SDEV 300 6381 Building Secure Web Applications (2178)

JOHN DELANEY

FALL 2017

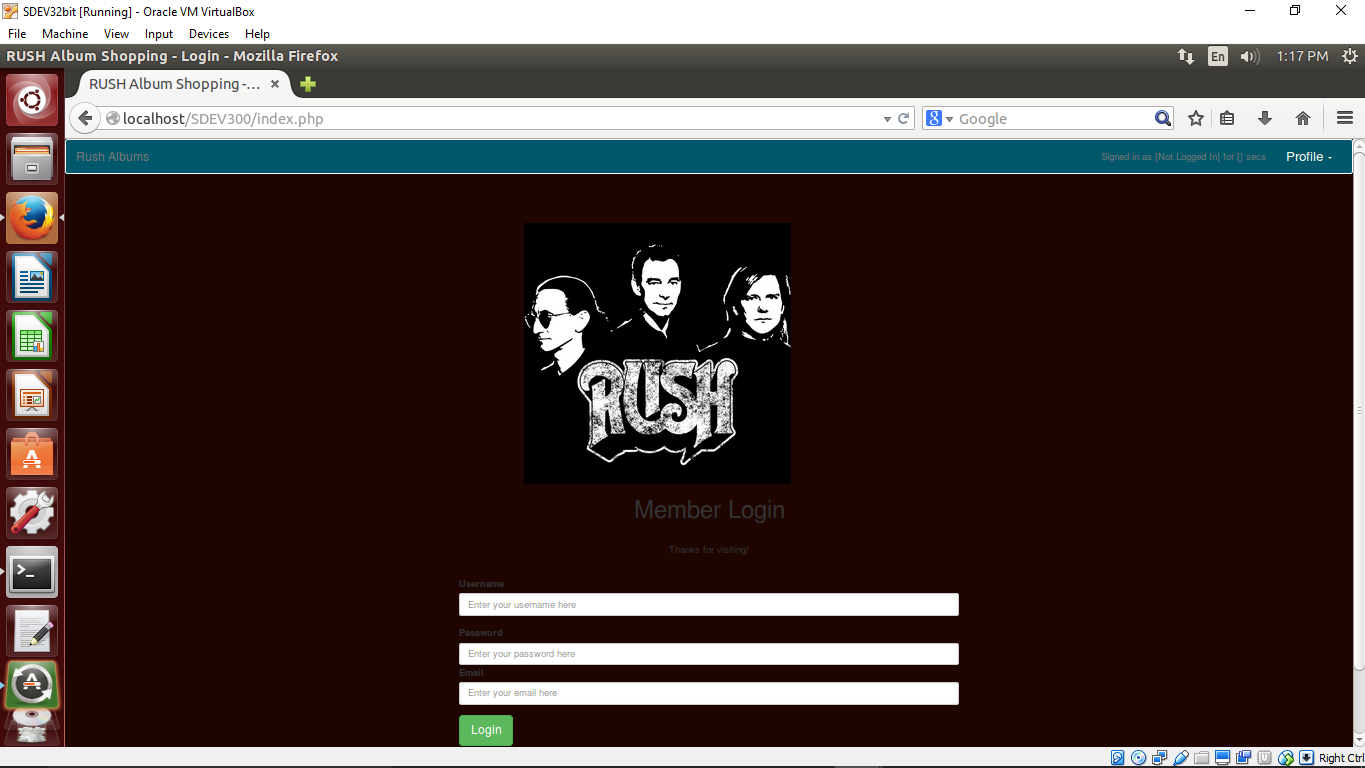
9/17/2017

**Description**

I enhanced the small website that sells Rush albums from Lab 2. The php server side portion has been added to enhance the functionality to match the requirement of Lab 4. Session variables were used in the login process and session timing portion of the website. After 30 minutes or 1800 secs the session will expire and return the user to the index.php page to login again. All session variables are destroyed after the expiration. The website was built on the LAMPP/XAMPP stack on Linux and Windows. GitHub was the version control system used to share the information between the two servers. The reason it was developed this way was the Virtual Box and Linux image runs very sluggish on my older laptop. All screenshots and testing was conducted on the Linux and LAMPP stack. The GitHub repository is <https://github.com/sd19surf/SDEV300> all changes and updates were committed in the event the virtual system had to be restored or reset. Bootstrap, jQuery, CSS and JavaScript were used in the creation of this website. Some features are available in HTML5 only and are not currently supported in the Firefox browser. The one feature of note is the date selection for the order form.

