ELEC 5220 Information Networks and Technology  
Project Report

Robert Skelton

# Abstract

The goal of the project was to create a basic eCommerce website. We were to use the programming languages we have been using throughout the semester, such as HTML, php, MySQL databases, and XML. I created an electronics store called Robertmart, and edited lots of code and put many new items into an already existing database.

# Introduction

I used Sublime Text 3 as my text editor to modify all code, Google Chrome as my Internet browser, and my Operatin System is Windows 7 Ultimate 64-bit. XAMPP powered the Apache server and MySQL, and I used phpMyAdmin for SQL administration.

First, I imported the bobby.sql file into a new database I created called cd\_store. The user I will use is called bob, with a password of bob. We’re going for efficiency here, not security. I changed the login information and database name in all files. I moved the allanimals.php and display\_animal.php files into the same folder in htdocs, but renamed them something more appropriate for my store’s inventory. So the requirements for the project were really vague, and the provided code seemed tangential. Also, most of the code didn’t work, and needed to be heavily modified. Well, so much for this being an easy project like we were told in class. I added about 10 items to start with, which were all classic rock CDs. I added their associated album art from albumart.org, so the image sizes are all the same. You can view these on a website in the allItems.php page. This code was originally meant for displaying animals or something, and I’m not sure if we were even supposed to do this step, but it was in the Project folder so I did it anyways. I decided to call my store Robertmart. I think it’s a pretty awesome name. Anyways, we sell CD’s, DVD’s, books, audio books, and digital books. Some people call digital books ebooks. I also decided to call them ebooks, and those are my five categories. I decided to stock Robertmart with my favorite CD’s and movies and books, then the top ebooks and audiobooks from Amazon.com!

