**Project 2 E-Commerce Website**

-Wang Kaida

**Project Overview**

For this project, I implement an E-commerce website that sales shoes online using PHP and MySQL database.

The requirements I fulfilled in this project are as follows:

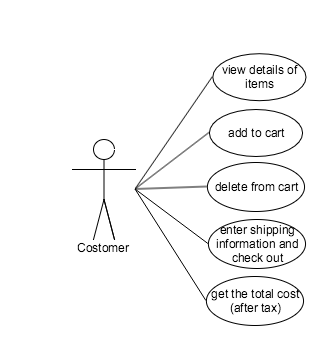
* Database with 10 or more products with name, description, and cost
* Web page that allows a user to select multiple products to purchase and the quantity of each product
* Enter shipping address
* Calculate tax based on the state being shipped
* Calculate total cost
* **Empty the shopping cart** (there is a minor problem with this function: you need to press the empty button twice to delete all items in the cart)
* **UI design using CSS** (I used some CSS templates on the internet)

**Overview of the Architecture**

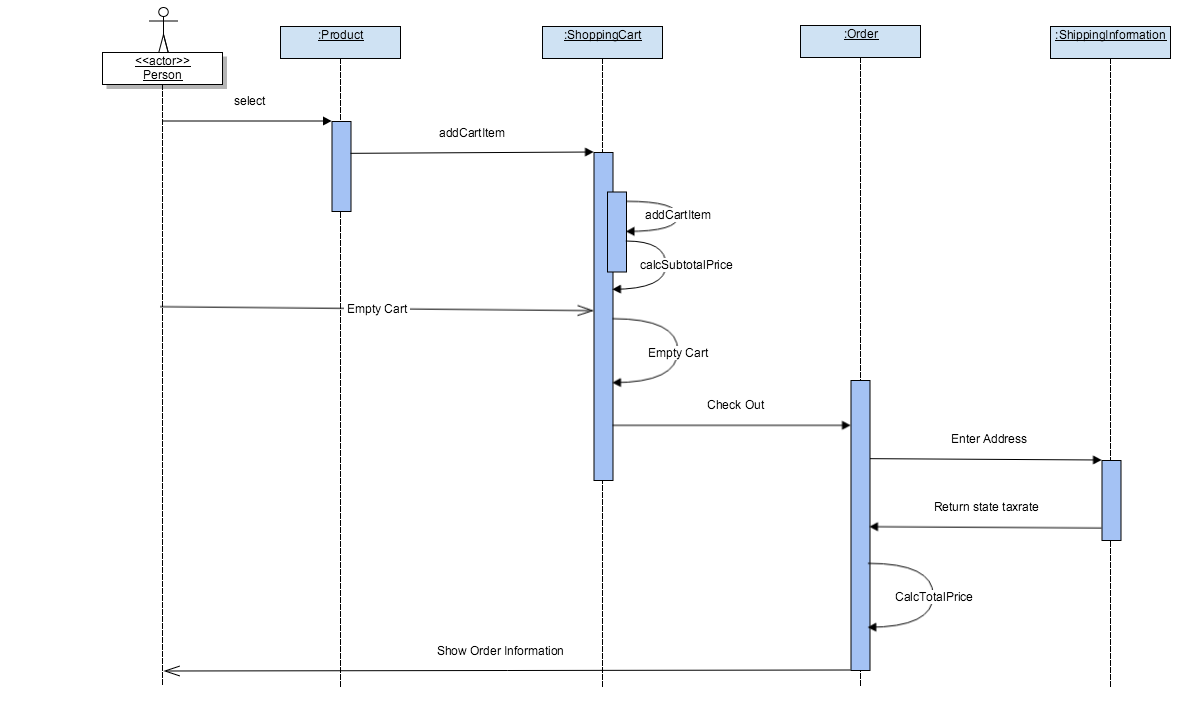
Software Architecture I Chose

I use **two-tier Client/Server architecture** to implement the project.

