Cookie Run: Sweets Shop

Cassie Ihekwaba
ihekwabac1@montclair.edu
(https://cyan.csam.montclair.edu/~ihekwac1/)

User Accounts

User Side:

nayohminty@sweetshop.bake	itwillbecool7
cquaba@sweetshop.bake	123meloncookie

Admin Side:

admin01@sweetshop.admin	sourmelon12345@
admin02@sweetshop.admin	lazybonesjones

These will be the user and admin accounts made to access the respective sides of the website - this information stored in separate tables since they are both made to access specific views.

Project Requirements

Business Plan: Enhanced

1. What are you developing?
   What I decided to develop is a merchandise website solely created around a mobile game series I personally enjoy called Cookie Run. This is meant to be a website where players of the game can gain access to special merchandise surrounding the characters (known as “Cookies”) of the game in three product types - Keychains, Shirts, and Stuffed Plushies. The main idea of the site that I wanted to focus on was its simplicity and accessibility for Users, so the user side is left highly simplified and easy to navigate.

2. Where did I get my materials? Am I modeling my site after another site?
   My material from this site was mainly harvested from the official Wiki page for the game purely for images of the characters when making the products. The design of the customer side mainly and some of the admin side came from a past project that I had worked on with a group from a past class, and some existing templates that I had found on certain domains that mainly dealt with how to construct the PHP side of login/signup pages. A rough idea for what to include in my receipt came from how Amazon invoices look, but mine are much more simplistic and include minimal information.

   Excluding the project website that I had worked on in a group last semester, I am not modeling this site off of another currently existing domain. The only similarity that is shared is that this is a merchandise website for a mobile game series, however that is the end of the similarities.

3. The sources that I used in the project are listed in the chart below along with what I used them for and their trustability:

Websites and Sources	Role in Project	Worthiness
Internet Computing Final Project and Homework	Helped in construction of admin and customer side	Highly worthy; past work from group assignment that I knew I could use as soon as this project was announced. Most of this code is used throughout the entire project - mostly modified from this code so It is highly valued/worthy.
https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php	Helped with creating login and sign up pages	Highly worthy - from a domain that does offer tutorials and templates on how to construct these site elements - When I was running into issues with how to fix certain elements of the login page to be object-oriented and not PDO, this site helped me.
https://www.php.net/manual/en/function.password-hash.php	Hashing password help	Official manual helped me figure out password hash

		issue in most of my code
https://www.w3schools.com/howto/howto_html_file_upload_button.asp	Learned file upload	Helped solve the issue on how to enable hole upload
https://www.geeksforgeeks.org/define-multipart-form-data/	Set up HTML to take image as input	Helped figure out half my file upload problem on HTML with actually accepting a media file as input with form data modifier for multi data
https://www.w3schools.com/howto/tryit.asp?filename=tryhow_html_file_upload_button	Set up code to prepare image for upload to database	Helped me figure out how to prepare statements to upload images to site and database properly
https://www.tutorialspoint.com/how-to-validate-an-email-address-in-php	Email address validation for checkout and signup	Had helpful information on how to validate email in PHP since I didn't know how to implement that function.
https://cookierunkingdom.fandom.com/wiki/List_of_Cookies	Pictures of cookie characters	Trusted domain for accurate illustrations of official cookie characters since it is official
Minimal Source Code from Welling Book given in Database Class	Just helped with basic PHP/HTML	Worthy examples from source code helped visualize what I should be doing if I get

		stuck

Analysis of Problems

For this website, my aim was to make it as simple as possible with the basic components of a small shopping portal. That’s the keyword - simple. I just needed the bare essentials for any shopping site to make a basic construction in diagrams and simple implementation as will be explained below:

1. Anchor Concepts: The anchor concepts for this site were originally longer than what was seen in the old V3 from November: The anchor concepts as they are now are CUSTOMER, for customer data like account ID, email and password. ADMINISTRATORS, which consists of administrator-specific login details separate from the CUSTOMER side since they have different views. PRODUCTS, that is just the site's products that have their respective product id, followed by other identifiers for the specific item, ORDERS, which hold orders for the customers under their email and randomly generated order numbers, and RECEIPTS that makes a randomly generated receipt number with the inherited order number followed by the order total, address, and the name and customer email the order was placed under. A CART originally existed as an anchor table, but I opted to change it to a class for reasons that will be explained later.

2. Anchor Interactions: Data interactions are as follows below:

CUSTOMER browses PRODUCTS
CUSTOMER can view CART*
CUSTOMER can edit CART*
CUSTOMER can make an ORDER

CUSTOMER can cancel an ORDER
CUSTOMER can lookup RECEIPTS
ADMINISTRATORS can cancel ORDERS
ADMINISTRATORS can remove CUSTOMER
ADMINISTRATORS can edit PRODUCTS
ADMINISTRATORS can lookup RECEIPTS
ORDERS generate RECEIPTS

All of these are fairly simple interactions for a small shopper site between entities and the class of CART (CART is denoted with a star to differentiate it from the entities that will become tables in the DB).

3. Meaningful Words: There are mainly Required words such as “can only”, which indicates that the action that is about to be done can be completed under certain conditions. And “must”, being for any dependencies in my project. “At a time” being another that occurs pretty frequently, as well. Number Indicator words such as “As least one”, “one”, and “one or more”. Most of these words are stating an optional cardinality between tables and their corresponding relationships, and one stating a required minimum and maximum of “1…1” on the table - this being for carrying out certain actions such as placing an order or removing a user/admin. Only one dependency exists in my project where the others that use “must” are simply required words for constraining cardinality ranges.This mainly revolves around ORDERS and RECEIPTS where an order has to exist before a receipt can be generated. Since a receipt is also an order record, once an order is canceled, the receipt record is removed with it.

4. Assumptions: All of my assumptions are listed below - all following the meaningful words and being reflect via cardinality and participation in the logical diagram to follow later:

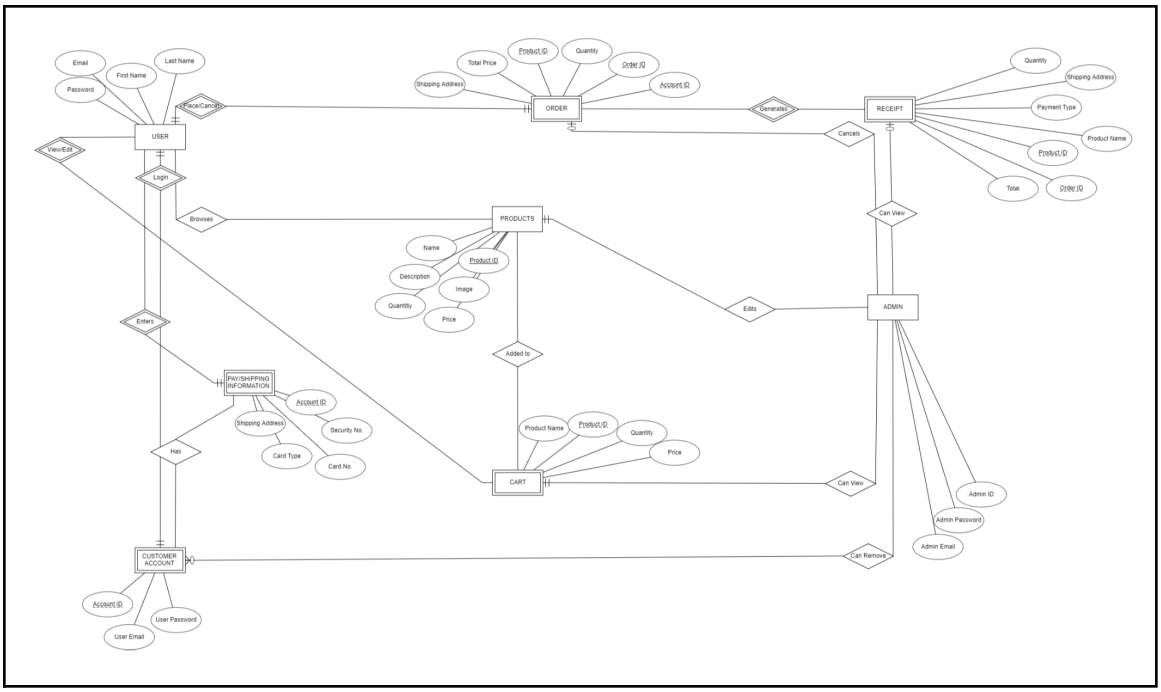
CUSTOMER can only make one ORDER at a time
CUSTOMER can cancel one or more ORDERS
CUSTOMER can view one or more of their RECEIPTS
There can be more than one ADMINISTRATOR
ADMINISTRATOR can remove ADMINISTRATORS
ADMINISTRATORS can only remove one ADMINISTRATOR at a time
ADMINISTRATORS can view one or more RECEIPTS
ADMINISTRATORS can only cancel one or more ORDERS
ADMINISTRATORS can only remove one CUSTOMER at a time
ADMINISTRATORS can only add/edit/delete one PRODUCT at a time from PRODUCTS
At least one PRODUCTS must be added to cart to place an ORDER
At least one PRODUCTS must be in an ORDER
ORDER generates one receipt per ORDER
ORDERS must be placed before a RECEIPT is made

With the previous information, it helped me formulate my newer diagrams and eventually solution to the project.

Solutions

ER Diagrams:

1. November Version:



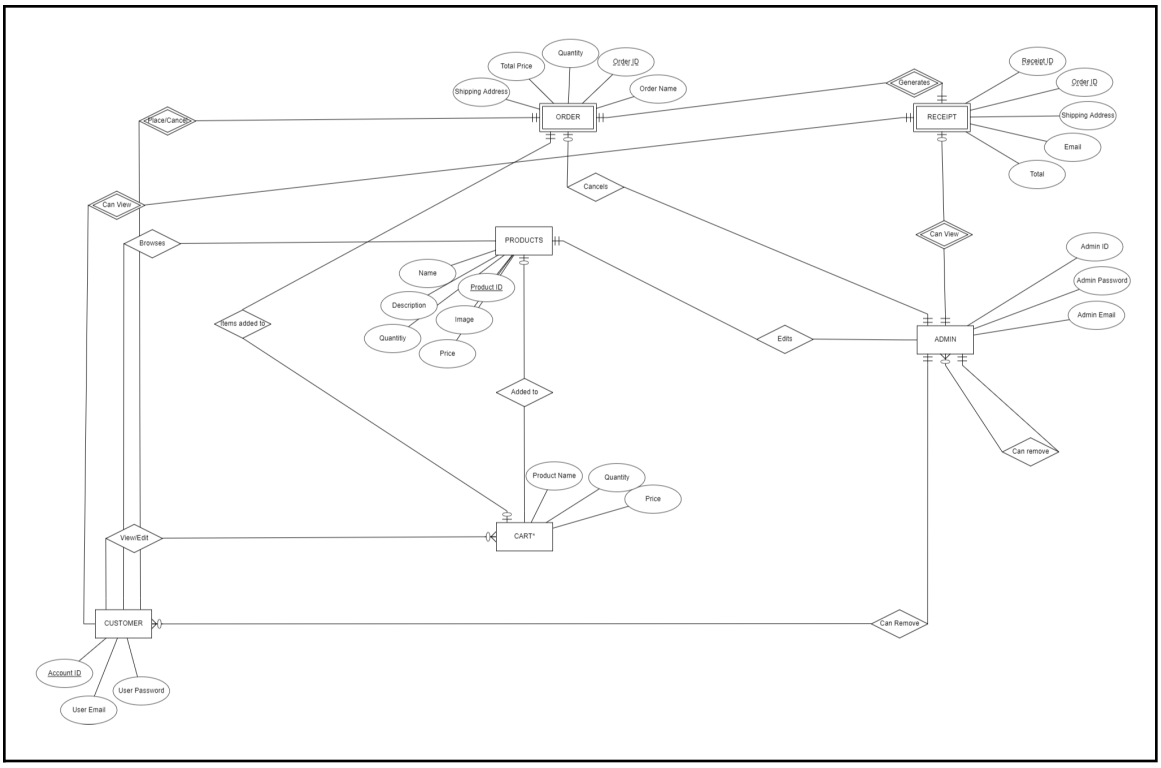
Element Explanation: In this November version of the ER diagram, there are as stated previously, more entities, and CART is still considered a class in this instance. Since it was an earlier stage, I thought it to be possible to have as many tables as I did and have them all interact the way that I envisioned before the later version and newer updates that were more realistic and more simplistic than what I had thought previous:

Element	**Explanation**
User	This would be a storage point for user data based on who already had an account with the gamer series, Cookie Run. It was initially going to be an exclusive shopping site for players, not too dissimilar to how a “Fanclub” would work where you’d get exclusive access to certain merchandise that wasn’t able to be attained by the public - this holds customer information to be used in orders. Here is the name of the player, along with

	basic login information.
Customer Account	This was an extension of customer that was supposed to have a “site-only” email address - basically a custom handle attached to your in-game user name since the login credentials fros USER were meant to be the email and password you used for the game - it’d be cross referenced with the player database from the games, and once a username/password was found that matched, it would pull that and then generate the email handle to go with your site credentials. It would have an account id, username and password.
Pay/Shipping Info	This is simply shipping information that would be entered by each user on the site and stored into a table to be pulled for certain orders that they’d make on the site. It was meant to be faster than retyping all credentials in. However, as will be seen later, this was a massive oversight on my part since this was disallowed for the SPECs of the assignment. At this moment I had not thought about it, so it was meant to be its own standalone table.
Products	This is just the products table for everything on the website.
Order	This holds all the orders of people who have placed any orders on the website while using it.
Cart	For this component, my idea was to have to hold all cart items of each user under an account id - so if a cart search was needed to be done by an admin - a SPEC is grossly misunderstood, they could do a lookup

	by account ID and find what each user had and was ready to buy. This was meant to be constantly updating as well, so anyone checking out, adding or removing items from their cart, on the database side it would be seen that the cart is constantly active (like live querying). - This would utilize the AccountID, ProductID, Product Name, Quantity and Price of each item.In the diagram, it does not have Account ID as an foreign key - this was a correction that I would make when starting to implement my database.
Receipt	This would hold all receipts from orders that were generated from completed orders. Holding quantity of the order, payment type, shipping address, Product name, product ID, Order ID, and the total price of the cart.
Admin	This is just the admin control section of the website - it would contain the admin id, admin email and password. This would grant access to everything on the site with permissions of an admin - they could add, edit, and remove products, look at carts, remove users, remove admins, remove orders, and view receipts.

2. Current Version: In this current version of the diagram, there are much less entities, slightly more relationships, and a more simple, less hectic look. A key component of this new diagram, CART, I kept hinting at becoming a class in this later design, will be explained in depth here:



Element Explanation:

Element	**Explanation**
Customer	This is where customer data is stored for login validation on the site - instead of making the site exclusive, this would make it so that anyone can sign up, since that was stated in the SPEC to begin with.
Order	This is where all orders on the website are stored after checkout. After an order is complete, a receipt is generated and injected into the receipts table with the data from the order with a receipt id attached to it.
Products	A product table that contains all products currently listed on the site - up to 30, in order of product type, easily discernible by name, rather than

	category.
Cart (Class)	An easier to manage version of the cart table, being converted into an actual class. This was more logical since it falls in line with what was supposed to happen with the cart, since there is supposed to be one per user, rather than a shared one. Users can see into it, modify it, get a message when it's empty, and checkout with it. It is also easier to manage since it is an array, rather than a constantly updating table, so there is no danger of products getting mixed up and possibly a strange table failure resulting from the table processes becoming overwhelmed, and even admin error where the entire cart table can be mistakenly emptied from PHPmyAdmin. This, amongst being odd to implement, was just not practical.
Receipt	This holds all data from Orders in a tuple with a receipt ID attached to it. Since these are connected to orders, they double as an "order tracker". So if a user wanted to remove an order, the receipt would disappear with it. These cannot exist until an order is placed - they have a 1:1 existence. One can't exist without the other.
Admin	A table that holds administrator login credentials for login verification. They are able to oversee all functions on the website such as product modification, user moderation and admin removal.