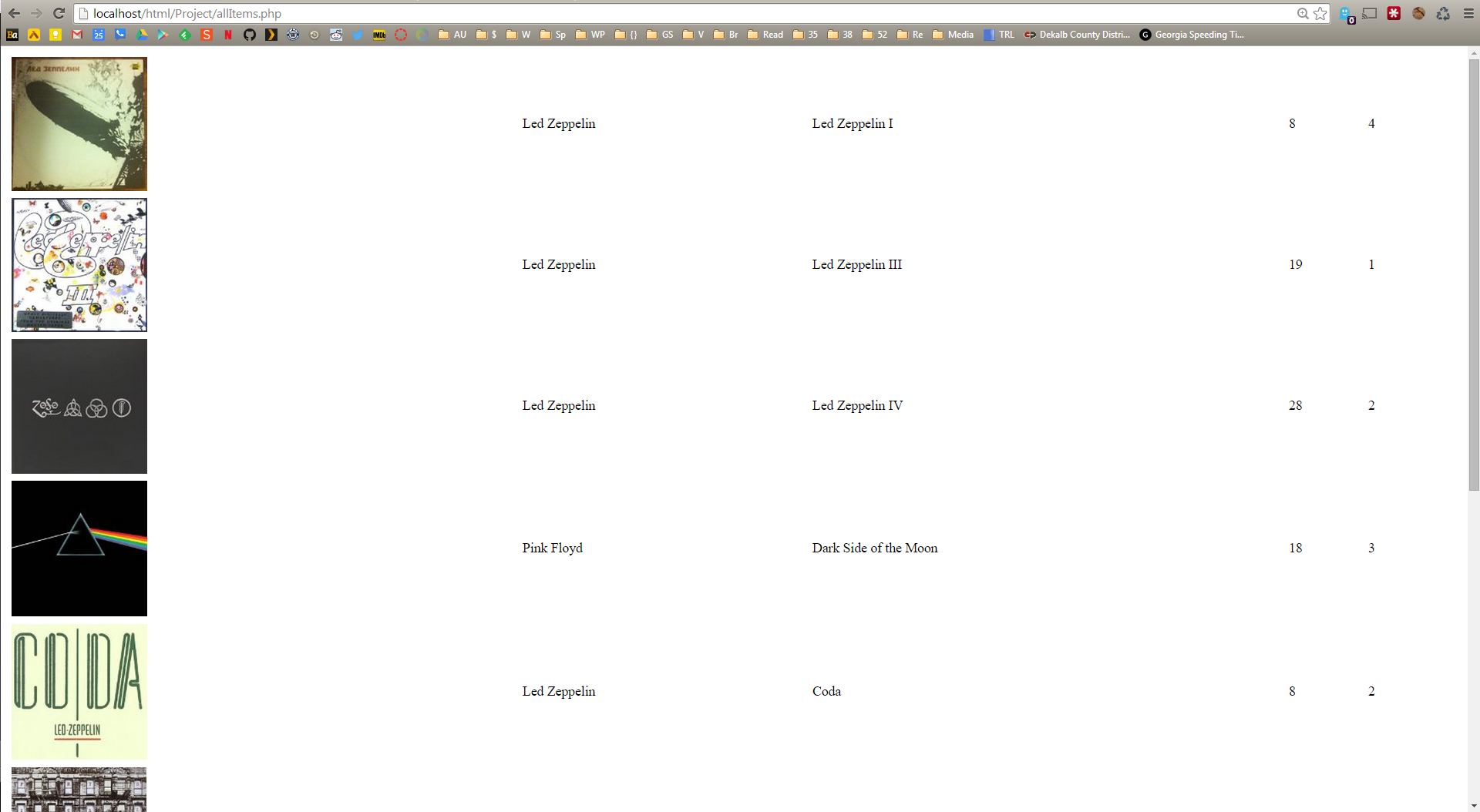


Figure 1. Loading allItems.php in a browser displays pictures of the current Inventory and associated information.