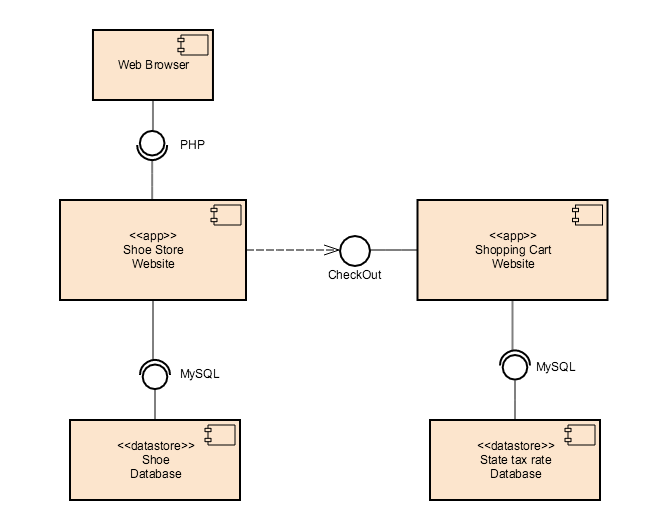
**Scenario View**



**Logic View**



**Deployment View**



**Architecture Selection**

**Analysis**

The first requirement of this project is to provide a database that saves information of all products, we can store all databases involved in this website to the server, and we can consider the user of the website as the client. After analyzing all the requirements, it is pretty simple to come up with the idea to implement the project using client/server architecture. And because our website won’t involve a lot of databases and interactions, I choose to use two tiers, that is the client communicates directly to the server without any third layer between them. The following are some elements in this architecture.

**Components**: database storing information of each shoe; database storing state tax rates; PHP files that show the content of websites.

**Connectors**: the components in this system are interacted by client clicking events. Whenever the client clicks a link, according to the specific link, a function of PHP file would be invoked and bring the required data in database to the front end.

**Trade Study**

The purpose of the trade study is to compare and analysis different software architecture and choose the best one to implement the project. Before making the final decision, I’ve chosen several often used software architecture to compare and analysis: Client/Server architecture, Object-Oriented architecture, Service-Oriented architecture, Component-Based architecture.

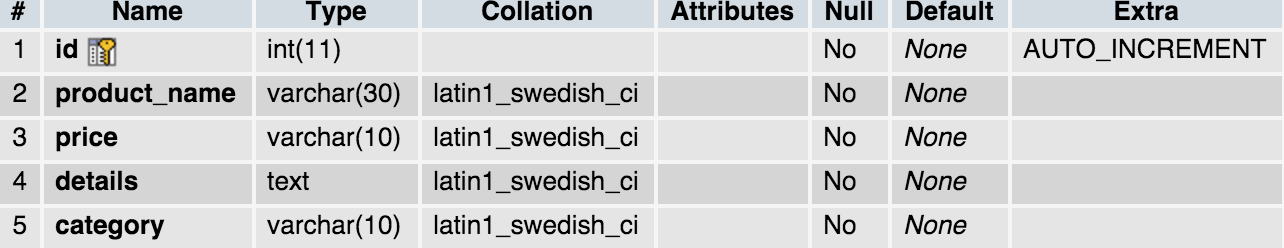
**Object-Oriented Architecture**: There are two objects we can extract from the requirements of the project: customer and the products. Because we don’t need to implement the login system for the customers, so object customer only has methods and no properties. And instead of using class, we can easily use database to store all the information of the products, which is easier to for implementation and maintenance. So I don’t think object-oriented architecture is a suitable for the project.

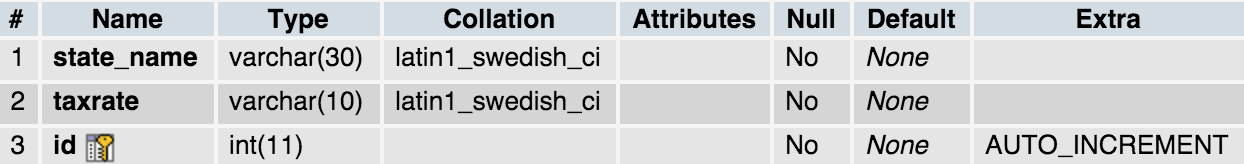
**Service-Oriented Architecture**: The goal of SOA is to achieve “loose coupling” among interacting and contracted services via communication protocols. It is usually meant for large-scale system. In our system, the only interaction is between the customer and the website invoked by clicking links. So the coupling of components is already loose. Therefore, SOA is a bit too complicated for our system.

**Component-Based Architecture**: I think component-based architecture has a lot of similarity to object-oriented architecture: They both decompose the system into components(classes) and are both good for reusability in the future. But component-based architecture can allow multiple levels of encapsulation and multiple interfaces between components, which makes it more flexible and useful. Actually component-based architecture is a good way to implement the E-commerce website, because the components in the system can be easily extracted and reused in the future version of the E-commerce website. But as I mentioned before, for this project, S/C architecture is the optimal option.

