

Little Big Planet E-Commerce Website

Grey DeLeon – deleong5@mail.montclair.edu

[https://cyan.csam.montclair.edu/~deleong5/ecommerce-project-final/
login.php](https://cyan.csam.montclair.edu/~deleong5/ecommerce-project-final/login.php)

Demo Accounts

Admin:

webadmin

admin0039

Customer:

iloveshopping

shopping2283

Business Plan

Participants/Client

Grey DeLeon – deleong5@mail.montclair.edu

Types of Users:

- Customers – browse products, add/remove items to cart, edit cart (change quantity, size, color), checkout, send feedback
- User Admins – monitor product stock, add new product listing, process orders, update inventory

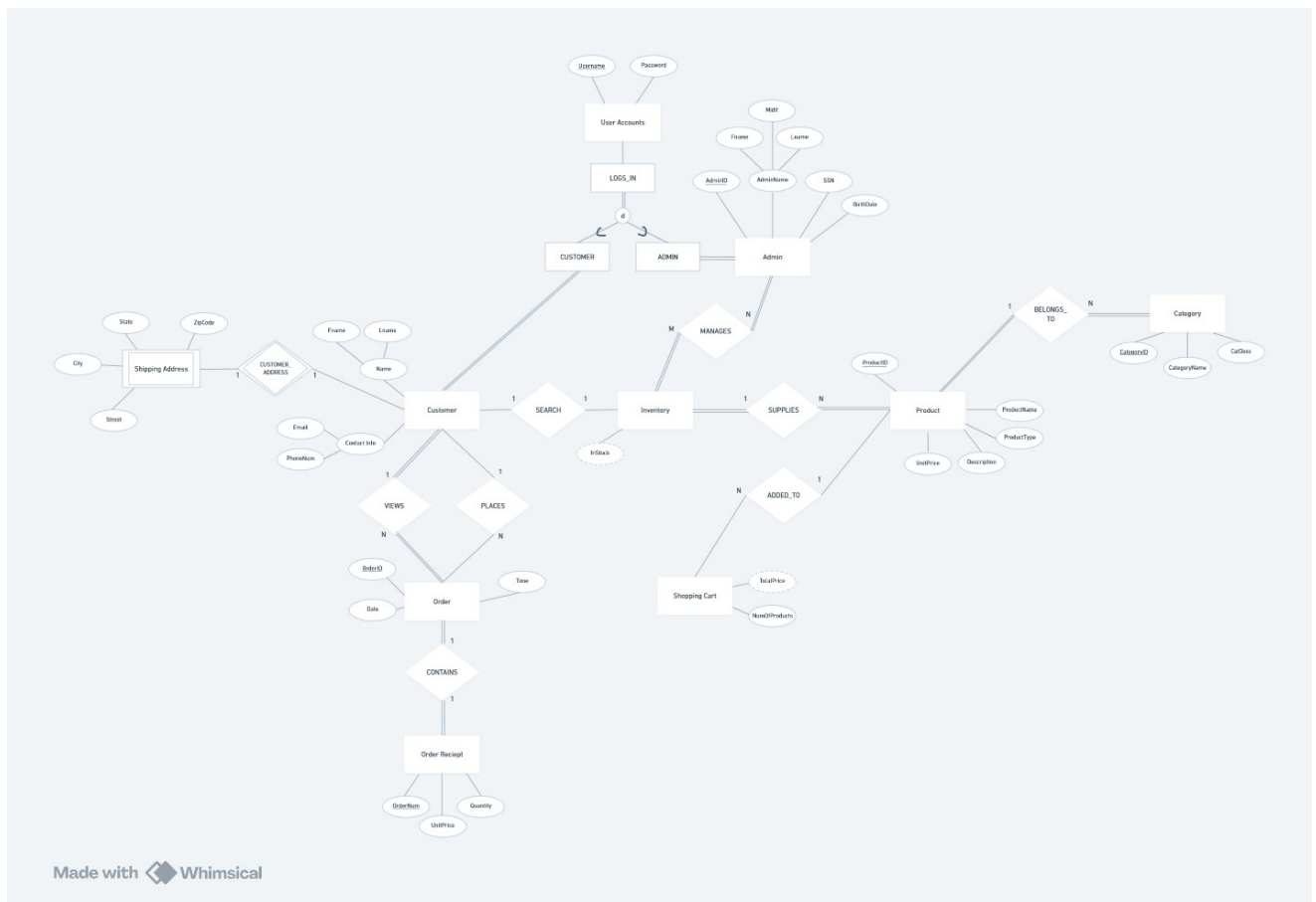
Business and Practices:

My business will be an e-commerce website where fans of the Little Big Planet series will be able to purchase Funko POP! figures surrounding the characters and iconic elements from the game franchise. I will be working with a client who is interested in selling Funko POP! figures and products surrounding the Little Big Planet series to customers and fans of the franchise mainly for their deep love, appreciation, and nostalgia for the series. Customers are to expect a variety of products for sale varying from figures to wearable merchandise to keychains and so on. Customers are to view the products available for sale and their shopping cart when they are ready to checkout. Admins of the e-commerce website are to see the products available as well but with an extended view. They should be the only ones to see additional features such as adding/removing products from the storefront, editing product details (in stock, low stock, out of stock/sold out), implementing/removing categories from the product listings, and so on.

Basic Plan of Implementation

To gather the necessary requirements and information for my miniverse, I will be conducting research on other e-commerce businesses and focusing heavily on what makes these sites successful in terms of features and design. All aspects of the e-commerce website project will be fulfilled by myself as I am completing this assignment alone.

Initial ER Diagram:



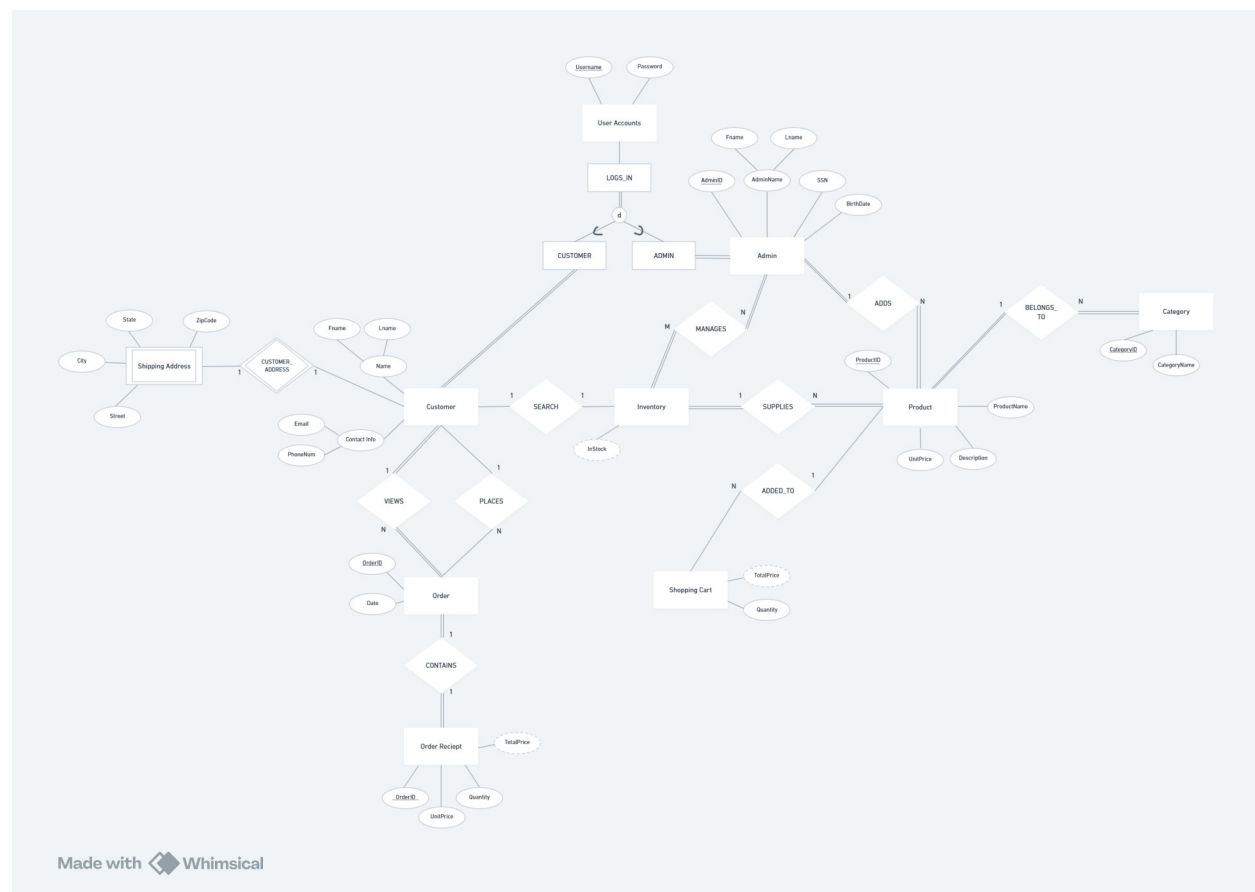
The ER diagram shown above is what was initially planned for the ecommerce website prior to development. There are ten entities shown in the diagram, all of which with their own attributes and relations with other entities. At the very top, we have the User Accounts entity that consists of attributes Username and Password, with Username being a key attribute. Based upon whether the logged in user is a customer or an administrator, different views will be shown (disjoint relation). If the logged in user is a customer, the relation will follow to the Customer entity. The Customer entity contains the attributes Name and Contact Info. The Name attribute and the Contact Info attribute are both composite attributes, Name having Fname and Lname while Contact Info has Email and PhoneNumber. The Customer entity contains a weak entity Shipping Address that carries attributes Street, City, State, and ZipCode. A Customer can have one address at a time. The Customer entity also has three different relations with other entities. There are two relations (VIEWS, PLACE) with the Order entity.

A Customer can view multiple previous orders and place multiple orders at a time. The Order entity contains attributes OrderID, Date, and Time, with OrderID being a key attribute. Orders also contain an Order Receipt where more information can be viewed. Order Receipt contains attributes OrderNum, UnitPrice, and Quantity, with OrderNum being a key attribute. Customers can search Products in the website's Inventory. The Inventory entity consists of the

InStock attribute which is calculated based on website activity. The Product entity consists of attributes ProductID, ProductName, ProductType, UnitPrice, and Description, with ProductID being a key attribute. A Product may belong to a Category at a time. The Category entity consists of attributes CategoryID, CategoryName, and CatDesc, with CategoryID being a key attribute. Products can be added to a Shopping Cart. The Shopping Cart entity consists of attributes TotalPrice and NumOfProducts.

If a user logging in is an Admin, then the user will view the interface for admins. The Admin entity consists of attributes AdminID, AdminName, SSN, and BirthDate. The AdminName attribute is a composite attribute containing Fname, Minit, and Lname. The attribute AdminID is a key attribute. Admins can manage Inventory.