3. Comparison:

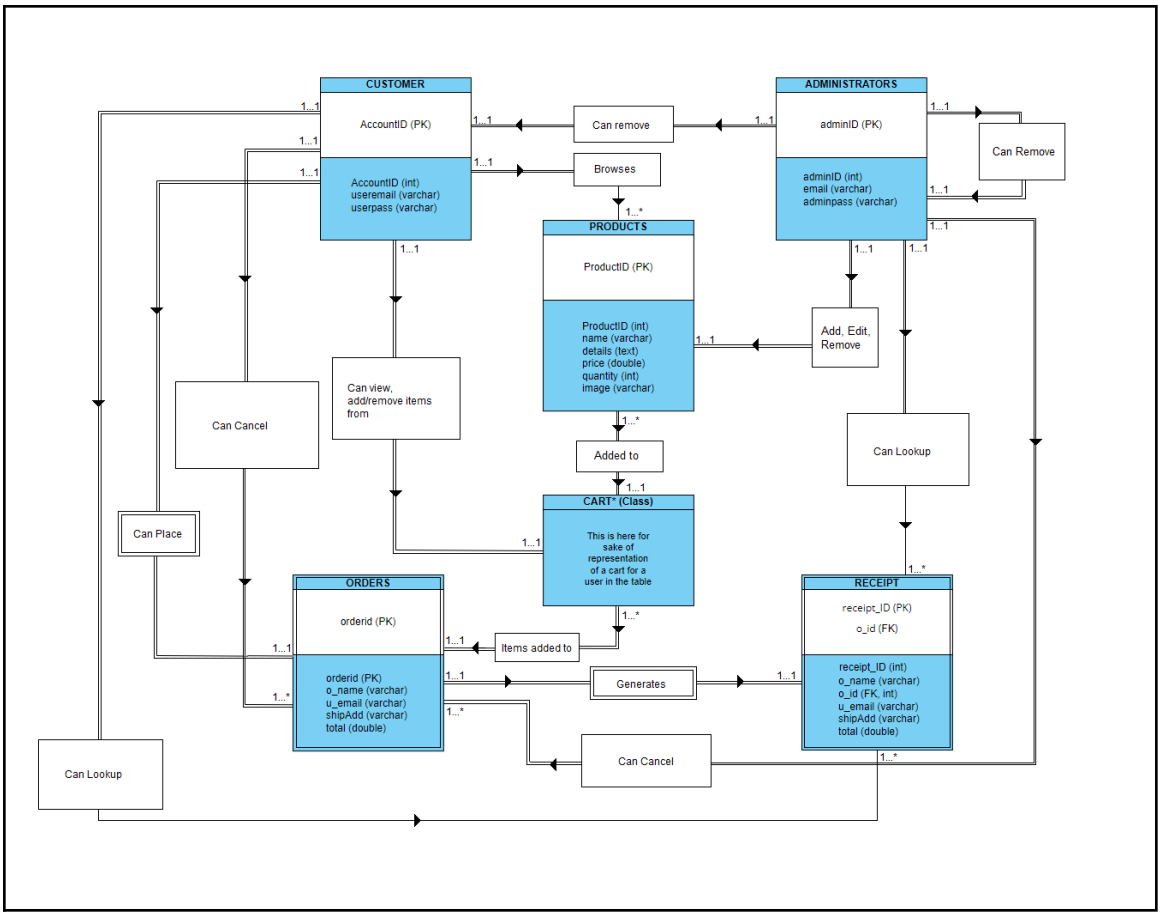
Old	**New**

User and Customer Table holds separate data for player identification to be allowed onto site	Only Customer table - easier to understand, allows for signup, and less redundant.
Cart is a table to allow for monitoring on Admin side and keeps user's products organized via account ID. Cannot exist unless filled with products.	Cart is now a class that gives users in sessions their own session cart to keep it separate and more easily manageable and safer. If kept in its original state, this would not allow for users to see their own products in the way I thought because as a table, it would show everyone's cart products, rather than their own. A system of identifying the products by account ID would be too cumbersome to implement and overall makes no sense. This old version would also be open to too many failures that can be brought on by human error - an entire dump of the CART table, or even drop can cause more issues than I initially thought. The cart was also initially weak - when it is not.
Pay/Shipping Table holds all shipping and payment information.	Shipping information is entered at orders and any reference to payment has been removed. Having a separate table for all the information that would be entered into the checkout page makes no sense in any capacity - I decided to move this information to the ORDERS

	table since at checkout, it will be injected there. Since this is not an in-depth website, I was gravely overlooking what was possible and focused on what would be “cool” to have.

Logical Model (Final Version)

1. In the diagram below, it is very different from a typical relational diagram - this was mainly done to fully demonstrate what was being asked of me in the documentation because a relational diagram I feel, would get too cluttered and wouldn’t fully explain or demonstrate how everything is flowing/working within my system. This setup came from a google search of a “logical diagram” to give me an idea of what it actually was, and from a powerpoint earlier in the semester on these diagrams where on one page it showed a ER diagram adapted into a UML diagram - I took both of these concepts into mind and came up with the final result below:



2. As stated previously, I made my diagram this way because it was easier to reflect the relationships, cardinality, and participation this way, rather than on a relational diagram, or redoing the ER diagram to have more information on it, rendering it more unreadable than it already is. It also helps map out how everything is actually working within the system better than the ER diagram from before was - it also made it easier to make corrections to cardinality constraints if need be so this could be a finalized, latest representation of how everything stands in my database and website:

Element	**Explanation**
Customer	This is where customer data is stored for login validation on the site - allows for

	sign up.
Order	This is where all orders on the website are stored after checkout. After an order is complete, a receipt is generated and injected into the receipts table with the data from the order with a receipt id attached to it.
Products	A product table that contains all products currently listed on the site - up to 30, in order of product type, easily discernible by name, rather than category.
Cart (Class)	Holds customer’s products based on “session”, so each user can have their own cart, and not share it in a massive table.
Receipt	This holds all data from Orders in a tuple with a receipt ID attached to it.
Admin	A table that holds administrator login credentials for login verification. They are able to oversee all functions on the website such as product modification, user moderation and admin removal.

Relationship	**Cardinality**	**Participation**
Customer Places Order	Customer: 1 Orders: 1	Customer: Total Orders: Total
Customer Cancels Order	Customer: 1 Orders: N	Customer: Total Orders: Total
Customer Can View Cart	Customer: 1 Cart: 1	Customers: Total Orders: Total
Customer Browses Products	Customer: 1 Products: N	Customer: Total Products: Partial
Customer Looks up	Customer: 1	Customer: Total

Receipts	Receipt: N	Receipt: Partial
Products added to Cart*	Products: N Cart: 1	Products: Total Cart: Total
Products Added To Order	Products: 1 Order: 1	Products: Total Order: Total
Order Generates Receipt	Order: 1 Receipt: 1	Order: Total Receipt: Total
Admin Cancels Order	Admin: 1 Orders: N	Admin: Total Orders: Total
Admin Adds/Edits/Deletes Products	Admin: 1 Products: N	Admin: Total Products: Total
Admin Can Remove Customer	Admin: 1 Customer: 1	Admin: Total Customer: Total
Admin Can Remove Admin	Admin: 1 Admin: 1	Admin: Total Admin: Total
Admin Can Lookup Receipts	Admin: 1 Receipt: N	Admin: Total Receipt: Partial

Weak Entity	**Explanation**
Receipt	Receipt cannot exist unless an order is placed and matched back to that order
Order	Orders cannot exist unless placed by a customer

Relationship	**Explanation**
Customer Places Order	Need to make a connection between customers and order since this is an ecommerce site Can’t just assume they can make

	orders - needs to be laid out and be visible in diagram
Customer Cancels Order	Needs to be clear who can and can’t cancel orders - if this is not clear, it can be assumed that orders are final, and cannot be canceled - this can’t be something that happens.
Customer Can View Cart	Customers should be able to see into and modify their cart as need be, so this relationship is needed. This is something that cannot be assumed.
Customer Browses/views Products	On the landing page, a customer can immediately browse products using the scroller on the side of the page. If they are unable to browse these products, they wouldn’t be able to add these items to their cart, effectively locking them out of shopping.
Customer Views/Edits Cart	Customer views items in cart and is able to increase or decrease the amount of product in cart, or remove it from their shopping cart entirely.
Products Added To Order	In order to place an order, items must be carried over from cart and put into order/checkout screen. Without this, orders would always be empty.
Order Generates Receipt	After order is complete, information from order is put into receipt to store order under that email and name.

Admin Cancels Order	If a user places an order, and for some reason an admin must remove it, they can reverse an order by removing it from the system, as well as the receipt.
Admin Edits Products	Admin must be able to modify site products
Admin Can Remove User	Any user that violates site terms must be removed from site - admins must be able to do this action.
Admin Can Remove Admin	If admins abuse powers or don't follow their tasks, an admin can flag them for removal for administration of the site.
Admin Can Lookup Receipts	Can track user orders based on email of the customer to find all orders under their name and email address.

There are no compound attributes or multivalued attributes present in my diagram - I initially had one for shipping address, but that could just be entered on one line and stored as such since I was looking back at the COMPANY database we had used as an example in class before. Not necessarily the smartest move, but I was looking to streamline parts of my site and keep them as tidy as possible without having too many attributes in one spot. I was essentially remapping the project to match the amount of manpower on it - the size of which being 1 since it was an independent project on my end.

EER Concepts in Table - there are no EER concepts in the table directly, but the only one that I am conscious of is the USER superclass consisting of Admin and Customer. This is only a concept, however, since I was intentionally

keeping the level of complexity as low as possible to keep from having more elements than I initially needed for a simple shopping domain.

Primary Keys:
The **primary keys** of the table are AccountID, adminID, receipt_ID, orderid, and ProductID.

The **Secondary Key/Candidate Key** in the tables are the email addresses attributes of useremail, email(ADMINISTRATOR), and u_email. They are the 2nd-most unique element in the table next to their primary key compared to the more generalized attributes and excluding the foreign keys.

I chose the primary keys the way that I did so I could at bare minimum be able to insert, modify or delete elements in a table since without a key, I could not do so. The AccountID, adminID, and ProductIDs are all auto incrementing keys - I just needed a basic primary key that would allow me to modify the tables at will. I set it to AUTO INCREMENT to keep an ordered record of which user joined last, and keep track of increment jumps when a user is removed from the site's database (1 that jumps to 3, due to the 2nd user being removed by an admin).

Initially, this was going to be for order/cart tracking purposes, but with the impracticality of having a constantly updating cart table that would have to mark each individual item in a cart with the user's ID to know that those are that user's "cart inventory", then log the same items under a receipt based on product ID, issues began to mount and goals began to look more nonsensical. For the incrementing keys, it is purely for keeping track of what came first and what came last in the database.

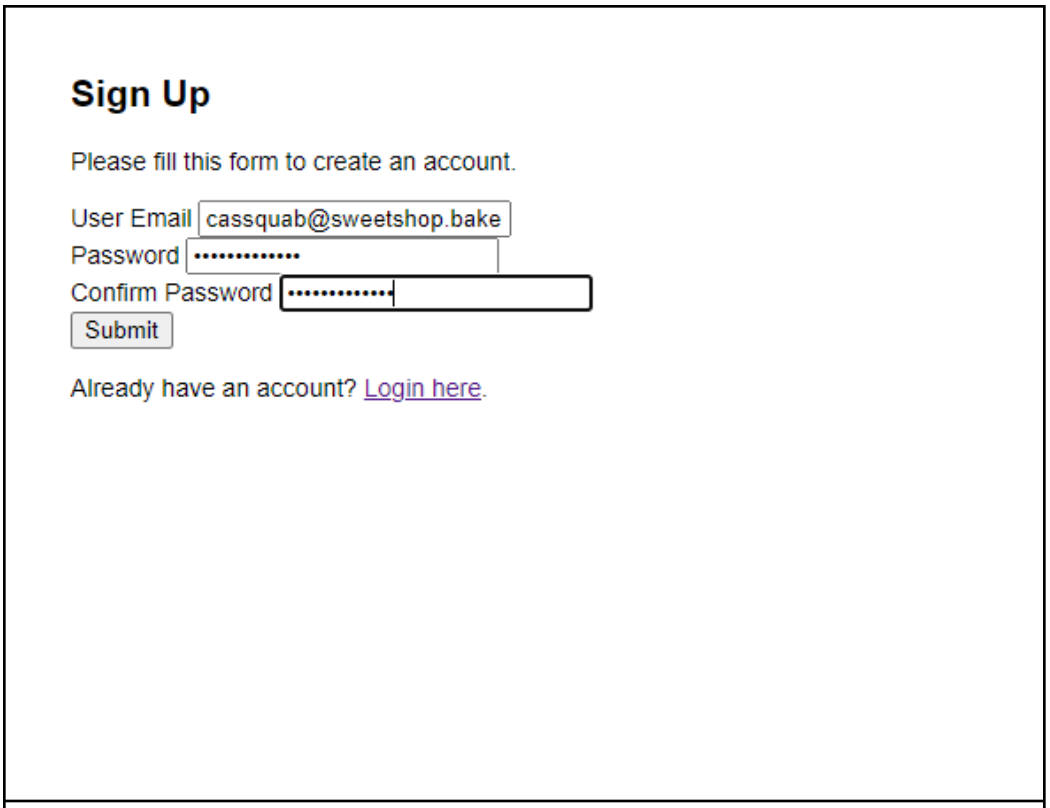
For receipt_ID and order_ID, these are randomized since I thought it would make more sense to do this, rather than

mark each order as 1,2,3, or 4. When a record gets deleted for instance, a jump in order records from 3 to 5 would not look clean/organized when it comes to database design like this in my understanding. So for post-order processing, I opted to keep this randomizer in since it made the most sense logically. Also since this was from a template of a past project I did with a group in Internet Computing where the same randomizer for a receipt/order was used, it was easier to implement than once again using an incrementer.

