

# Software Engineering Project: E-commerce Web Application

Kassie Garcia (Writer / Editor), Nkeiruka Nwogu, James Le, Azra Al Rabeeah (Main Writer),

Walter Schmidt, and Herman Garza, Members of Group 2, CS-3773-002 Spring 2022

**Abstract:-** E-commerce Websites are great for businesses, big and small, to allow their customers to buy / sell goods, services, and products over the internet rather than at a physical store location. In this project, we have developed an e-commerce web application for a coffee company named *Hug With Mug*. At Hug With Mug, the goal is to sell various coffee related items to customers from all over the world. The Hug With Mug web application offers customer functionality such as making an account, displaying best seller items, adding an item to a wish list, adding an item to a shopping cart, placing an order, etc. The Hug With Mug web application also offers admin functionality for the Hug With Mug store owner. This functionality includes viewing currently placed orders, adding/deleting/modifying products for sale, adding/modifying coupon codes, and much more! This application was designed by us as a group to not only help customers buy and browse Hug With Mug products, but also to help the store owner (admin) to manage the website and the orders with ease.

**Part 1a: Group Workflow:-** We started out as a group unsure of how to go about this project.

Many of us did not have any web development experience prior to taking this course and started off without a clue as to what developing a website entailed. Our project manager, Kassie Garcia, was chosen to lead our group, as it was discussed it would be best to have the student with development experience manage the project.

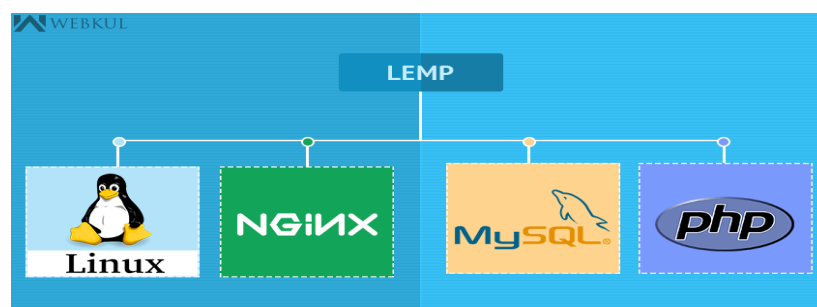
Since the beginning of February, our project manager has planned out sprints for every 2-week period. These sprints were planned in documents, which consisted of the group's overall goal for the 2-week period, as well as a list of the product backlog items to be worked on by the front-end & back-end developers. The group met every Wednesday of the week (over Discord) to discuss the product backlog items to be implemented, as well as any updates, questions, and issues we were going through. If any of the backlog items from the sprint were not finished, it would then get added as a higher priority item to be implemented in the next sprint.

At first, we encountered a few difficulties in deciding how to best develop the web application.

We considered using WordPress but was unsure as to how to implement the admin backend functionality when WordPress itself acted as the backend. We considered using AngularJS but found the framework to be more overwhelming than it was helpful. Eventually, we landed on

**LEMP.**

**LEMP (L-Linux E-Nginx M-MySQL P-PHP)** is the stack we used to develop our web application.



The **LEMP** stack is an open-source web development stack which is used to serve dynamic web pages and web applications. This means that for our project, we used a Linux operating system with a Nginx web server. The data for our project was stored in a MySQL database and the dynamic processing / functionality was implemented with PHP.

Most of the functionality was implemented with the language PHP, while we did use the combination of PHP/SQL to interact with our database, and we also used JavaScript for some client-side features.

To publicly host the website, we used an Amazon EC2 Instance to host our application on the Amazon Web Services Infrastructure. This means that only one of us (the project manager) made an EC2 instance, while everyone else worked on the project from their own local environments.

To share the project and keep the EC2 instance updated, we would push / pull the project files and the MySQL database file from / to our GitHub repository. This repository served as our version control software to collaborate as a group.

### **Part 1b: Group Project Roles:-**

At the beginning, we each had roles assigned to us at the start of the semester. However, these roles changed drastically as time progressed, and some members ended up contributing little to no work to the project. Here are the roles which reflect the actual contributions:

**Kassie Garcia**- Project Manager (management, write-up, and presentation slides), Main Project Developer for both front and back-end work

**James Le**- Worked on a document consisting of a list of products for sale

**Nkeiruka Nwogu**- Worked on the same document and designed some of our product images

**Azra Al Rabeeah**- Project Write Up

**Walter Schmidt**- No code contributed

**Herman Garza**- Worked on the search functionality

As this report continues, we will discuss these contributions in greater detail, as well as who exactly contributed to the work being discussed. This is to give proper credit for the work done.

**Part 2: The Creation of Hug With Mug:-** In our first sprint, we decided to base our e-commerce website on a made-up coffee company named *Hug With Mug*. We brainstormed items to sell, and landed on selling items such as coffee bags, tea, mugs, tote bags, and t-shirts. We decided that in order to have our UI reflect the theme of our company, it should then consist of a clean, simple look with mainly white, brown, and green colors. We also decided that we will need to gather 15 products to sell on the website, as well as create the imagery necessary to contribute to the website design. It took some effort to collect these as images, and we sought to collect ones that best suited our store's concept. We also designed our own apparel and display photos that highlight the items we now offer for sale.

Here is the logo image, designed by Nkeiruka Nwogu, as well as some of the products curated by both Nkeiruka and James Le:



**Part 3: UI/UX Design:-** As discussed in part 2, we decided that the UI theme should be kept simple and natural in terms of design elements. And we focused on using mainly white, gray, brown, and green colors. For the UI, we used **bootstrap** / other CSS styles to design our website. This means that many of the style classes we used are already existing classes offered in bootstrap. We decided to use bootstrap because it is typically used to make web applications responsive and reactive, which was a big goal for us. This meant our website will then look good on a variety of screen sizes- which would ensure the usability and satisfaction of our website. However, we had some difficulty using bootstrap for our design. Some design elements, such as forms or the navigation bar, were easily implemented with bootstrap and could be easily scalable to any window size. But when it came to our product blocks (the blocks which display the product image), we could not get them to be as scalable as intended without them looking extremely “wonky” and out of place from their style grid. However, developer Nkeiruka Nwogu made her best efforts to make the website responsive and reactive from the styles and design Kassie Garcia initially implemented.

