Documentation

Comp 3340-Group 34
Brittany Vanderlip, Selin Ozoglu, Ivana Jamina

For Reference:

The entire website can be viewed and interacted with on **ecomall.vanderlb.myweb.cs.uwindsor.ca**

Note: Admin username**: adminUser**

password: **group34** for access to Admin Panel

Database setup:

1. On your account through myweb.cs.uwindsor.ca, setup a new database
2. “Account Manager”-->”My SQL Management”-->”Create New Database”
3. Create your new database name
4. Before moving on, a popup of your username and password will be revealed. Use this information to fill in $dBUsername, $dBPassword, and $dBName in “**dbh.inc.php**” (Replace the “YOUR_DB….”) and save. The file has been left blank purposefully as to not breach any confidentiality on our own database.

```php
<?php
$dBServername = "localhost";
$dBUsername = "YOUR_DBUSERNAME";
$dBPassword = "YOUR_DBPASSWORD";
$dBName = "YOUR_DBNAME";

// Create connection
$conn = mysqli_connect($dBServername, $dBUsername, $dBPassword, $dBName);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
```

5. “Extra Features”-->”phpMyAdmin”
6. On the left hand panel, click on the database you just created
7. Click on the SQL tab at the top and copy in the sql command in **users.sql** to create the user table. Click “Go” on the bottom right.

8. Run the following SQL code to get started with a basic database template to be used by this website. The following sql codes are also stored in **inventory.sql** and **ratings.sql**:

```
CREATE TABLE inventory (
  productID int(3) PRIMARY KEY NOT NULL,
  productName VARCHAR(25) NOT NULL UNIQUE
);

INSERT INTO inventory
VALUES(1, 'Boxed Water');

INSERT INTO inventory
VALUES (2, 'Wooden Cutlery');

INSERT INTO inventory
VALUES (3, 'Reusable Coffee Capsules');

INSERT INTO inventory
VALUES (4, 'Eco Soap');

CREATE TABLE ratings (
  ratingID int(11) AUTO_INCREMENT PRIMARY KEY NOT NULL,
  idUsers int(11) NOT NULL,
  productID int(3) NOT NULL,
  Rating int(1) NOT NULL,
  comment VARCHAR(100),
  FOREIGN KEY(productID) REFERENCES inventory(productID)
  ON DELETE CASCADE,
  FOREIGN KEY(idUsers) REFERENCES users(idUsers)
  ON DELETE CASCADE
);
```

Domain setup:

1. On myweb.cs.uwindsor.ca, set up your own domain under “Account Manager”-->”Domain Setup”
2. “System Info & Files”--> “File Manager”-->”domains”-->your_website
3. Files *intended* to be in the private_html folder are the login.inc.php, signup.inc.php, contactform.php, dbh.inc.php since all of these files access the database directly.
4. The remaining files are meant to be in the public_html folder.
5. You should now be able to view your site!

Website Handling:

1. Before accessing any features on the site, it is important that the admin user is created on the site first. (it has already been created on our site with the credentials above, however since you are building this site from scratch you need to add it into the database first).
    a. Click on the Login/Signup button at the top and click on Signup
    b. Username **must be** "**adminUse**r" and create an email and password as you like
    c. No usernames can be duplicated on this website.
    d. Only adminUser can access the Admin Panel
2. A TINY MCE account must also be set up.
3. Create an account on Tiny cloud and copy and paste the tiny API key in the script for the tiny cloud in **contact.php**. Replace "qrcozjz7l1jua34j7bxytq22ucprsrzwxrwavotiig122d9j" with your own api key.
4. Add the domain name that you created the website for to your tiny cloud

Website Concept & Purpose:

The purpose of this website is to provide an e-commerce platform to the company Ecomall, so their customers may easily purchase products, track purchase history, and provide/see feedback to the store. It tracks all important user/inventory data and stores it into a database, making it consistent, portable, and easy to access. It is extremely helpful to customers on the user-end because it provides an extremely convenient way to shop, read feedback, provide feedback, and directly communicate further with a sales associate should they wish to do so. It is also extremely helpful to the company on the backend because it provides them with valuable consumer feedback and constructive criticism, makes using their products extremely convenient, and helps advertise their brand. Online presence is essential for nearly any modern business, and this website provides all the basic needs to satisfy this need!

The concept for this website is fairly straightforward. Users can easily browse through publicly available information on the site, such as reviews, or make themselves an account to log in and access even more exclusive features. With an account, users can additionally browse through the store's catalogue of items, shop, adjust which items are in their cart as well as the quantity of each item, and even place an order. Then, after receiving their orders, users can leave a detailed review of each item, and can always come back to edit, delete, or add reviews for everyone to see. Finally, users can also contact the administrators of the site directly by sending them a message. Administrators of the websites (who are the only people with access to these messages) can also read what potential customers have sent them and directly email them a response. This website

provides basic but essential services to any company that wishes to have an online, interactive shopping and feedback platform.

Website Walk Through:

Home Page:

Files: index.html, ecoMall.css, img.jpg

The home page is where you can direct yourself to other pages on the website. The navigation bar at the top of the website will take you to the various locations.

The “about us” section is also available on this page.

