User Manual

**Database to use:**

Andy

**Group Member:**

Rocherno de Jongh

Andy Lau

**To start the server:**

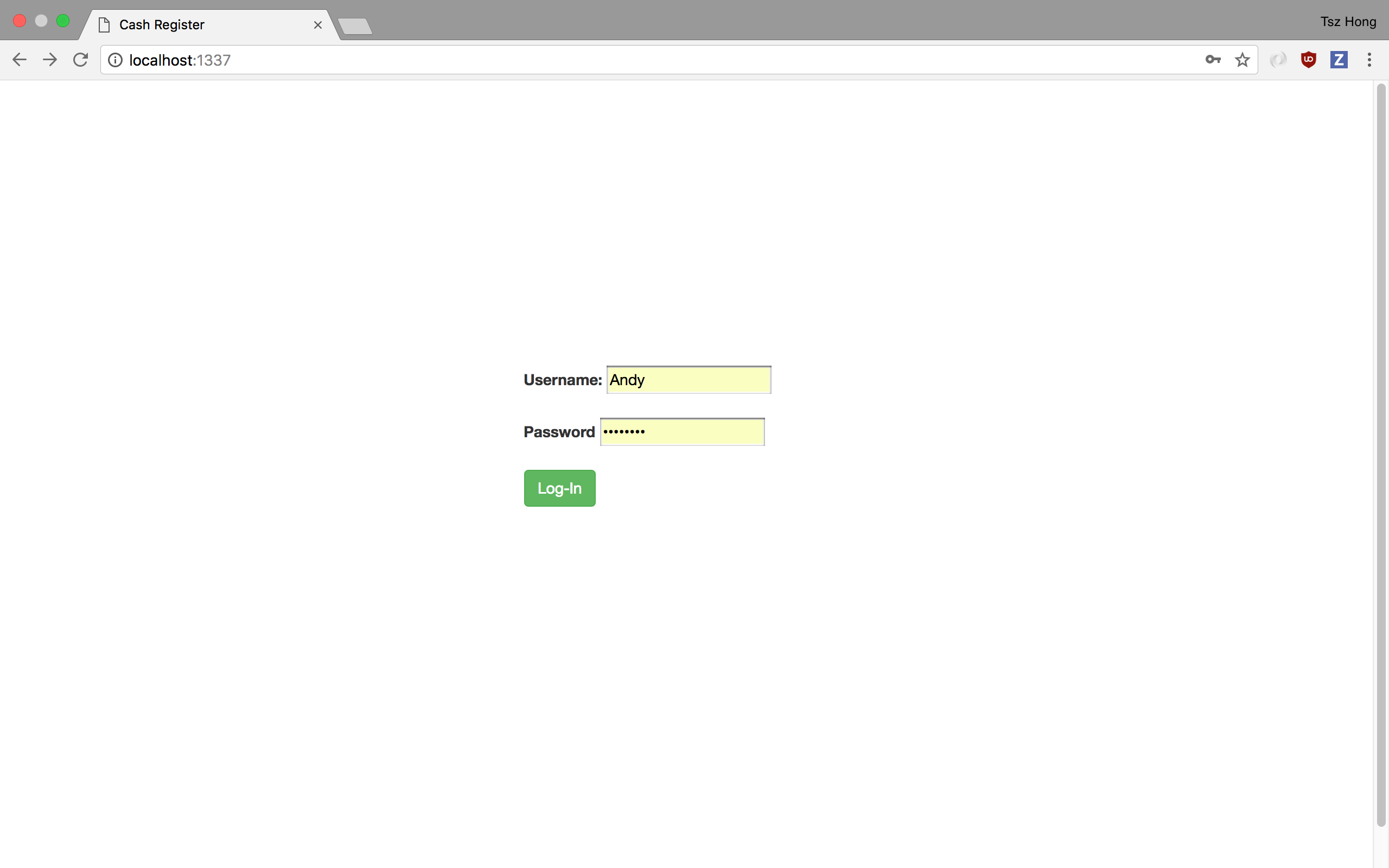
Simply type “node express.js” at the main directory of the repository.