**Part 4a: Functionality:-** As discussed in our abstract, our goal with developing our web application was to help customers buy and browse Hug With Mug products, and to help the store owner (admin) to manage the website and the orders with ease. In **Part 4b**, we will discuss the **User Experience**. This User Experience will consist of the details surrounding the functionality and web pages implemented for customers to use. In **Part 4c**, we will discuss the **Admin Experience**. This Admin Experience will consist of the details surrounding the functionality and web pages implemented for the admin user (store owner) to use.

Note here, that *all* the functionality we will discuss was implemented by Kassie Garcia. Except for the search functionality, which was implemented by Herman Garza.

## Part 4b: Functionality: User Experience:-

### 4b.01.01: Registering a New User

When a user would like to create a new account, they can do so by clicking on the “register” tab on the top left of the homepage or by clicking the “Register Here!” link in our login page. Our website will take a user’s full name, unique username, unique email, password, mobile phone number, and home address. In our code, we cross check with our database table customers and ensure there is not a user that already exists for the given username or email.

If all is good to go, we will update our customer database table with the entered information, encrypt the password for safety concerns, and inform the user that they have been registered successfully. If there was an error, the user will be redirected to try registering again.

Register A New Account

Enter your full name:

Enter a Unique username:

Enter your email address:

Enter a Strong password:

Confirm Password:

Customer Mobile Number

Customer Address

Register

Snippet of our *registration form*, which takes in all the user input for creating a Hug With Mug User

customer_id	full_name	username	customer_email	customer_pass	customer_contact	customer_address
10	Kassie Garcia	hamtaro	kassiejgarcia@gmail.com	\$2y\$10\$YitACqQ5Ijd2UTYgYAqv2.sdCfbF3vKL09.72v9p8St...	210210210210	san antonio
11	demo customer	demo	demo@demo.com	\$2y\$10\$RMKg94c/zXaDH5LCgzqdOO9bmP6iS/ie9bC9vejaz8Y...	210102102	280 Brighton Avenue
12	frog	frog	frog@gmail.com	\$2y\$10\$mBfoOVMpgGAO3344XMsE8utkgLKTAgHlpJbEtAIKOsg...	555555555	832 Kinwest Parkway

Snippet from our *customers* database table, which stores all the information on a Hug With Mug User

### 4b.02.01: Login an Existing User

If a user would like to sign in with an existing account, they can do so by clicking on the “sign in” link on the top left of the homepage. Users may even get redirected to the sign in page if they are trying to add a product to a wish list, or if they are trying to checkout without signing in. In our sign in page, we take in the user's username and their password, then we cross check with our database table customers to ensure a given user exists for that username and that the password given matches the encrypted one in the database.

If there was an error, the user will be redirected to try logging in again. If all is good to go, a session is created for that user, and they will be redirected to their account page. Once signed in, the user can now access certain features (i.e., checkout, wish list, account page, etc.)

### Login

Are you a Hug With Mug Customer?

Sign in down below with your Hug With Mug Username and Password!

**Username**

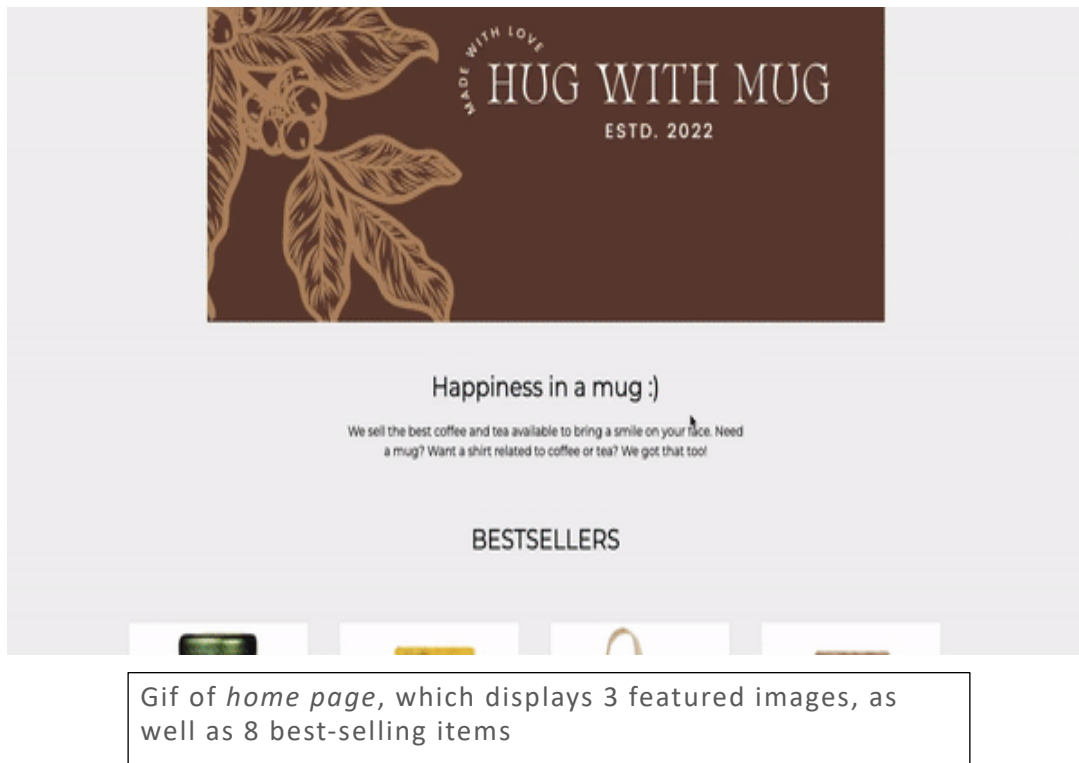
**Password**

Are you a new customer? Register Here!