Login Page:

Files: loginHeader.php, login.inc.php, logout.inc.php, loginSignup.css, footer.php, index2.php

Here is where a user can login to their account if they already have an account made. If no account has been made, an error in the url will prompt the user and the page will read “you are logged out!”.

Once the user logs into their account they will be able to access the shopping page. Only authenticated users can access this page.

In addition, if the admin logs into the site, only they can access the Admin Panel (Username: adminUser).

Signup Page:

Files: signup.php, signup.inc.php, loginSignup.css, footer.php

On the Signup page, a user will enter in their username, email, and password. Any errors will redirect the user back to the signup page and auto-repopulate the fields that were correct.

Errors that are checked include if the username is taken, if the passwords match, if the email is valid, and if any of the fields are left empty. This information is stored into the database table named “users”.

Contact us Page:

Files: contact.php, contact.css, contactform.php

The contact us page is where a user can use the form to input their name, email, subject of the message and message. The text area for the message section is used using TINYMCE. Read “Website Handling” above to use TinyMce yourself.

The message is handled through a php file called contactform.php. The name, subject, email and message are all obtained and sent to an administrative user's email. If you would like to change which email this is sent to, go to contactform.php, and change $adminEmail = "vanderlb@vanderlb.myweb.cs.uwindsor.ca"; to the email of your choice.

The message will be received by the admin user's email once the "Let's Talk" button is clicked.

Admin Panel:

Files: adminPanel.php, adminPanel.css

The admin page is only accessible to the user "**adminUser**" which needs to be created when first setting up the site as indicated above. The admin user is able to view all the user information (username and email) that is stored in the user database on the Admin Panel page.

This page is not accessible to normal users.

Products Page:

Files: shop.php, shop.js, shop.css, products.json, bamboo.jpg, soap.jpg, coffee.jpg, boxed-Water.jpg

The products page is only accessible to users that are logged in. If the user is not logged in they will view the message stating that they are not logged in and they will be prompted to the login page. From the login page they can then continue onto shopping. Once they are logged in the products that are currently available will now be visible to the users.

The current products can then be added to the bag, and in the bag the quantity of the products can be adjusted up or down. In the shopping cart you can clear the cart or place an order. When you clear the cart all products are reset and available to shop again. When you place an order a prompt will pop up and ask if you are sure you'd like to place an order when the user clicks yes the shopping cart is emptied and the order is considered to be placed. If there are no products in the shopping cart at the time of placing an order an alert will come up informing the user. This is a page where users can create orders by adding to cart, modify the orders by changing the quantities, and remove products they no longer would like.

This page also involves non-AJAX javascript interactivity through adding products to the cart, changing quantities of products, placing orders and clearing the cart.

Reviews Page:

Files: reviews-all.php, reviews-load.php, reviews.css

The Reviews page is where any users (logged in or not) can see a list of all reviews within the database. The product, rating, user who reviewed it, their comments, and an image of the item are all displayed. Additionally, more comments can be loaded without refreshing the page by clicking the "Load More Reviews" button. This action is done using AJAX, and will continue to load 2 additional

comments until all are displayed. The reviews can also be sorted by the type of product by using the drop down menu. Again, this uses AJAX to load in the appropriate, filtered reviews without having to refresh the page.

My Reviews Page:

Files: reviews-myreviews.php, reviews-new.php, reviews-submit.php, reviews-edit.php, reviews.css, loginHeader.php

The My Reviews page is *only accessible to logged in users!* If one attempts to access this page without being logged in, they are prompted to the login page to continue the usual login process. If they are logged in, however, this page is similar to the Reviews page. The difference is that these reviews are filtered so that only the currently logged-in user's reviews are listed. Additionally, each review has the option to be edited or deleted. Upon deletion, the page is refreshed to show one less review and the database is updated. Upon editing, the user is taken to an Edit Review page where they can update their comment and rating value. If successful, they will again be brought back to an updated version of their personal review page, and the database will be updated. At the bottom of the page, there is also an "Add New Review" button. This button leads to a page similar to the Edit Review page, called the Add Review page. Here, the user can additionally choose which product they would like to leave a review for using a drop down menu. In both the Add and Edit Review pages, AJAX is used once more to create an interactive star rating system, which follows the user's mouse and lights up the number of stars they chose to rate the product at. This value of lit up stars is then transferred to the database.

Website Security:

The files dbh.inc.php, login.inc.php, signup.inc.php, contactform.php, and logout.inc.php are intended to be in the private_html folder because they contain sensitive information. Dbh.inc.php contains the database username and password to access the database through the website. All the other files intended to be in this folder.

The files login.inc.php and signup.inc.php protect against sql injection. In these php files, prepared statements were used in order to prevent hard coded access to the user's information. Placeholders ('?') were used in the SELECT sql statements rather than the variables containing user's information and a binding function was used to bind the user's information to the SQL statement.

In addition, the function "mysqli_real_escape_string" was used when declaring the variables that hold the user's information. This is a security feature in order to ensure that in the login and signup form that input is read as text and not as code. This contributes to both html injection and sql injection protection.

Pages with user input text, such as the pages used to add and edit user reviews, also use the `filter_var()` function to sanitize the string (`FILTER_SANITIZE_STRING`) the user entered before it is processed. Users are also limited to the amount of text input they may contribute, and instead use drop-down menus and interactive AJAX segments to contribute the appropriate input.