**Web Browser to use**

Google Chrome; Firefox seems to have issues with some of the CSS formatting we used

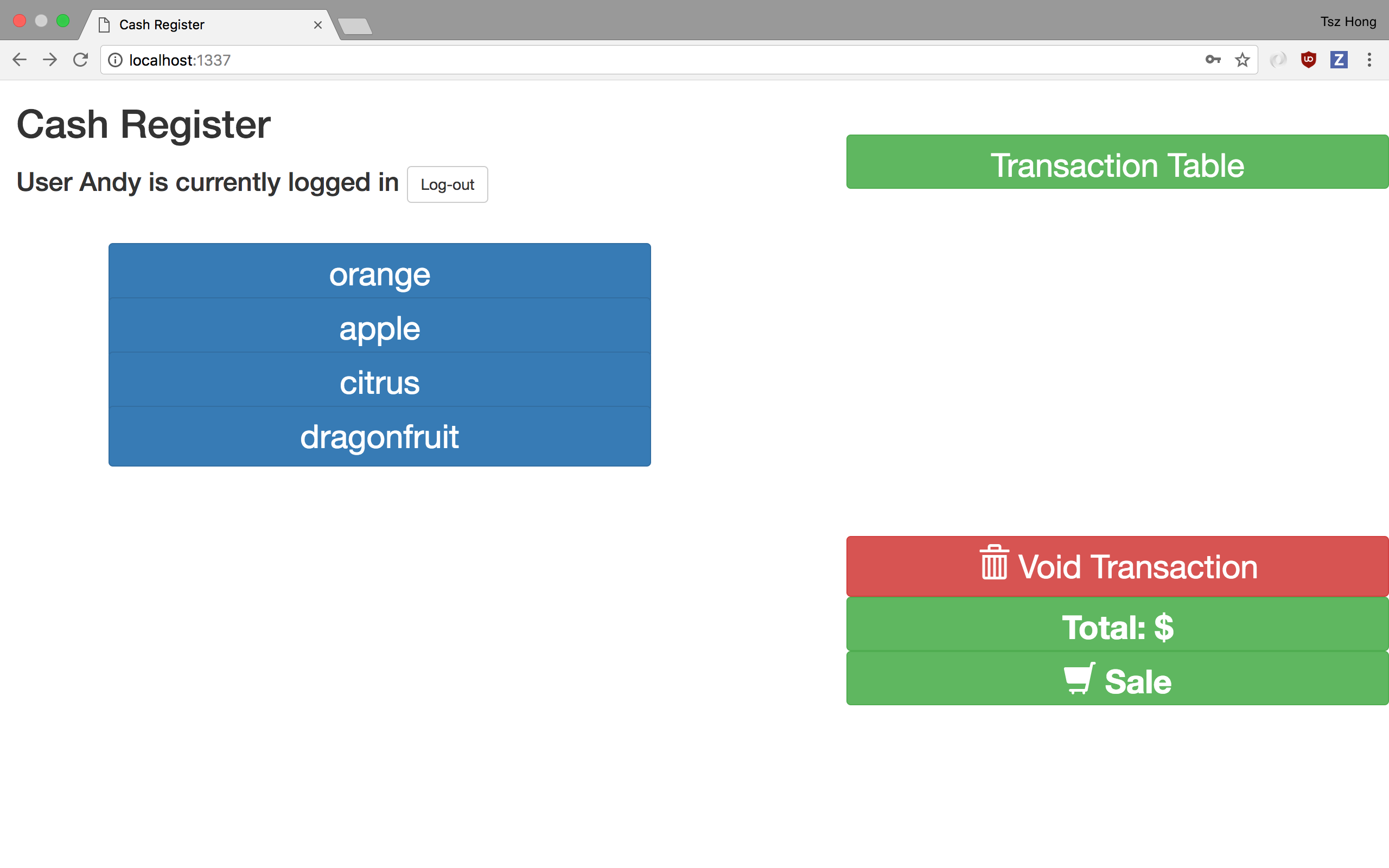
**Overview**

The goal of this project is to create a point of sales system using a simple user interface that interacts with a database. Currently, the system could handle log-in and log-off functionality, the display of inventory on the web page, the ability to add to the transaction table, a simple sale/void functionality, as well as creating a json object that displays the receipt of the previous transaction.