**System Implementation**

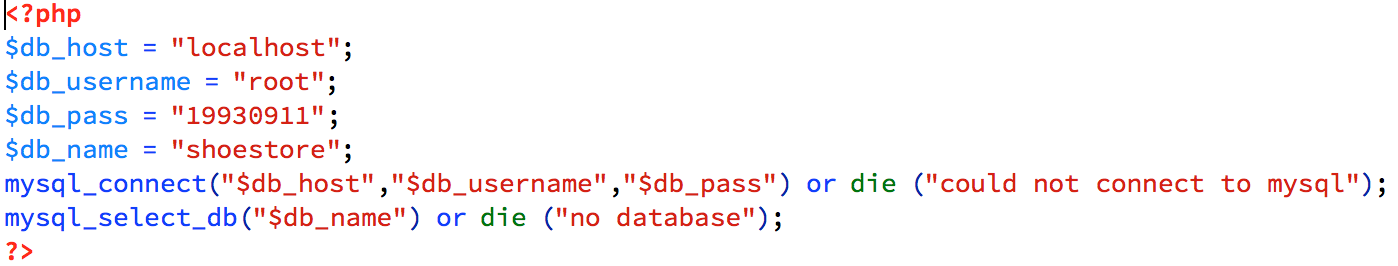
1. Based on the architecture views that was drew in the design phase, we can see there are 3 website pages that are necessary for our website: the homepage that shows all categories and products(index.php), a page that shows the detail information of the product(product.php), a shopping cart page that shows all items currently in the cart and can show the user the total price once the user entered the shipping information(shoppringcart.php).
2. We need two databases: a database that store the information of shoes and another one to store the tax rate for every US state. In this project, I used MySQL to store the data. The pictures below show the data structure of those 2 databases.





**Figure 1.** Data structure of the databases

1. Since all pages need to show the product information which is stored in the database, it is necessary for all pages to connect to the server. It is convenient to simply create a new PHP file(connect\_to\_mysql.php) that connects to the database and include that file in every page. The code for connect\_to\_sql.php is shown below.



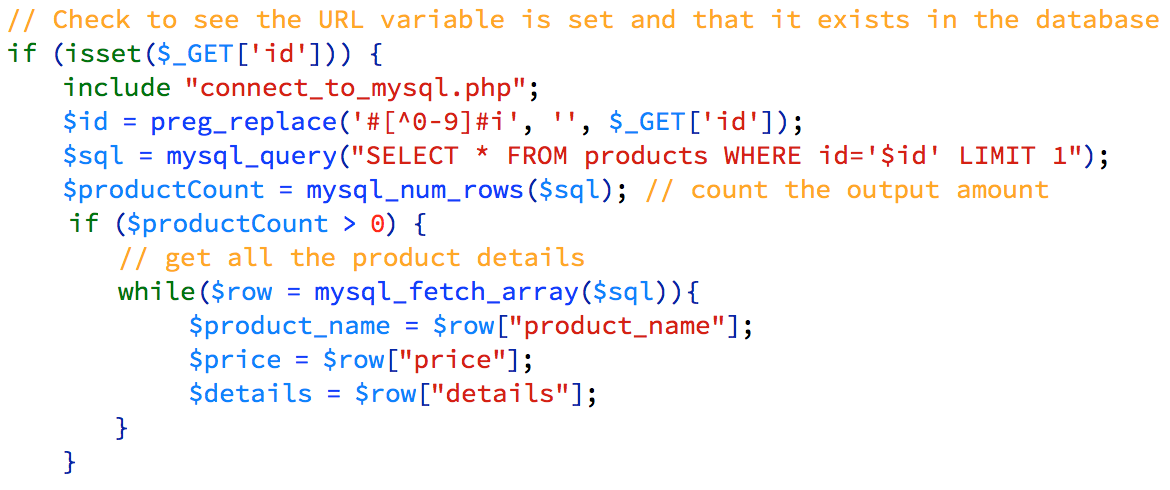
**Figure 2.** connet\_to\_sql.php

1. To show all products in the homepage, I fetched the data from MySQL using PHP. For example, in the figure 3, to show all shoes under the category Reebok, I store all data in the variable “$sql3”, and use while loop to output the price, the name of the product to the website.



