General:

Our header displays dynamically based on the identity of users. Different identities have different access as being customers have no access to administrator, and customers cannot access a user profile. Accessibility control is implemented in the head of every site which keeps track of accessibility by session variables. If conditions are not passed, the site will stop displaying more sensitive contents.

Main-page:

In this project, our homepage lists and displays images of products dynamically based on the products in the database. The products can be folded into categories, so the website could be more organized if many products are existing the database. Also, if customers have placed ordered in recent 15 days, there will be a category called popular products. The popular products are ranked based on the amount sold in 15 days, and there will be maximum ten items placed in the section. Moreover, the page utilizes the prepare statement, so it also prevents SQL injection attack. Features above are produced by "main-page.php," which retrieves all available product information from the database and prints them upon the homepage. "Userview.js" unfolds products from category bar or folds them into category bar.

Register:

Our register page requests various user information, and it includes uploading a user icon. The user image is stored in the database as a blob because it is restricted to be small and flexible. The textarea blocks are checked from JavaScript as well as php. Therefore, it is secured that registration is valid for both client-side and server-side usage. Moreover, the user validation checks if such user exists in our database via username or email address, because username and email are designed to be unique in our database. The validations are mainly produced by "process-register.php" and "RegVal.js."

Login:

Similarly, we validate username and password front-end and back-end so that it ensures the existence of the customer. Also, if a customer forgets his/her password, "forgot password" link can reset the customer password in the database and sends a notification to the customer's email address. The features are accomplished by having "process-login.php," "login-screen.php," "LogVal.js" and PHP Mailer.

itemList:

Our itemList not only displays products based on the category of the product which selected from the Category List on the left side of the page but also displays products based on the top search bar.

ItemList can handle searching for category, description and name of the product from the search bar. Moreover, it is implemented with prepare statement, so it prevents SQL injection as well. For different identity of users, the itemList also provides different access. For instance, a guest or admin cannot add an item to their cart because they

have none, but customers can add products directly and asynchronously to their carts. The features above are produced by “itemList.php,” “category-list.php.”

Item-page:

Item-page shows products’ image, description, price and comments correspond to the information stored in the database. We have restricted the user to view the existing products, which means if a user searches pid = 100, and there is no such item, the page will notify the user as accessing an invalid item page. Item-page also runs AJAX for the comment section, which is produced by using “item-ajax.js,” “comment-api.php.” Also, our comment section checks users’ comment status which could be disabled from admin. A guest and banned customer cannot comment on products, and the comment block notifies user’s comment status dynamically. The feature is accomplished in “item-page.php”.

Checkout:

Checkout page shows the products exist in the customer’s cart. It displays item’s information and provides a link for the customer back to the item-page and review the product. The checkout page also allows the customer to remove or update the number of certain products. Meanwhile, the subtotal of the cart changes dynamically. The feature is achieved by using

Checkout-quantity.js which performs AJAX update on quantity, and checkout.php which displays the subtotal dynamically as if the cart has updated.

Admin-main:

We use Google API to visualize the market shares of each product customers have ordered. The Google API takes a result set of product and generates a pie chart. Also, our main admin page can view all the orders submitted from customers using “listorder.php,” which prints a table of order summary for each placed order. Moreover, our admin can reset the entire database by clicking “reset database” button in main admin page, which is not recommended, but it is available.

Admin-user-control:

Admin-edit-item:

We use “edit-item-admin.php” and “process-edit-item-admin.php” to remove/ update the existing product in our database. We use prepare statement for each block of updating so that the database is safe from SQL injection. We validate admin input for product information so that admin must need to know the product id from the database to perform a valid update. Also, we validate the size of the product images and extension so it will not take up too much upload stream. If a new category is created while updating the product, the left side category list will update dynamically as well. Similarly, admin is

required to know the correct product id to perform deletion of the product. If such a product is the last one in the category, such category will disappear in the category list as well.

Admin-add-item:

We use “admin-add-item.php” and “process-add-item-admin.php” to insert a new product into our database. We validate admin input for product information so that there will be no blank on the description, price, name and image for the new product. Also, we validate the size of the product images and extension so it will not take up too much upload stream. If a new category is created while inserting a new product, the left side category list will update dynamically as well.

Limitations:

1. In our user control’s search, the admin has to copy the comment in advance to edit a comment at the current stage.
2. Our registration page is lack of validation for detail, for instance, the postcode input does not validate format, and new users can write random city, a state which location does not exist in the real world.