Snippet of our *login form*, which takes in a username and password from user input

#### 4b.03.01: Home Page

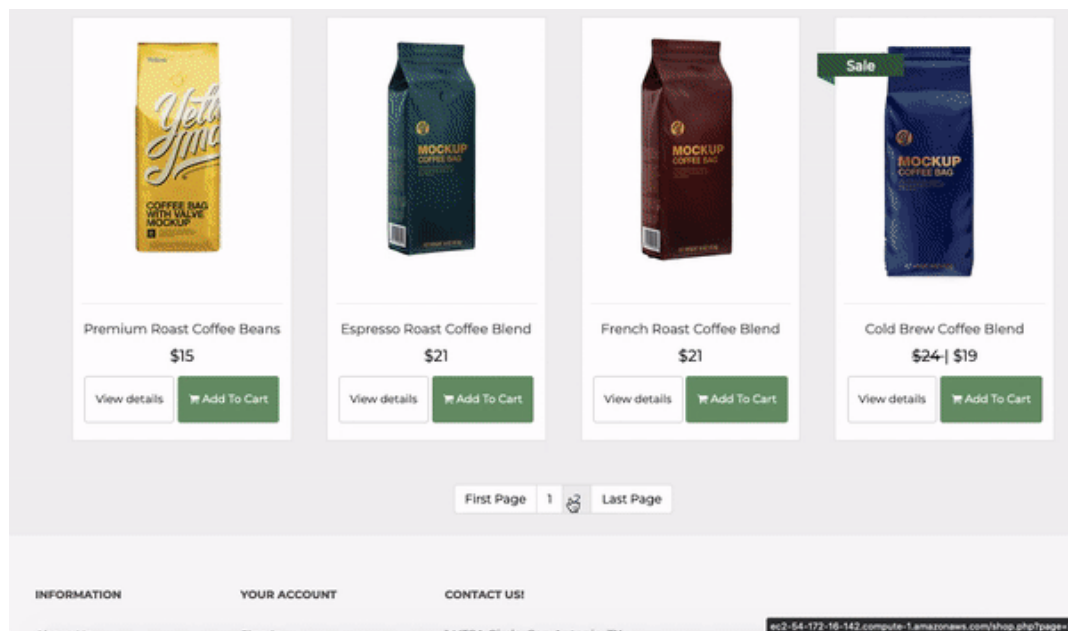
Our homepage displays a slideshow consisting of three images showcasing the products that we sell. Below this, we show our “bestsellers”- which consists of 8 products shown in display blocks. These blocks include the first product image, a banner if a product is on sale or not, a view details button, and an add to cart button which links the user to the product display page.



#### 4b.04.01: Shop All page

The shop page displays all the products we have stored in our products database table. These products are all displayed in blocks. These blocks include the first product image, a banner if a product is on sale or not, a view details button, and an add to cart button which links the user to the product display page. There are 8 products to a page. The user can also switch back and forth between pages via the *paginator* to view all the products.





Gif of *shop all* page, which displays 8 product blocks per page

#### 4b.04.02: Coffee / Tea / Merch Pages

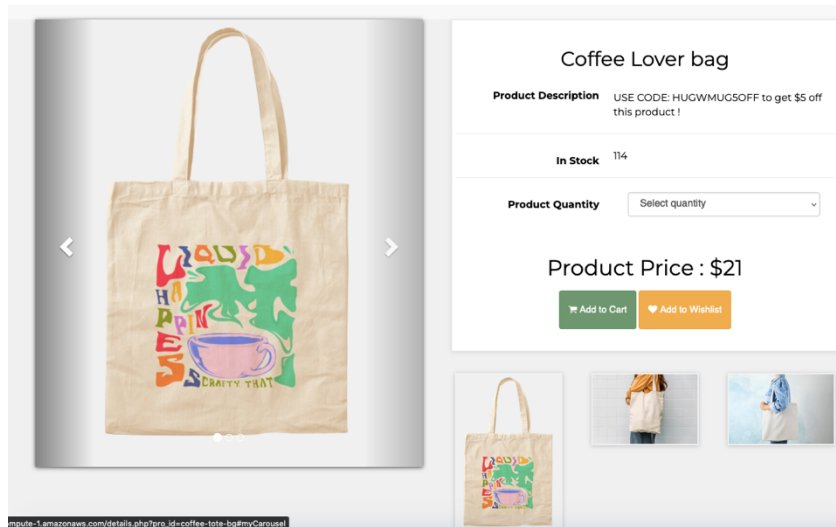
They are all simply the shop page, except they are filtered by their respective categories, which are indicated in the products and categories database tables.

For example, for the *coffee* page, we pass in the URL variables: `page=1&cat[]=1`. This indicates that we should display the first page in which the category is equal to 1. In our database, we stored *coffee* to be category 1, so this means we should only display coffee related products. We know a product is a coffee product if it is stored with `cat id = 1` in the product database table.

#### 4b.05.01: Product Display Page

Our product display/details page displays all the information about a single product. Depending on the “product URL” stored in our products database table, we will display the corresponding products title, description, quantity in stock, a form selection for the user to select the quantity they desire, the product price, the sale price (if on sale), the add to cart button, the add to wish

list button, and 3 “recommended” products that the user may also click on. We also display the 3 product images for the corresponding product. These 3 images rotate amongst a carousel that the user can traverse through.







Snippet of *the product display page*, which the details surrounding a single product

#### 4b.06.01: Add to Wishlist

Anytime a user is visiting a product’s display page, they have the option to add the product to a wish list. If the user is not signed in, but clicks “add to wish list”, they will be redirected to login with their account credentials. If the user is signed in, they will be notified that the product was added successfully. Then, the user may see the product in their wish list by visiting the ‘my wish list’ page, located in account settings. From there, they can see an image of the product, the product title, and they even have the option to delete the product from the list. This functionality interacts with the wish list table in the database.