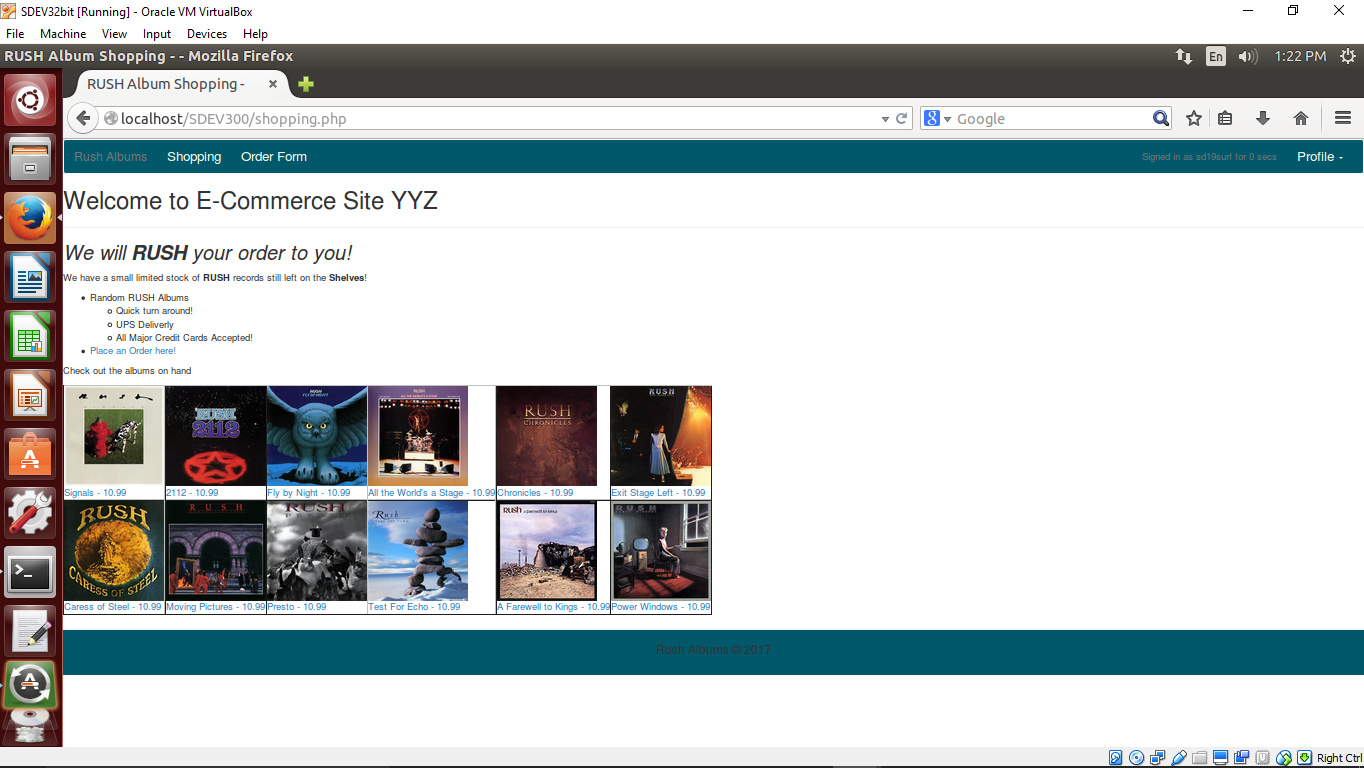
**Screenshot 1.**

Index.php has the login information and creates the session once the user clicks the submit button. The password field is available but the comparison feature to a database is lacking since there were no requirements for a database backend. Any password will work when entering the site.