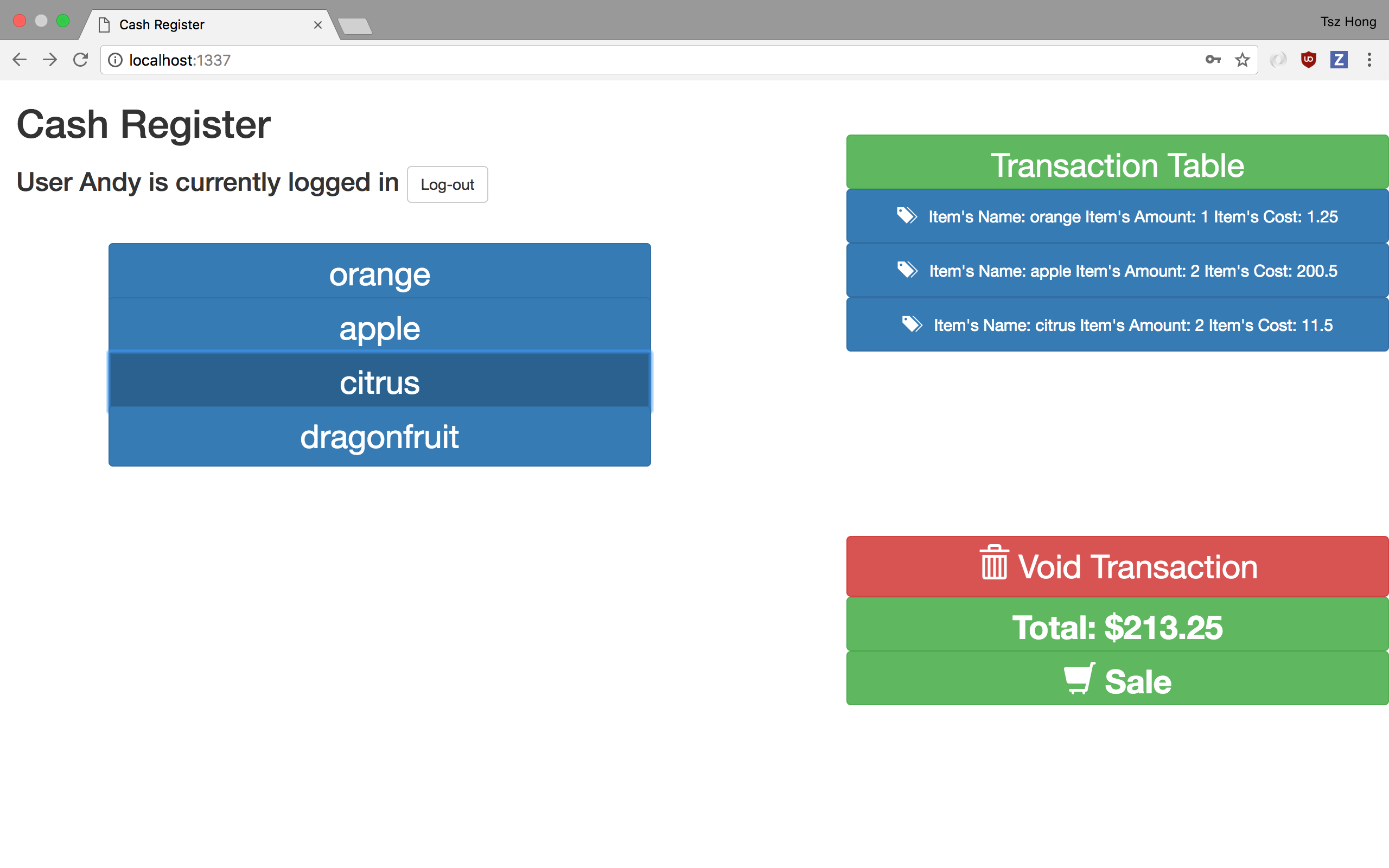


Above is the page user will see when they first arrive at the web page. The user will be asked to login in order to access other functionality of the point of sales system. A failed login will prompt a warning message indicating that the user entered the wrong username/password.