Final ER Diagram:



The ER diagram shown above is the final version after development. Not much has changed from what was initially conceptualized. The changes made were to remove some attributes from certain entities as they were not needed or were awkward to utilize. OrderNum from the Order Receipt entity was scrapped and replaced with ProductID. Minit from the composite AdminName of the Admin entity was also removed from the ER diagram. The Admin entity gained an additional relation, being the ADDS relation where Admins can add products to

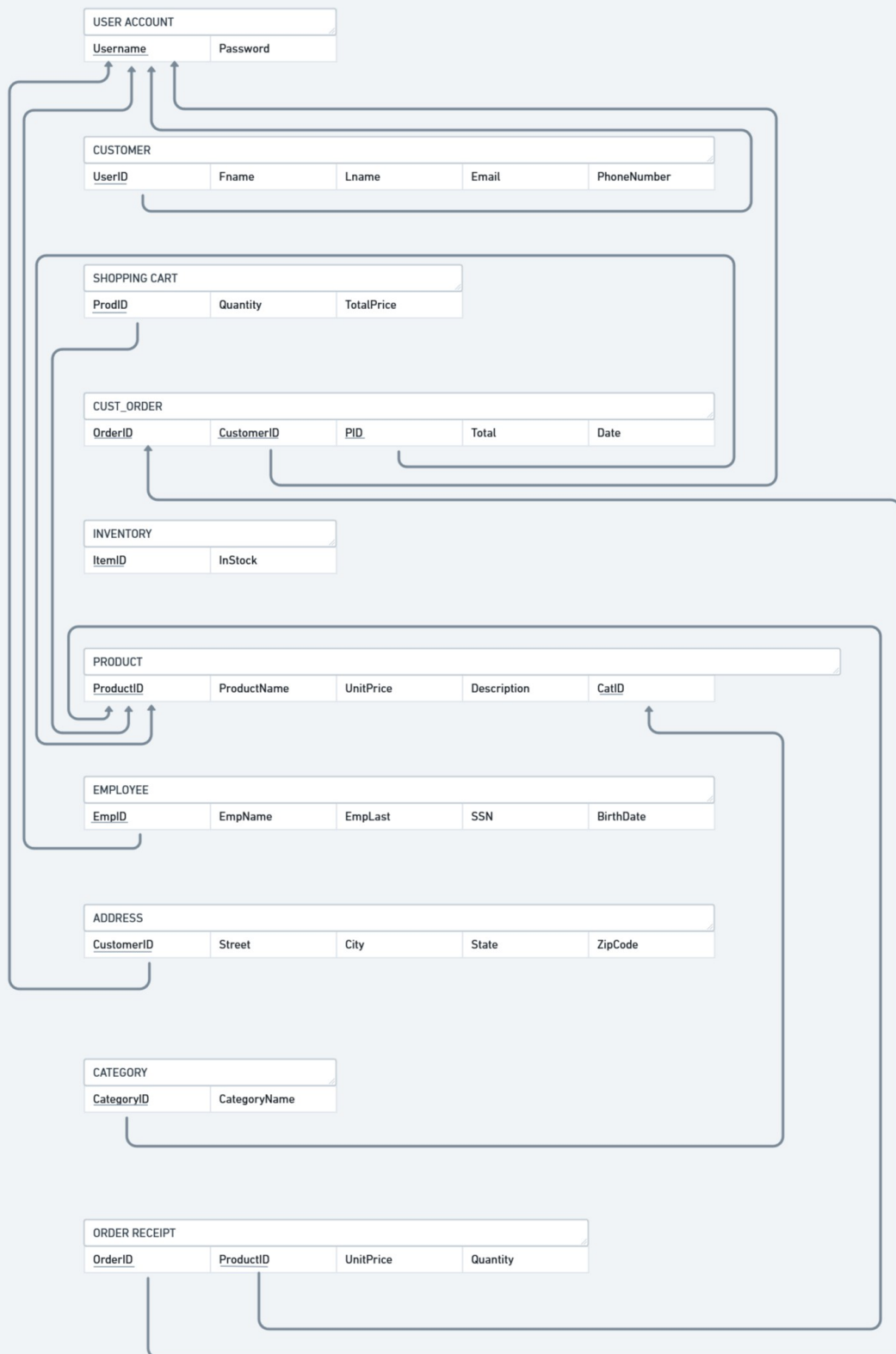
Initial Relational Diagram:



Above is the initial relational diagram I had created prior to building the website. For the first table above, the LOGIN table, its attributes consist of Username and Password with Username being the primary key. The CUSTOMER table consists of attributes UserID, FName, Lname, Email, and PhoneNumber with UserID being both a primary key and a foreign key reference to Username from the LOGIN table. The SHOPPING_CART table consists of attributes ProdID, NumOfProducts, and TotalPrice with ProdID being a foreign key references to ProductID from the PRODUCT table. The ORDER table consists of attributes OrderID, OrderAmount, Date, Time, and OrdNum with OrderID being a primary key and OrdNum being a foreign key reference to OrderNum from the ORDER_RECEIPT table.

The INVENTORY table consists of attributes ItemID and InStock with ItemID being a foreign key reference to ProductID from the PRODUCT table. The PRODUCT table consists of attributes ProductID, ProductName, ProductType, UnitPrice, Description, and CatID. ProductID is a primary and CatID is a foreign key reference to the CATEGORY table attribute CategoryID. The EMPLOYEE table consists of attributes EmpID, EmpName, EmpLast, SSN, and BirthDate. EmpID is a foreign key referencing Username from the LOGIN table. The ADDRESS table consists of attributes CustomerID, Street, City, State, and ZipCode. The attribute CustomerID is a foreign key reference to the LOGIN attribute Username. The CATEGORY table consists of attributes CategoryID, CatItemID, CategoryName, and CategoryDesc. CategoryID is a primary key and CatItemID is a foreign key reference to the PRODUCT table attribute ProductID. The ORDER_RECEIPT table contains attributes OrderNum, OrdID, PurchasedProdID, UnitPrice, and TotalQuantity. OrderNum is a primary key and there are two foreign key references. OrdID references OrderID from ORDER and PurchasedProdID references ProductID from PRODUCT.

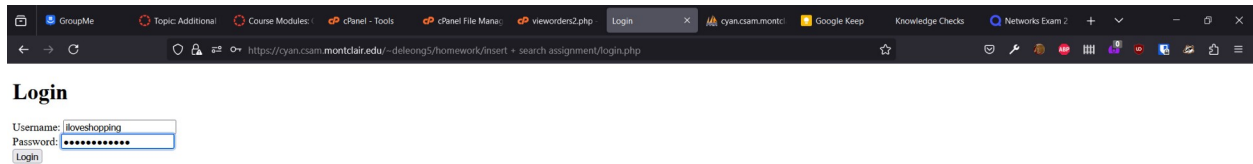
Final Relational Diagram:



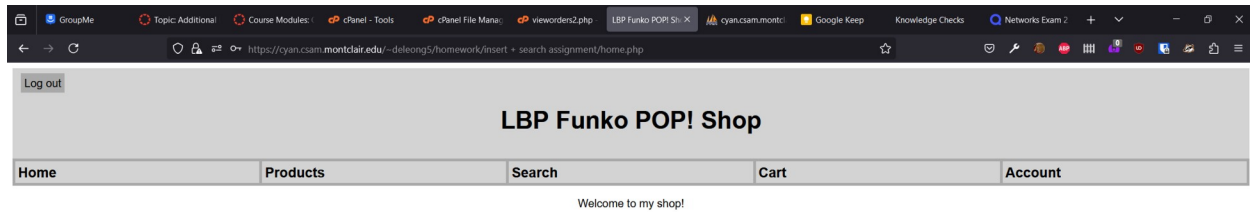
Above is the final relational diagram after development for the ecommerce website. Several changes were made to the design of the database during development. Some attributes were cut entirely while others were changed. In the case of the ORDER table, the foreign key OrdID was removed completely as the original primary key, OrderNum, was removed and replaced with PID (foreign key reference to ProductID) for orders placed in the CUST_ORDER table. This change was made to simplify matching orders with their receipt based on the products purchased. Another change made was the elimination of the attribute ProductType. It's function lines up with CatID, therefore there was no need for an attribute that functions practically the same as another attribute in the same table. CatDesc was also scrapped as there was not much of a need for category descriptions.

Walkthrough:

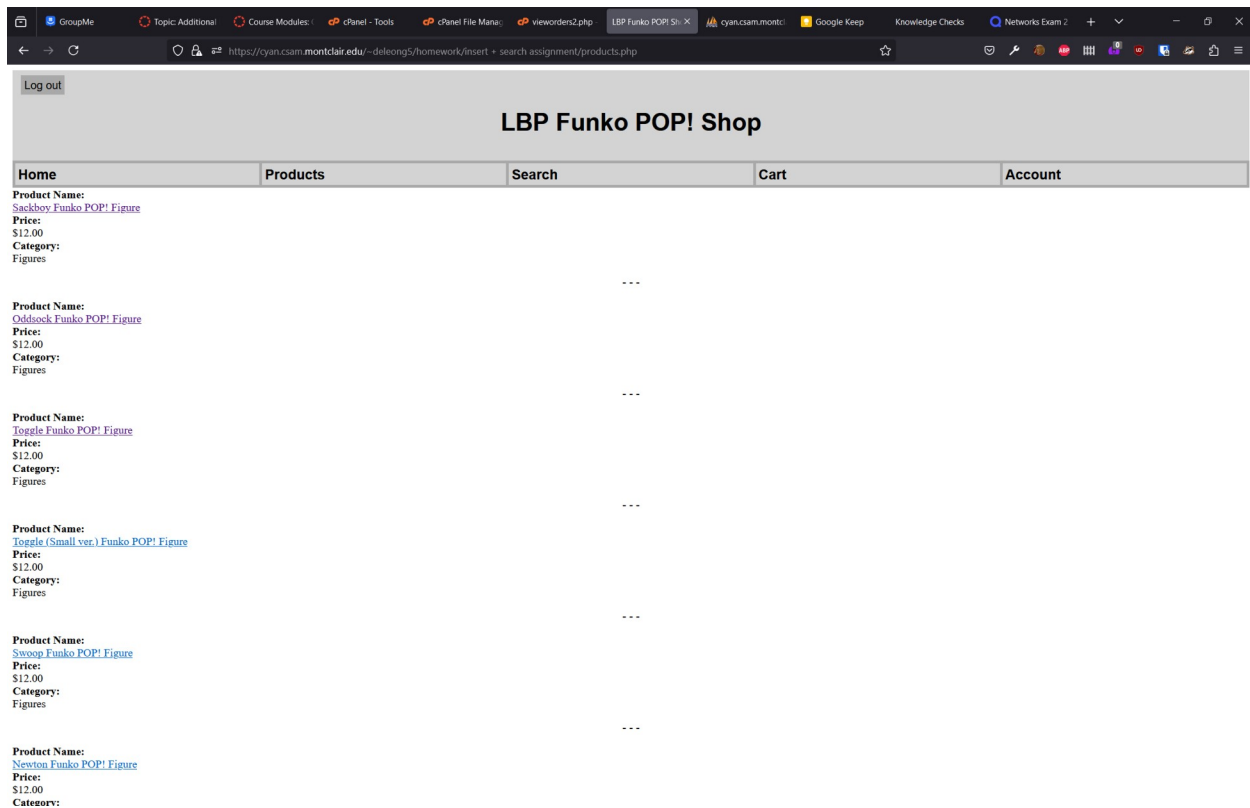
Walkthroughs of both customer and administrator views are to be followed.