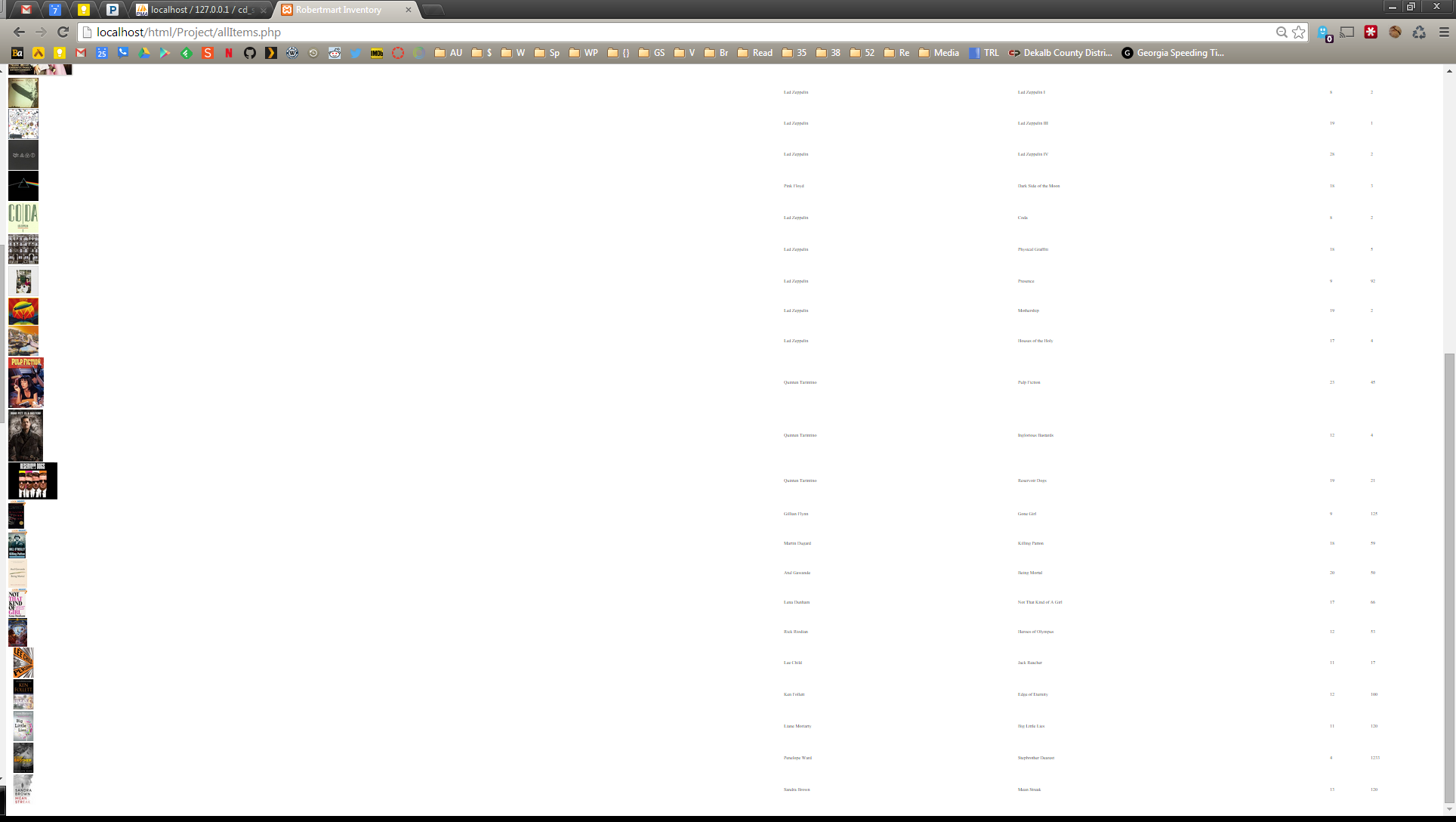


Figure 2. A zoomed out view of allItems.php once over 30 items of 5 different categories were added.

Then I realized that all these files were included for no reason, and had wasted a day working on something that wasn’t required. Then I looked at the supplied project by Rhodes and Thomlison and imported the laptop database. I called the database laptop\_store like in the example so that I had to change less code. I created a user called Web\_customer with password “password” for this new database, which is the same as the example. I let 15 items display on the main page by default. There were 5 categories required. I used laptops, desktops, phones, tablets, and phablets. I set item\_number to auto incremented so I didn’t have to worry about it. I changed the weight table to Category; so much of the code could stay functional. To update weight per brand, use the following snippet of code:

[UPDATE](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fupdate.html&server=0&token=60d09a2008b2292a0ba7c298ba632b47) `laptop\_store` [SET](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fset.html&server=0&token=60d09a2008b2292a0ba7c298ba632b47) `weight`='light' WHERE `brand`='Google'

To update the category of an already placed item, like I had to for almost every item before I destroyed everything, run something similar to the following:

[UPDATE](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fupdate.html&server=0&token=60d09a2008b2292a0ba7c298ba632b47) `laptop\_store` [SET](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fset.html&server=0&token=60d09a2008b2292a0ba7c298ba632b47) `Category`='Tablet' WHERE `item\_number`='20'

To delete an item from the database based on item number, run the following statement:

[DELETE](http://localhost/phpmyadmin/url.php?url=http%3A%2F%2Fdev.mysql.com%2Fdoc%2Frefman%2F5.5%2Fen%2Fdelete.html&server=0&token=60d09a2008b2292a0ba7c298ba632b47) FROM `laptop\_store` WHERE `item\_number`=27

# Design

The lab had nine parts, most of which were already correctly implemented when the lab was assigned. We basically just had to add 25 products in 5 different categories, and keep the website completely functional. I will detail these nine parts of the lab below, and show appropriate screenshots to verify that my implementation is correct.

First of all, I restarted with a new database, using phones, phablets, tablets, laptops, and desktops as my categories. You can search by category and brand of electronic. I used most of my pictures from Google Images or Amazon, and a lot of the specifications were made up because I’m lazy. At one point I was modifying the database and I accidentally submitted an SQL command that dropped the entire laptop\_store table. That wasn’t fun. So I started over from scratch, and learned how to start backing up my SQL databases. Then I started making sure the website was correct according to the project specifications. Here goes!

**Step 1**

The goal of this step was to have multiple users with different session IDs, and use AJAX to compute the total cost in your shopping cart.