## My Wishlist

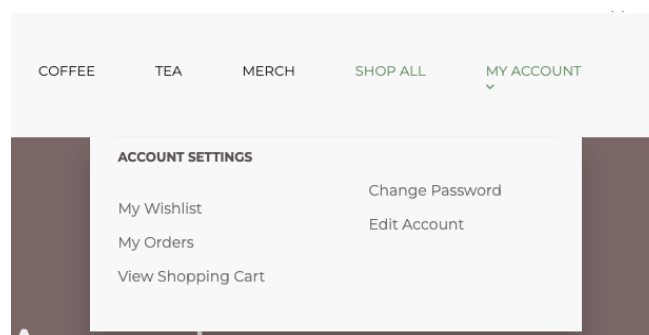
Your all Wishlist Products on one place.

Wishlist No:	Wishlist Product	Delete Wishlist
1	 Coffee Lover bag	<a href="#">Delete</a>
2	 Organic Lemon Tea	<a href="#">Delete</a>
3	 Coffee & work mug	<a href="#">Delete</a>
4	 Whole Bean Espresso Blend	<a href="#">Delete</a>

Snippet of the 'my wish list' page, which lists the products stored in a customer's wish list

### 4b.07.01: My Account Features

Once a user is logged in, they will be able to access a link to different account features through the "my account" drop down menu. The user can view a list of all their placed orders- which gets its information from the customer\_orders database table. They can change their password, they can view their wish list, they can edit their account, and they can log out as well.



Snippet of the 'account settings' drop-down menu

### 4b.07.02: My Wishlist Page

As talked about previously, the user can view their wish list saves on this page. This page is accessible through the account settings drop down menu or sidebar.

#### 4b.07.03: View Orders Page

The user can view a list of all their placed orders- which gets its information from the customer\_orders database table.

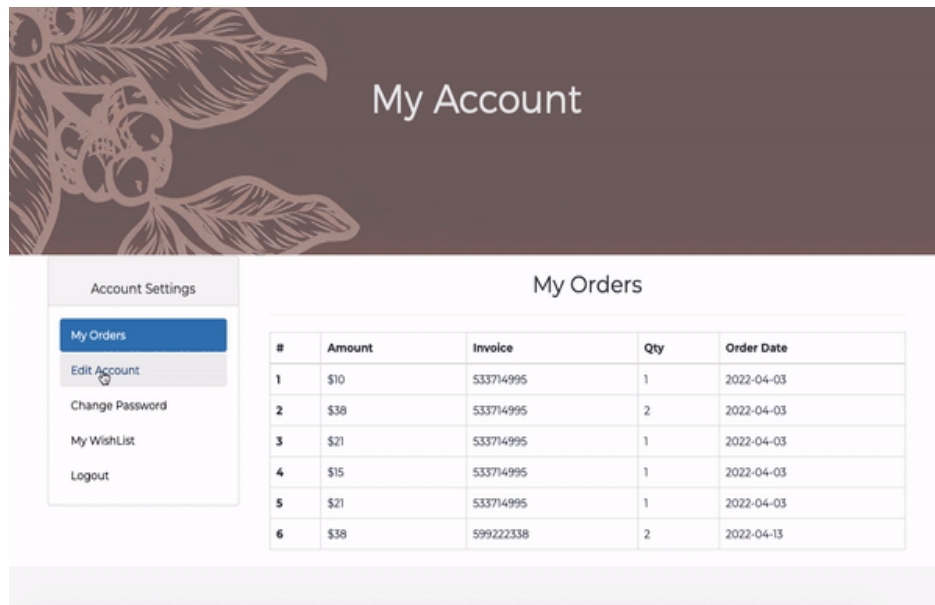
This list displays a list of product purchases, their corresponding order/invoice number, the quantity purchased, and the order date.

#### 4b.07.04: Edit Account Page

The user can edit their account. Their account information will be in a pre-filled form, then they can edit this information to their liking, then submit the changes. After the submission, the user will be prompted to log into their account again.

#### 4b.07.05: Change Password Page

Here, the user can easily change their password. Once submitted, the new password will be stored in the customers database for that existing user. Then, the user will be prompted to log in again.



Gif of the account settings panel, which is where a user can access various account features

#### 4b.08.01: Add to Cart Functionality + Order Summary

When the user is on the product's display page, they have the option to add the product to their cart. They can add to their cart even if they are not signed in, but they will eventually have to sign-in to checkout. When the user "adds to cart", we are updating the cart database table which will store the products added, their quantity, and an ip address which corresponds to the user's ip.

To visit their shopping cart, a user can click on the cart icon on the top right of the homepage.

The shopping cart page will display the products that the user has added, as well as other product information that is found in the products database. This cart table will display the products image, title, quantity selected, price, total, and an option to select the product for deletion.




There is also a **discount code** option which will apply to certain products enabled for discounts.

- For example, "HUGWMUG5OFF" will take 5 dollars off the "coffee lover bag".

There is also an order summary, which displays the true calculated total that the user will "pay" for at checkout (subtotal + tax + shipping (- discount)).

### Shopping Cart

You currently have 3 item(s) in your cart.

Product	Quantity	Unit Price	Delete	Sub Total
 Coffee Lover bag	<input type="text" value="1"/>	\$16.00	<input type="checkbox"/>	\$16.00
 Yellow Instant Coffee Blend	<input type="text" value="1"/>	\$6.00	<input type="checkbox"/>	\$6.00
 Matcha Tea Powder	<input type="text" value="3"/>	\$10.00	<input type="checkbox"/>	\$30.00
<b>Total</b>				<b>\$52.00</b>