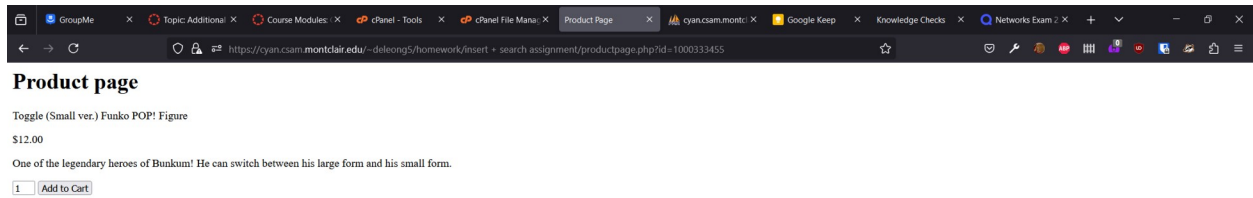
Above shows the first page users will be directed to once they follow the link to the ecommerce website. Users will be prompted to login, otherwise they will not be able to access anything on the website. This is mainly for security purposes and for copyright purposes as well. How this works behind the scenes is that a SELECT statement will be prepared which includes the login information users have input to log in. A query will be made where the database will search and select for an entry that matches the username entered. If no such information exists, users will be given an error. In the instance that an account does exist, the website will then compare the entered password with the password stored in the database, verifying if it is a match. The password is encrypted using MD5. Password entries will be converted into MD5 and compared with what is in the database. For the first walkthrough, we will be exploring the view of a consumer who would like to make a purchase on our website. First, they must log in using their account.



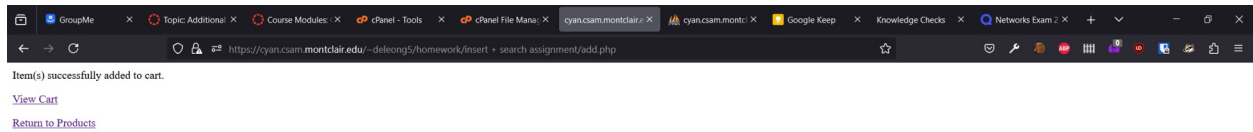
Once the consumer has logged in, they will be greeted with the home page for the website. This is where a user's session on the website begins. Whenever a user logs in to the website, a session is created that maintains the credentials entered for each page visited. When these sessions expire, users will have to log in again. From here, we can see the header for the website which includes the name of the website, a logout button, and a navigation menu. On the navigation menu, there are five options that users can navigate through: home, products, search, cart, and account. The home option brings users back to the home page. The products option will bring users to a page where all products will be displayed. The search option will bring users to a search page where they can search for specific items based on category and information entered through the search bar. The cart option brings users to their shopping cart. The account option brings users to their account information page. They can also access previous orders from there as well.



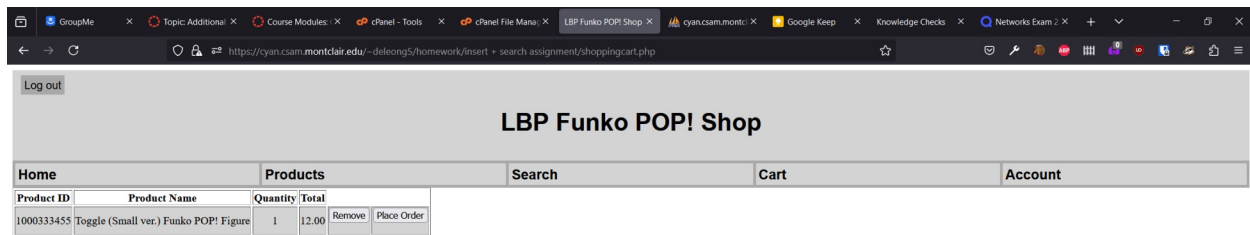
This is the page users will see when viewing the complete collection of products available for retail. For each product stored in the database, the web page will display its name, unit price, and category. This is done using a SELECT statement. A VIEW is being utilized in this instance that contains information from both the PRODUCT and CATEGORY tables. Based on shared category ID, each entry will display the product name, unit price, and category name. Users can browse through the complete collection of products available. When a user would like to see more details on a product, they can click on the name of the product, and it will redirect them to the product page for the selected product. In this case, our user would like to see more information on the Toggle (small ver.) Funko POP! Figure.



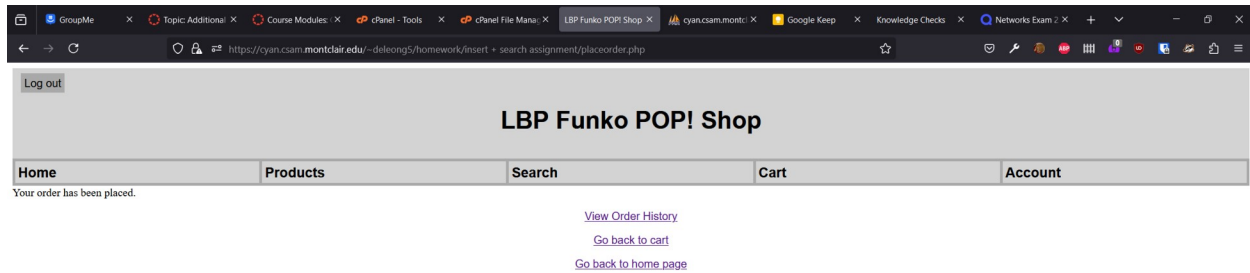
After clicking on the product name, users will be brought to this page. Here, they can view more information about the product and add the product to their shopping cart. Each product page will display the product name, its unit price, a description about the product, and the option to add whichever quantity of the product a user would like to add to their shopping cart. Following adding whichever quantity of the product a user adds to their shopping cart, they will be redirected to a page which will confirm whether their product has been added or not.



Once users add the quantity of their desired product, they will be redirected to a confirmation page which will notify if their product has successfully been added to their shopping cart. Items are added to the shopping cart using an INSERT statement to the SHOPPING_CART table. The statement will include the product ID, quantity, and the total price. The total price is calculated by multiplying the unit price by the entered quantity of the item. Below the confirmation message are two options users can choose from based on whether they want to continue browsing through products or go straight to their cart. For the next part of the walkthrough, we will be viewing the shopping cart of the website.



Now we are viewing the shopping cart page for the ecommerce website. Here, we can see the item that has been added from the previous page. The shopping cart displays some details about the item in addition to two options customers can perform. Items in the cart display their product ID, name, quantity, and total price for that quantity. There is the option to remove items from the cart with the Remove option. This will invoke a DELETE statement that will delete a set of items from the cart based on the product ID. Next to it is the Place Order option. Customers can move forward and place their order using that option.



Following placing an order on the products stored in the cart, the website will redirect users to an order confirmation page. This page will display whether the order has been successfully placed or not. Orders are placed by using an INSERT statement that includes the customer ID, product ID, total, and date that the order has been placed. This will all be inserted into the CUST_ORDER table. Order IDs are auto incremented as each order is placed. Below the message are three options to choose from. The first option is the view the complete order history of the logged in customer. The second option is to go back to the shopping cart in case a customer would like to place more orders. The third option is to go back to the home page.

Log out

LBP Funko POP! Shop

[Home](#)
[Products](#)
[Search](#)
[Cart](#)
[Account](#)

Order History

Order Number: 2
 Product: Sackboy Funko POP! Figure
 Unit Price: \$12.00
 Quantity: 1
 Total Price: \$12

...

Order History

Order Number: 3
 Product: Oddsock Funko POP! Figure
 Unit Price: \$12.00
 Quantity: 2
 Total Price: \$24

...

Order History

Order Number: 4
 Product: Toggle Funko POP! Figure
 Unit Price: \$12.00
 Quantity: 3
 Total Price: \$36

...

Order History

Order Number: 5
 Product: Sackboy Funko POP! Figure
 Unit Price: \$12.00
 Quantity: 1
 Total Price: \$12

...

This is what the complete order history of a customer looks like. All previously placed orders will show on this page. Previous orders are stored within the ORDER_RECEIPT table in the database. Entries in the table are displayed using a SELECT statement that retrieves data from both the ORDER_RECEIPT table and the PRODUCT table. Product data is matched with products previously purchased with the constraint that their product IDs match. Previous orders show their order ID, product name, unit price, quantity, and the total price of the order placed.

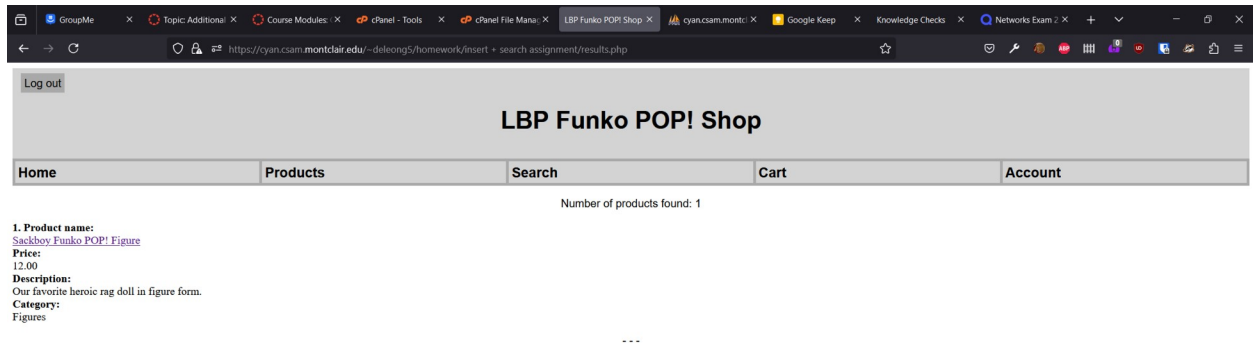
Log out

LBP Funko POP! Shop

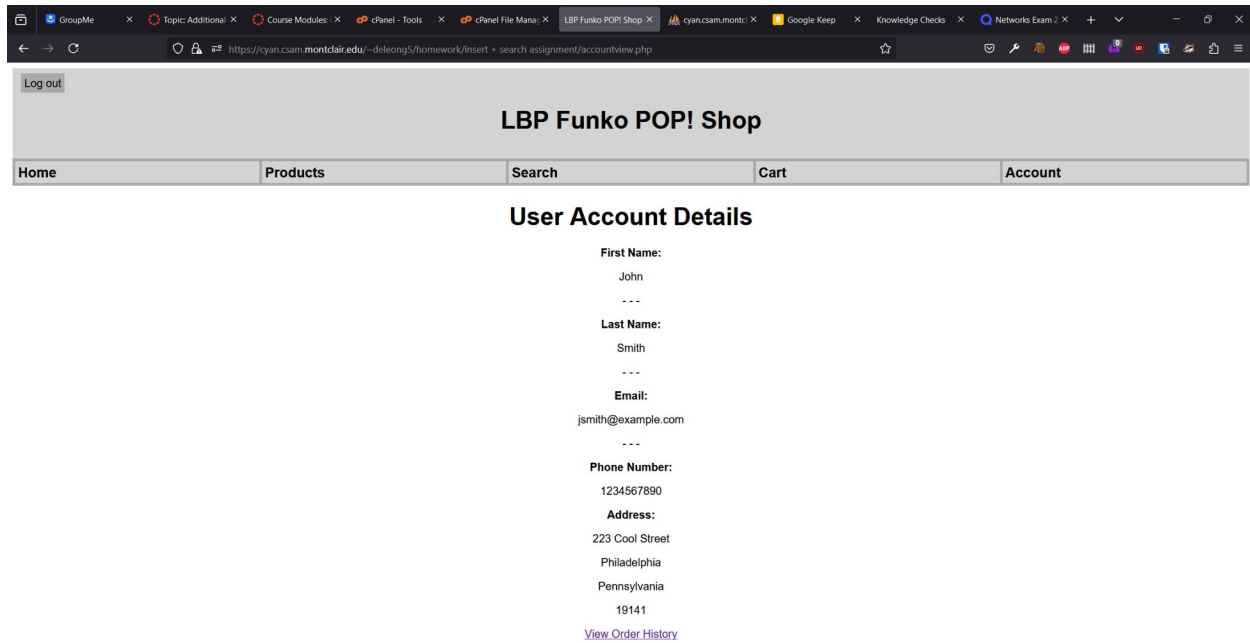
Home	Products	Search	Cart	Account
------	----------	--------	------	---------

Choose search type:
Figures
Enter search term:
sackboy
Submit

Above is the search page that would allow customers to search for specific products based on their category using keywords. If a user attempts to search without filling out all the necessary details for their search, an error will be thrown that they have not filled out their search details. To prevent SQL injections, search terms and categories entered will go through the addslashes() function. When all details are entered, a SELECT statement will be created using the information entered by the user. The statement will then be queried to the PRODUCT table to select entries from the table with the input category ID and with product names like the search terms entered by the user.