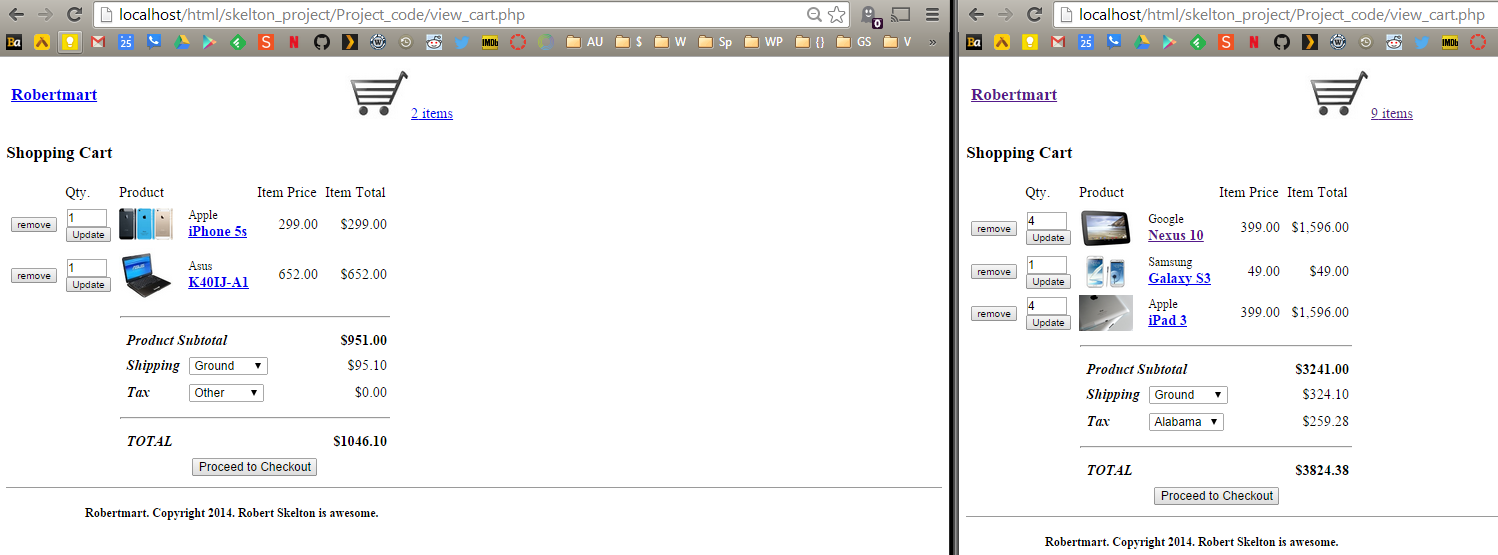


Figure 1. Display of two different users with different shopping carts, checking out at the same time.

**Step 2**

The goal of this step was to have a shopping catalog with a search function, and images for each product. There should be at least 25 products spread out over 5 categories. I implemented my categories to be laptops, desktops, phones, tablets, and phablets. You can import my database using phpMyAdmin on your own computer to verify. This step took a long time and was tedious. I also messed up at some point in this step and dropped the entire laptop\_store table, so I restarted.The final database is contained in the attached laptop\_webstore.sql. I kept the laptops that were originally in the database just to have more items, but I also added 25 or more items of my own to fulfill the requirements for this step.