Discount Code

Coupon Code:  Apply Coupon Code

< Continue ShoppingUpdate CartProceed to Checkout >

### Order Summary

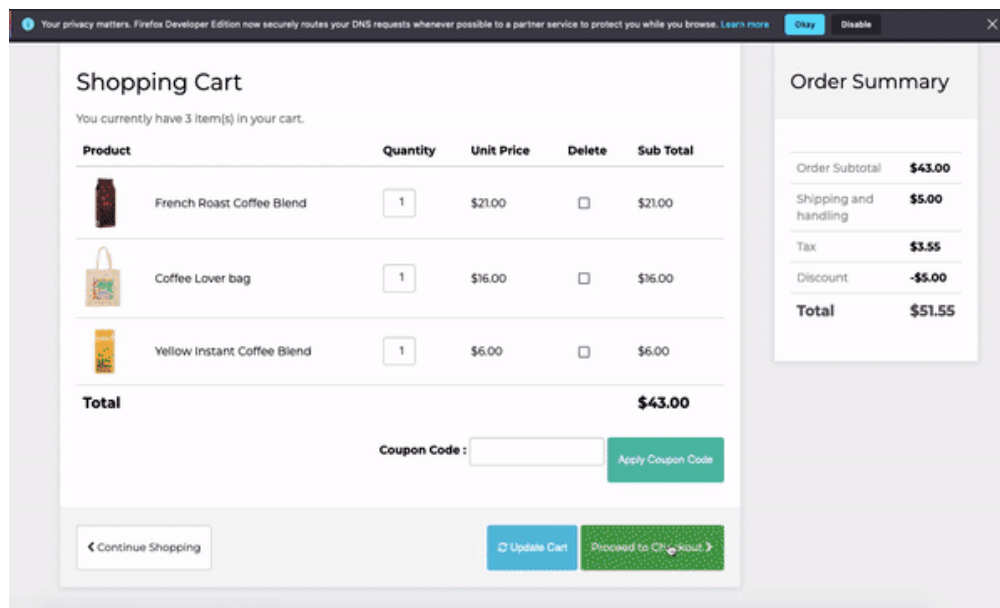
Order Subtotal	<b>\$52.00</b>
Shipping and handling	<b>\$5.00</b>
Tax	<b>\$4.29</b>
Discount	<b>-\$5.00</b>
<b>Total</b>	<b>\$61.29</b>

Order Summary

Snippet of the 'cart' page, which lists the products the customer has added to their shopping cart.

#### 4b.09.01: Placing the Order

Once a user is signed in, and they have products in their cart. They will be able to “proceed to checkout”. From there, they can view the payment options available, then click to place their order. Once the order is placed, the customer\_order database table is updated with the customer’s order, and the user will be notified that their order was successfully placed. They can then visit the details of their order in their account settings → view orders.



Gif of the 'proceeding to checkout', which is how orders are placed & saved to the DB.

#### 4b.10.01: Search

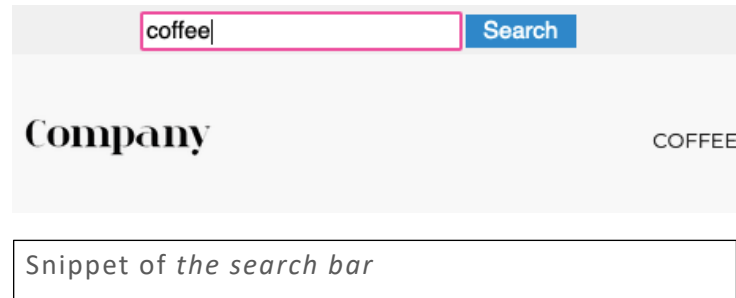
The search engine works as a function for the user to search for a specific product or anything related to the website. The user can search for things such as coffee, tea, and merchandise products.

The user can type into the search box located on the home page and click the “search” button.

After the “search” button is clicked it will display the search results to the user if any keywords

are matched from the database. If no matches are found, then the search feature will ask the user to search again. The results that will be displayed will depend on the matching keywords.

For example: if the user searches for coffee, then the search feature will display the results found for “coffee”, the keyword “coffee”, and a functioning link that the user can click on to be directed to the web page related to the search.



#### **Part 4c: Functionality: Admin Experience:-**

To demo this experience, Just visit [our admin panel](#) and login with:

*username: admin@mail.com | password: Password@123*

##### **4c.01.01: Admin Login**

If an admin user wants to visit the admin backend panel, they can do so by visiting url/ admin\_area. From there, they will be prompted to log in with their admin credentials. The credentials entered are cross checked with the admin database table. If the credentials entered match with what is in the database, then the admin user can successfully access the admin panel. If not, the admin will be prompted to try logging in again.

##### **4c.02.01: Admin Panel**

The admin panel is the homepage for the admin user. It is the page the admin immediately sees as they are logged in.

#### 4c.02.02: Admin Panel: Dashboard

This dashboard conveniently displays important information for the admin. It displays the number of products currently for sale, the number of customers, the number of categories of the products, the number of orders currently placed, the total amount of money earned, and the total number of coupon codes currently available to use. Within each of these blocks, the admin can click to view more details, which will take them to their perspective pages. At the bottom of the dashboard, the admin can readily see the most recent orders placed by customers.

#### 4c.02.03: Admin Panel: Sidebar

This sidebar is where the admin user can access *all* the admin features- such as viewing / inserting / editing products for sale, viewing orders, and more!

The screenshot displays the Admin Panel interface. On the left is a dark sidebar with navigation links: Dashboard, Products (with a dropdown), View Categories, Edit About Us Page, Coupons (with a dropdown), View Orders, Users (with a dropdown), Edit Your Profile, and Log Out. The main area is the Dashboard, which features seven colored blocks with icons, counts, and labels: Products (15), Customers (12), Products Categories (3), Orders (17), Earnings (345), Pending Orders (17), and Total Coupons (2). Each block has a 'View Details' link. Below these blocks is a section titled 'New Orders' containing a table with 5 rows of order data.

Order #	Customer	Invoice No	Product ID	Qty	Status
1	k_garcia@email.com	1459750310	33	1	pending
2	k_garcia@email.com	1459750310	32	1	pending
3	k_garcia@email.com	1459750310	28	1	pending
4	kassiejgarcia@gmail.com	1452656771	31	1	pending
5	kassiejgarcia@gmail.com	1748626578	34	3	pending

At the bottom right of the dashboard area is a link 'View All Orders'.