Here is the result of performing a search in the Figures category for products with the term “sackboy” in its name. The page will display the number of products found from the search at the top. Below would be a list of products that match the user’s input along with information about the product. The displayed information includes the product name, unit price, description, and category.



When users click on the Account tab of the navigation bar, they will be brought to a page where they can view their account information and find the link to their previous orders. User information is displayed using a SELECT statement. A view, CUSTOMER_INFO, is used in this instance. When the statement is queried, the system will be able to retrieve the customer's first name, last name, email address, phone number, and full address.

Login

Username: webadmin

Password: ••••••••

Login

After logging out of our session as a customer, users will be redirected back to the login page. We will explore the view of a website administrator for our ecommerce website. As with the first session, we will need to login again but with an admin account. Behind the scenes, our login page will verify the login information and verify whether the user logging in is an admin account or not. If so, we will be brought to a different page only authorized personnel will be allowed to view. Otherwise, users will be brought to the home page of the website again.

Log out			
LBP Funko POP! Shop - Inventory			
View Orders			
Add Product			
Product ID	Product Name	In Stock	Add Stock
1234567890	Sackboy Funko POP! Figure	32	Add
1231234564	Sackboy Funko POP! Keychain	35	Add
1344556677	Sackboy Funko POP! Pin	28	Add
1023478599	Oddsock Funko POP! Figure	12	Add
1000333444	Toggle Funko POP! Figure	9	Add
1000333455	Toggle (Small ver.) Funko POP! Figure	10	Add
1009988776	Swoop Funko POP! Figure	5	Add
1004567833	Newton Funko POP! Figure	5	Add
1079884733	Nana Pud Funko POP! Figure	5	Add
1255748932	Captain Pud Funko POP! Figure	5	Add
1010998877	Sackbot Funko POP! Figure	7	Add
1000001937	Oddsock Funko POP! Pin	14	Add
1380048329	Toggle (both ver.) Funko POP! Pin Bundle	5	Add
1099334422	Swoop Funko POP! Pin	5	Add
1008847583	Sackbot Funko POP! Pin	5	Add
1000338859	Oddsock Funko POP! Keychain	5	Add
1055667788	Toggle (both ver.) Funko POP! Keychain Bundle	5	Add
1011225577	Swoop Funko POP! Keychain	13	Add
1022223333	Newton Funko POP! Keychain	5	Add
1993747385	Sackbot Funko POP! Keychain	11	Add
1008334421	Funko POP! Foldable Figure Protector	5	Add
1098234556	Funko POP! Sackboy Backpack	5	Add
1223377485	Funko POP! LBP Wallet	10	Add
1010112233	Funko POP! LBP Shirt	5	Add
1003034556	Funko POP! LBP Hoodie	13	Add

Above is the first page administrators will view once they have successfully logged in. They will view the inventory of all the items that are in the PRODUCT database. Products and their stock are retrieved using a SELECT statement. The statement will retrieve data from both the PRODUCT and INVENTORY tables on the condition that the product IDs from both tables match each product. From there, the inventory page displays every product's ID, name, and how many are in stock. Next to each product entry is the option to add items to its stock.

Log out

LBP Funko POP! Shop - Inventory

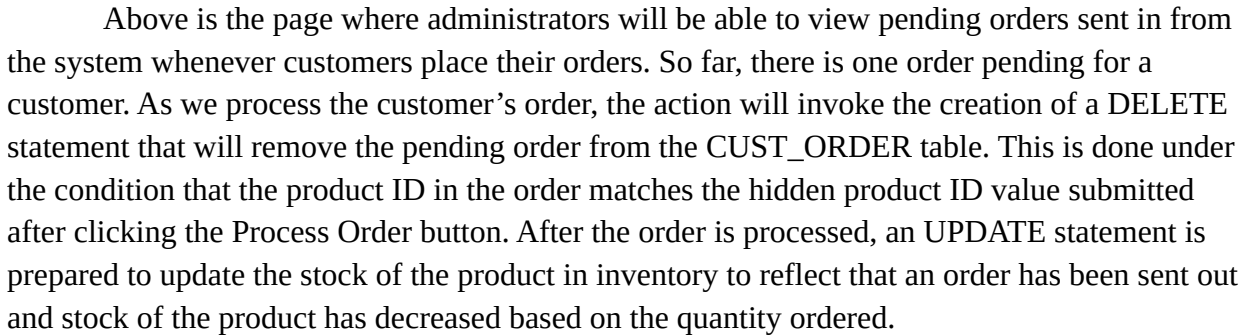
[View Orders](#)
[Add Product](#)

Product ID	Product Name	In Stock	Add Stock
1234567890	Sackboy Funko POP! Figure	32	0 Add
1231234564	Sackboy Funko POP! Keychain	35	0 Add
1344556677	Sackboy Funko POP! Pin	28	0 Add
1023478599	Oddsock Funko POP! Figure	12	0 Add
1000333444	Toggle Funko POP! Figure	9	0 Add
1000333455	Toggle (Small ver.) Funko POP! Figure	10	0 Add
1009988776	Swoop Funko POP! Figure	5	10 Add
1004567833	Newton Funko POP! Figure	5	0 Add
1079884733	Nana Pud Funko POP! Figure	5	0 Add
1255748932	Captain Pud Funko POP! Figure	5	0 Add
1010998877	Sackbot Funko POP! Figure	7	0 Add
1000001937	Oddsock Funko POP! Pin	14	0 Add
1380048329	Toggle (both ver.) Funko POP! Pin Bundle	5	0 Add
1099334422	Swoop Funko POP! Pin	5	0 Add
1008847583	Sackbot Funko POP! Pin	5	0 Add
1000338859	Oddsock Funko POP! Keychain	5	0 Add
1055667788	Toggle (both ver.) Funko POP! Keychain Bundle	5	0 Add
1011225577	Swoop Funko POP! Keychain	13	0 Add
1022223333	Newton Funko POP! Keychain	5	0 Add
1993747385	Sackbot Funko POP! Keychain	11	0 Add
1008334421	Funko POP! Foldable Figure Protector	5	0 Add
1098234556	Funko POP! Sackboy Backpack	5	0 Add
1223377485	Funko POP! LBP Wallet	10	0 Add
1010112233	Funko POP! LBP Shirt	5	0 Add
1003034556	Funko POP! LBP Hoodie	13	0 Add

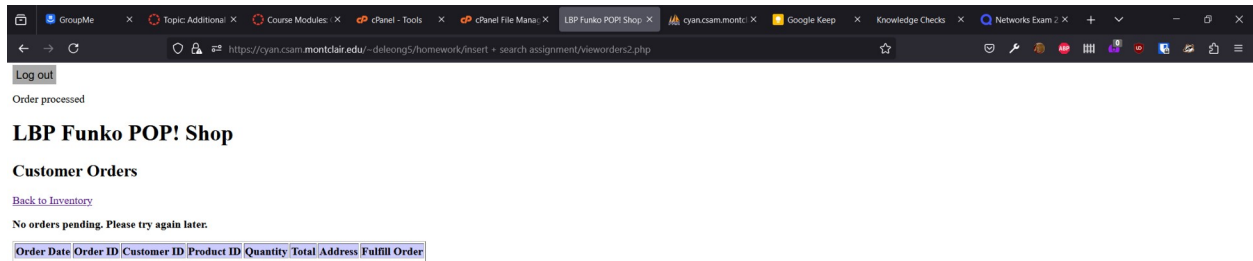
In this instance, we have received 10 additional stock of the Swoop Funko POP! Figure and will be adding it to what is already in stock. For this, we will input 10 to the entry bar and add it to our inventory by clicking the Add button. The way this works within the system is whenever stock has been added, an UPDATE statement is created using the entered quantity. The query statement will update that item's stock by adding the entered quantity to the existing quantity. This is on the condition that the product ID in the INVENTORY database matches with the hidden value product ID submitted whenever users add to the inventory.

1009988776	Swoop Funko POP! Figure	15	0	Add
------------	-------------------------	----	---	-----

Changes made to the inventory can be reflected in the screenshot above. In this case, we have updated the inventory for the Swoop Funko POP! Figure to increase from 5 to 15. An UPDATE statement was queried where the product ID of the product shown was matched with the same product ID in the INVENTORY table and the stock available was updated to add 10 to what was already stored.



Above is the page where administrators will be able to view pending orders sent in from the system whenever customers place their orders. So far, there is one order pending for a customer. As we process the customer's order, the action will invoke the creation of a DELETE statement that will remove the pending order from the CUST_ORDER table. This is done under the condition that the product ID in the order matches the hidden product ID value submitted after clicking the Process Order button. After the order is processed, an UPDATE statement is prepared to update the stock of the product in inventory to reflect that an order has been sent out and stock of the product has decreased based on the quantity ordered.



When orders are processed, the page will refresh and display a message stating that the order has been processed. Processed orders will trigger two different queries to two different database tables. The first query being a DELETE query which will remove the pending order entry from the CUST_ORDER table based on a hidden product ID value input after a user clicks on the “Process Order” button. The second query is an UPDATE statement to the INVENTORY table. Here, a statement is created to update the stock of the purchased product based on the quantity entered by customers for their order. Changes to the purchased product’s stock will be reflected on the inventory page.

1000333455	Toggle (Small ver.) Funko POP! Figure	9	0	Add
------------	---------------------------------------	---	---	-----

After an order has been processed, that product’s inventory updates to reflect how many items have been sent out to a customer. In the example above, after an administrator processed the order that came in for 1 Toggle (small ver.) Funko POP! Figure, the system queries an UPDATE statement to subtract 1 item from stock.

[Log out](#)

LBP Funko POP! Shop - New Product Entry

[Back to Inventory](#)

Category:

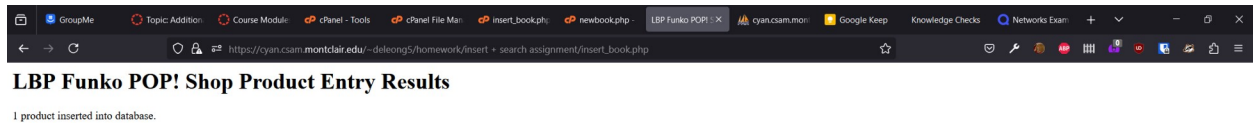
Product ID:

Product Name:

Unit Price \$:

Description:

This is where administrators will be brought when they would like to add new products to the website. Users will be prompted to provide certain details about the new product, such as what category the new product will belong to, the product ID, what the product is called, its unit price, and a description about the new product. Once all the details are included, users can register the product and will be redirected to a page that displays whether the product has been successfully added to the database or not. If there is missing information from the new product entry, the website will notify users of one or more entry details missing.