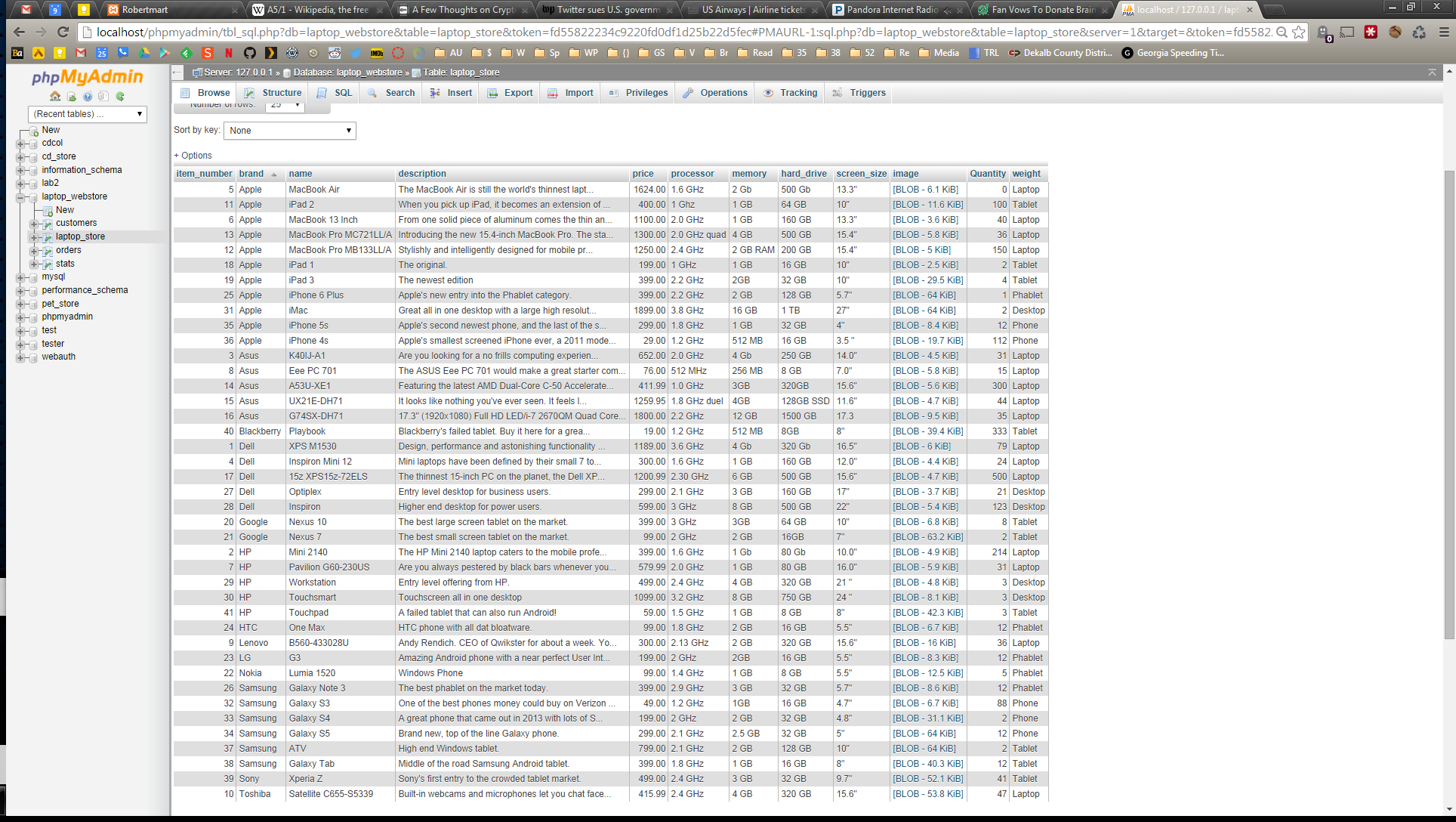


Figure 4. Screenshot of the finalized laptop\_store table for laptop\_store.

**Step 3**

There should be three types of shopping: ground, 2 day, and overnight. There should be tax implemented, and AJAX should be used to compute the total without reloading the page. This was already implemented in the inherited code.