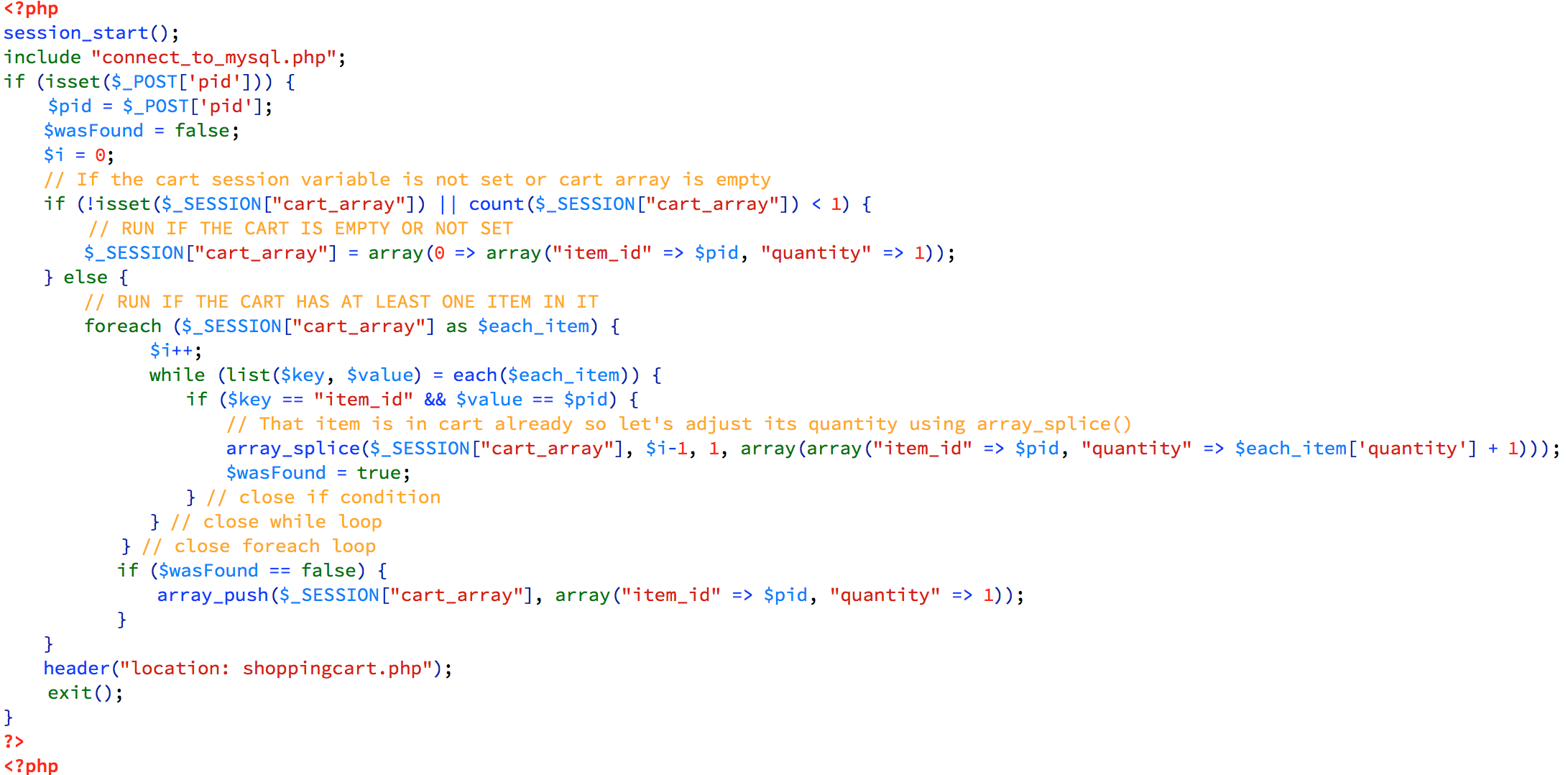
**Figure 3.** Get and show information of Reebok shoes on the site

1. To show the detailed information of a specific product, I passed the “id” value of a product from the index page to the product page, find the product of the same id in the database and show it to the user.