When users successfully fill out the entry details for a new product, they will be brought to this page where the product is added to the database. Behind the scenes, an INSERT statement is used where details about the new product will be added to the PRODUCT database. A query statement is made which includes the product ID, product name, unit price, description, and category ID. Following this is another INSERT statement query, this time for the INVENTORY database. The statement includes the product ID information from the user input product ID and, by default, 0 for the InStock entry.

Log out

LBP Funko POP! Shop - Inventory

[View Orders](#)
[Add Product](#)

Product ID	Product Name	In Stock	Add Stock
1234567890	Sackboy Funko POP! Figure	32	0 Add
1231234564	Sackboy Funko POP! Keychain	35	0 Add
1344556677	Sackboy Funko POP! Pin	28	0 Add
1023478599	Oddsock Funko POP! Figure	12	0 Add
1000333444	Toggle Funko POP! Figure	9	0 Add
1000333455	Toggle (Small ver.) Funko POP! Figure	9	0 Add
1009988776	Swoop Funko POP! Figure	15	0 Add
1004567833	Newton Funko POP! Figure	5	0 Add
1079884733	Nana Pud Funko POP! Figure	5	0 Add
1255748932	Captain Pud Funko POP! Figure	5	0 Add
1010998877	Sackbot Funko POP! Figure	7	0 Add
1000001937	Oddsock Funko POP! Pin	14	0 Add
1380048329	Toggle (both ver.) Funko POP! Pin Bundle	5	0 Add
1099334422	Swoop Funko POP! Pin	5	0 Add
1008847583	Sackbot Funko POP! Pin	5	0 Add
1000338859	Oddsock Funko POP! Keychain	5	0 Add
1055667788	Toggle (both ver.) Funko POP! Keychain Bundle	5	0 Add
1011225577	Swoop Funko POP! Keychain	13	0 Add
1022223333	Newton Funko POP! Keychain	5	0 Add
1993747385	Sackbot Funko POP! Keychain	11	0 Add
1008334421	Funko POP! Foldable Figure Protector	5	0 Add
1098234556	Funko POP! Sackboy Backpack	5	0 Add
1223377485	Funko POP! LBP Wallet	10	0 Add
1010112233	Funko POP! LBP Shirt	5	0 Add
1003034556	Funko POP! LBP Hoodie	13	0 Add
1111111111	Oreo Funko POP! Figure	0	0 Add

After receiving confirmation that our new product has been added to our product database, we can now go back to the inventory page and see the changes reflected. Our new product is now listed in the inventory. By default, new products will have 0 inventory until an admin/employee adds product into the inventory. Along with this, our new product will now show up for customers when they browse the ecommerce website. They will be able to purchase the product, if it is in stock, however.

Conclusion:

To wrap this documentation up, it was quite frankly a valuable learning experience creating an ecommerce platform that interacts with a functioning database from scratch. As with learning new skills, it was not a feasible task to complete. For starters, I began working with no prior knowledge of PHP and little experience of HTML/CSS. This was also the first time I truly began working with a database using what was taught during this course. Honestly, I am rather proud of myself for what I have accomplished and learned during the limited amount of time given to work on this assignment.

There is room for improvement with the website built, however, time was one of the main concerns I kept in mind when working. Aesthetic-wise, the website is rather bare bones to look at but if I had more time to work with it or if this were something I were working on as a personal

project, I would have given more care and attention to the design of the whole website. That and if I also had a better understanding of HTML/CSS/JavaScript.

In terms of future work, I would like to continue this project and improve the features implemented so far. Instead of having individual quantities of products be purchased at a time, I would like to have it where everything in a user's cart is purchased at the same time. For time and simplicity's sake, only individual quantity of a particular item can be purchased at a time. I would like to keep learning on working with databases while building websites and to keep practicing SQL for future employment. There are various improvements I would like to make to the website in my own personal time. It would be a great learning opportunity to continue as the semester wraps up and a potential project to include in my resume.

Ecommerce Website Code Dump:

accountview.php

```
<?php
    include('session.php');
    include('config.php');
    require('header.php');

    $user = $_SESSION['login_user'];

    $query = "select * from customer_info where customer_info.userid =
'{"$_SESSION['login_user']}";
    $result = $db->query($query);

    $row = $result->fetch_assoc();
?>

<html>
    <head>
        <title>Account Details</title>
    </head>
    <body>
```

<h1>User Account Details</h1>

<p>

First Name:

<?php

echo "<p>".\$row['fname']."</p>";

?>

</p>

<p>- - -</p>

<p>

Last Name:

<?php

echo "<p>".\$row['lname']."</p>";

?>

</p>

<p>- - -</p>

<p>

Email:

<?php

echo "<p>".\$row['email']."</p>";

?>

</p>

<p>- - -</p>

<p>

Phone Number:

<?php

echo "<p>".\$row['phonenumber']."</p>";

?>

</p>

<p>

Address:


```

<?php
    echo "<p>".$row['street']."</p>";
    echo "<p>".$row['city']."</p>";
    echo "<p>".$row['state']."</p>";
    echo "<p>".$row['zipcode']."</p>";
    ?>
</p>
<p>
    <a href="orderhistory.php">View Order History</a>
</p>
</body>
</html>

```

add.php

```

<?php
    include('session.php');
    include('config.php');

    $quan = $_POST['quantity'];
    $pid = $_POST['productid'];

    $query = "select * from product where ".$pid." = product.productid";
    $result = $db->query($query);

    $row = $result->fetch_assoc();

    $total = $row['unitprice'] * $quan;

    $intoCart = "insert into shopping_cart values

```



```

        ("'.$pid.'", "'.$quan.'", "'.$total.'");
$result2 = $db->query($intoCart);
?>
<html>
    <p>
        <?php
            if($result2){
                echo "Item(s) successfully added to cart.";
            } else{
                echo "An error has occurred. Your item(s) has not been added.";
            }

            $db->close();
        ?>
    <p>
        <a href="shoppingcart.php">View Cart</a>
    </p>
    <p>
        <a href="products.php">Return to Products</a>
    </p>

</p>
</html>

```

authenticate.php

```

<html>
    <head>
        <title>Login Status</title>
    </head>
    <body>

```

```
<?php
@ $db = new
mysqli('localhost','deleong5_bookorama','bookorama123','deleong5_bookorama');

if(mysqli_connect_errno()){
    echo "Error: Could not connect to database. Please try again later.";
}

$username = $_POST['username'];
$password = $_POST['upassword'];

$username = stripslashes($username);
$password = stripslashes($password);
$username = mysqli_real_escape_string($db,$username);
$password = mysqli_real_escape_string($db,$password);

$query = "select * from login where username = '".$username.'" and upassword =
''.md5($password).''";

$result = mysqli_query($db,$query);
$row = mysqli_fetch_assoc($result);

if(mysqli_num_rows($result) == 1){
    echo "Login successful";
} else{
    echo "Login failed";
}

mysqli_free_result($result);
mysqli_close($db);
?>
```

```
</body>
</html>
```

config.php

```
<?php
    @ $db = new
    mysqli('localhost','deleong5_bookorama','bookorama123','deleong5_bookorama');

    if(mysqli_connect_errno()){
        echo "Error: could not connect to database. Please try again later.";
    }
?>
```

header.php

```
<html>
    <head>
        <title>LBP Funko POP! Shop</title>
    </head>
    <style>
        h1 {
            font-family: Helvetica;
            text-align: center;
            font-size: 35px;
        }
        .menu {
            font-weight: bold;
            font-size: 20px;
            text-align: center;
        }
        .menu a{
```

```
text-decoration: none;
font-family: sans-serif;
color: black;
}
td {
background-color: lightgray;
}
p {
font-family: arial;
text-align: center;
}
.prod {
background-color: white;
}
.acc a{
font-family: sans-serif;
text-decoration: none;
padding: 5px 5px;
background-color: darkgray;
color: black;
}
</style>
<body>
<table width="100%" cellpadding="12" cellspacing="0" border="0">
<tr>
<td width="25%">
<span class="acc"><a href="logout.php">Log out</a></span>
</td>
</tr>
</table>
```

```

<table width="100%" cellpadding="12" cellspacing="0" border="0">
  <tr bgcolor="lightgray">
    <td>
      <h1>LBP Funko POP! Shop</h1>
    </td>
  </tr>
</table>
<table width="100%" bgcolor="darkgray" cellpadding="4" cellspacing="4">
  <tr>
    <td width="20%">
      <span class="menu"><a href="home.php">Home</a></span>
    </td>
    <td width="20%">
      <span class="menu"><a href="products.php">Products</a></span>
    </td>
    <td width="20%">
      <span class="menu"><a href="search.php">Search</a></span>
    </td>
    <td width="20%">
      <span class="menu"><a href="shoppingcart.php">Cart</a></span>
    </td>
    <td width="20%">
      <span class="menu"><a href="accountview.php">Account</a></span>
    </td>
  </tr>
</table>
</body>
</html>

```

home.php

```
<?php
    include('session.php');
    require('header.php');
?>
<html>
    <p>Welcome to my shop!</p>
</html>
```

insert_book.php

```
<html>
<head>
    <title>LBP Funko POP! Shop Product Entry Results</title>
</head>
<body>
<h1>LBP Funko POP! Shop Product Entry Results</h1>
<p><a href="newbook.php">Back to Add Product</a></p>
<?php
    // create short variable names
    $productid=$_POST['productid'];
    $productname=$_POST['productname'];
    $unitprice=$_POST['unitprice'];
    $description=$_POST['description'];
    $category=$_POST['category'];

    if (!$productid || !$productname || !$unitprice || !$description || !$category) {
        echo "You have not entered all the required details.<br />"
        . "Please go back and try again.";
        exit;
    }
```

```

if (!get_magic_quotes_gpc()) {
    $productid = addslashes($productid);
    $productname = addslashes($productname);
    $unitprice = doubleval($unitprice);
    $description = addslashes($description);
    $category=addslashes($category);
}

@ $db = new mysqli('localhost', 'deleong5_bookorama', 'bookorama123',
'deleong5_bookorama');

if (mysqli_connect_errno()) {
    echo "Error: Could not connect to database. Please try again later.";
    exit;
}

$query = "insert into product values
        ('".$productid."', '".$productname."', '".$unitprice."', '".$description."', '".$category."')";
$result = $db->query($query);

$inv = "insert into inventory (itemid) values ('".$productid."')";
$db->query($inv);

if ($result) {
    echo $db->affected_rows." product inserted into database.";
} else {
    echo "An error has occurred. The item was not added.";
}
$db->close();
?>

```

```
</body>
```

```
</html>
```

inventory.php

```
<?php
```

```
    include('session.php');
```

```
    require('logoutbutton.php');
```

```
    include('config.php');
```

```
    if(isset($_POST['submit'])){
```

```
        $quan = $_POST['quantity'];
```

```
        $pid = $_POST['productid'];
```

```
        $update = "update inventory set instock = instock + ".$quan." where inventory.itemid = ".  
$pid."";
```

```
        $db->query($update);
```

```
    }
```

```
    $query = "select * from inventory, product where inventory.itemid = product.productid";
```

```
    $result = $db->query($query);
```

```
?>
```

```
<html>
```

```
    <head>
```

```
        <title>LBP Funko POP! Shop - Inventory</title>
```

```
    </head>
```

```
    <body>
```

```
        <h1>LBP Funko POP! Shop - Inventory</h1>
```

```
        <p>
```

```
            <a href="vieworders2.php">View Orders</a>
```

```
        </p>
```



```

<p>
    <a href="newbook.php">Add Product</a>
</p>
<?php
    echo "<table border='1'>\n";
    echo "<tr><th bgcolor='#CCCCFF'>Product ID</th>
        <th bgcolor='#CCCCFF'>Product Name</th>
        <th bgcolor='#CCCCFF'>In Stock</th>
        <th bgcolor='#CCCCFF'>Add Stock</th>
    <tr>";

    $count = $result->num_rows;
    for($i = 0; $i < $count; $i++){
        $row = $result->fetch_assoc();

        echo "<tr>
            <td align='right'>".$row['itemid'].</td>
            <td align='left'>".$row['productname'].</td>
            <td>".$row['instock'].</td>
            <form action='inventory.php' method='post'>
                <td><input type='text' name='quantity' value='0' size='2'>
                    <input type='hidden' name='productid' value='{ $row['itemid']}'>
                    <input type='submit' name='submit' value='Add'></td>
            </form>
            </tr>";
        }
    ?>

<?php
    $result->free();