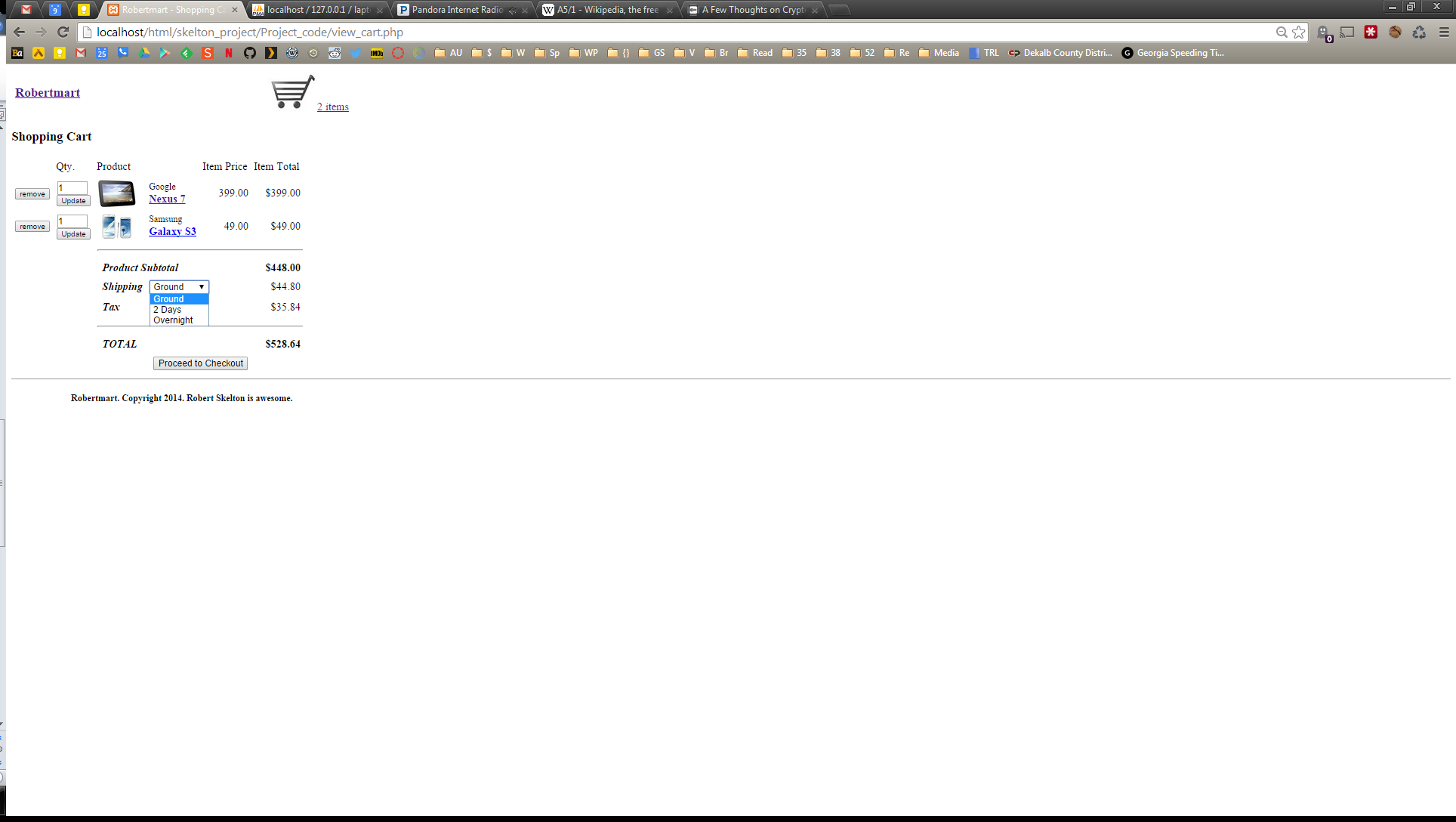


Figure 5. Displaying Ground, 2-Day, and Overnight shipping options.



Figure 6. Removing an item automatically updates the cost using AJAX.

**Step 4**

Mask the credit card number. The last four digits should be the only part of the credit card number that isn’t hidden.

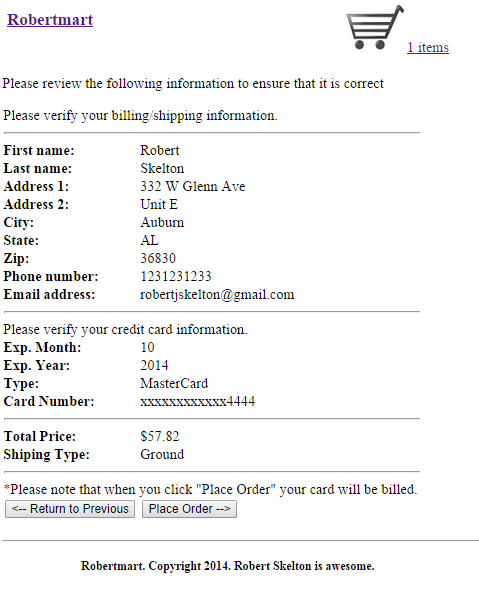


Figure 7. On the checkout page, the Credit Card information is hidden, except for the last 4 digits.

**Step 5**

Inventory control should be implemented, so that a customer can’t buy more of an item than there are in stock. We did not have to implement anything for rain checks, or oversold or anything like that, just a simple inventory count check. As seen in Figure 8 below, I tried adding 17 iPad 3’s to my cart, when only 4 are in stock. The website added 4 items to my cart, then told me that there is not enough iPad 3’s in stock for me.