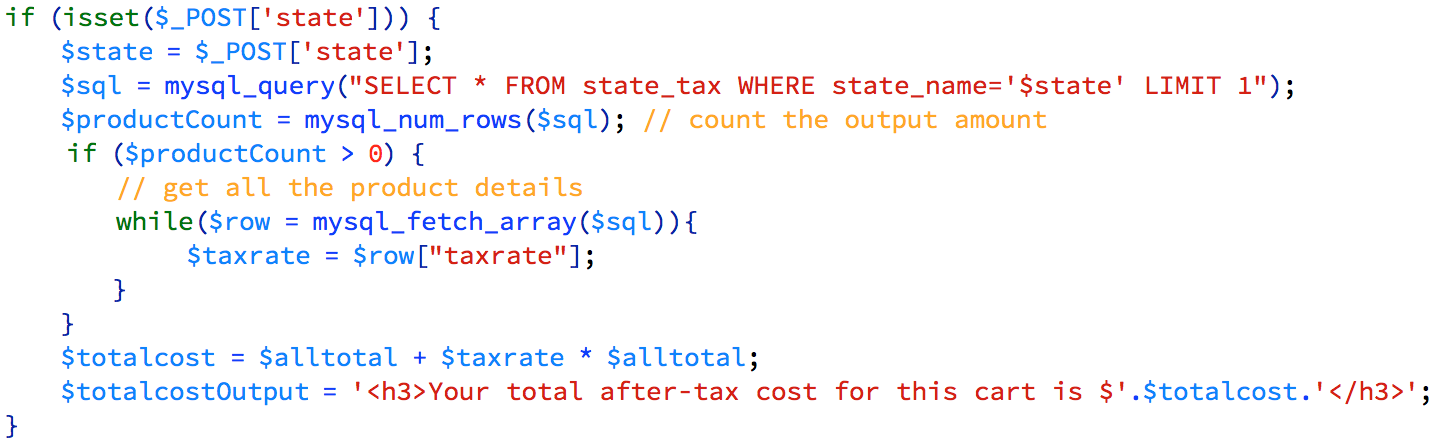
**Figure 4.** Check the passed value of “id” and store its data in variables

1. To implement the shopping cart of the website, it is necessary to store what is in the cart for some time. To make this happen, I used “session” in the shopping cart page. The session started once an item has been added to the cart, so that we can store multiple items in the cart without destroying the already exiting ones. The way to determine which item is added is basically the same as the technique I used in step 5.



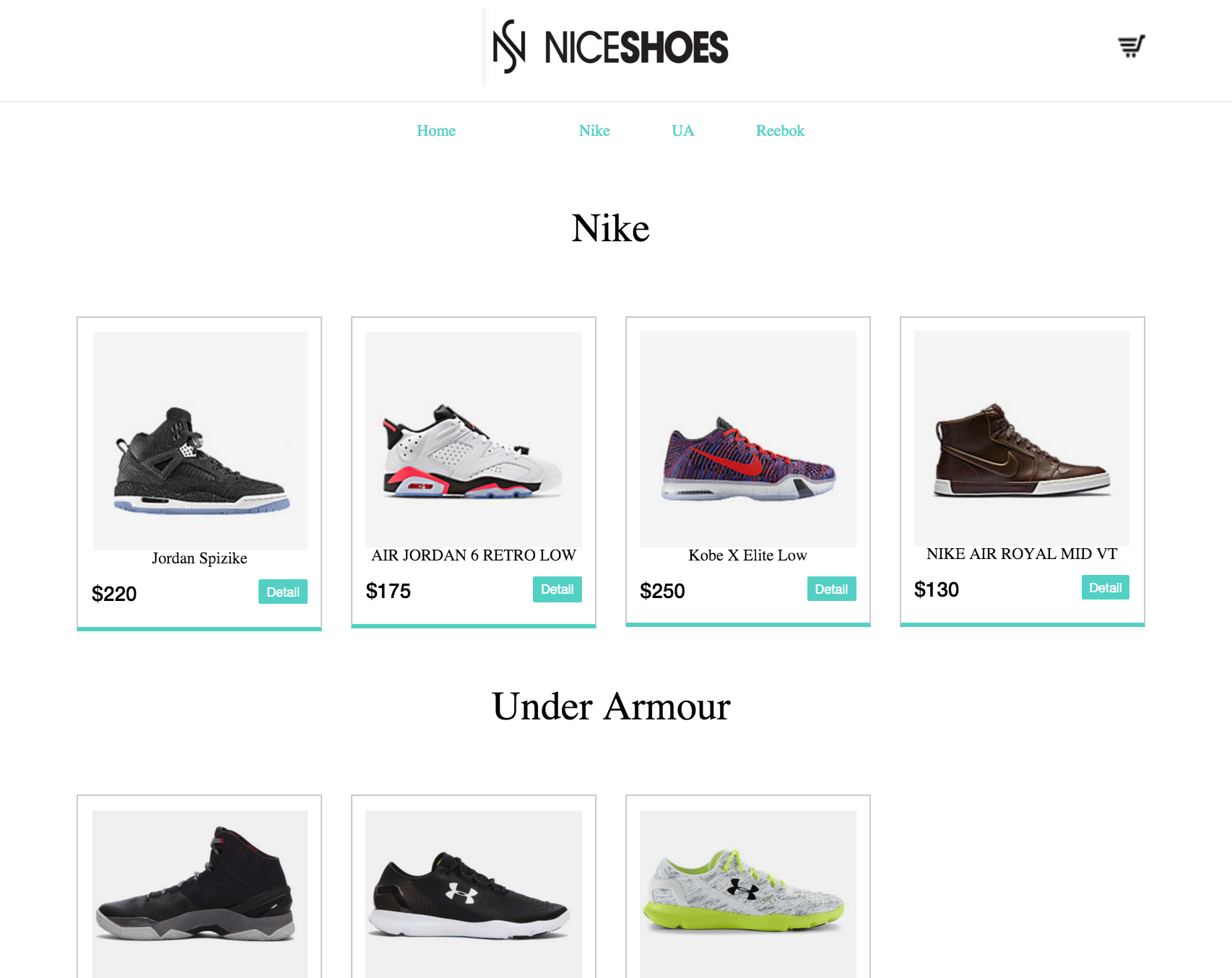
**Figure 5.** start the session and store the item information on a 2-demention array