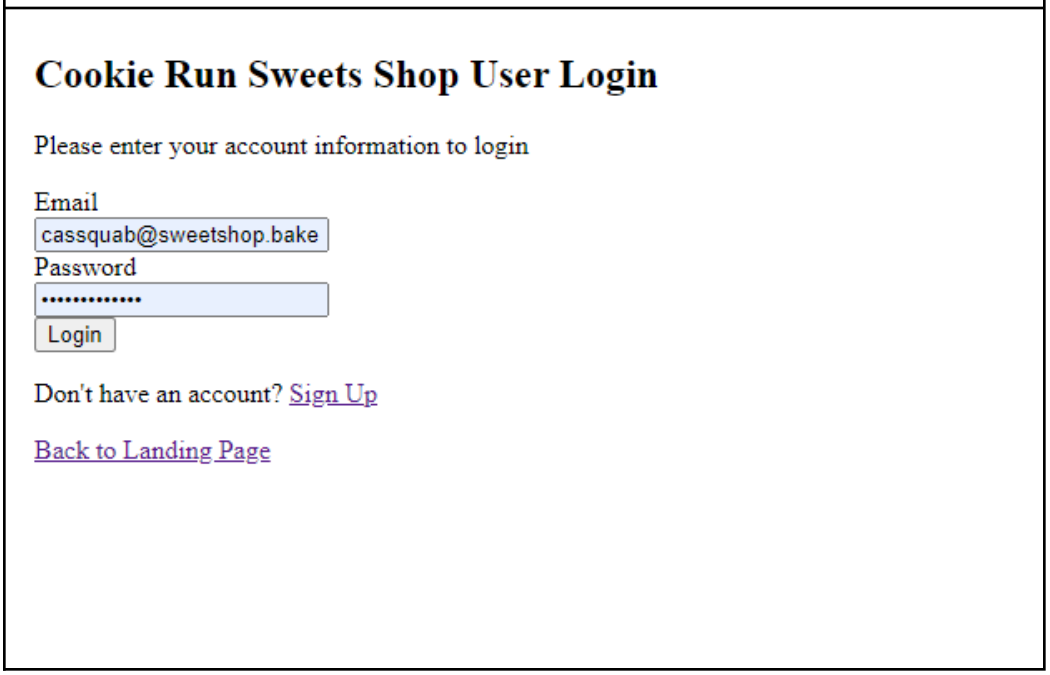
The only **foreign key** is o_id, from the ORDERS table. This is injected into the RECEIPTS table for post-processing and creation of the user's order history/receipts containing this receipt. If they wanted to cancel an order, they could look up their order via their email, get the individual number from the order (since they are ordered based on what was ordered first to last, or 1st to 3rd), then cancel it on another page - either with the individual number, or an email address that would cancel all their orders, removing them from receipt history since the receipts also act as an "Order History" page, as well.

Application Development

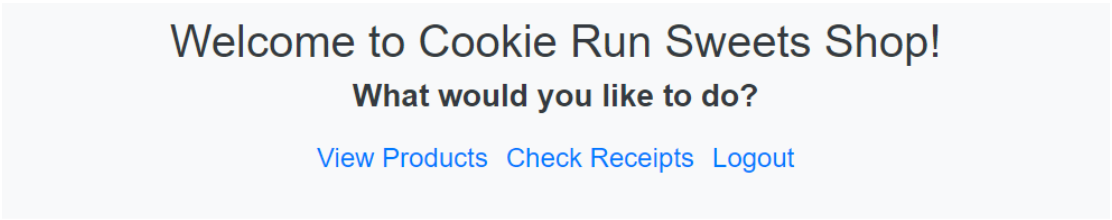
User Side (before on Top, After on Bottom): All explanations for screenshots are at the end of this section. Each paragraph is in order of appearance.



SQL: **$sql = "INSERT INTO CUSTOMER (useremail, userpass) VALUES (?, ?)";** (? is the useremil and hashed password generated by SQL code. It inserts the final values into the CUSTOMER table)

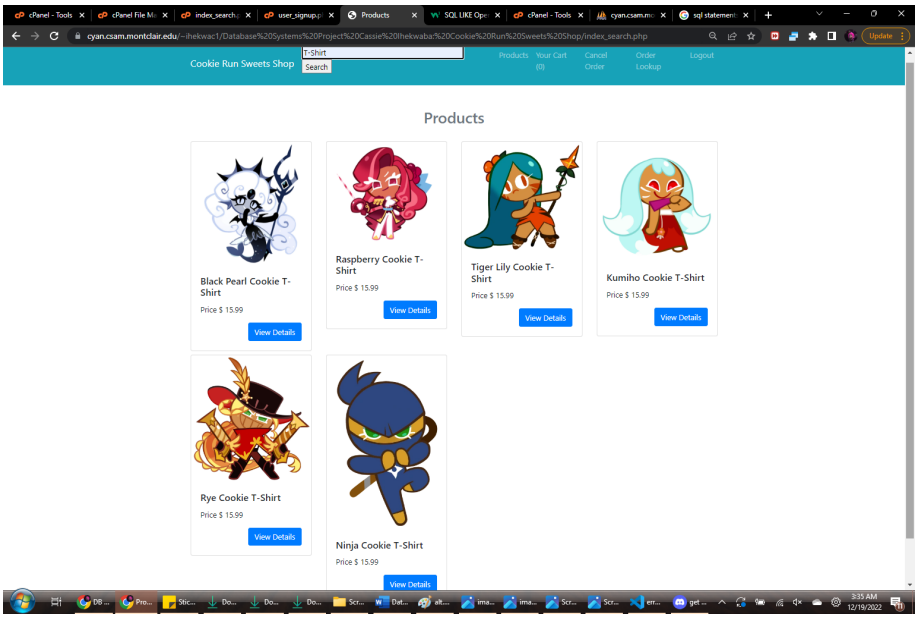



```
$sql = "SELECT * FROM CUSTOMER WHERE useremail = ?";
```

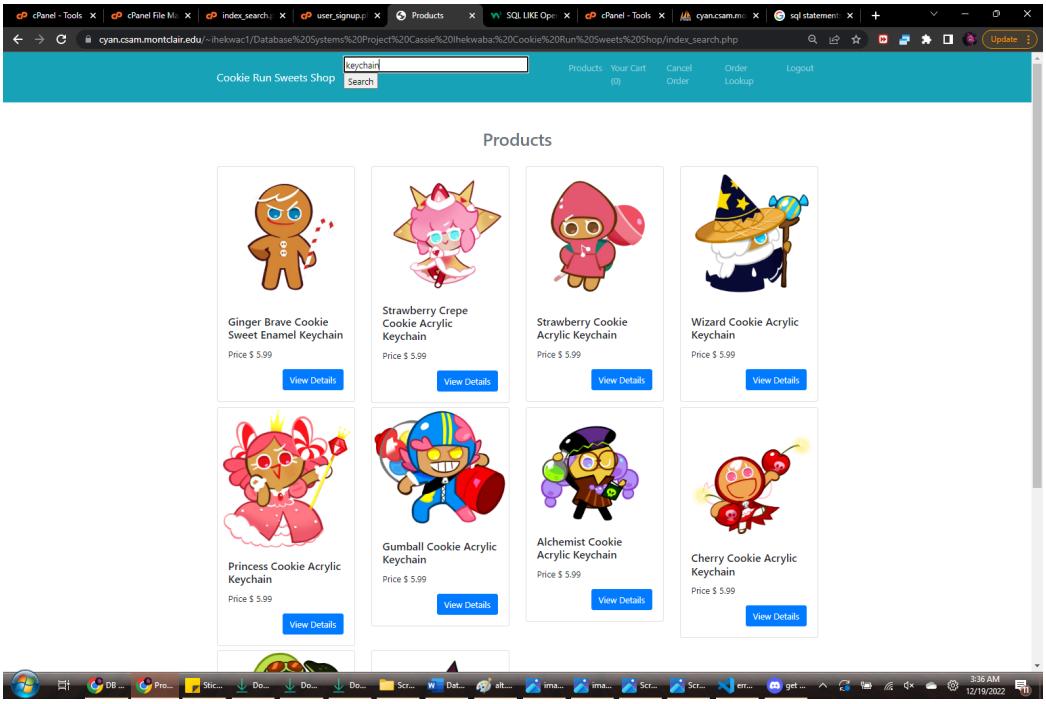


$sql="select * from PRODUCTS where name like '%".$searchterm."%'"; or select * from PRODUCTS; - This is how I split up categories - purely by name of each product - search a certain term at the end to find the products under that product type or scroll normally. They are all in order.

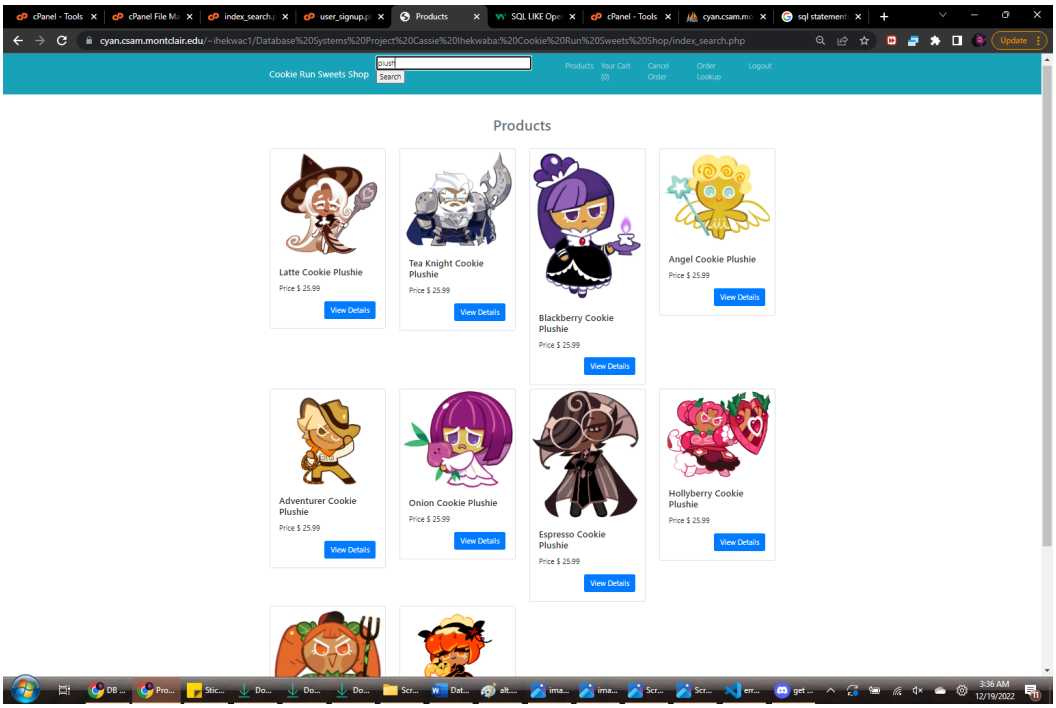
Searchterm = T-shirt:



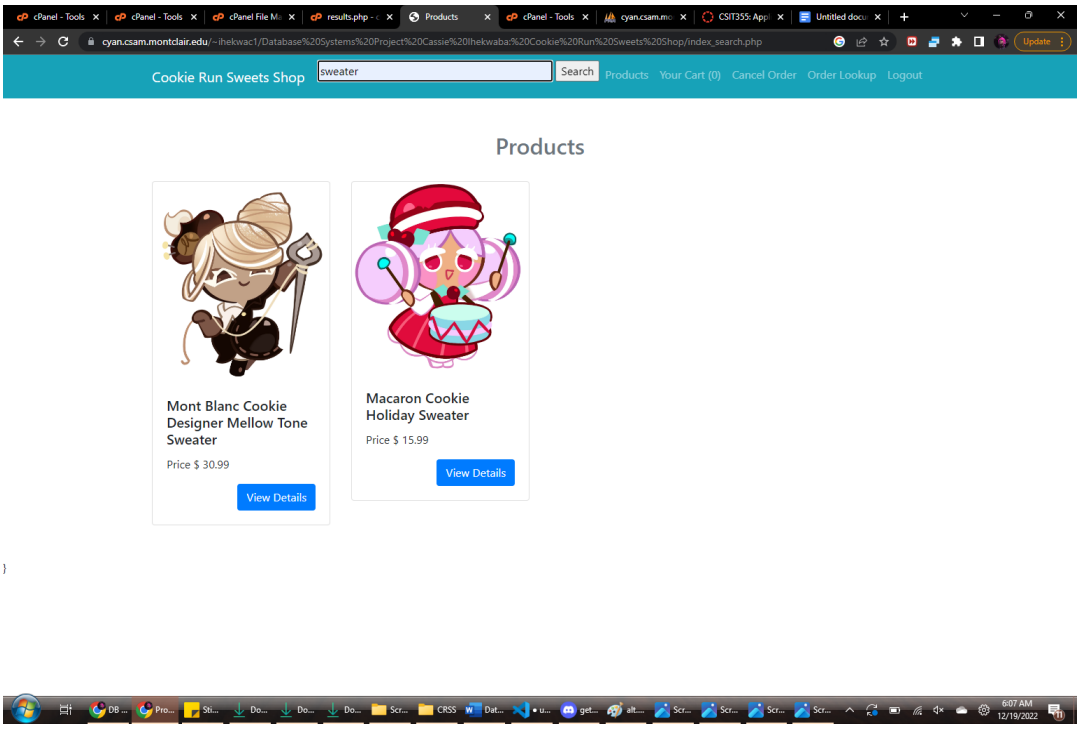
Searchterm = keychain:



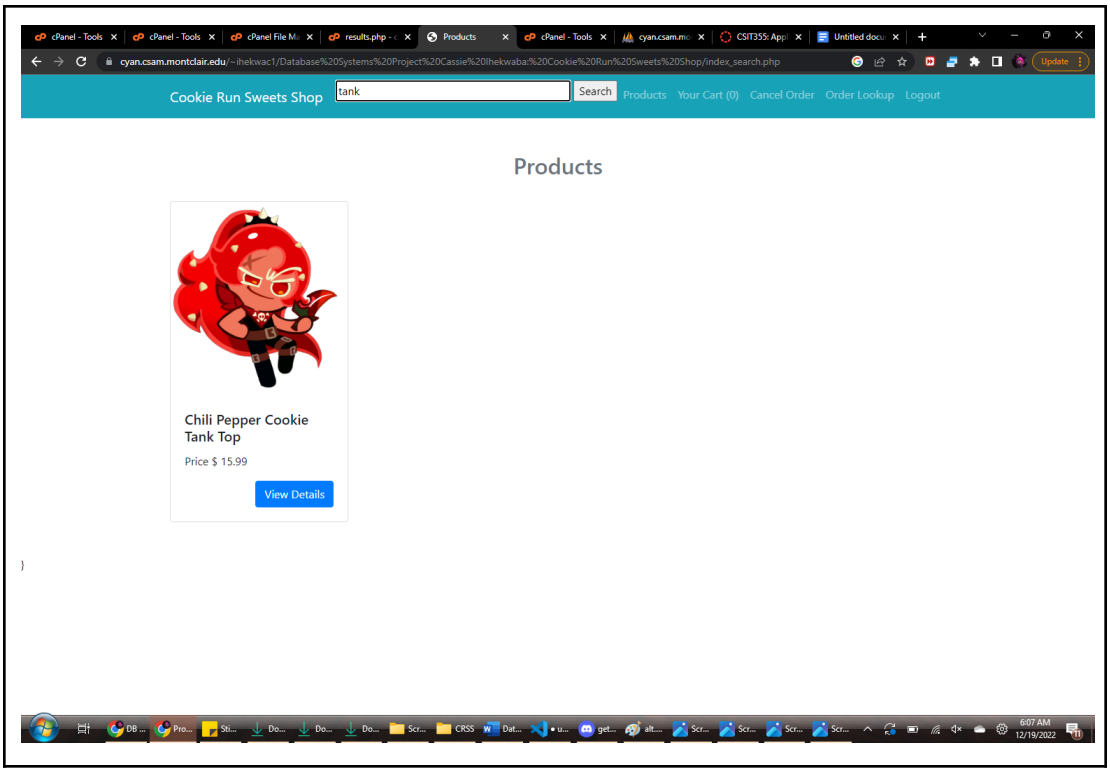
Searchterm = plush:

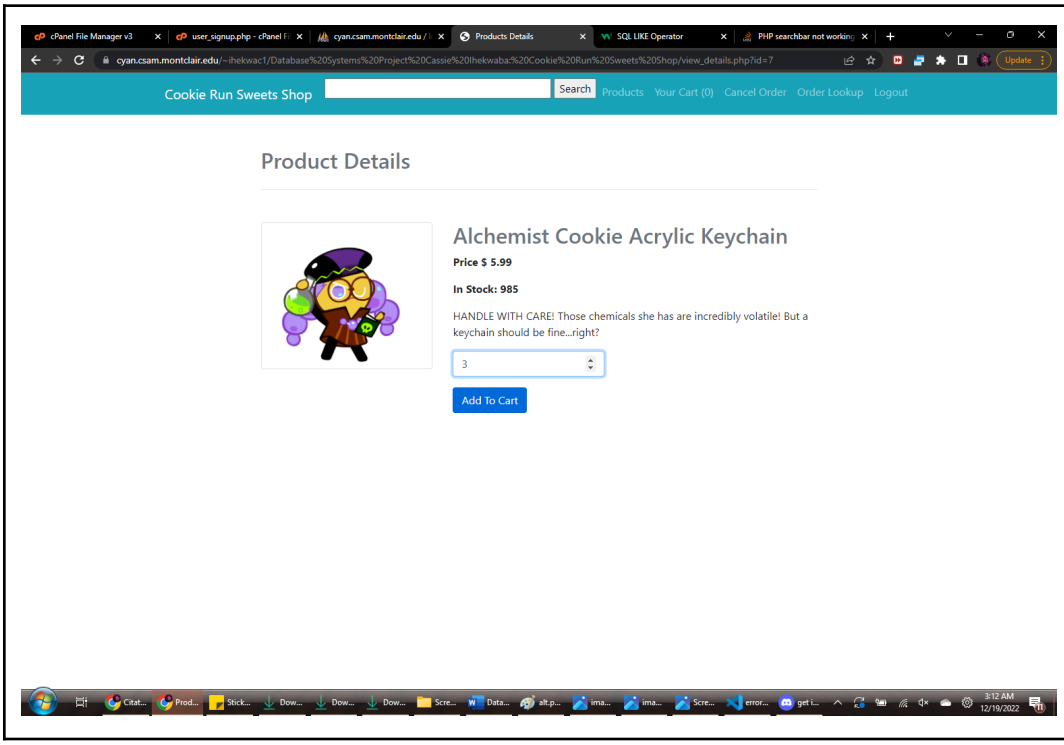


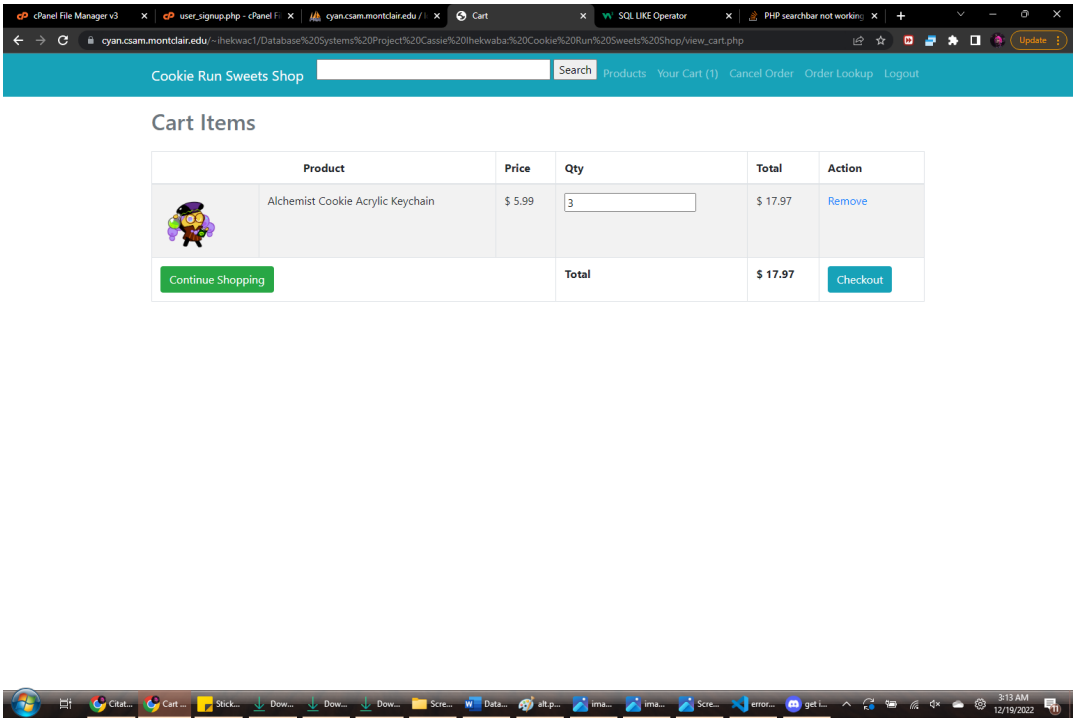
Searchterm = sweater:



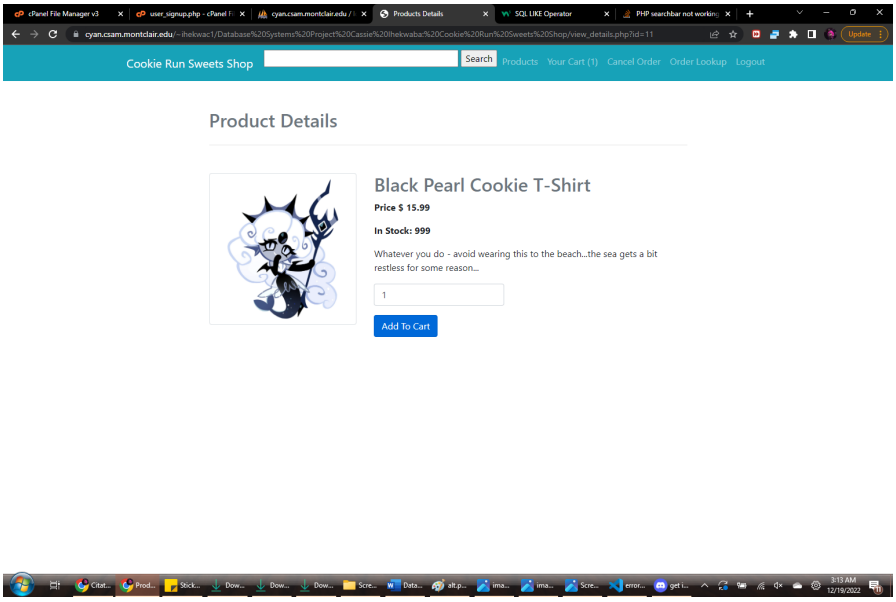
Searchterm = tank:

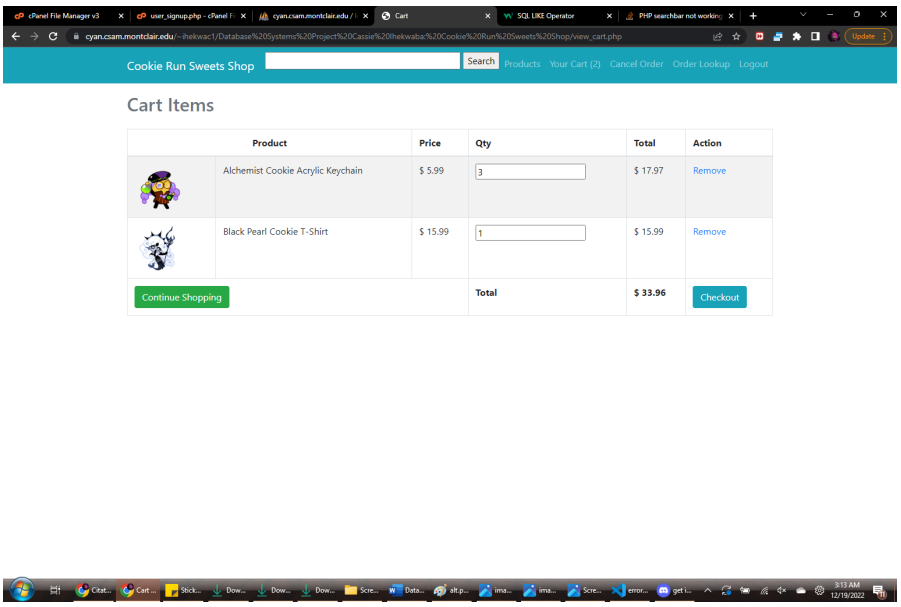

cyan.csam.montclair.edu/~ihekwac1/Database%20Systems%20Project%20Cassie%20Ihekwaba:%20Cookie%20Run%20Sweets%20Shop/index_search.php
Cookie Run Sweets Shop
tank
Search
Products
Your Cart (0)
Cancel Order
Order Lookup
Logout
Products
Chili Pepper Cookie Tank Top
Price $ 15.99
View Details
}

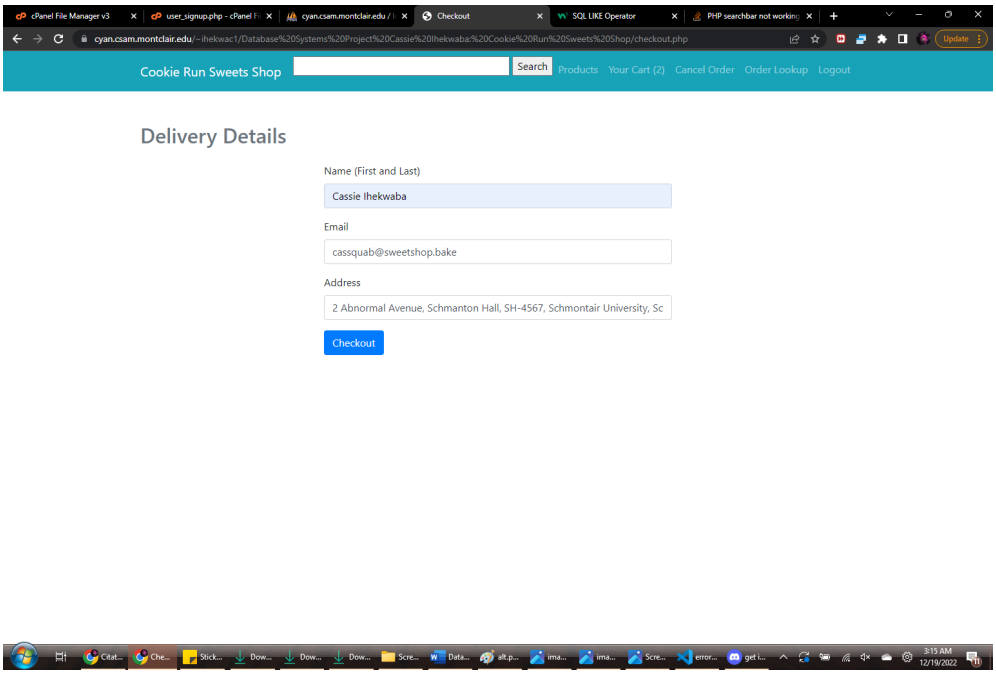

cyan.csam.montclair.edu/~ihekwac1/Database%20Systems%20Project%20Cassie%20Ihekwaba:%20Cookie%20Run%20Sweets%20Shop/view_details.php?id=7
Cookie Run Sweets Shop
Search
Products
Your Cart (0)
Cancel Order
Order Lookup
Logout
Product Details
Alchemist Cookie Acrylic Keychain
Price $ 5.99
In Stock: 985
HANDLE WITH CARE! Those chemicals she has are incredibly volatile! But a keychain should be fine...right?
3
Add To Cart



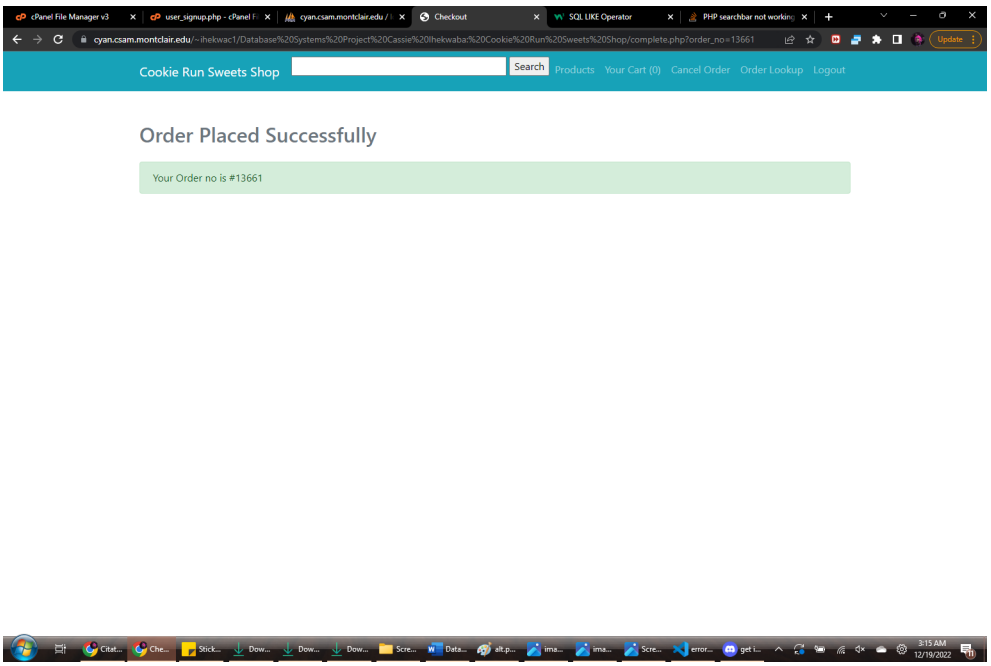
Since this was a class as I explained before, all of these items are currently being stored into an array - this allows each user to have their own cart, and avoid possible backend failure, destroying all carts. **There are no SQL queries for this cart function.**








```
$sql = "insert into ORDERS (orderid, o_name, u_email, shipAdd, total)
values ('{$order_no}','{$name}','{$email}','{$address}', '{$total_amt}')";
$sql = "insert into RECEIPT (receipt_ID, o_id, o_name, u_email, shipAdd,
total) values ('{$rec_no}','{$order_no}','{$name}','{$email}','{$address}',
'{$total_amt}')";
$invupd = "update PRODUCTS set quantity='".$upd_qty."' where
ProductID = " .$item["id"]; ($upd_qty = $stock - $quant;)
```



Order ID is 13661 - remember this for next page

Order Search

Back to Main Page | Cancel Order | Logout

Enter an Email to find your **Order no. and receipt** to delete an **individual order**:

Or to delete **multiple orders**, enter **email** associated with order(s)

on "**Cancel Order**": Page:

Manual Search

Enter Email Infomation:

cassquab@sweetshop.bake

Search

```
$query = "select * from RECEIPT where u_email = '".$searchterm."'";
```

***RECEIPT is just the name I chose for the table - it's still an**

order history. It orders from Oldest (1) to newest (3)

Receipt

Orders with Receipts for Given Criteria: 3

These are you order records/receipts - if none exists, no orders have been placed
Or have been cancelled by user or site admin.