```

```
        $result2->free();
        $db->close();
    ?>
</body>
</html>
```

login.php

```
<?php
    @ $db = new
    mysqli('localhost','deleong5_bookorama','bookorama123','deleong5_bookorama');

    if(mysqli_connect_errno()){
        echo "Error: Could not connect to database. Please try again later.";
    }

    session_start();

    $error = "";
    if($_SERVER["REQUEST_METHOD"] == "POST"){

        // username and password sent from form
        $username = $db->real_escape_string($_POST['username']);
        $password = $db->real_escape_string($_POST['password']);

        $query = "select * from login where username = '". $username. "' and upassword =
        '".md5($password)."'";

        $result = $db->query($query);
        $row = $result->num_rows;
        $count = $result->num_rows;
```

```
if($count == 1){
    $_SESSION['login_user'] = $username;
    if($username == "webadmin"){
        header("location: inventory.php");
    } else{
        header("location: home.php");
    }

} else{
    $error = "Your Login Name or Password is Invalid";
}
}
?>
```

```
<!DOCTYPE html>
<html>
    <head>
        <title>Login</title>
    </head>
    <body>
        <h1>Login</h1>
        <?php if(isset($error)): ?>
            <div><?php echo $error; ?></div>
        <?php endif; ?>

        <form action="" method="post">
            <div>
                <label>Username:</label>
                <input type="text" name="username" required>
```

```
</div>
<div>
  <label>Password:</label>
  <input type="password" name="password" required>
</div>
<div>
  <button type="submit" name="login">Login</button>
</div>
</form>
</body>
</html>
```

logout.php

```
<?php
    session_start();

    if(session_destroy()){
        header("location: login.php");
    }
?>
```

logoutbutton.php

```
<html>
  <style>
    .acc a{
      font-family: sans-serif;
      text-decoration: none;
      padding: 5px 5px;
      background-color: darkgray;
      color: black;
```

```
    }
</style>
<body>
    <table width="100%" cellpadding="12" cellspacing="0" border="0">
        <tr width="25">
            <span class="acc"><a href="logout.php">Log out</a></span>
        </tr>
    </table>
</body>
</html>
```

newbook.php

```
<?php
    include('session.php');
    require('logoutbutton.php');

    @ $db = new mysqli('localhost', 'deleong5_bookorama', 'bookorama123',
'deleong5_bookorama');

    if(mysqli_connect_errno()){
        echo 'Error: could not connect to database. Please try again later.';
        exit;
    }

    $query = "select * from category";
    $result = $db->query($query);
?>

<html>
    <head>
        <title>LBP Funko POP! Shop - New Product Entry</title>
```

</head>

<body>

<h1>LBP Funko POP! Shop - New Product Entry</h1>

<p>Back to Inventory</p>

<form action="insert_book.php" method="post">

<table border="0">

<tr>

<td>Product ID</td>

<td> <input type="text" name="productid" maxlength="13" size="13"></td>

</tr>

<tr>

<td>Product Name</td>

<td> <input type="text" name="productname" maxlength="50" size="50"></td>

</tr>

<tr>

<td>Unit Price \$</td>

<td><input type="text" name="unitprice" maxlength="7" size="7"></td>

</tr>

<tr>

<td>Description</td>

<td> <input type="text" name="description" maxlength="80" size="80"></td>

</tr>

<tr>

Category:

<select name="category">

<?php

while(\$category = \$result->fetch_array(MYSQLI_ASSOC));

?>

<option value="<?php echo \$category['categoryid']?>">

```

                <?php echo $category['categoryname'];?>
            </option>
        <?php
            endwhile;
        ?>
    </select>
</tr>
<tr>
    <td colspan="2"><input type="submit" value="Register"></td>
</tr>
</table>

</form>
</body>
</html>

```

orderhistory.php

```

<?php
    include('session.php');
    include('config.php');
    require('header.php');

    $order = "select * from order_receipt, product where order_receipt.productid =
product.productid";
    $result = $db->query($order);
?>

<html>
    <head>
        <title>Order History</title>

```

```

</head>
<body>
    <?php
        $count = $result->num_rows;
        for($i = 0; $i < $count; $i++){
            $row = $result->fetch_assoc();
            $total = $row['quantity'] * $row['unitprice'];
            echo"
                <h1>Order History</h1>
                <strong>Order Number: </strong>{$row['orderid']} <br />
                <strong>Product: </strong>{$row['productname']} <br />
                <strong>Unit Price: $</strong>{$row['unitprice']} <br />
                <strong>Quantity: </strong>{$row['quantity']} <br />
                <strong>Total Price: $</strong>{$total} <br />
                <p>- - </p>
            ";
        }
    ?>
</body>
</html>

```

placeorder.php

```

<?php
    include('session.php');
    include('config.php');
    require('header.php');

    // gets product id from placed order
    $pid = $_POST['prodid'];

```



```
$item = "select * from shopping_cart where ".$pid." = shopping_cart.prodid";  
$result = $db->query($item);
```

```
$row = $result->fetch_assoc();
```

```
$date = date('Y-m-d');
```

```
$order = "insert into cust_order (customerid, pid, total, orderdate)values  
(('{$_SESSION['login_user']}','.$row['prodid'].', '.$row['totalprice'].', '$date'))";  
$status = $db->query($order);
```

```
$query = "select * from cust_order, product, shopping_cart where cust_order.pid =  
shopping_cart.prodid and cust_order.pid = product.productid";  
$result2 = $db->query($query);
```

```
$row2 = $result2->fetch_assoc();
```

```
$receipt = "insert into order_receipt values ('.$row2['orderid'].', '.$row2['productid'].', ".  
$row2['unitprice'].', '.$row2['quantity'].')";  
$stat = $db->query($receipt);  
?>
```

```
<html>
```

```
<head>
```

```
<title>Order Processed</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
if($status){
```

```
    echo "Your order has been placed.";
```

```
    echo "<p><a href=\"orderhistory.php\">View Order History</a><p>";
```

```

        echo "<p><a href=\"shoppingcart.php\">Go back to cart</a></p>";
        echo "<p><a href=\"home.php\">Go back to home page</a></p>";

        $purchased = "delete from shopping_cart where ".$pid." = shopping_cart.prodid";
        $db->query($purchased);
    } else{
        echo "An error has occurred. Your order has not been placed.";
    }
    ?>
</body>
</html>

```

productpage.php

```

<?php
    include('session.php');
    include("config.php");

    $prodid = $_GET['id'];

    $query = "select * from product where ".$prodid." = product.productid";
    $result = $db->query($query);

    $query2 = "select * from inventory where ".$prodid." = inventory.itemid";
    $result2 = $db->query($query2);

    $row = $result->fetch_assoc();
    $row2 = $result2->fetch_assoc();
    ?>

<html>

```

```
<head>

    <title>Product Page</title>

</head>


<body>
    <h1>Product page</h1>
    <div class="product-name">
        <?php
            echo "<p>".$row['productname']. "</p>";
        ?>
    </div>
    <div class="product-price">
        <?php
            echo "<p>$".$row['unitprice']. "</p>";
        ?>
    </div>
    <div class="product-desc">
        <?php
            echo "<p>".$row['description']. "</p>";
        ?>
    </div>


    <div class="cart-action">
        <form action="add.php" method="post">
            <input type="text" name="quantity" value="1" min="1" max="<?=$row2['instock'];
?>" size="2">
            <input type="hidden" name="productid" value="<?=$row['productid']; ?>">
            <input type="submit" value="Add to Cart" class="btnAddAction" >
        </form>
    </div>
</body>
</html>
```

</div>

</body>

</html>

products.php

<?php

include('session.php');

require('header.php');

?>

<html>

<head>

<title>LBP Funko POP! Shop Products</title>

</head>

<body>

<?php

include("config.php");

\$query = "select * from product_category";

\$result = \$db->query(\$query);

\$num_results = \$result->num_rows;

for(\$i = 0; \$i < \$num_results; \$i++){

\$row = \$result->fetch_assoc();

?>

<div class="product-item">

<div class="product-name">

Product Name:


```
<a href="productpage.php?id=<?php echo $row['productid']; ?>">
    <?php
        echo $row['productname'];
    ?>
</a>
```

```
</div>
```

```
<div class="product-price">
    <strong>Price: <br /></strong>
    <?php
        echo "$".$row['unitprice'];
    ?>
```

```
</div>
```

```
<div class="product-category">
    <strong>Category: <br /></strong>
    <?php
        echo $row['categoryname'];
    ?>
```

```
</div>
```

```
<div>
    <p align="left">- - -</p>
</div>
```

```
</div>
```

```
<?php
    }
```

```
$result->free();
```

```
$db->close();
```

```
?>
```

```
</body>
</html>
```

results.php

```
<?php
    include('session.php');
    require('header.php');
?>

<html>
<head>
    <title>LBP Funko POP! Shop Search Results</title>
</head>
<body>
<?php
    // create short variable names
    $category=$_POST['category'];
    $searchterm=trim($_POST['searchterm']);

    if (!$category || !$searchterm) {
        echo 'You have not entered search details. Please go back and try again.';
        exit;
    }

    if (!get_magic_quotes_gpc()){
        $category = addslashes($category);
        $searchterm = addslashes($searchterm);
    }
```

```
@ $db = new mysqli('localhost', 'deleong5_bookorama', 'bookorama123',
'deleong5_bookorama');
```

```
if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database. Please try again later.';
    exit;
}
```

```
$query = "select * from product where product.catid = ".$category." and product.productname
like '%".$searchterm.%'";
```

```
$result = $db->query($query);
```

```
$num_results = $result->num_rows;
```

```
$query2 = "select category.categoryname from category where ".$category." =
category.categoryid";
```

```
$result2 = $db->query($query2);
```

```
$row2 = $result2->fetch_assoc();
```

```
echo "<p>Number of products found: ".$num_results."</p>";
```

```
for ($i=0; $i <$num_results; $i++) {
```

```
    $row = $result->fetch_assoc();
```

```
    ?>
```

```
<div>
```

```
    <strong> <?php echo ($i+1).". "; ?> Product name: <br /></strong>
```

```
    <a href="productpage.php?id=<?php echo $row['productid']; ?>">
```

```
        <?php echo htmlspecialchars(stripslashes($row['productname'])); ?>
```

```
    </a>
```

```

</div>
<div>
    <strong>Price: </strong><br />
    <?php echo stripslashes($row['unitprice']); ?>
</div>
<div>
    <strong>Description: <br/></strong>
    <?php echo stripslashes($row['description']); ?>
</div>
<div>
    <strong>Category: <br/></strong>
    <?php echo stripslashes($row2['categoryname']); ?>
</div>
<div align="left">
    <p>- - -</p>
</div>
<?php
}
$result->free();
$db->close();
?>
</body>
</html>