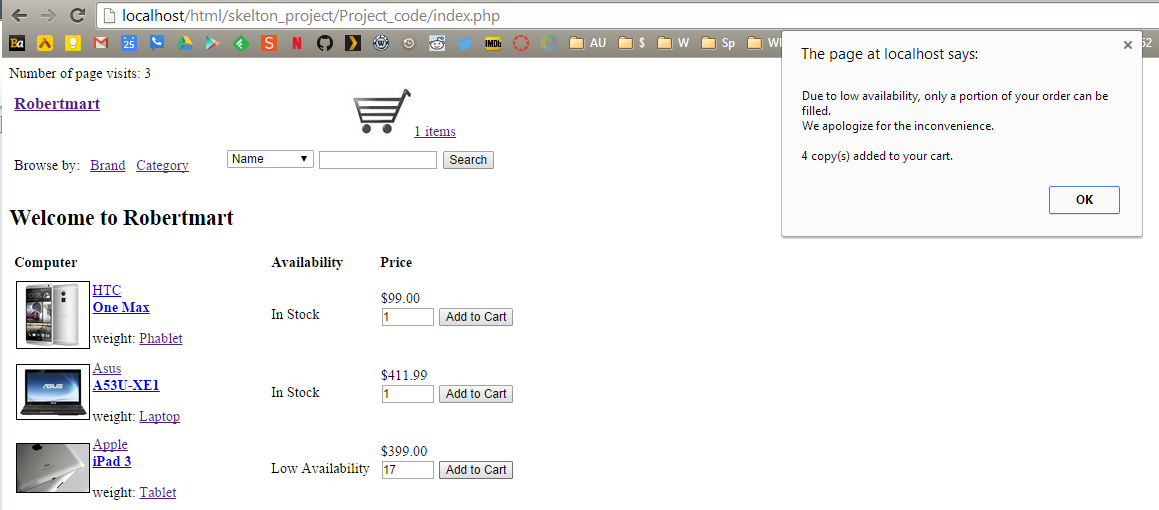


Figure 8. Inventory control is implemented so that you cannot purchase more items than there are currently in stock.

**Step 6**

Check all user input when checking out. This includes checking the zip code, phone number, email address and credit card number using AJAX functions. This was accomplished by using already written regular expressions, or regexes, as we learned in a previous lab. You can see in Figure 9 below that all fields in the checkout portion, including the credit card number and expiration date, are checking using Javascript functions in AJAX.

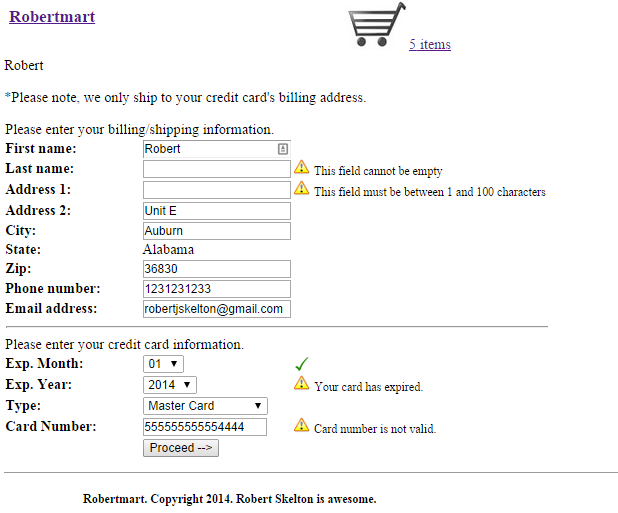


Figure 9. All input is checked using AJAX before checking out.

**Step 7**

Implement multiple cookies, so that each user has a counter for the number of visits he or she has. The counter should be displayed at the top of the page. I used the Google Chrome Extensions “Clear Cache” and “Remove cookies for site” to assist on this step.



Figure 10. Two different users with different values for the page view counter.

**Step 8**

I used Wireshark to capture three things. The first was the initial cookie, set on the first visit of the site. The second was submitted a new cookie by the browser on the second visit. The third and final Wireshark screenshot is when a new user has a cookie set.

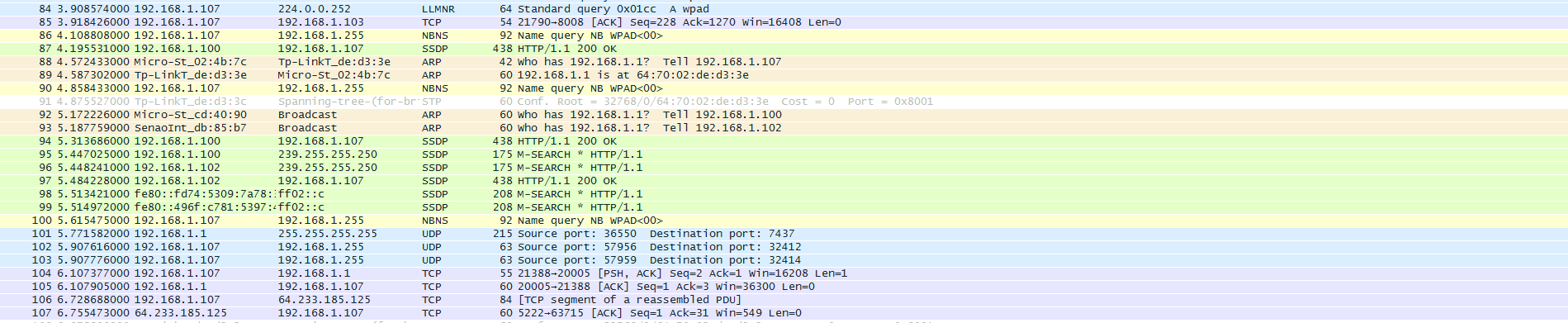


Figure 11. Setting the initial cookie on the first visit on Robertmart website.