Snippet of the *admin panel*, where the *sidebar* is on the left, and the *dashboard* is the main page on the right



#### **4c.03.01: Admin\_Area Products**

From the admin panel sidebar, the user can access the products section. In this section, the admin user has the option to either visit the insert products page or the view products page.

#### **4c.03.02: Insert a new Product**

From this page, the admin user can easily insert new products to be displayed on the website.

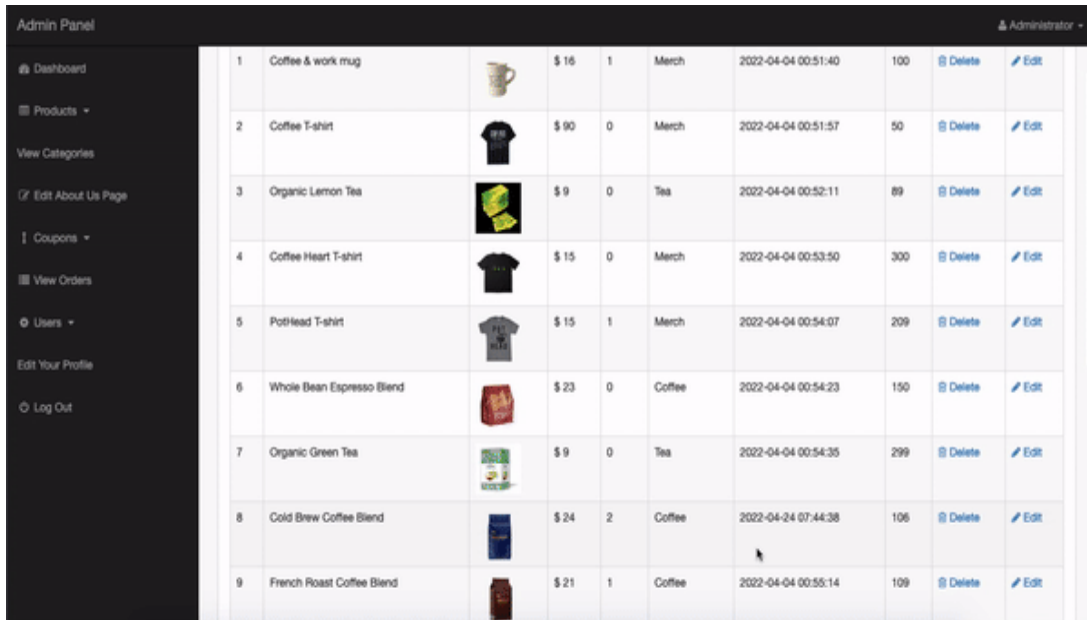
The page takes in the products title, url, category, 3 images, price, sale price, keywords, quantity, description, and a label which will indicate if the product is on sale or not. If all the input is correct, the images will be uploaded to our image directory, and the information will all be entered into the products database table. Then, the admin user will be notified of its success, and can immediately see the newly added product in the product list and in the shop pages. If there was an error, the admin user will be notified to try again.

#### **4c.03.03: View List of Products for sale**










From this page, the admin user can view all the products that are currently for sale on the website. The list displays the product information from the product database table, such as: product title, product first image, price, number sold, keywords, date added, quantity, and links to either *delete the item* or to *edit / modify the item*.

#### **4c.03.04: Modify an existing Product**

Through the view product page, the admin user can select to “edit” a product. This link will take them to the modify product page. From this page, the form will pre-fill with all the product information currently stored in the products database table. Once a change is made, the admin user can click to update the product and will be notified if that update was successful. These updates will be immediately reflected in the list of products as well as on the website directly.



The screenshot shows the Admin Panel interface. On the left is a sidebar with navigation links: Dashboard, Products, View Categories, Edit About Us Page, Coupons, View Orders, Users, Edit Your Profile, and Log Out. The main content area displays a table of products. The table has 10 columns: ID, Name, Image, Price, Stock, Category, Date, Quantity, and Actions. There are 9 rows of product data.

ID	Name	Image	Price	Stock	Category	Date	Quantity	Actions
1	Coffee & work mug		\$ 16	1	Merch	2022-04-04 00:51:40	100	<a href="#">Delete</a> <a href="#">Edit</a>
2	Coffee T-shirt		\$ 90	0	Merch	2022-04-04 00:51:57	50	<a href="#">Delete</a> <a href="#">Edit</a>
3	Organic Lemon Tea		\$ 9	0	Tea	2022-04-04 00:52:11	89	<a href="#">Delete</a> <a href="#">Edit</a>
4	Coffee Heart T-shirt		\$ 15	0	Merch	2022-04-04 00:53:50	300	<a href="#">Delete</a> <a href="#">Edit</a>
5	PotHead T-shirt		\$ 15	1	Merch	2022-04-04 00:54:07	209	<a href="#">Delete</a> <a href="#">Edit</a>
6	Whole Bean Espresso Blend		\$ 23	0	Coffee	2022-04-04 00:54:23	150	<a href="#">Delete</a> <a href="#">Edit</a>
7	Organic Green Tea		\$ 9	0	Tea	2022-04-04 00:54:35	299	<a href="#">Delete</a> <a href="#">Edit</a>
8	Cold Brew Coffee Blend		\$ 24	2	Coffee	2022-04-24 07:44:38	106	<a href="#">Delete</a> <a href="#">Edit</a>
9	French Roast Coffee Blend		\$ 21	1	Coffee	2022-04-04 00:55:14	109	<a href="#">Delete</a> <a href="#">Edit</a>

Gif showcasing the functionality of *viewing the listed products & updating a single product*

#### 4c.04.01: Admin\_area View Categories

An admin user can click on the categories section of the panel sidebar to visit the view categories page. This page simply displays the categories in use (located in the categories database table) and includes the option to delete these categories as well.

#### 4c.05.01: Admin\_area Edit the About Us Page

The admin user can click on the edit about us page from the panel sidebar to visit the edit page. This page will contain a form which will prefill with the “about us” data stored in the about\_us database table. From here, the user can make changes to this “about us” description (which is displayed in the “about us” link located in the website’s footer) and update the database.

