`item_id` int(11) NOT NULL
`quantity` int(11) NOT NULL

`bsanchez_stores`

`store_id` int(11) NOT NULL AUTO_INCREMENT
`user_id` int(11) NOT NULL
`name` varchar(45) NOT NULL
`description` varchar(45) NOT NULL
`img_dest` varchar(100) NOT NULL