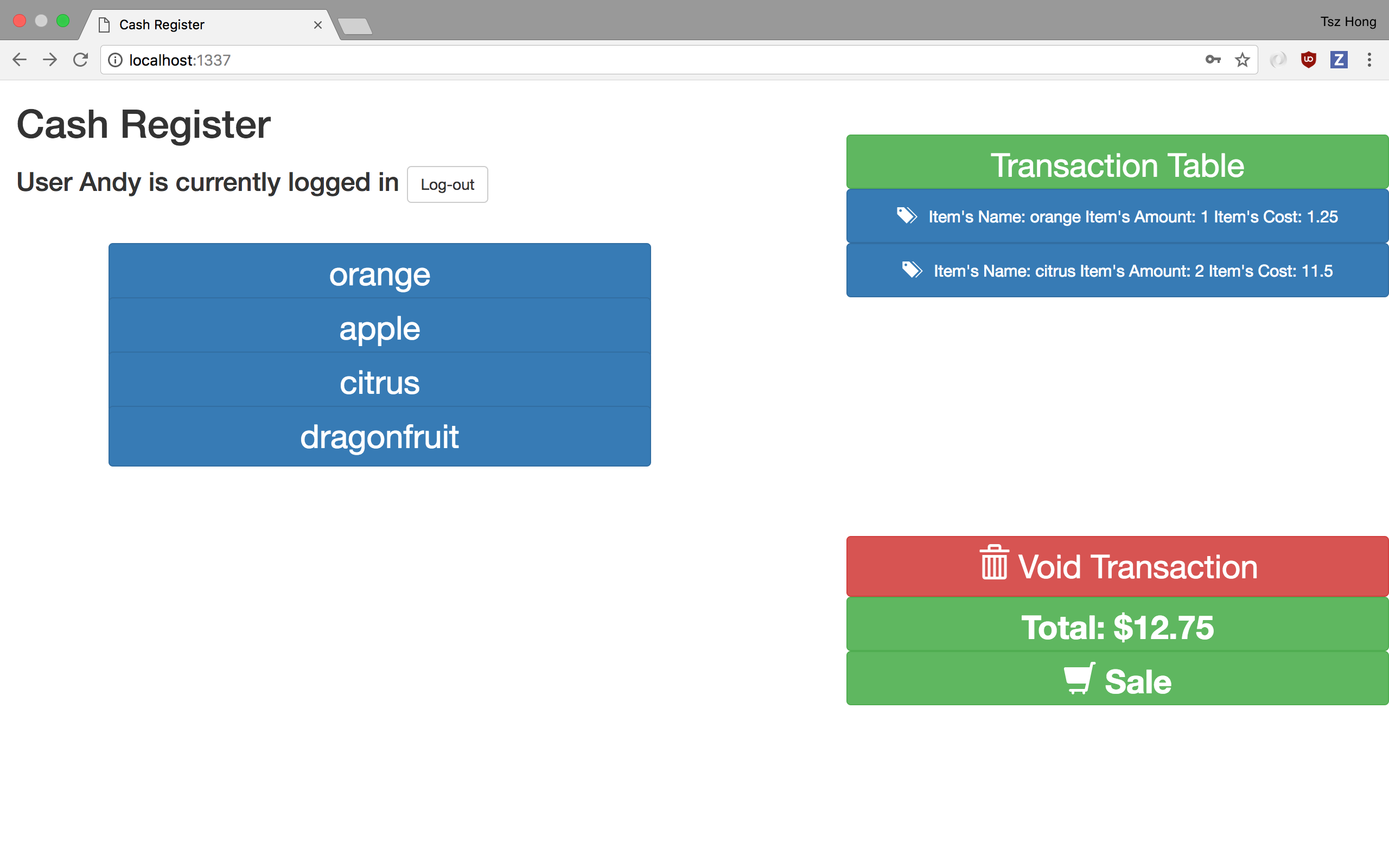
Once logged-in, the user will see the following page.



To the left of the web page, there is a list of inventory currently being offered in the store. Clicking on one of the buttons will add that item to the transaction list. The inventory data is query from the database table till\_buttons.