```

search.php

```

<?php
    include('session.php');
    require('header.php');

```



```
@ $db = new mysqli('localhost', 'deleong5_bookorama', 'bookorama123',  
'deleong5_bookorama');
```

```
if(mysqli_connect_errno()){  
    echo 'Error: could not connect to database. Please try again later.';  
    exit;  
}
```

```
$query = "select * from category";  
$result = $db->query($query);  
?>
```

```
<html>  
    <head>  
        <title>LBP Funko POP! Shop Catalog Search</title>  
    </head>
```

```
<body>  
    <form action="results.php" method="post">  
        Choose search type:<br />  
        <select name="category">  
            <?php  
                while($category = $result->fetch_array(MYSQLI_ASSOC));  
                ?>
```

```
            <option value="<?php echo $category['categoryid']?>">  
                <?php echo $category['categoryname'];  
                ?>  
            </option>  
            <?php
```

```

        endwhile;
    ?>
</select>
<br />
Enter search term:<br />
<input name="searchterm" type="text" size="40">
<br />
<input type="submit" value="Submit" name="submit">
</form>
<br>
</body>
</html>

```

session.php

```

<?php
    session_start();
    if(!isset($_SESSION['login_user'])){
        header("location: login.php");
        die();
    }
    $login_session = $_SESSION['login_user'];
?>

```

shoppingcart.php

```

<?php
    include('session.php');
    include('config.php');
    require('header.php');

    //removes item from shopping cart

```

```
if(isset($_POST['remove'])){  
    $item = $_POST['productid'];
```

```
    $revitem = "delete from shopping_cart where shopping_cart.prodid = ".$item."";
```

```
    $db->query($revitem);  
}
```

```
if(isset($_POST['place-order'])){  
    if(empty($_POST['place-order'])){  
        echo "<p>There is nothing in your cart. Add items in your cart to place an order.</p>";  
    }  
}
```

```
}
```

```
$query = "select * from shopping_cart, product where shopping_cart.prodid =  
product.productid";
```

```
$result = $db->query($query);
```

```
?>
```

```
<html>
```

```
    <head>
```

```
        <title>LBP Funko POP! Shop - Shopping Cart</title>
```

```
    </head>
```

```
    <body>
```

```
        <div id="shopping-cart">
```

```
            <table border="1">
```

```
                <tr>
```

```
<th>Product ID</th><th>Product Name</th><th>Quantity</th><th>Total</th>
</tr>
```

```
<?php
```

```
    $count = $result->num_rows;
```

```
    if($count == 0){
```

```
        echo "<p>Your shopping cart is empty</p>";
```

```
    }
```

```
    for($i = 0; $i < $count; $i++){
```

```
        $row = $result->fetch_assoc();
```

```
?>
```

```
<tr>
```

```
    <td align="right">
```

```
        <?php echo $row['prodid']; ?>
```

```
    </td>
```

```
    <td align="left">
```

```
        <?php echo $row['productname']; ?>
```

```
    </td>
```

```
    <td align="center">
```

```
        <?php echo $row['quantity']; ?>
```

```
    </td>
```

```
    <td>
```

```
        <?php echo $row['totalprice']; ?>
```

```
    </td>
```

```
    <td>
```

```
        <form action="shoppingcart.php" method="post">
```

```
            <input type="hidden" name="productid" value="<?= $row['prodid']; ?>">
```

```
            <input type="submit" name="remove" value="Remove">
```

```
        </form>
```

```

        </td>
        <td>
            <form action="placeorder.php" method="post">
                <input type="hidden" name="prodid" value="<?=$row['prodid']; ?>">
                <input type="submit" name="place-order" value="Place Order">
            </form>
        </td>
    </tr>
</table>
<?php
    }
    $result->free();
    $db->close();
?>
</div>
</body>
</html>

```

vieworders2.php

```

<?php
    include('session.php');
    require('logoutbutton.php');
    include('config.php');

    if(isset($_POST['order-process'])){
        $pid = $_POST['productid'];
        $bought = $_POST['quantity'];

        $inv = "update inventory set instock = instock - ".$bought." where inventory.itemid = ".
        $pid."";
    }
}

```

```
$process = $db->query($inv);

if($process){
    echo "<p>Order processed</p>";

    $processed = "delete from cust_order where cust_order.pid = ".$_POST['productid']."";
    $res = $db->query($processed);

} else{
    echo "<p>Order has not been processed</p>";
}

}

$query = "select * from cust_order, order_receipt, address where cust_order.orderid =
order_receipt.orderid and cust_order.customerid = address.customerid";
$result = $db->query($query);

?>
<html>
<head>
    <title>LBP Funko POP! Shop - Customer Orders</title>
</head>
<body>
<h1>LBP Funko POP! Shop</h1>
<h2>Customer Orders</h2>
<p><a href="inventory.php">Back to Inventory</a></p>
<?php
    // count the number of orders in the array
    $number_of_orders = $result->num_rows;
```

```

if ($number_of_orders == 0) {
    echo "<p><strong>No orders pending.
        Please try again later.</strong></p>";
}

```

```

echo "<table border='1'>\n";
echo "<tr><th bgcolor='#CCCCFF'>Order Date</th>
    <th bgcolor='#CCCCFF'>Order ID</th>
    <th bgcolor='#CCCCFF'>Customer ID</th>
    <th bgcolor='#CCCCFF'>Product ID</th>
    <th bgcolor='#CCCCFF'>Quantity</th>
    <th bgcolor='#CCCCFF'>Total</th>
    <th bgcolor='#CCCCFF'>Address</th>
    <th bgcolor='#CCCCFF'>Fulfill Order</th>
    <tr>";

```

```

for ($i=0; $i<$number_of_orders; $i++) {
    $orders= $result->fetch_assoc();
    // output each order
    echo "<tr>
        <td>".$orders['orderdate'].</td>
        <td align='right'>".$orders['orderid'].</td>
        <td align='right'>".$orders['customerid'].</td>
        <td align='right'>".$orders['productid'].</td>
        <td align='right'>".$orders['quantity'].</td>
        <td>".$orders['total'].</td>
        <td align='right'>".$orders['street'].$orders['city'].$orders['state'].
        $orders['zipcode'].</td>
        <form action='vieworders2.php' method='post'>

```

```
<td>
  <input type=\"hidden\" name=\"productid\" value=\"{$orders['productid']}\">
  <input type=\"hidden\" name=\"quantity\" value=\"{$orders['quantity']}\">
  <input type=\"submit\" name=\"order-process\" value=\"Process Order\">
</td>
</form>

</tr>";
}
echo "</table>";
?>
</body>
</html>
```


Database Dump:

-- phpMyAdmin SQL Dump

-- version 5.2.1

-- <https://www.phpmyadmin.net/>

--

-- Host: localhost:3306

-- Generation Time: Dec 13, 2024 at 11:49 AM

-- Server version: 5.7.44

-- PHP Version: 8.1.31

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

START TRANSACTION;

SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8mb4 */;

--

-- Database: `deleong5_bookorama`

--

-- -----

--

-- Table structure for table `address`

--

```
CREATE TABLE `address` (  
  `customerid` varchar(50) NOT NULL,  
  `street` varchar(30) NOT NULL,  
  `city` varchar(30) NOT NULL,  
  `state` varchar(25) NOT NULL,  
  `zipcode` char(5) NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

--

-- Dumping data for table `address`

--

```
INSERT INTO `address` (`customerid`, `street`, `city`, `state`, `zipcode`) VALUES  
('iloveshopping', '223 Cool Street', 'Philadelphia', 'Pennsylvania', '19141');
```

-- -----

--

-- Table structure for table `category`

--

```
CREATE TABLE `category` (  
  `categoryid` char(15) NOT NULL,
```

```
`categoryname` varchar(50) DEFAULT NULL,  
`catdesc` text  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Dumping data for table `category`
```

```
--
```

```
INSERT INTO `category` (`categoryid`, `categoryname`, `catdesc`) VALUES  
(109876543210123, 'Figures', 'Here are all the figures available for purchase.'),  
(222333444555888, 'Pins', 'Decorate with all these pins available!'),  
(765123958578039, 'Keychains', 'Get some cool keychains for your house keys!'),  
(123456789002345, 'Accessories', 'Accessorize yourself with these cool accessories!'),  
(666777888999222, 'Apparel', 'Walk out in fashion with our apparel!');
```

```
-- -----
```

```
--
```

```
-- Stand-in structure for view `customer_info`
```

```
-- (See below for the actual view)
```

```
--
```

```
CREATE TABLE `customer_info` (  
  `userid` varchar(50)  
, `fname` varchar(20)  
, `lname` varchar(50)  
, `email` varchar(50)  
, `phonenum` char(10)
```

```
,`customerid` varchar(50)
,`street` varchar(30)
,`city` varchar(30)
,`state` varchar(25)
,`zipcode` char(5)
);
```

```
-- -----
```

```
--
-- Table structure for table `cust_order`
--
```

```
CREATE TABLE `cust_order` (
  `orderid` int(15) NOT NULL,
  `customerid` varchar(50) NOT NULL,
  `pid` char(12) NOT NULL,
  `total` float(4,2) DEFAULT NULL,
  `orderdate` date NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
-- -----
```

```
--
-- Table structure for table `employee`
--
```

```
CREATE TABLE `employee` (  
  `empid` varchar(50) NOT NULL,  
  `empname` varchar(20) NOT NULL,  
  `emplast` varchar(20) NOT NULL,  
  `ssn` char(9) NOT NULL,  
  `birthdate` date NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Dumping data for table `employee`
```

```
--
```

```
INSERT INTO `employee` (`empid`, `empname`, `emplast`, `ssn`, `birthdate`) VALUES  
('webadmin', 'Eric', 'Jones', '321123444', '1994-08-20');
```

```
-- -----
```

```
--
```

```
-- Table structure for table `inventory`
```

```
--
```

```
CREATE TABLE `inventory` (  
  `itemid` char(15) NOT NULL,  
  `instock` int(11) DEFAULT '0'  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Dumping data for table `inventory`
```

```
--
```

```
INSERT INTO `inventory` (`itemid`, `instock`) VALUES
```

```
('1234567890', 32),
```

```
('1231234564', 35),
```

```
('1344556677', 28),
```

```
('1023478599', 12),
```

```
('1000333444', 9),
```

```
('1000333455', 9),
```

```
('1009988776', 15),
```

```
('1004567833', 5),
```

```
('1079884733', 5),
```

```
('1255748932', 5),
```

```
('1010998877', 7),
```

```
('1000001937', 14),
```

```
('1380048329', 5),
```

```
('1099334422', 5),
```

```
('1008847583', 5),
```

```
('1000338859', 5),
```

```
('1055667788', 5),
```

```
('1011225577', 13),
```

```
('1022223333', 5),
```

```
('1993747385', 11),
```

```
('1008334421', 5),
```

```
('1098234556', 5),
```

```
('1223377485', 10),
```

```
('1010112233', 5),  
('1003034556', 13),  
('1111111111', 0);
```

```
-- -----
```

```
--
```

```
-- Table structure for table `login`
```

```
--
```

```
CREATE TABLE `login` (  
  `username` varchar(50) NOT NULL,  
  `upassword` varchar(100) NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Dumping data for table `login`
```

```
--
```

```
INSERT INTO `login` (`username`, `upassword`) VALUES  
('iloveshopping', 'b2ef636fbb4a278d15858fdf6f0a8b96'),  
('webadmin', '16c81a9cd3365c535750cc15a72bfa79');
```