#### 4c.06.01: Admin\_area Coupon / Discount Codes

From the admin panel sidebar, the user can access the coupons section. In this section, the admin user has the option to either visit the insert coupon page or the view coupons page.

#### 4c.06.02: Insert a new Coupon / Discount Code

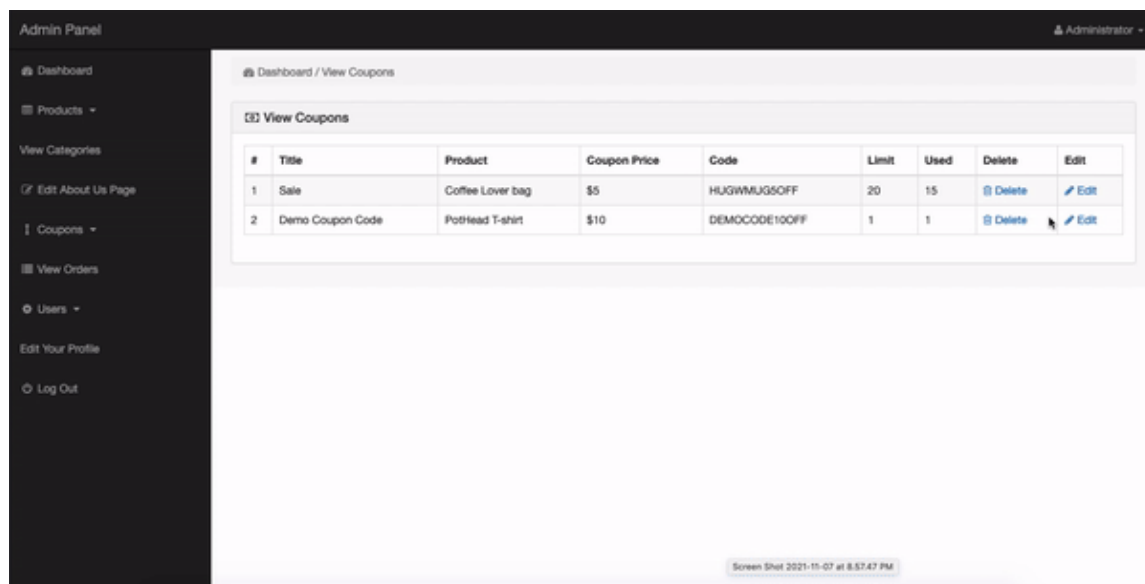
In this page, the admin user can create a new coupon for the website. This page will take in the coupon title, the amount that should be taken off, the code, the usage limit to the code, and a selection box to select which product this code should apply to. If there are no errors, the code information will be stored in the coupons database table and will be ready for usage.

#### 4c.06.03: View the list of Coupons / Discount Codes

From this page, the admin user can view all the coupon codes available for customers to use. The list displays the coupon title, product it applies to, price off, code, limit, how many times it has been used, an option to delete the coupon code, and an option to edit the coupon code.

#### 4c.06.04: Modify an existing Coupon / Discount Code

Through the view coupon page, the admin user can select to “edit” a coupon. This link will take them to the edit coupon page. From this page, the form will pre-fill with all the coupon info currently stored in the coupon database table. Once a change is made, the admin user can click to update the coupon and will be notified if that update was successful. These changes will be automatically reflected onto the coupon list and can be used immediately on the website itself.



Gif showcasing the functionality of *viewing the listed coupon & making changes to a selected coupon*

#### 4c.07.01: Admin\_area Viewing Customer Orders

From the admin panel sidebar, the user can access the view orders page. In this page, the admin user can see a list of all the currently placed orders from all customers. This list displays the order's customer, the invoice / order number, the product ordered, the quantity of the product ordered, the date the order was placed, the total amount spent on the product ordered, the order status, and an option to delete the order from the customer\_orders database.

#### 4c.06.02: Sorting Customer Orders

In this order page, the admin user can sort the orders either by the order size (dollar amount), the order date, or by the customer in alphabetical order. All the admin users must do is click on the sort buttons at the bottom of the list and they can sort how they please.

View Orders

#	Customer ID	Customer Email	Invoice	Product	Qty	Order Date	Total Amount	Status	Action
1	10	kassiejgarcia@gmail.com	533714995	Coffee & work mug	1	2022-04-03 05:29:49	\$10	Pending	Delete
2	10	kassiejgarcia@gmail.com	533714995	Coffee Lover bag	1	2022-04-03 05:29:49	\$21	Pending	Delete
3	11	demo@demo.com	1222349279	Espresso Roast Coffee Blend	2	2022-04-04 07:47:05	\$42	Pending	Delete
4	11	demo@demo.com	1222349279	Yellow Instant Coffee Blend	1	2022-04-04 07:47:05	\$6	Pending	Delete
5	14	toad@email.com	1582428953	PotHead T-shirt	1	2022-04-05 03:23:40	\$5	Pending	Delete
6	14	toad@email.com	1582428953	Coffee Lover bag	2	2022-04-05 03:23:40	\$42	Pending	Delete
7	10	kassiejgarcia@gmail.com	1748626578	Coffee Lover bag	1	2022-04-24 05:07:34	\$16	Pending	Delete
8	10	kassiejgarcia@gmail.com	1748626578	Yellow Instant Coffee Blend	1	2022-04-24 05:07:34	\$6	Pending	Delete
9	10	kassiejgarcia@gmail.com	1748626578	Matcha Tea Powder	3	2022-04-24 05:07:34	\$30	Pending	Delete
10	10	kassiejgarcia@gmail.com	1452656771	Breakfast Coffee Blend	1	2022-04-24 05:49:44	\$12	Pending	Delete
11	21	k_garcia@email.com	1459750310	French Roast Coffee Blend	1	2022-04-24 05:52:33	\$21	Pending	Delete
12	21	k_garcia@email.com	1459750310	Coffee Lover bag	1	2022-04-24 05:52:33	\$16	Pending	Delete
13	21	k_garcia@email.com	1459750310	Yellow Instant Coffee Blend	1	2022-04-24 05:52:33	\$6	Pending	Delete

Sort by Order Size \$\$   Sort by Customer   Sort by Date

admin\_area/index.php?view\_orders#

snippet which shows the *list of placed orders* as well as the *sort buttons* which can sort the orders by different options

#### 4c.07.01: Admin\_area Users

From the admin panel sidebar, the user can access the ‘users’ section (customers). In this section, the admin user has the option to either visit the insert users page or the view user page.