Orders processed below:

1. Order ID: **41632**

Receipt ID: 28411

User Email: cassquab@sweetshop.bake

Name: Cassie Ihekwaba

Address: 2 Abnormal Avenue, Schmanton Hall, SH-4567, Schmontair University, Schmontair, NJ, 99086

Total: 61.98

2. Order ID: **29615**

Receipt ID: 65114

User Email: cassquab@sweetshop.bake

Name: Cassie Ihekwaba

Address: 2 Abnormal Avenue, Schmanton Hall, SH-4567, Schmontair University, Schmontair, NJ, 99086

Total: 15.99

3. Order ID: **13661**

Receipt ID: 43807

User Email: cassquab@sweetshop.bake

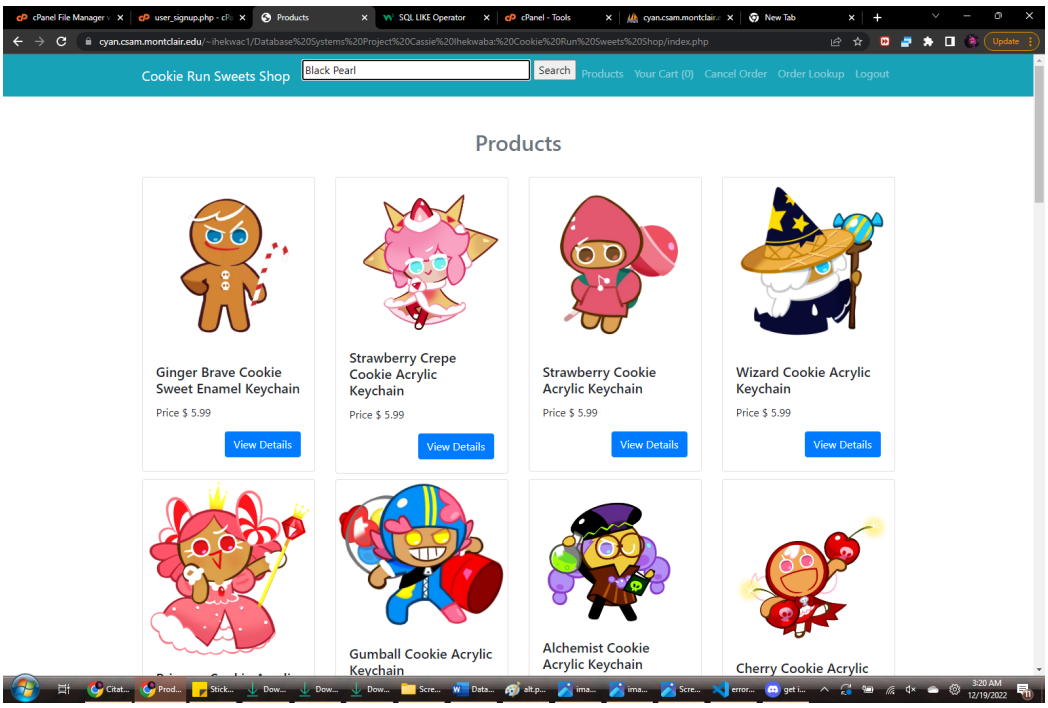
Name: Cassie Ihekwaba

Address: 2 Abnormal Avenue, Schmanton Hall, SH-4567, Schmontair University, Schmontair, NJ, 99086

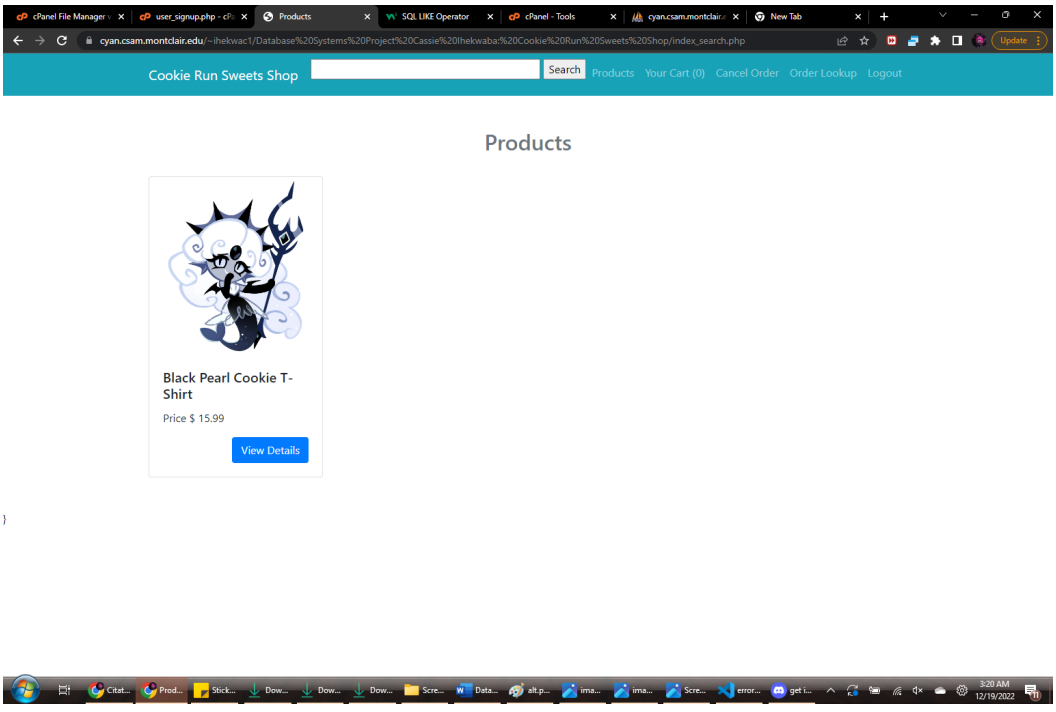
Total: 33.96

Back to Receipt Search

13661 is seen as latest order below as #3




```
$sql="select * from PRODUCTS where name like '%".$searchterm."%'";
```



All of the screenshots from above display the user side controls of the webpage. The first group of screenshots displays user signup, where an input takes the email, and password from a user into the input, and to commit it to the database, must first prepare statements to avoid SQL injection, then hash the password that will be linked to that user email. This is why they are both question marks - the information must first be inserted into the hashing algorithm to be stored away or it will be a high security risk. Upon signup completion, it redirects to the user sign in page. I saved my login for the site, so it autofilled for me on my browser.

The next set of screenshots has to deal with login. On this side, it takes the user email and password (also in prepared statements to avoid SQL injection) and finds the tuple associated with the email. It then takes everything from the tuple, and using the password stored in the tuple associated with the email, crosschecks it with a hashed version of the plaintext password we typed in on the login page. If the hashes match up, it allows for login and takes the user to the landing page. The landing page will either redirect the user to the products page, or their receipt lookup for past orders - this landing page is an html page with no SQL attached to it.

For viewing one item to purchase in their respective categories, I opted to stay with a search bar (If I did hyperlinks, this would possibly result in having to rework parts of the database and php. Due to time constraints from class responsibilities outside this project, I opted to do this, rather than have category hyperlinks). Each category has 10 items each - under certain handles of “Keychain”, “T-Shirt”, and “Plushie”. To see items in their groups, you would have to type in each handle bit while paying attention to any hyphens or separators. Upon entry, you will get all respective items under that name category. The SQL takes the searchterm and compares it to tuples with a name

(or section of it) that have similar lettering, thus resulting in this return.

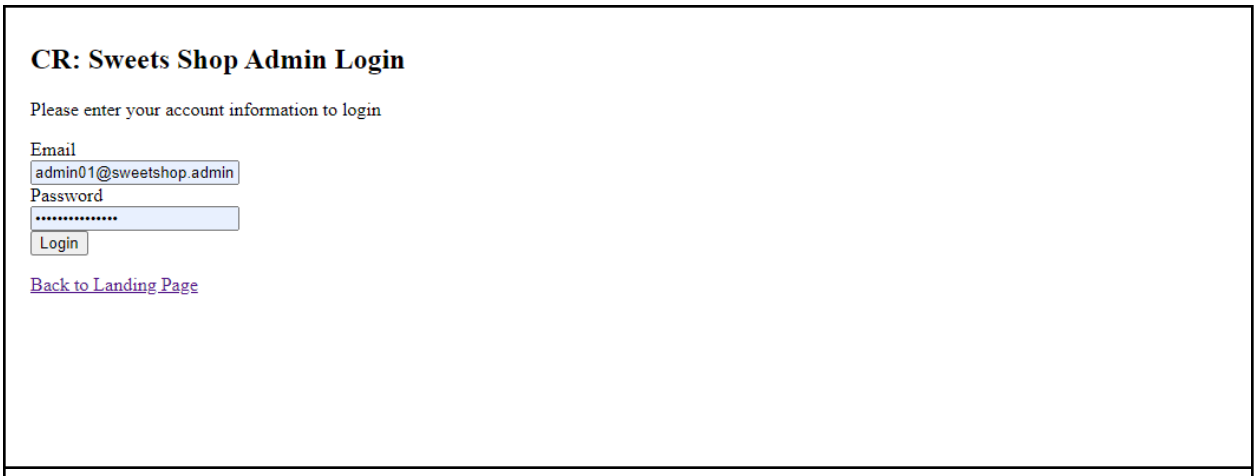
Adding products to the cart required no SQL - I had opted to use a class function for this side due to reasons explained earlier. But to give a quick summary again: Having a cart table would be live queries where items would be piling in and out of it. To keep track of user items, it would have to be linked to their email or account id for each item inside the table. This was my initial plan, but proved to be nonsensical to implement, and could possibly cause massive failure on the DB side if the table were to be dumped/dropped. This was too sensitive, so I used a cart class from a past project for this section. The cart items are stored into a cart array for that session the user is on, so each user has their own cart, can look into the cart, and can empty and checkout the cart as well. This was more versatile, and less cumbersome/time consuming compared to a table.

To purchase the cart, we are taken to a checkout page that has us entering our name, email (that requires the “sweetshop.bake” handle), and full address into the respective bars. The PHP active on this page total to three. The first one inserts the order itself into the order table, committing the order to the DB after purchase is complete. Update changes inventory levels based on how much of a product was bought by pulling the total quantity from PRODUCTS and subtracting the array value of “qty” from it, it would give us our new value for each product id the foreach loop would pass over based on the amount of items in the cart array. This new value is then reflected on whatever products were purchased. The final statement would create a receipt that also has the order details, but stores them into a receipt record that has a receipt id attached to it.

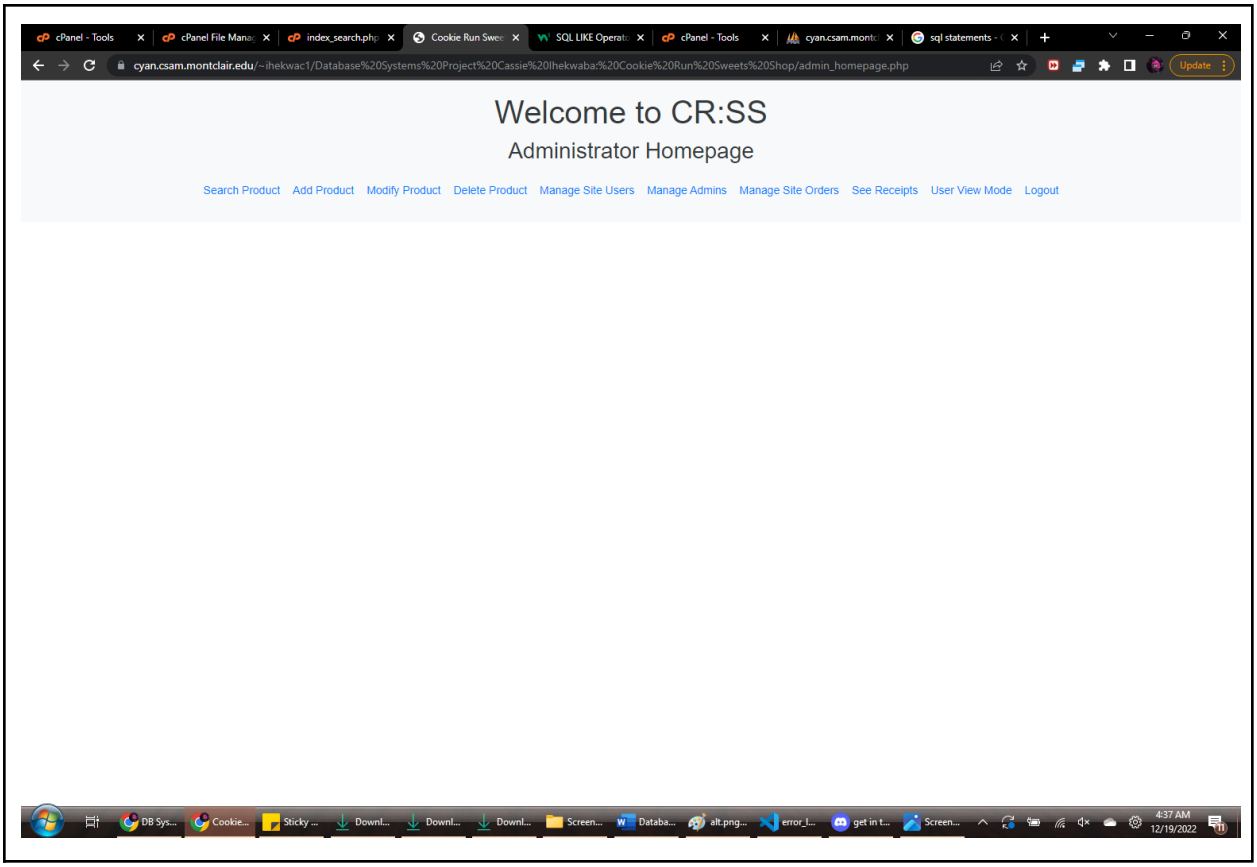
To find the orders in our history, this would be a search based off of the email associated with the account. The user would search this email, and the SQL statement, taking this search term, would apply it to find orders under that email in the

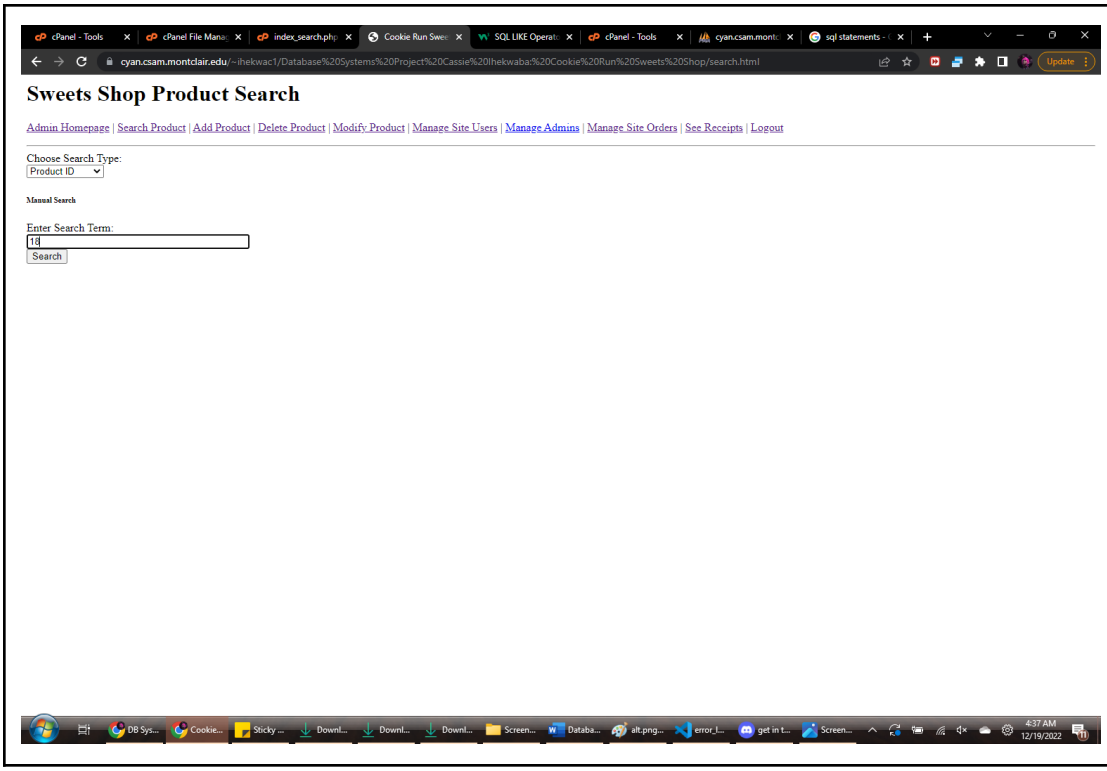
RECEIPT table. This would get every order back to the user that they made. This also helps with order deletion, where a user can pull an individual order number to delete one order, or delete all orders based on their email that was used on each order.