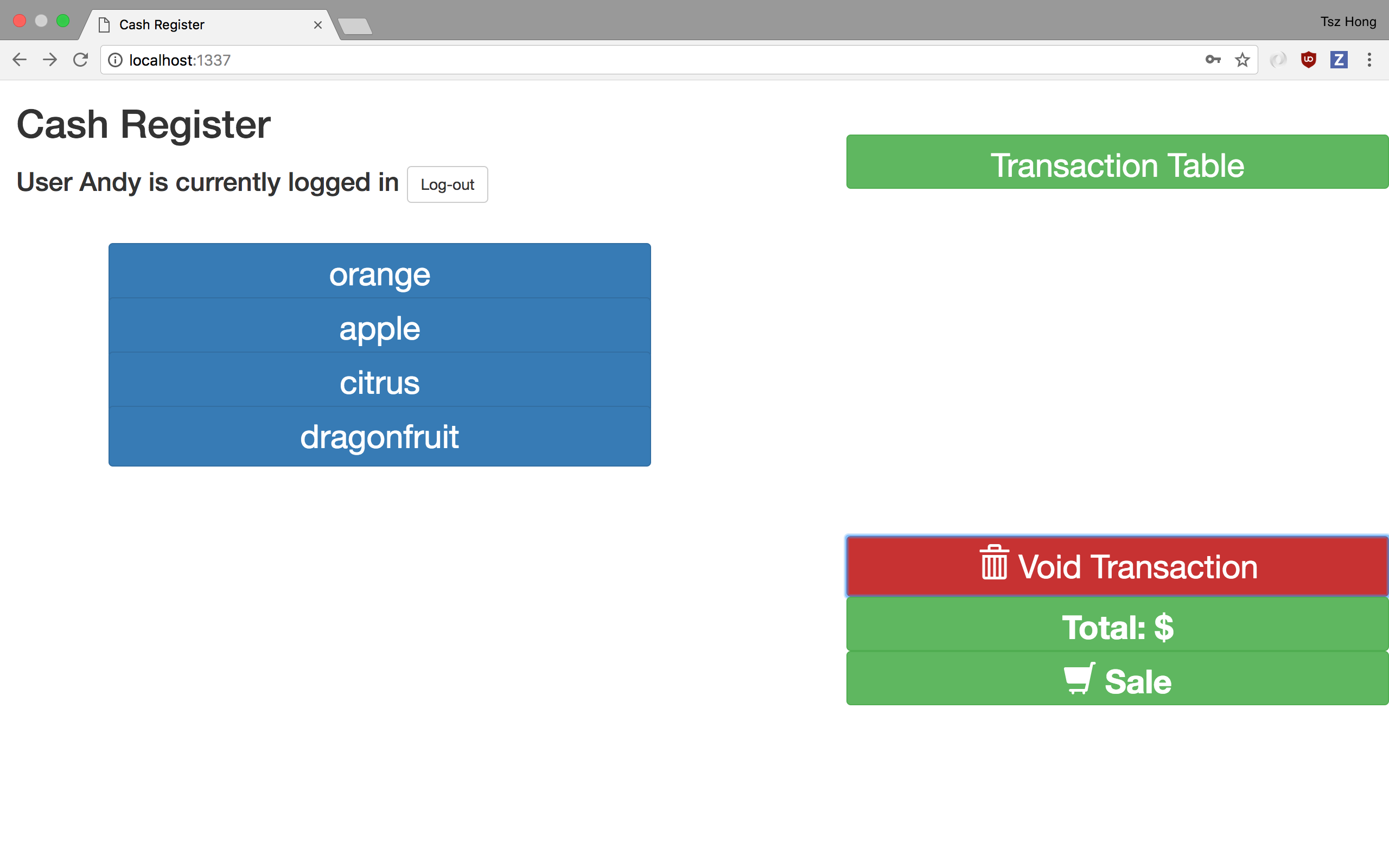


The above image shows all the items added to the transaction table. Now, let’s say we want to eliminate one of the items off the transaction table. Simply press one of the buttons of the transaction table, and that item will not be stored in the transaction table.