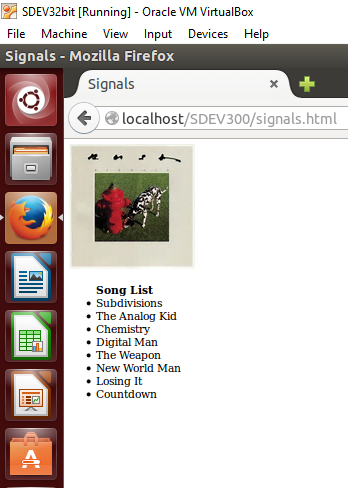
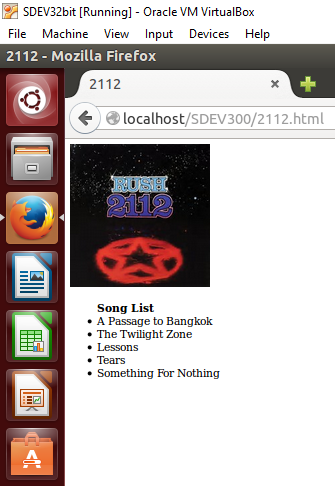
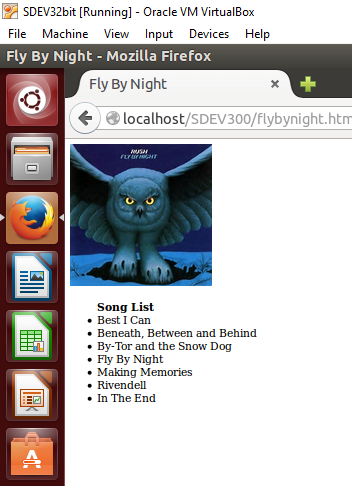
**Screenshot 2.**

This screenshot shows the result of logging in. The username is displayed at the top right in the navigation bar. The links to the Shopping and Order form are displayed in the navigation bar and the page is redirected to the Shopping.php page.

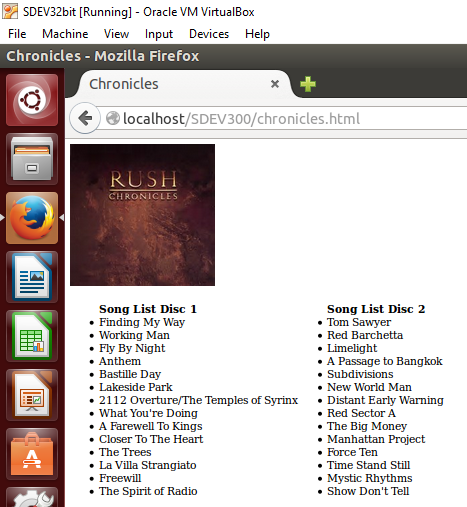
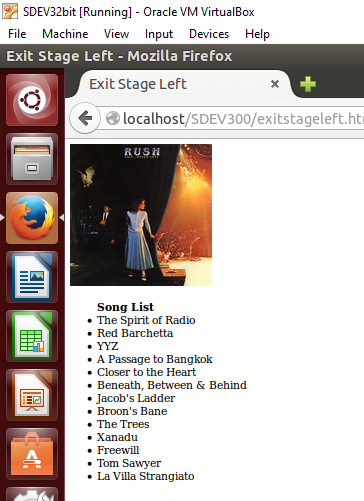
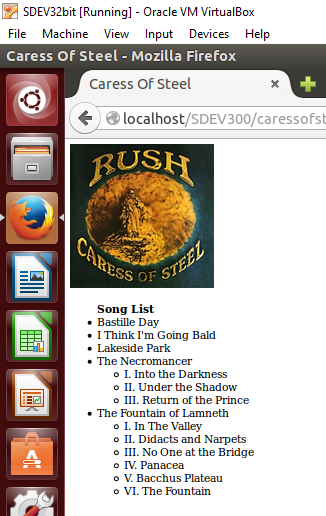
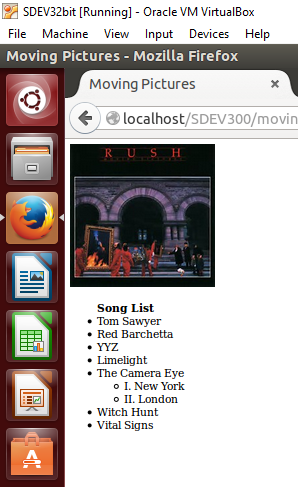