Searching for a product works the same way as explained before - it runs a search query based on a search term and runs it against the product names in PRODUCTS and returns anything similar to it. In this case, if we type in “Black Pearl”, we get the product associated with that item and that item only.

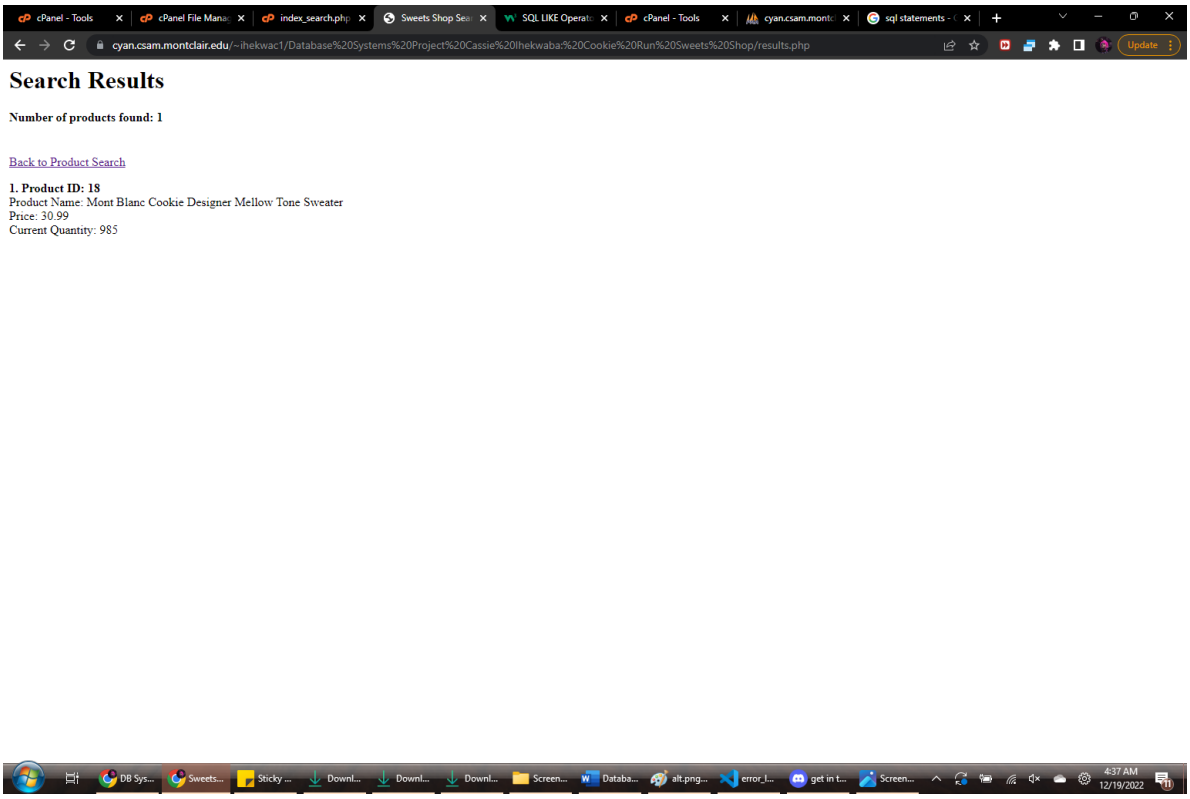


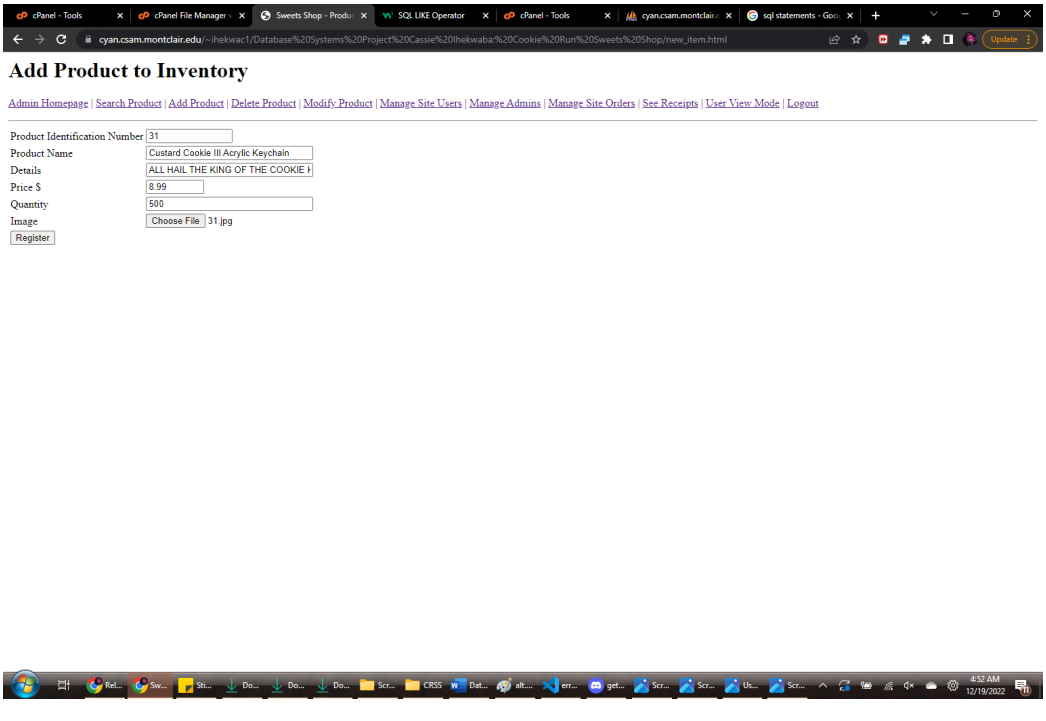

```
$sql = "SELECT * FROM ADMINISTRATORS WHERE email = ?";
```


cyan.csam.montclair.edu/~ihekwac1/Database%20Systems%20Project%20Cassie%20Ihekwaba:%20Cookie%20Run%20Sweets%20Shop/admin_homepage.php
Welcome to CR:SS
Administrator Homepage
Search Product Add Product Modify Product Delete Product Manage Site Users Manage Admins Manage Site Orders See Receipts User View Mode Logout

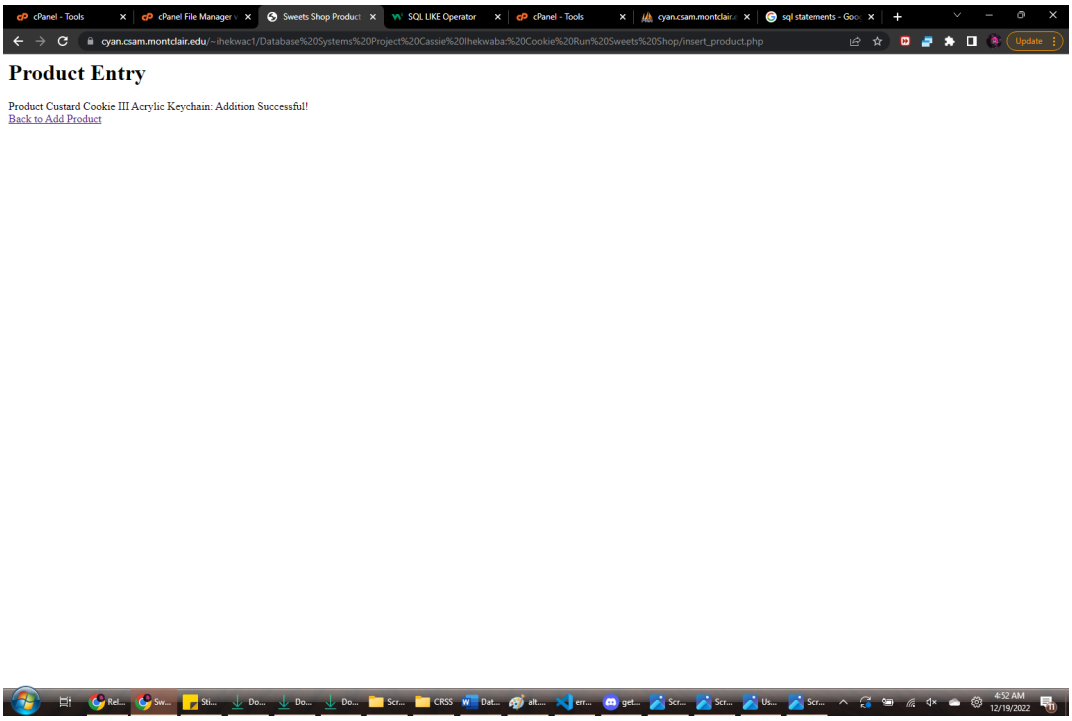

cyan.csam.montclair.edu/~ihekwac1/Database%20Systems%20Project%20Cassie%20Ihekwaba:%20Cookie%20Run%20Sweets%20Shop/search.html
Sweets Shop Product Search
Admin Homepage | Search Product | Add Product | Delete Product | Modify Product | Manage Site Users | Manage Admins | Manage Site Orders | See Receipts | Logout
Choose Search Type:
Product ID
Manual Search
Enter Search Term:
18
Search

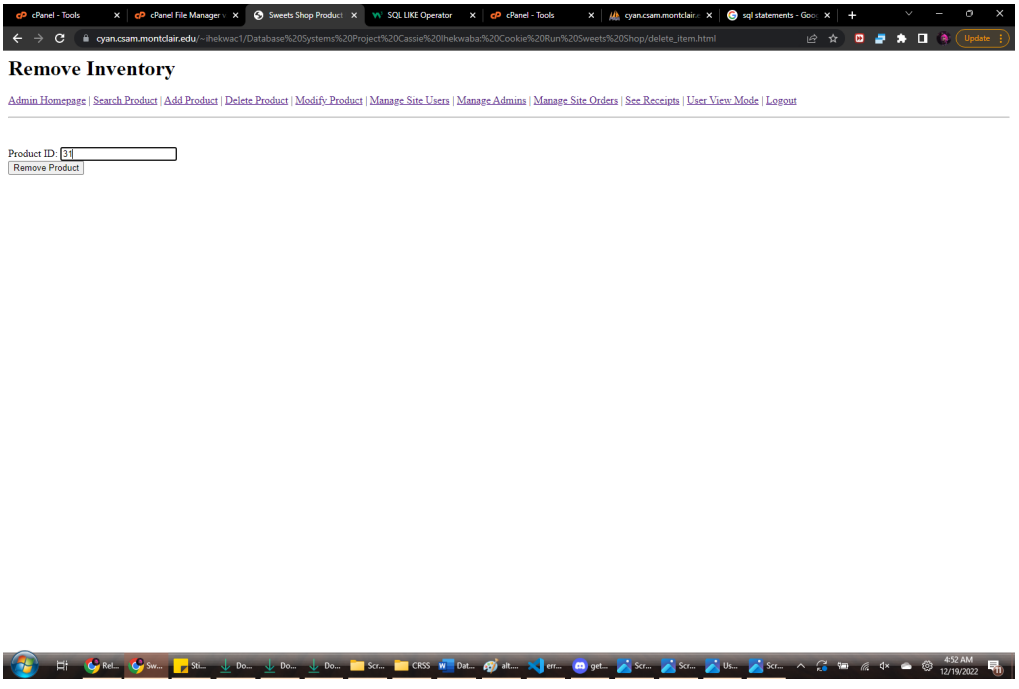
```
$query = "select * from PRODUCTS where ".$searchtype." like '%".$searchterm."%'";
```