#### 4c.07.02: Insert a new User

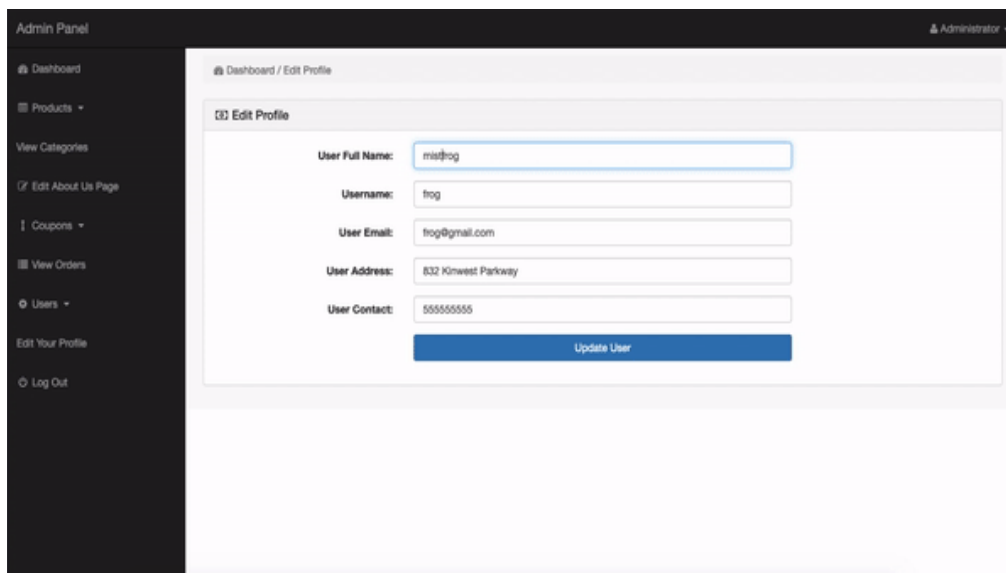
In this page, the admin can insert a new user. The admin must enter a name, username, email, password, address, and contact info. If the information entered is correct, the customers database table will be updated with a new user inserted and the admin will be notified of its success.

#### 4c.07.03: View List of Users

In this page, the admin user can view a list of users with displays a user’s name, username, email, address, and phone number. There are also 2 options to either delete the user from the customers database table or to edit the user via edit user page.

#### 4c.07.04: Modify an User

Through the view users page, the admin can select to “edit” a user. This will take them to the modify user page. From here, the form will pre-fill with all the users information currently in the customer database table. Once a change is made, the admin user can click to update the user and will be notified if successful. Any changes will be automatically reflected in the users list.

The image is a screenshot of a web application's admin panel. On the left is a dark sidebar with the title 'Admin Panel' and a list of navigation items: 'Dashboard', 'Products', 'View Categories', 'Edit About Us Page', 'Coupons', 'View Orders', 'Users', 'Edit Your Profile', and 'Log Out'. The 'Users' item is selected. The main content area has a header 'Dashboard / Edit Profile' and a sub-header 'Edit Profile'. It contains a form with the following fields: 'User Full Name' (pre-filled with 'mistfrog'), 'Username' (pre-filled with 'frog'), 'User Email' (pre-filled with 'frog@gmail.com'), 'User Address' (pre-filled with '832 Kinwest Parkway'), and 'User Contact' (pre-filled with '555555555'). Below these fields is a blue button labeled 'Update User'. In the top right corner of the admin panel, there is a user profile icon and the text 'Administrator'.

Gif which showcases the *list of customers (users)* as well as *modifying a single user named frog*

#### **4c.08.01: Edit Your Admin Profile**

From the admin panel sidebar, the admin user can access the edit your profile page. In this page, the admin user can edit their own profile and update their changes to the admins database table.

The admin user will be notified if their changes were successful. And will be prompted to sign in again.

**Part 5: Conclusion:-** In this project, there was a lot for everyone to learn- and we definitely had our highs and our lows as a group. In some meetings, we would have great communication and collaboration. In other meetings, barely anyone would show up or contribute towards the sprint goals. It was difficult, but in the end, we learned a great deal about self-discipline. We learned that as computer scientists, we should be able to handle a wide variety of tasks, no matter the programming language(s) we must learn. So, even though the majority of the group started off not knowing anything about web development, we were able to learn and handle developing in brand new languages- and we became better computer scientists because of it.

All in all, we achieved our goal of creating a web application dedicated to the Hug With Mug shopping experience, and we are all immensely proud of the work we have put into this project.

**Part 6: Relevant Links:-** Below, we have listed important links related to our project.

**GitHub Repository:** <https://github.com/kassiejgarcia/Software-Engineering-Project>

**Website URL:** <http://ec2-54-172-16-142.compute-1.amazonaws.com/shop.php>

**Website Customer Login:** <http://ec2-54-172-16-142.compute-1.amazonaws.com/checkout.php>

Username: hamtaro Password: UTSA2022se!

**Website Admin Login:** [http://ec2-54-172-16-142.compute-1.amazonaws.com/admin\\_area/login.php](http://ec2-54-172-16-142.compute-1.amazonaws.com/admin_area/login.php)

Username: [admin@mail.com](mailto:admin@mail.com) Password: Password@123

**Shared Google Folder:** <https://drive.google.com/drive/folders/1D6MyMsLhCdx8hKRCg7fmNKZWlOzuXh?usp=sharing>