**Screenshot 3**.

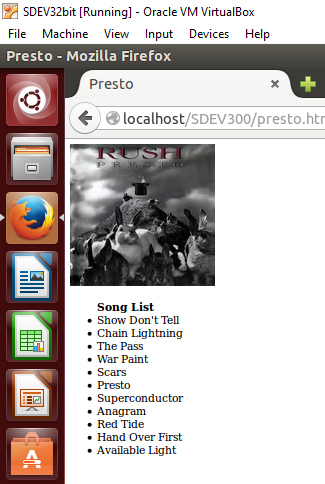
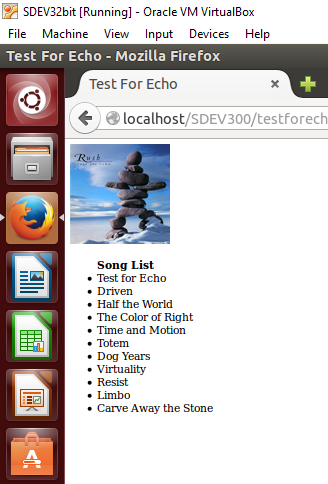
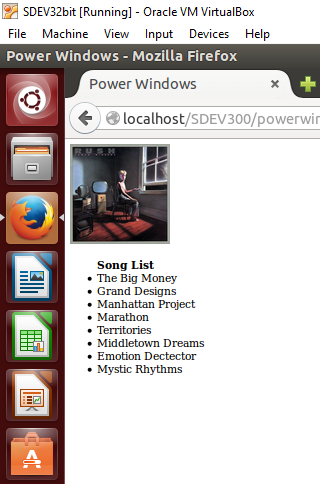
These screenshots are the sub-pages that are linked to the pictures on the index.html. To save time and scrolling smaller screenshots will be used to show all 12 sub-pages used.