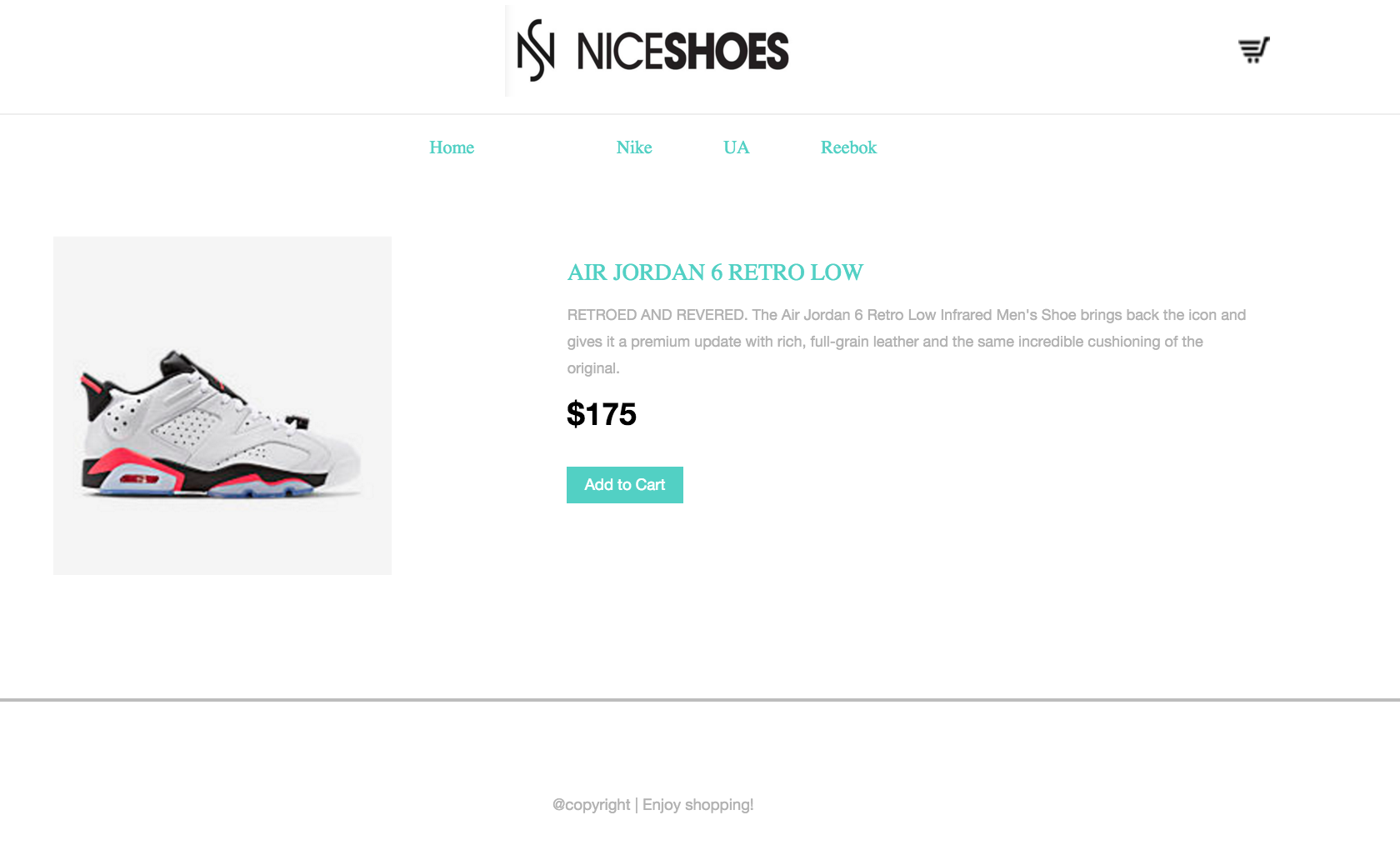
1. After the user entered the shipping address and the state, the value of the state would be compared to the state tax database and get the corresponding tax rate, then the total price of the shopping cart would be shown on the webpage based on that rate.