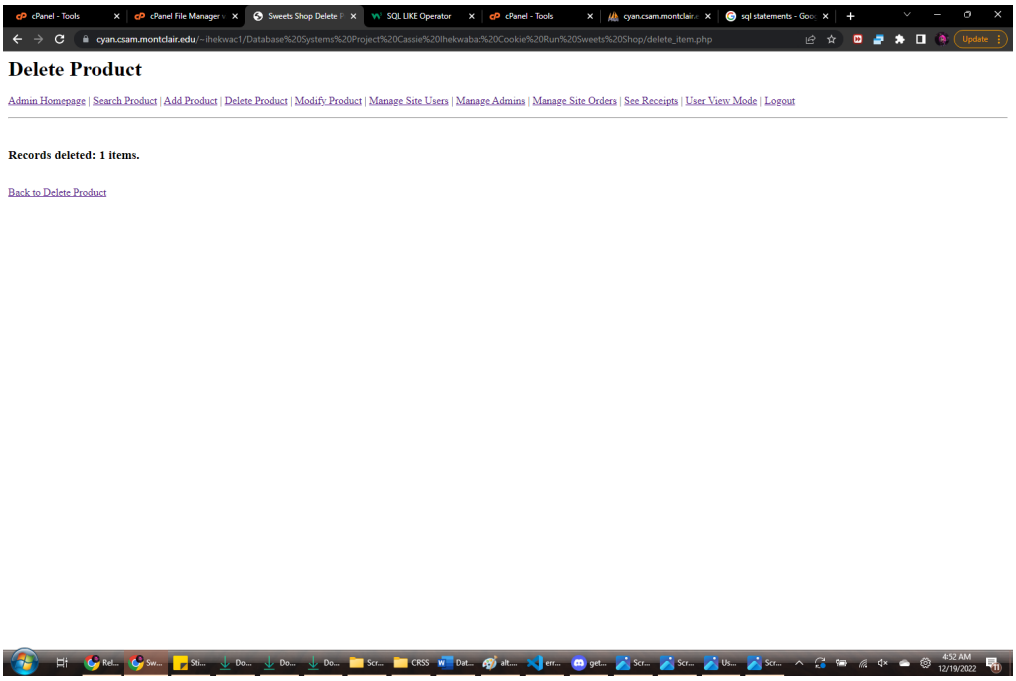


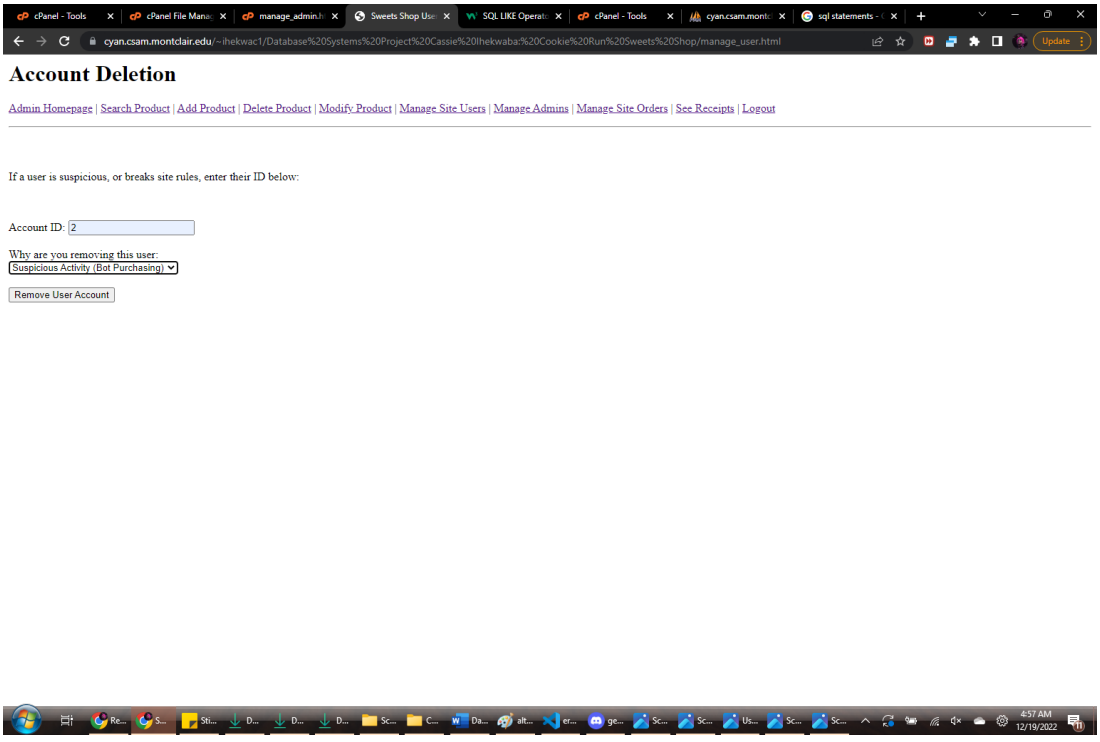
```
$query = "insert into PRODUCTS values ('".$product."', '".$name."', '".$details."',
'".$price."', '".$quant."', '".$imagename."')";
```



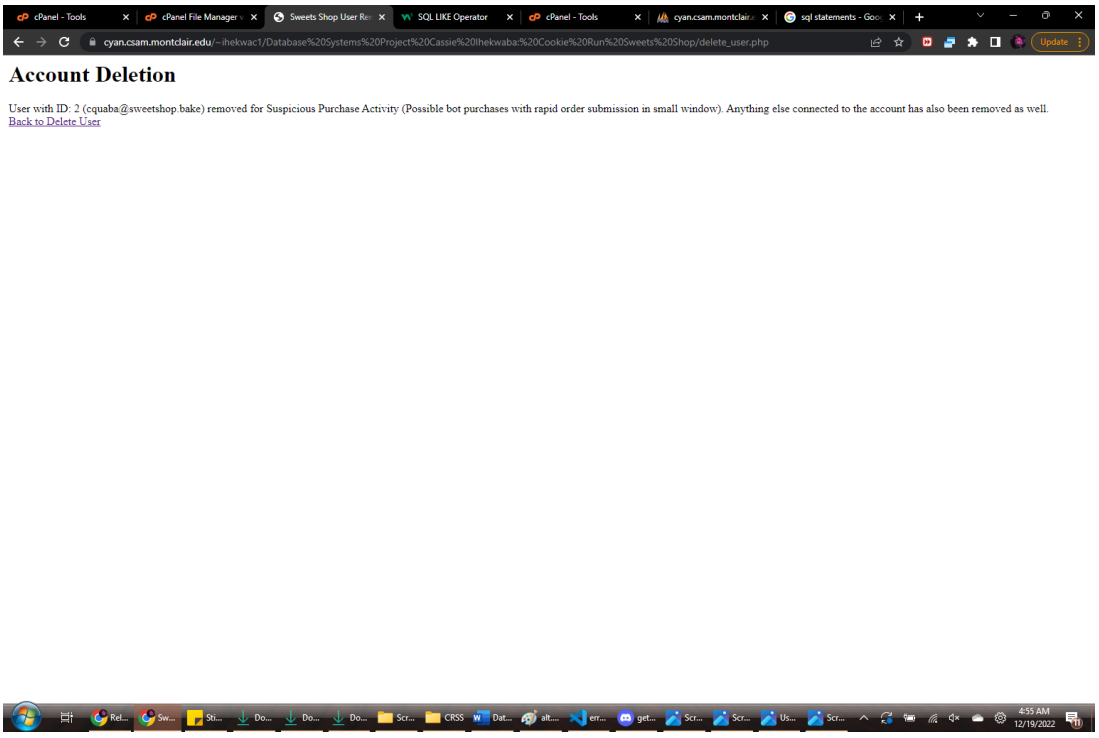


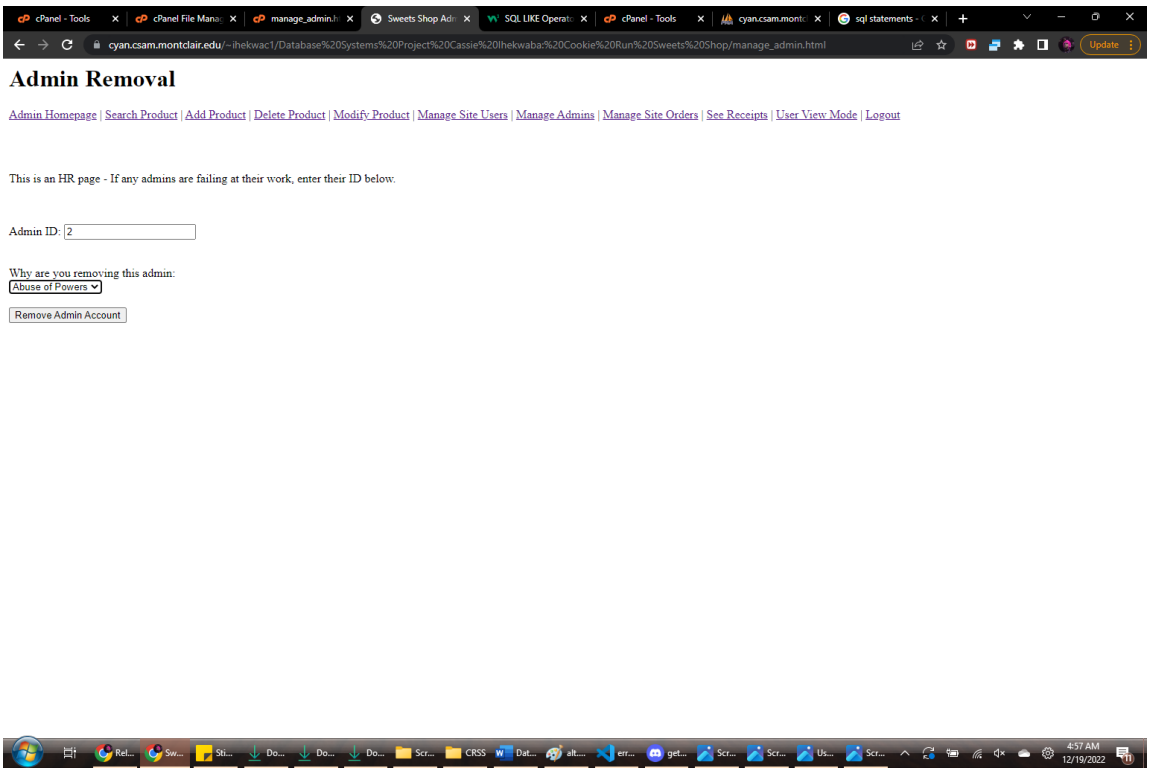
```
$count = "select * from PRODUCTS where productID = ".$productID;
$query = "delete from PRODUCTS where productID = ".$productID;
```



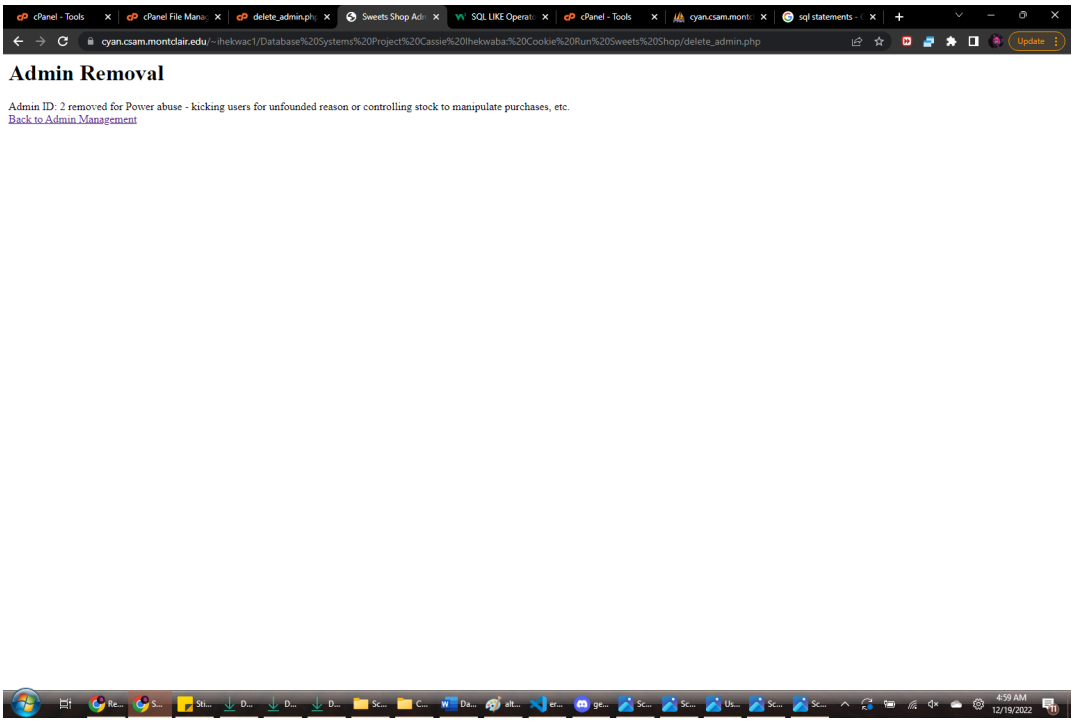


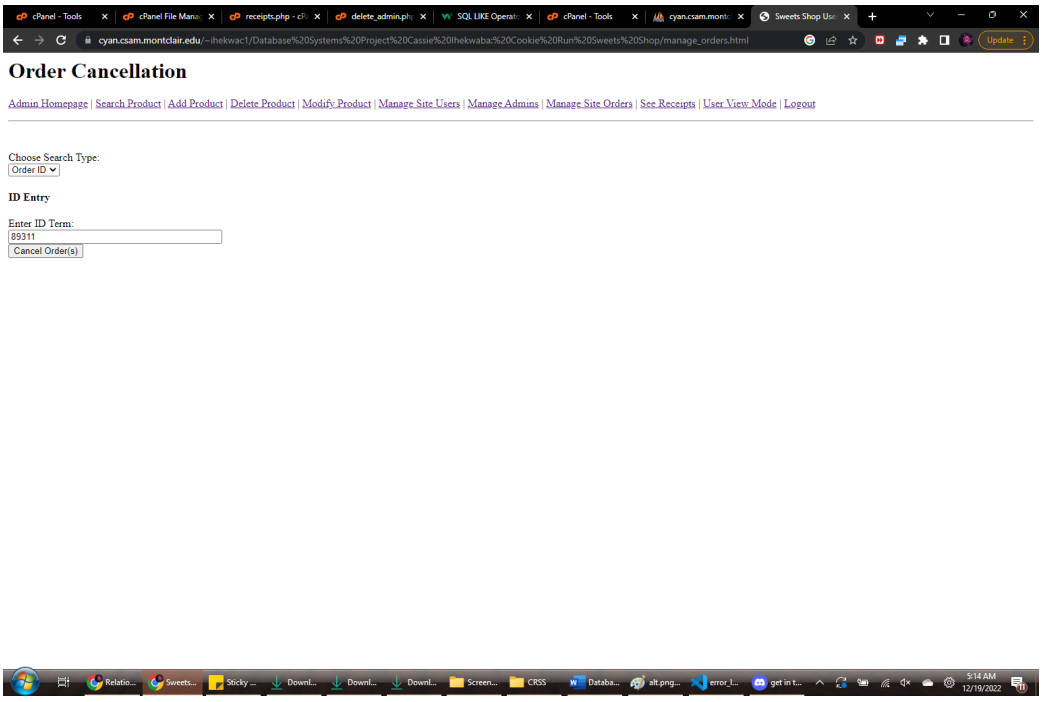
```
$count = "select * from CUSTOMER where AccountID = ".$AccountID;
$query1 = "delete from CUSTOMER where AccountID = ".$AccountID;
```