Signals.html 2112.html flybynight.html alltheworldsastage.html

chronicles.html exitstageleft.html caressofsteel.html movingpictures.html

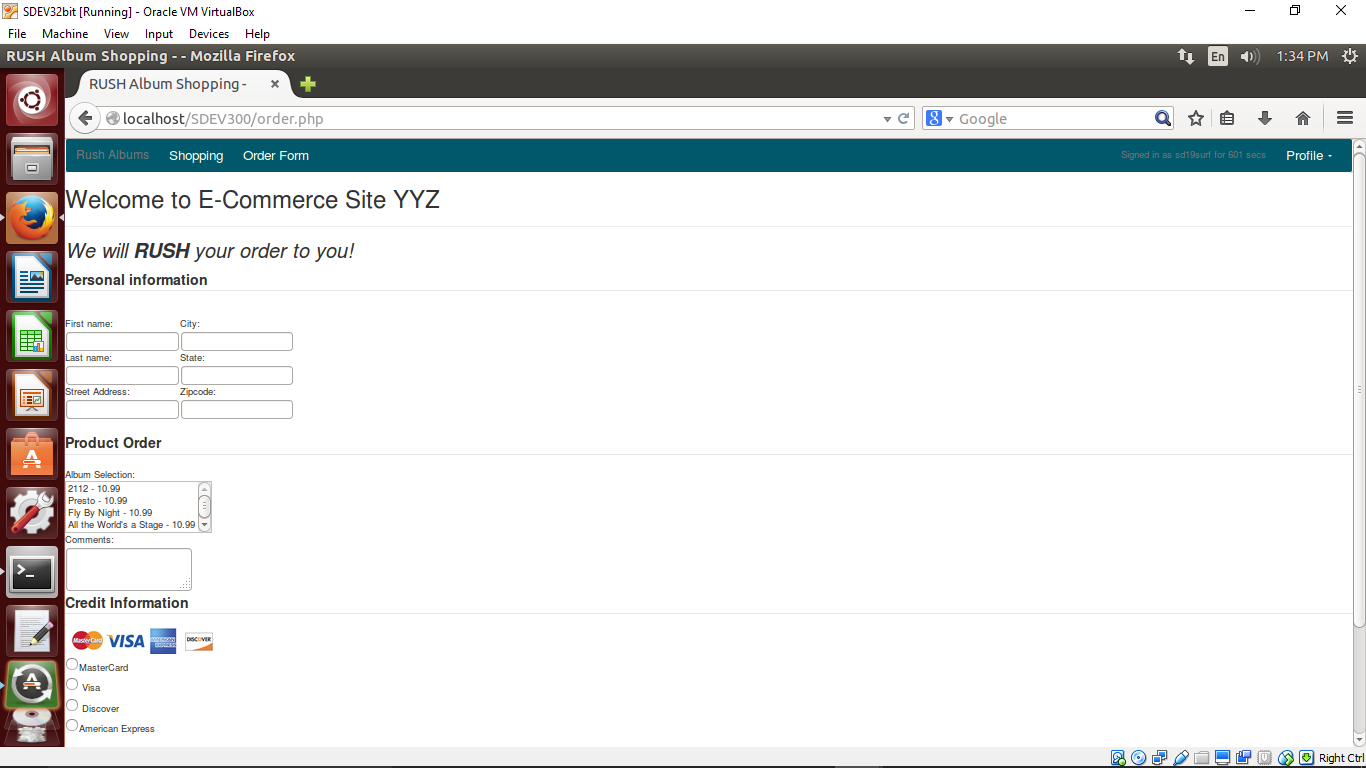
   

presto.html testforecho.html afarewelltokings.html powerwindows.html

There was a internal debate on whether I wanted to allow customers to purchase or add to cart from these pages that popup when clicked. I choose not to so that customers did not end up surprised at the purchasing page with purchases they did not intend to purchase and to avoid building a separate method to remove selections from the cart. JavaScript onClick is used to display the mini-pages from the shopping.php screen.