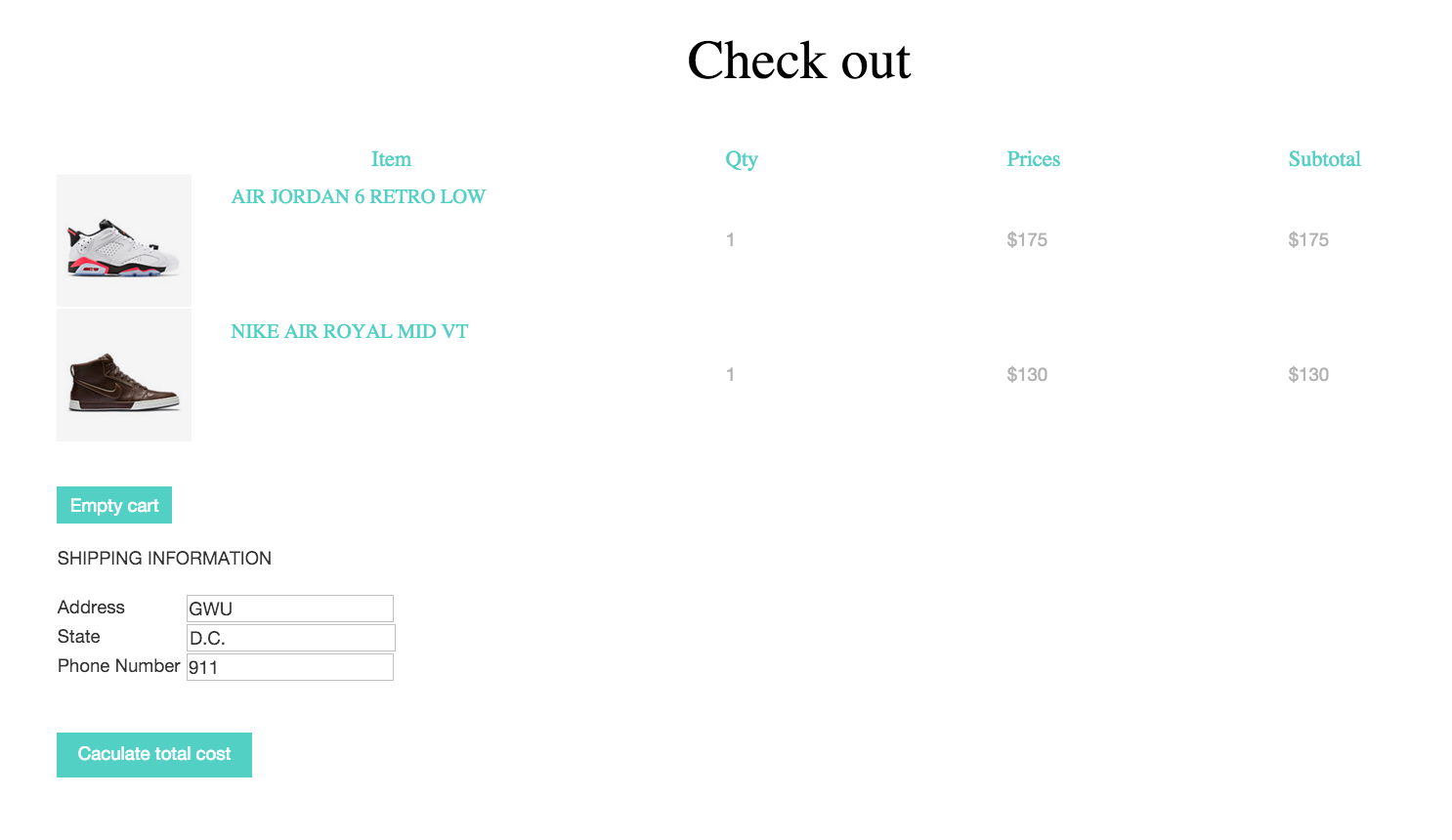


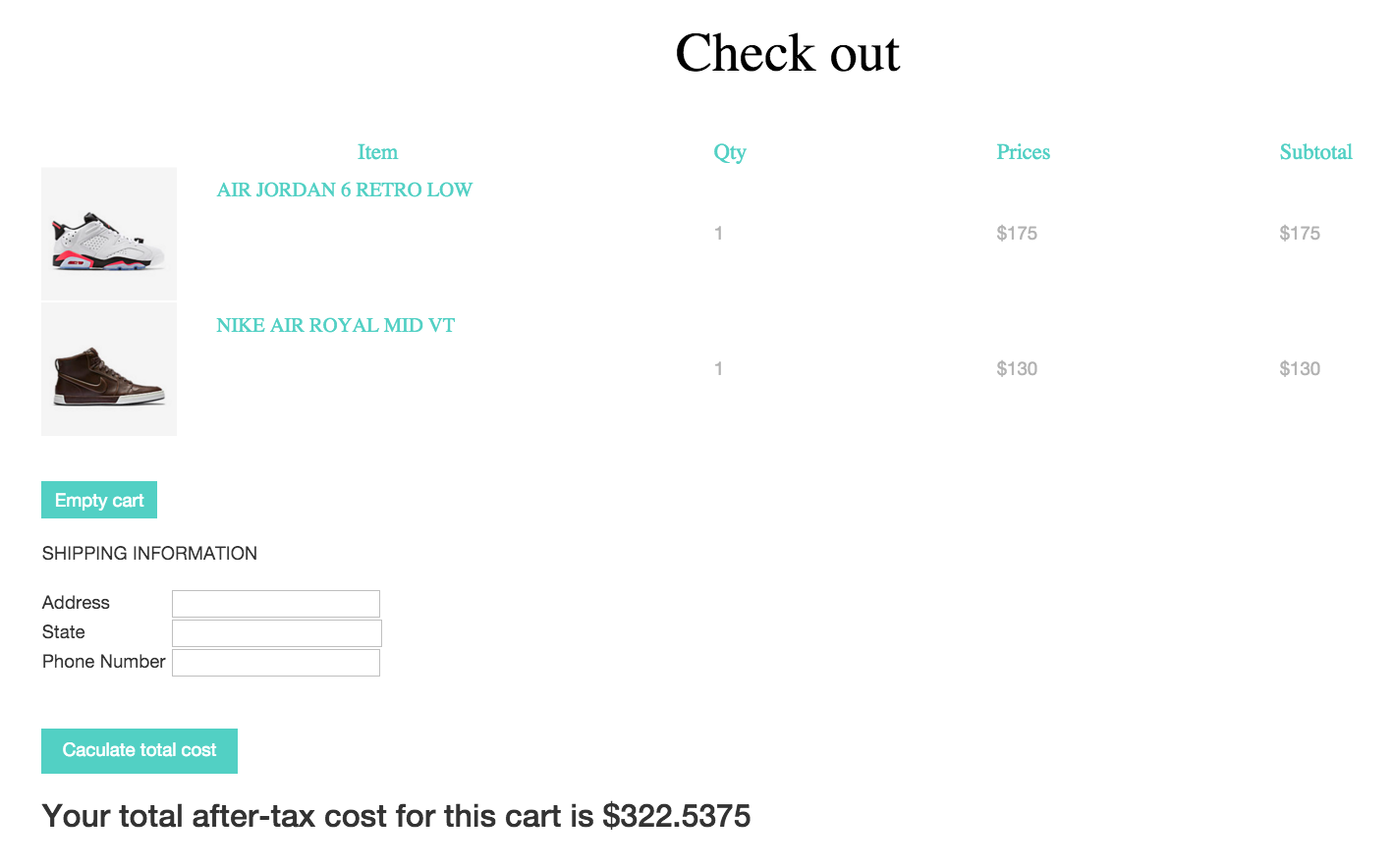
**Figure 6.** Output the total cost of the cart

**Screenshots of the Website**









**Conclusion**

Before this project, I don’t have much experience on web development, so I spent a lot of time learning HTML and PHP before starting. In the process, I find web development can be very funny and I certainly enjoy building my own website. For the last several lectures, I’ve learned a lot about software architecture and its importance. For all projected I worked before, I never spent time designing and chose architecture for the system and just blindly start coding without any plans ahead. In this project, however, I documented before coding, which is very helpful in the whole process. Although it is a little time-consuming to document first, but spending time analyzing and thinking about the project that I was to build really makes me understand the project more and makes coding really easy. Software architecture is already helpful for me to implement a relatively small project, I don’t doubt its importance for large-scale projects that involve lots of people.