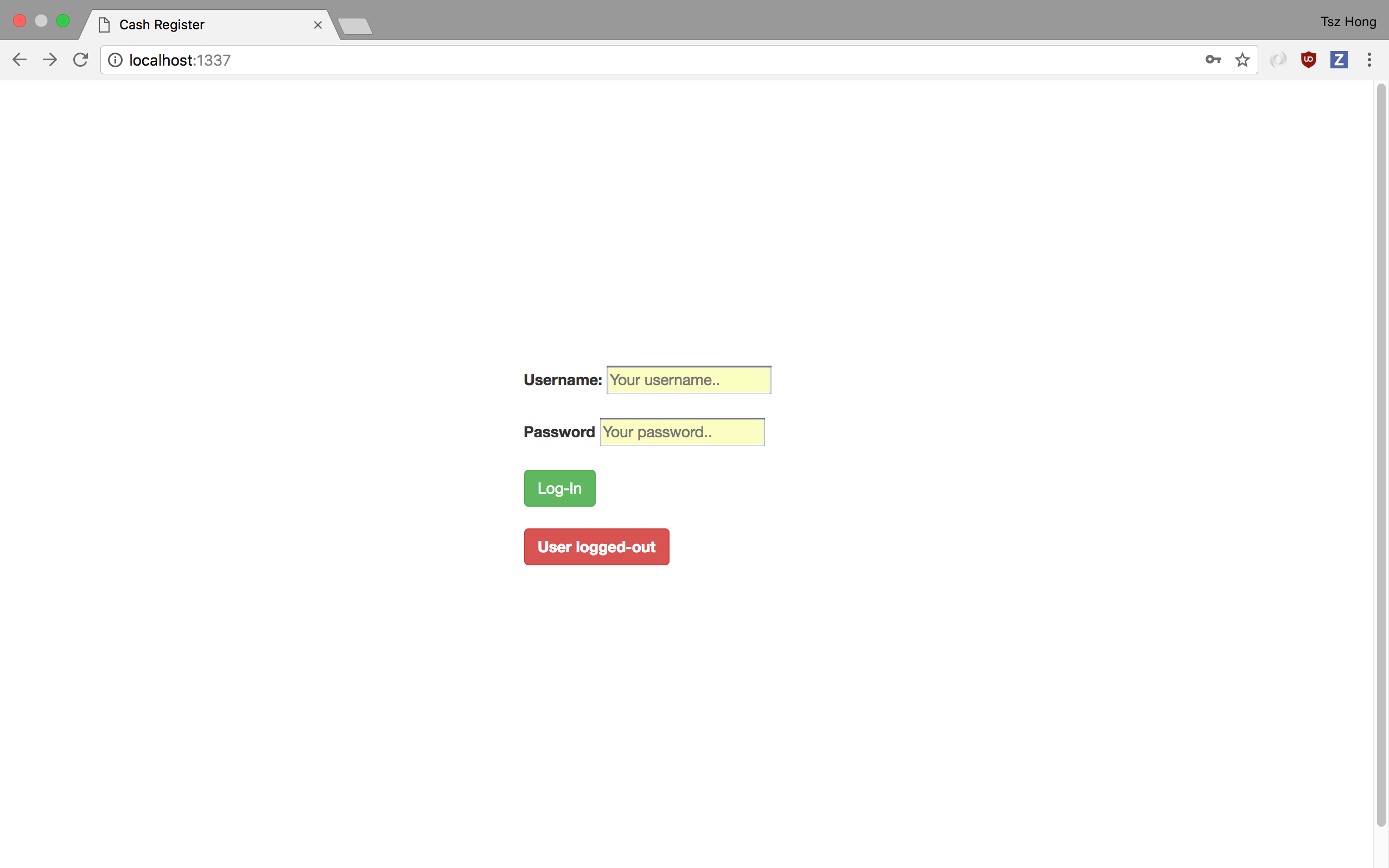


The button void transaction simply eliminate all items of the transaction table. This system also displays the current amount of the transaction. It updates the amount as items are added or removed. The sale functionality works by having the user click on the sale button and it will prompt the web server to call a procedure that moves all the temporary transaction data to the transaction\_archive table. The transaction\_archive table a one to many relationships to the transaction\_archive\_detail table. Since there could be multiple items under one transaction, the relationship is suitable.

All items will be removed from the transaction table. Shown below.



If user would like to log-out of the system, the user could click on the log-out button at the top left corner.



The user could also see the receipt for the previous transaction by typing localhost:1337/ticketize on the url of the web browser. It displays the startTime and endTime of the transaction, the duration, the total amount as well as the user.

**API endpoints**

/**buttons**

Return all buttons in Andy.till\_buttons

**/void**

Deletes all records in the Andy.transaction table

**/list**

Return all records in the Andy.transaction table; responsible for displaying the contents of the table in the web page

**/click**

Insert record into Andy.transaction table with a specific item selected from till\_buttons table

**/delete**

Delete a certain record in Andy.transaction, the user could decide which item to delete from the table

**/total**

Query the database for the total amount of the transaction

**/login**

Query to database to see if the username and password passed to the web server matched the ones displayed in the database

**/sale**

Calls on the Andy.SALE procedure that moves all records from the transaction table to the archive table

**/ticketize**

Calls on the Andy.RECEIPT procedure that returns a json object containing the necessary component of a receipt of the previous transction.