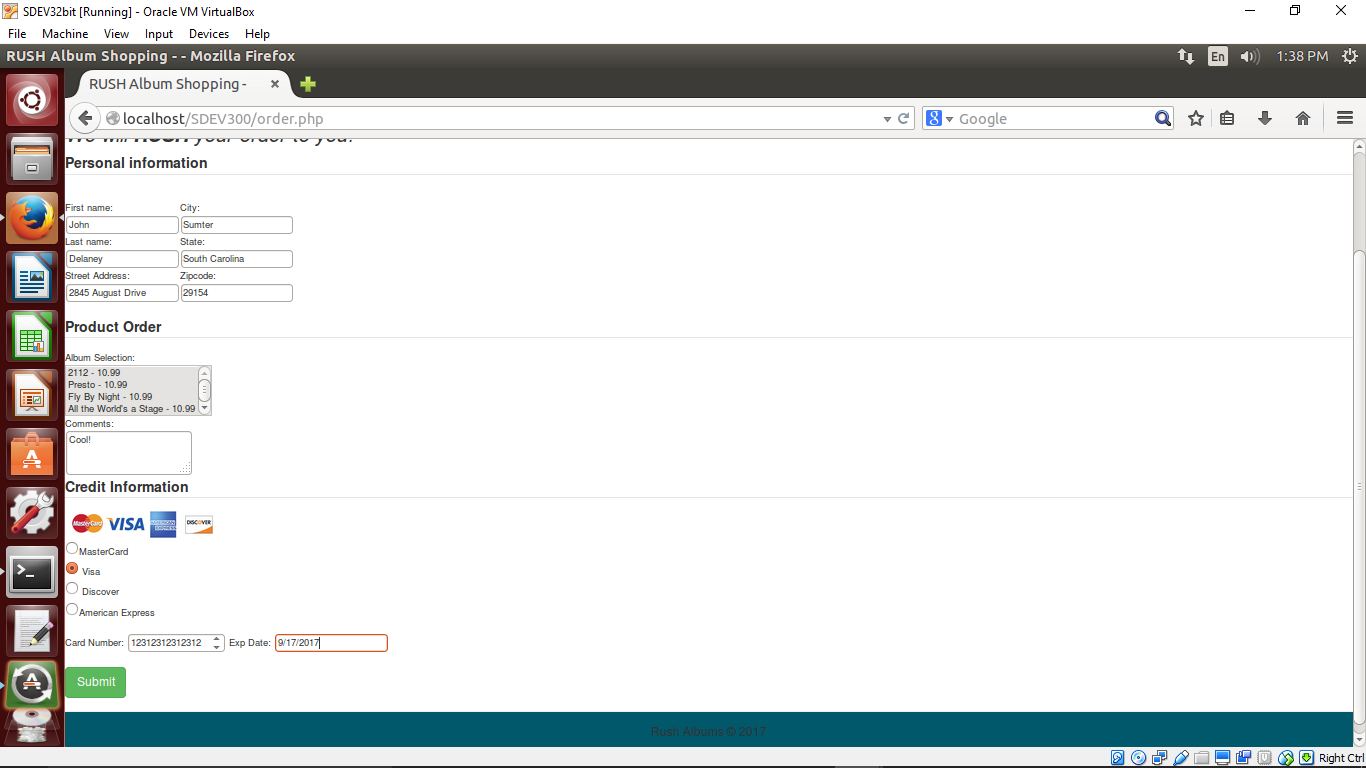
**Screenshot 4.**

Clicking the Order Form link in the navigation bar brings the customer to ordering page to select the items for purchase.



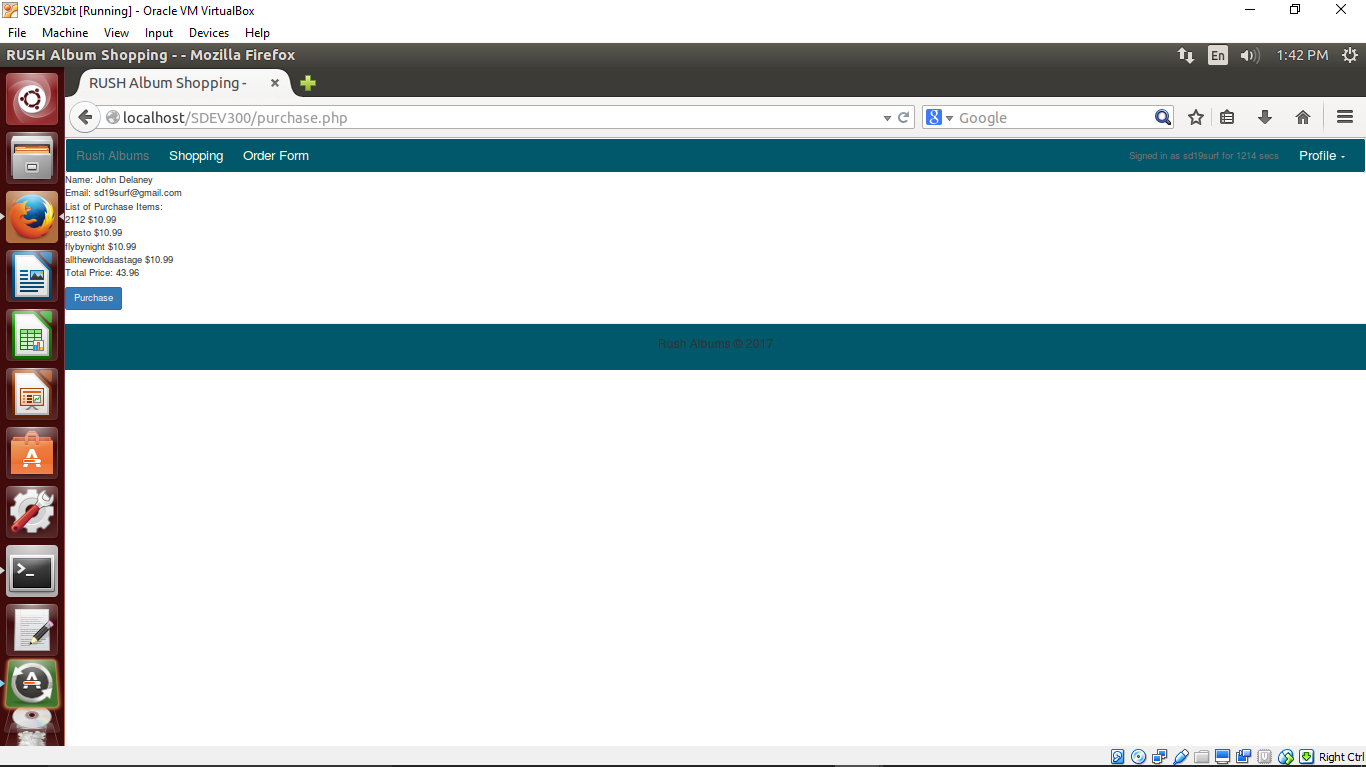
Screenshot 5

This screenshot shows the information entered before submitting the information to the purchase confirmation page. Here I selected the first four albums. All the albums are the same price this was a carry over from the previous assignment so it wasn’t done intentionaly. If the products were the same price I would carry over a product/price pair array over and process that to gain the same result.