```
-- -----
```

```
--
```

```
-- Table structure for table `order_receipt`
```

--

```
CREATE TABLE `order_receipt` (  
  `orderid` int(15) NOT NULL,  
  `productid` char(12) NOT NULL,  
  `unitprice` float(4,2) DEFAULT NULL,  
  `quantity` int(11) DEFAULT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

--

-- Dumping data for table `order_receipt`

--

```
INSERT INTO `order_receipt` (`orderid`, `productid`, `unitprice`, `quantity`) VALUES  
(2, '1234567890', 12.00, 1),  
(3, '1023478599', 12.00, 2),  
(4, '1000333444', 12.00, 3),  
(5, '1234567890', 12.00, 1),  
(6, '1000333455', 12.00, 1);
```

-- -----

--

-- Table structure for table `product`

--

```
CREATE TABLE `product` (  

```



```

`productid` int(12) NOT NULL,
`productname` varchar(100) DEFAULT NULL,
`unitprice` float(4,2) DEFAULT NULL,
`description` text,
`catid` char(15) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `product`
--

INSERT INTO `product` (`productid`, `productname`, `unitprice`, `description`, `catid`)
VALUES
(1234567890, 'Sackboy Funko POP! Figure', 12.00, 'Our favorite heroic rag doll in figure form.',
'109876543210123'),
(1231234564, 'Sackboy Funko POP! Keychain', 5.00, 'Now in keychain form! Carry Sackboy
around with your keys as you journey across your day!', '765123958578039'),
(1344556677, 'Sackboy Funko POP! Pin', 8.00, 'Need a little decoration? Pin Sackboy wherever
you would like!', '222333444555888'),
(1023478599, 'Oddsock Funko POP! Figure', 12.00, 'One of the legendary heroes of Bunkum!
An agile fella they are!', '109876543210123'),
(1000333444, 'Toggle Funko POP! Figure', 12.00, 'One of the legendary heroes of Bunkum! He
can switch between his large form and his small form.', '109876543210123'),
(1000333455, 'Toggle (Small ver.) Funko POP! Figure', 12.00, 'One of the legendary heroes of
Bunkum! He can switch between his large form and his small form.', '109876543210123'),
(1009988776, 'Swoop Funko POP! Figure', 12.00, 'One of the legendary heroes of Bunkum!
Watch her soar across the skies for new adventures!', '109876543210123'),
(1004567833, 'Newton Funko POP! Figure', 12.00, 'The antagonist of our story. Quite the
betrayal, was it not?', '109876543210123'),

```

(1079884733, 'Nana Pud Funko POP! Figure', 12.00, 'The mother of our main antagonist. She only wants to save her son from himself.', '109876543210123'),

(1255748932, 'Captain Pud Funko POP! Figure', 12.00, 'The father of our main antagonist. His son is quite the handful.', '109876543210123'),

(1010998877, 'Sackbot Funko POP! Figure', 12.00, 'Formerly made as a soilder for the Alliance, this robotic fella was brought to life to ultimately help save Craftworld.', '109876543210123'),

(1000001937, 'Oddsock Funko POP! Pin', 8.00, 'Our scampering hero now in pin form!', '222333444555888'),

(1380048329, 'Toggle (both ver.) Funko POP! Pin Bundle', 10.00, 'Stocky or small, our form-changing hero can now be pinned anywhere!', '222333444555888'),

(1099334422, 'Swoop Funko POP! Pin', 8.00, 'Our heroic avian now in pin form!', '222333444555888'),

(1008847583, 'Sackbot Funko POP! Pin', 8.00, 'Our expressive robotic friend now in pin form!', '222333444555888'),

(1000338859, 'Oddsock Funko POP! Keychain', 5.00, 'Carry Oddsock around with this keychain!', '765123958578039'),

(1055667788, 'Toggle (both ver.) Funko POP! Keychain Bundle', 5.00, 'Carry both forms of Toggle as keychain!', '765123958578039'),

(1011225577, 'Swoop Funko POP! Keychain', 5.00, 'Carry Swoop around as a keychain!', '765123958578039'),

(1022223333, 'Newton Funko POP! Keychain', 5.00, 'Carry Newton around as a keychain!', '765123958578039'),

(1993747385, 'Sackbot Funko POP! Keychain', 5.00, 'Carry Sackbot around as a keychain!', '765123958578039'),

(1008334421, 'Funko POP! Foldable Figure Protector', 6.00, 'Protect your figure cases with this foldable protector.', '123456789002345'),

(1098234556, 'Funko POP! Sackboy Backpack', 25.00, 'Perfect for the biggest fans of Sackboy and those who need some extra carry-on space.', '123456789002345'),

(1223377485, 'Funko POP! LBP Wallet', 15.00, 'Express your love for LBP while carrying your money.', '123456789002345'),

(1010112233, 'Funko POP! LBP Shirt', 20.00, 'Do you love LBP? We love LBP too! Show your love with this shirt!', '666777888999222'),

(1003034556, 'Funko POP! LBP Hoodie', 40.00, 'Do you love LBP? We love LBP too! Show your love with this hoodie!', '666777888999222'),

(1111111111, 'Oreo Funko POP! Figure', 12.00, 'This is Oreo. He is a cute little shih tzu. Be warned, he is very spoiled.', '109876543210123');

-- -----

--

-- Stand-in structure for view `product_category`

-- (See below for the actual view)

--

CREATE TABLE `product_category` (

`productid` int(12)

, `productname` varchar(100)

, `unitprice` float(4,2)

, `description` text

, `catid` char(15)

, `categoryid` char(15)

, `categoryname` varchar(50)

, `catdesc` text

);

-- -----

--

-- Table structure for table `shoppers`

--

```
CREATE TABLE `shoppers` (  
  `userid` varchar(50) NOT NULL,  
  `fname` varchar(20) NOT NULL,  
  `lname` varchar(50) NOT NULL,  
  `email` varchar(50) NOT NULL,  
  `phonenum` char(10) NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `shoppers`  
--
```

```
INSERT INTO `shoppers` (`userid`, `fname`, `lname`, `email`, `phonenum`) VALUES  
('iloveshopping', 'John', 'Smith', 'jsmith@example.com', '1234567890');
```

```
-- -----
```

```
--  
-- Table structure for table `shopping_cart`  
--
```

```
CREATE TABLE `shopping_cart` (  
  `prodid` char(12) DEFAULT NULL,  
  `quantity` int(11) DEFAULT NULL,  
  `totalprice` float(4,2) DEFAULT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
-- -----  
  
--  
-- Structure for view `customer_info` exported as a table  
--  
DROP TABLE IF EXISTS `customer_info`;  
CREATE TABLE `customer_info`(  
    `userid` varchar(50) COLLATE latin1_swedish_ci NOT NULL,  
    `fname` varchar(20) COLLATE latin1_swedish_ci NOT NULL,  
    `lname` varchar(50) COLLATE latin1_swedish_ci NOT NULL,  
    `email` varchar(50) COLLATE latin1_swedish_ci NOT NULL,  
    `phonenumber` char(10) COLLATE latin1_swedish_ci NOT NULL,  
    `customerid` varchar(50) COLLATE latin1_swedish_ci NOT NULL,  
    `street` varchar(30) COLLATE latin1_swedish_ci NOT NULL,  
    `city` varchar(30) COLLATE latin1_swedish_ci NOT NULL,  
    `state` varchar(25) COLLATE latin1_swedish_ci NOT NULL,  
    `zipcode` char(5) COLLATE latin1_swedish_ci NOT NULL  
);
```

```
-- -----  
  
--  
-- Structure for view `product_category` exported as a table  
--  
DROP TABLE IF EXISTS `product_category`;  
CREATE TABLE `product_category`(  
    `productid` int(12) NOT NULL,
```

```
`productname` varchar(100) COLLATE latin1_swedish_ci DEFAULT NULL,  
`unitprice` float(4,2) DEFAULT NULL,  
`description` text COLLATE latin1_swedish_ci DEFAULT NULL,  
`catid` char(15) COLLATE latin1_swedish_ci DEFAULT NULL,  
`categoryid` char(15) COLLATE latin1_swedish_ci NOT NULL,  
`categoryname` varchar(50) COLLATE latin1_swedish_ci DEFAULT NULL,  
`catdesc` text COLLATE latin1_swedish_ci DEFAULT NULL  
);
```

```
--
```

```
-- Indexes for dumped tables
```

```
--
```

```
--
```

```
-- Indexes for table `address`
```

```
--
```

```
ALTER TABLE `address`
```

```
ADD PRIMARY KEY (`customerid`);
```

```
--
```

```
-- Indexes for table `category`
```

```
--
```

```
ALTER TABLE `category`
```

```
ADD PRIMARY KEY (`categoryid`),
```

```
ADD UNIQUE KEY `categoryid` (`categoryid`);
```

```
--
```

```
-- Indexes for table `cust_order`  
  
--  
ALTER TABLE `cust_order`  
  ADD PRIMARY KEY (`orderid`),  
  ADD UNIQUE KEY `orderid` (`orderid`),  
  ADD KEY `customerid` (`customerid`),  
  ADD KEY `pid` (`pid`);  
  
--  
  
-- Indexes for table `employee`  
  
--  
ALTER TABLE `employee`  
  ADD PRIMARY KEY (`empid`);  
  
--  
  
-- Indexes for table `inventory`  
  
--  
ALTER TABLE `inventory`  
  ADD KEY `itemid` (`itemid`);  
  
--  
  
-- Indexes for table `login`  
  
--  
ALTER TABLE `login`  
  ADD PRIMARY KEY (`username`),  
  ADD UNIQUE KEY `username` (`username`);
```

```
--  
-- Indexes for table `order_receipt`  
--  
ALTER TABLE `order_receipt`  
  ADD KEY `orderid` (`orderid`),  
  ADD KEY `productid` (`productid`);
```

```
--  
-- Indexes for table `product`  
--  
ALTER TABLE `product`  
  ADD PRIMARY KEY (`productid`),  
  ADD UNIQUE KEY `productid` (`productid`),  
  ADD KEY `prod_cat` (`catid`);
```

```
--  
-- Indexes for table `shoppers`  
--  
ALTER TABLE `shoppers`  
  ADD PRIMARY KEY (`userid`);
```

```
--  
-- Indexes for table `shopping_cart`  
--  
ALTER TABLE `shopping_cart`  
  ADD KEY `prodid` (`prodid`);
```



```
--  
-- AUTO_INCREMENT for dumped tables  
--  
  
--  
-- AUTO_INCREMENT for table `cust_order`  
--  
ALTER TABLE `cust_order`  
  MODIFY `orderid` int(15) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;  
COMMIT;  
  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

References / Sources:

https://www.w3schools.com/sql/sql_datatypes.asp

https://www.w3schools.com/howto/howto_css_searchbar.asp

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model

<https://www.geeksforgeeks.org/create-a-drop-down-list-that-options-fetched-from-a-mysql-database-in-php/>

<https://www.dolthub.com/blog/2024-01-26-writing-mysql-views/>

https://www.w3schools.com/php/func_mysqli_real_escape_string.asp

https://www.tutorialspoint.com/php/php_mysql_login.htm

<https://stackoverflow.com/questions/59777249/update-stock-using-php-and-mysql>

<https://forums.phpfreaks.com/topic/262529-linking-to-a-single-product-page/>

https://www.w3schools.com/php/php_superglobals_get.asp

<https://www.geeksforgeeks.org/mysql-update-view/>

https://www.tutorialspoint.com/php/php_function_session_id.htm

<https://stackoverflow.com/questions/17532310/session-variable-in-select-statement>

https://www.w3schools.com/php/php_date.asp