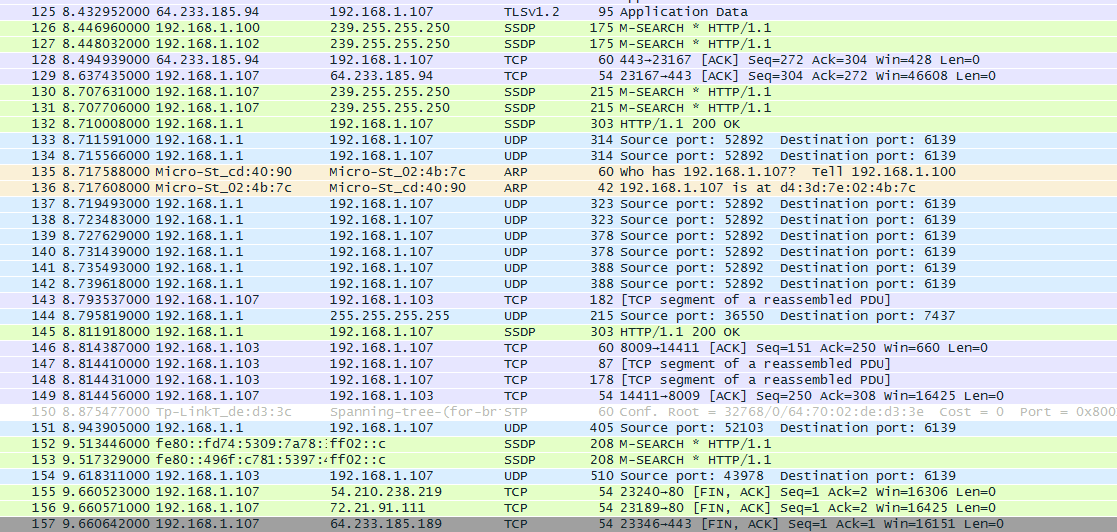


Figure 12. Second visit to Robertmart.

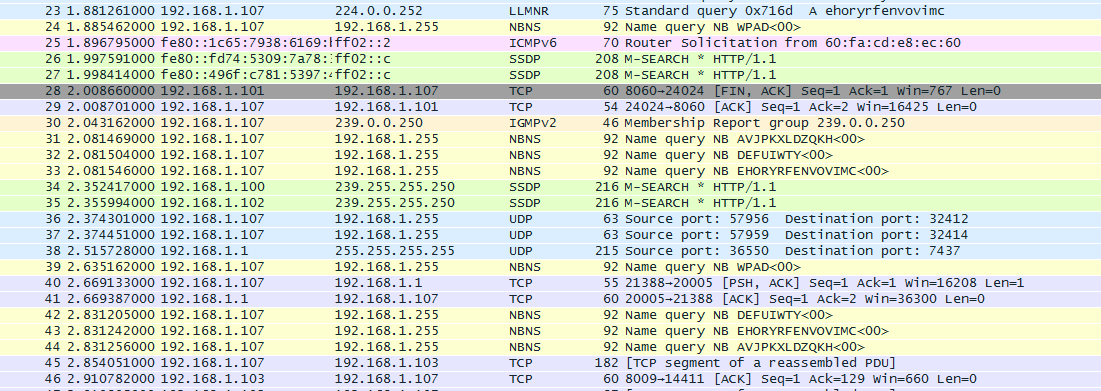


Figure 13. Setting a new user cookie upon visiting Robertmart.

**Step 9**

This was simply to document and write comments for all new code added. First, I moved all the brackets onto the same line, because the coding convention that the previous programmer used drove me crazy. That way uses up so much space, and is useless.

# Result

Eventually, I completed all required tasks of this lab and could display a fully functional, basic Amazon style website. I did not attempt any of the extra credit options since I don’t know Android and I’m not looking to do web development in the future.

# Conclusion

This project was a challenge, as it is easily the largest code base I have ever dealt with. Managing 20 tabs of code files is stressful, especially when you hate the coding style. Putting lots of new items was tedious, but I feel much more comfortable with phpMyAdmin and issuing MySQL commands because of it. This project taught me lots about working on a large project, and working with existing code. I know these are both things I will be doing once I obtain a full time job after graduation.

# References

Dr. Wu’s included slides and lab material ls

* Rhodes\_Tomlinson\_Project1
* http://www.albumart.org for CD album covers
* Google Images for some electronics pictures
* Amazon.com for all other images