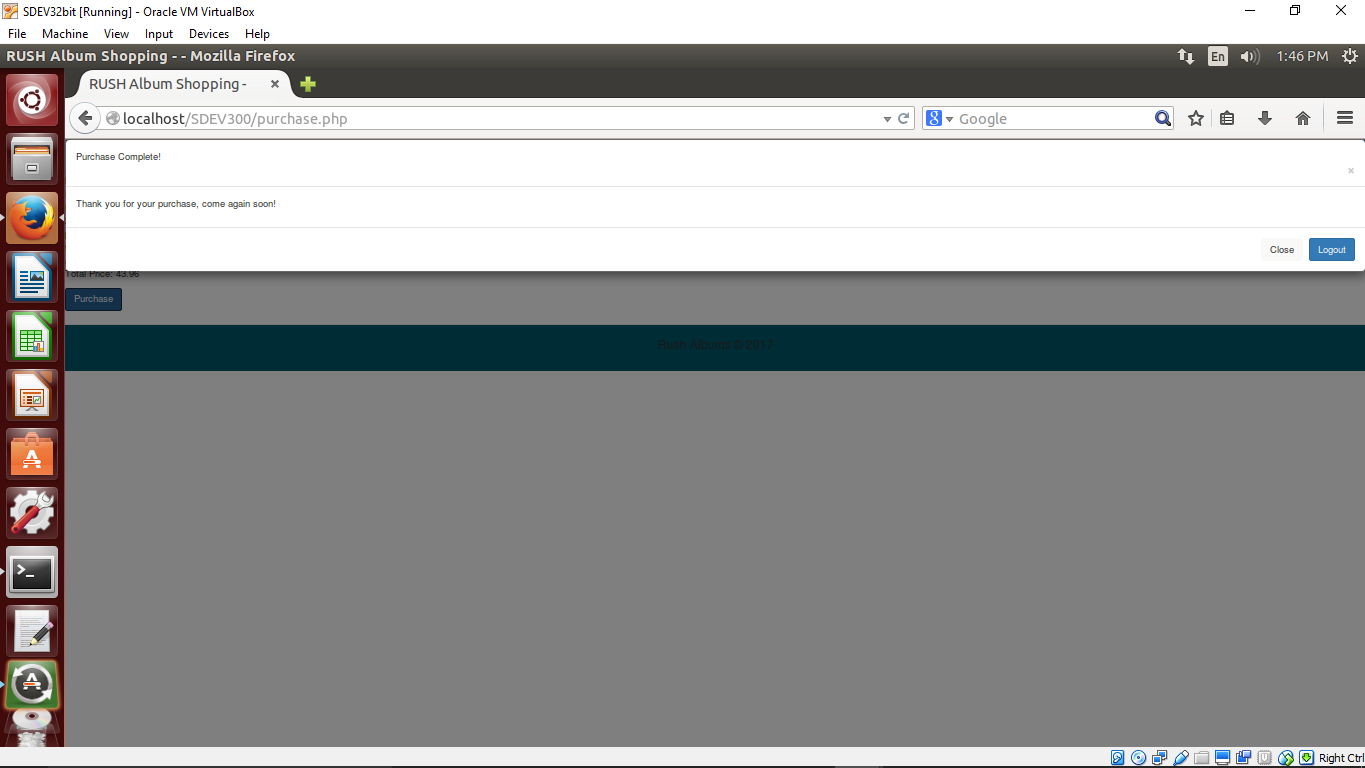
Screenshot 6.

Users are redirected to a purchase screen to confirm their purchase or back away if they so choose. A better display could be warrant here, however users are usually in a hurry to finish the purchase and only need a quick printable receipt.



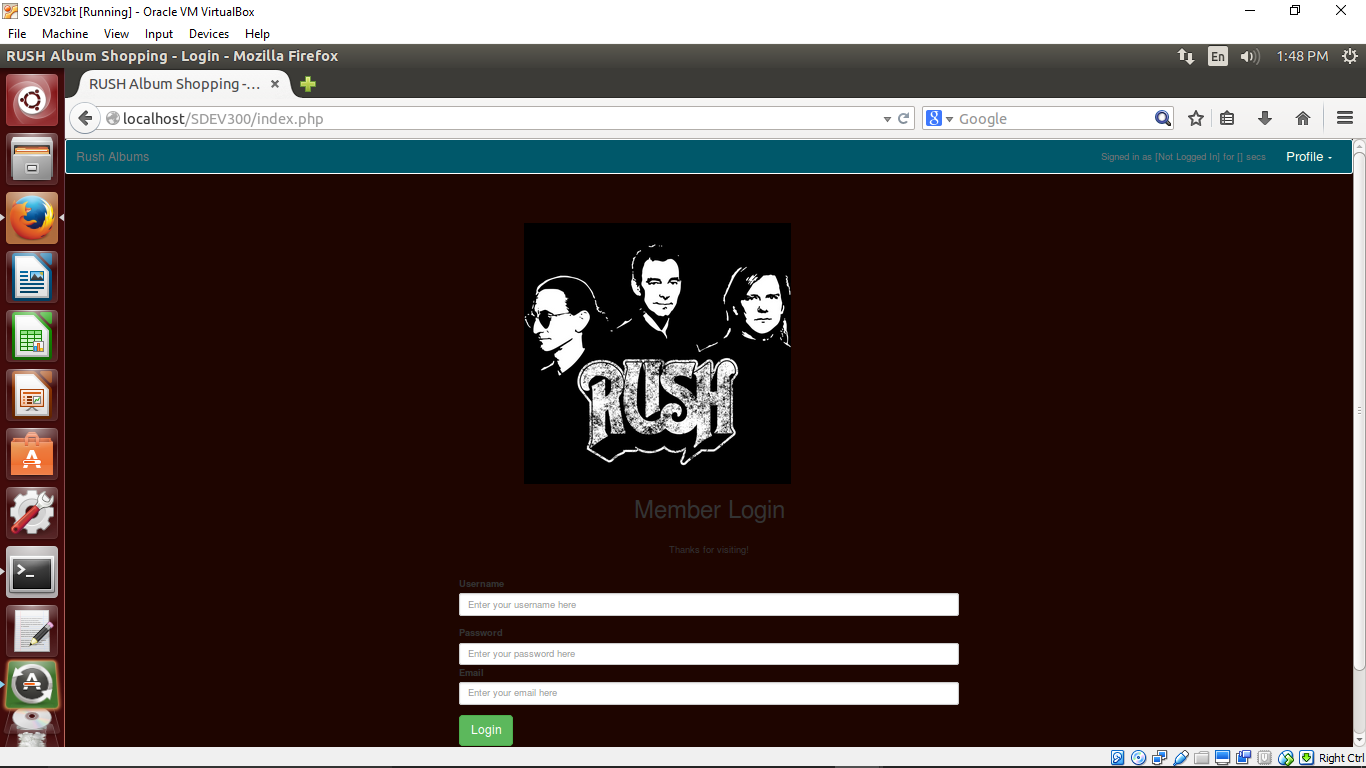
Screenshot 7

This screenshot is the result of clicking purchase. The modal shows Thanking the user and offers a logout or close option.



Screenshot 8

The clicking of logout redirects the user to the index.php screen and clears all session variables.



**Review of access.log**

Reviewing the access logs shows the POST and GET, I used POST for all SESSSION variable setting and password transfer so it does not show up in the logs and in the URL. I also put an unsalted MD5 hash on the password to ensure a plaintext password was not passed. The index.php, order.php and purchase.php all had the POST annotated in the log and that was expected. All the development errors were mostly malformed variables, a missing semi-colon, improper setting and looping of arrays. These showed up in the error.log of the XAMPP stack.

**Lessons Learned**

The most challenging portion of this lab was the session variable expiring. As you will see in the code I placed my time in session next to the username in the navigation bar to make sure I was tracking the right amount of time and to make the expiration logout the user, clear the variable states and redirect them back to the index.php to login again if necessary. The most fun was to pass a list of items through POST and set them to an array and loop through it for the shopping cart. A little tricky and I neede reference PHP documentation coupled with the HTML5 documentation on using a mutli-select list on a form. I feel a bit more comfortable with the session variables and will implement some of these practices on a system at work.