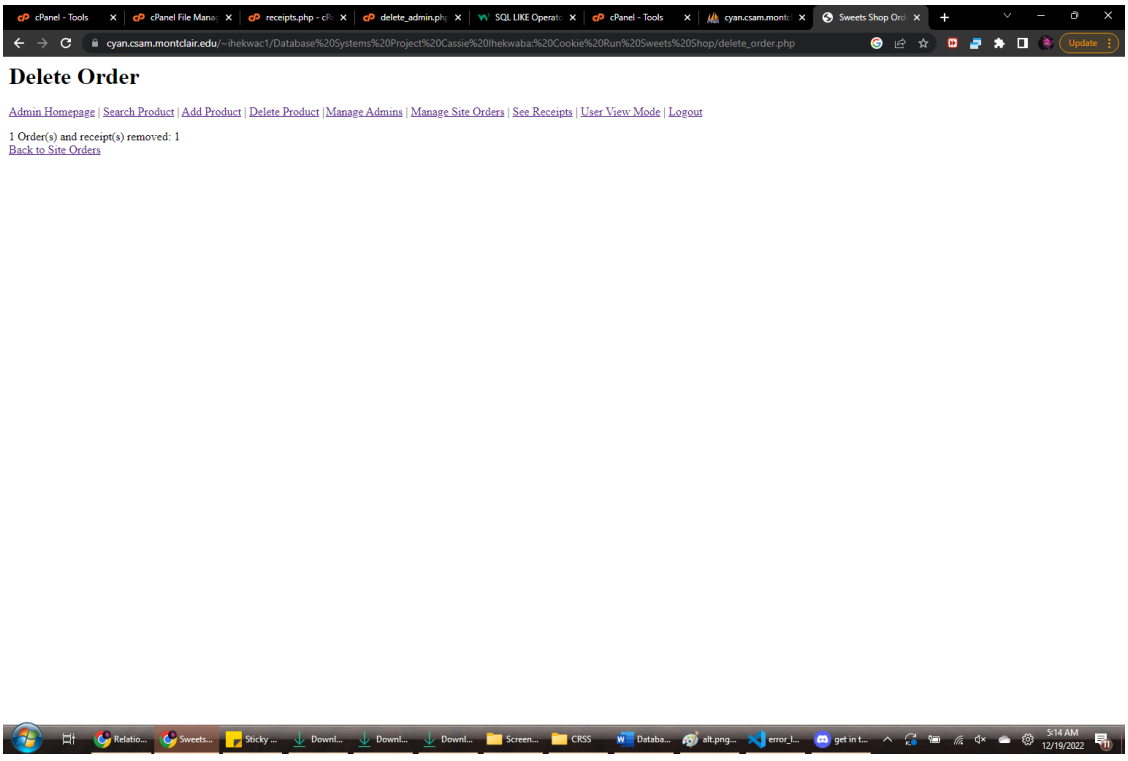


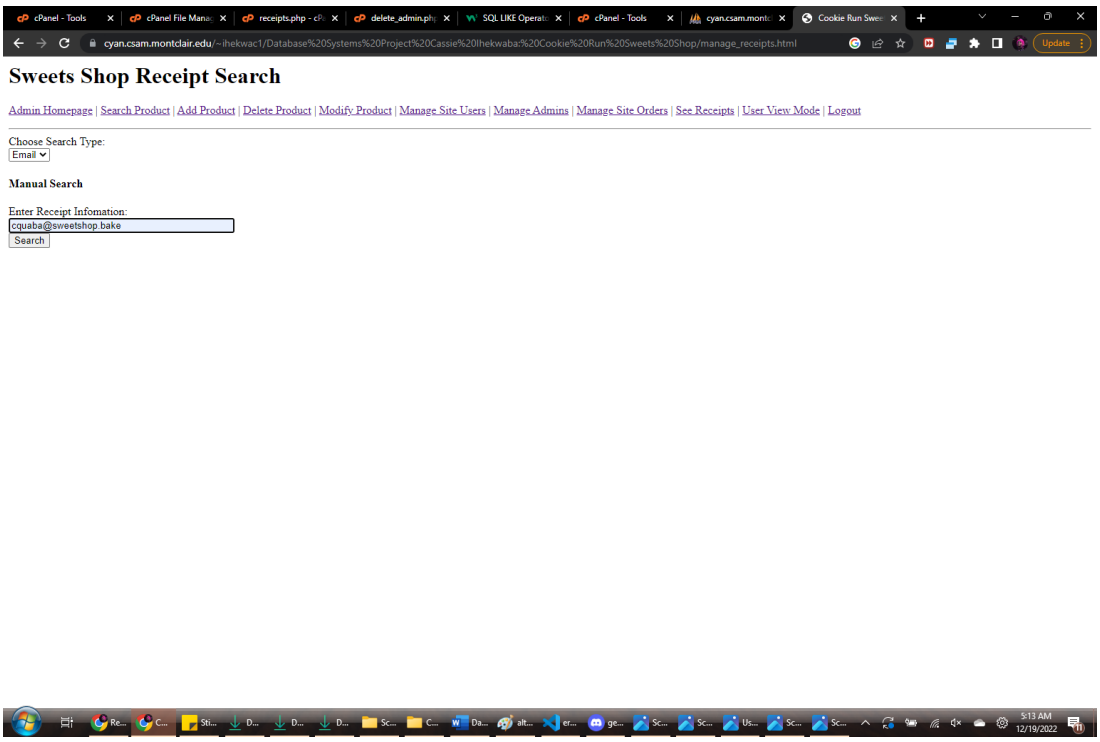

```
 $count = "select * from ADMINISTRATORS where adminID = ".$adminID;
$query = "delete from ADMINISTRATORS where adminID = ".$adminID;
```



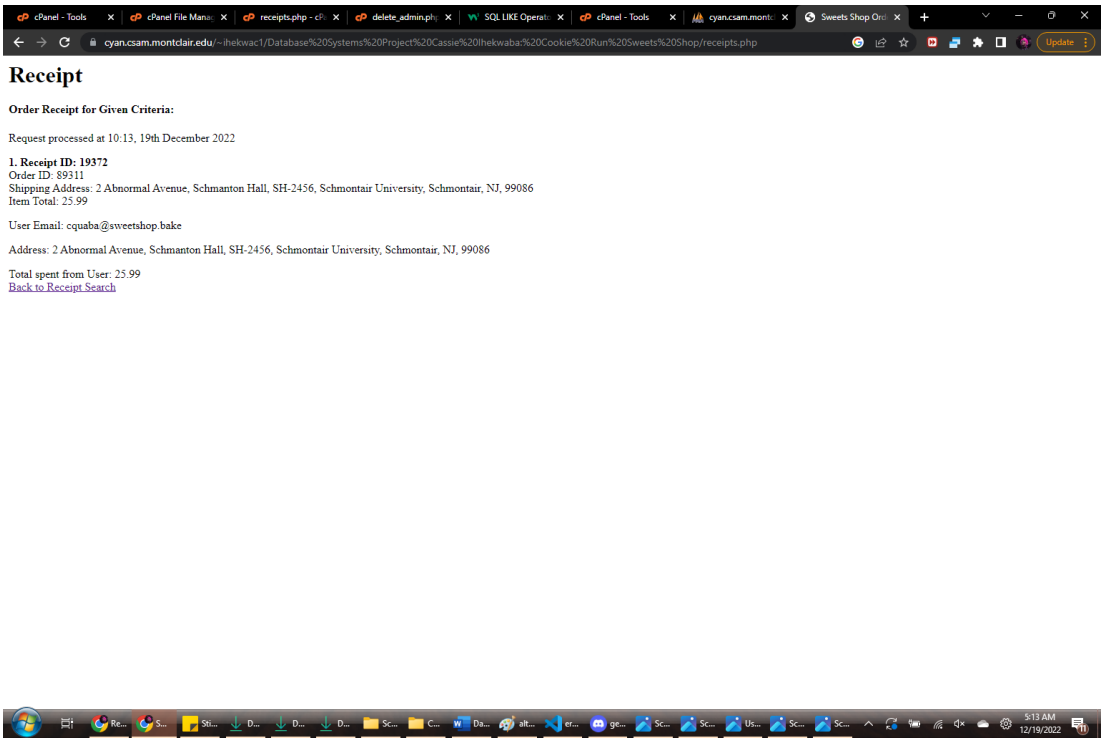


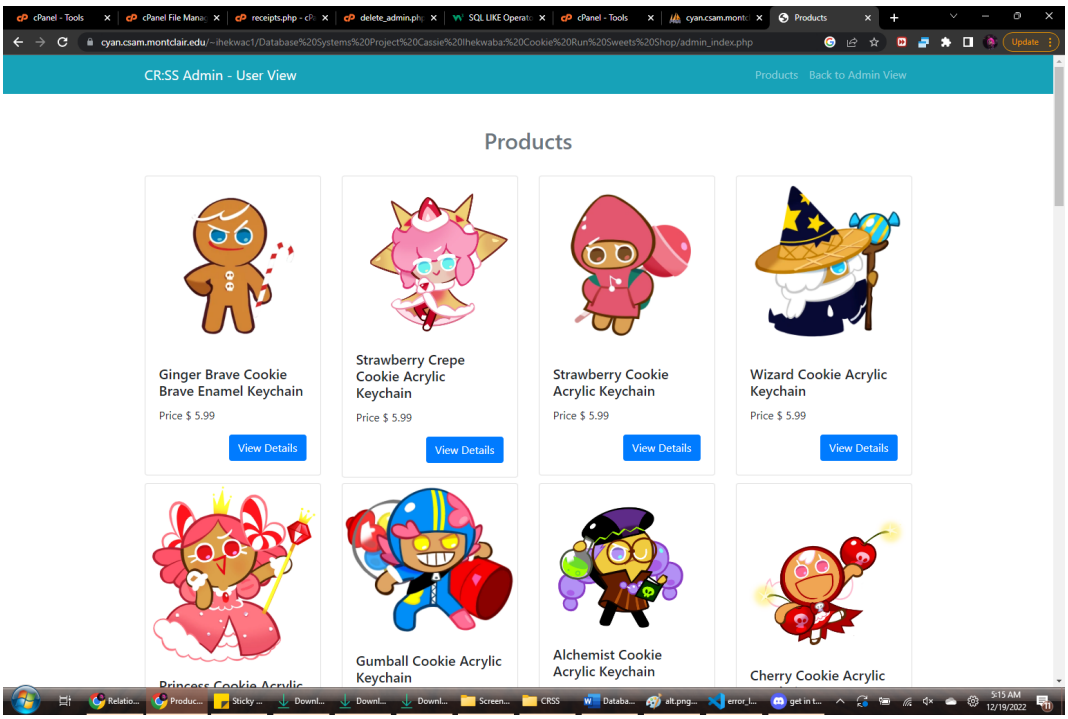

```
$count = "select * from ORDERS where ".$searchtype." = '".$searchterm."'";
$query = "delete from ORDERS where ".$searchtype." = '".$searchterm."'";
$receipt = "delete from RECEIPT where ".$searchtype." = '".$searchterm."'";
```

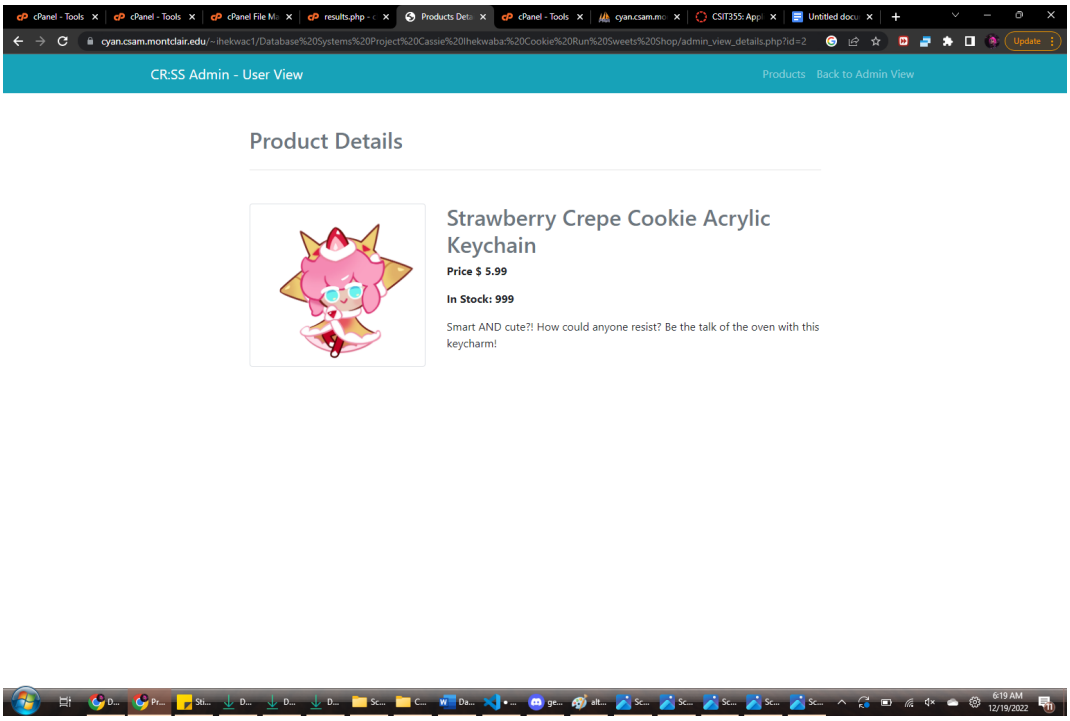





```
$query = "select * from RECEIPT where ".$searchtype." like '".$searchterm."'";
```







Simply a User View Mode page to see immediate changes to inventory as a user - can't add to cart, this just allows us to see changes

to stock and name changes that are entered into the DB from admin side. No SQL is active on this page beyond **select * from PRODUCTS;** which lists the products as you see them now.

Admin Side Screenshot Explanations:
For sign in, the function is the exact same as the user side - it compares the already existing admin's email to the entered email, and hashes the plaintext password with the stored hash password to allow access to the landing home page. Now one thing that should be obvious here is the lack of an admin sign up. This is because it would be a massive security risk to have such an option on a site where both login pages are meant to be reached normally. If I am protecting against SQL injection, allowing for employee signup could also open the site to bad actors, completely tearing the site apart. This is why I have intentionally opted to remove a signup option since this will be going on the live web. It is purely for safety.

For product searches on the database, the admin would select a search type, and add the specific term they are trying to look under - if they select "All" the SQL query of selecting everything under product would activate, and return every item. For more specific searches such as Name or ID, it would have to be selected, and the search term would then be matched with search type while going through the selected table of PRODUCTS to return what was queried.

For adding a product to the inventory, it was a simple insert page - To add a new product, the admin needed an ID, a name, description, price, quantity, and image. For image, an extra set of steps were required to implement it properly. If first needed to be taken in as file input on the form, next a target folder needed to be linked to to store the image file and name in, and then the query would be ran to insert all text information, and then the file itself would be moved over to the image folder at the end to make it visible on the product page.

For deleting a product, I needed to run two queries - one that selects the amount of rows based on the query sent, and the other that fulfills the deletion query. I needed two so I could report how many tuples were deleted from the PRODUCTS table. The select query and delete query are the same - both use the product ID that is entered in the page to find the tuple that is at the center of the query. Only one delete at a time can be done to avoid accidental mass deletion of stock.

Account deletion of users runs the same way (every deletion page requires a select and an delete to confirm the existence of the account to follow through with deletion), it takes admin input based on one row attribute - in this case it's account ID, and runs both queries one after the other in order to confirm an existence, and delete the account. On this site, a reason must be given for user removal. A blind ban of an account is not permitted, so a drop down menu of reasons is provided to the admin to select and subsequently remove the user for their offense.

Administrator removal functions the same way - an admin id is taken at the input line and a reason is required in order to fully delete an account. Admins are validated and added to the site from higher up the administrative ladder. The admin level in this case can remove admins, but their termination from the company is determined by the same people who validated their employment. An admin is simply removing their administrative powers for the time being so they cannot continue their misbehavior.

To delete from orders, it is the same idea - a select statement is ran to find the order that matches up with the entered order id or user email - however since a user email delete is too nuclear and should only be used when users break code of conduct, receipt searches are done to find individual orders to delete for the user if they are unable to delete from their side, or if an inventory issue that is not immediately visible needs to be

controlled, so any new orders must be removed. When orders are deleted, their receipt itself is also being removed. Since this receipt is also serving as a reachable order record, its deletion also signals to the user their order is deleted as well. This function is also available to users as will be explained in the SQL section later.

To look up receipts, admins find these orders under a user email. These emails will return every order that this user has placed on their site ordered from Oldest to Newest Order (1 to n). The query is the same as normal selection - it selects the tuples matching the search term based on the condition of the user email matching it. When matched, all order receipts will be sent back to the admin to look through as they need to.

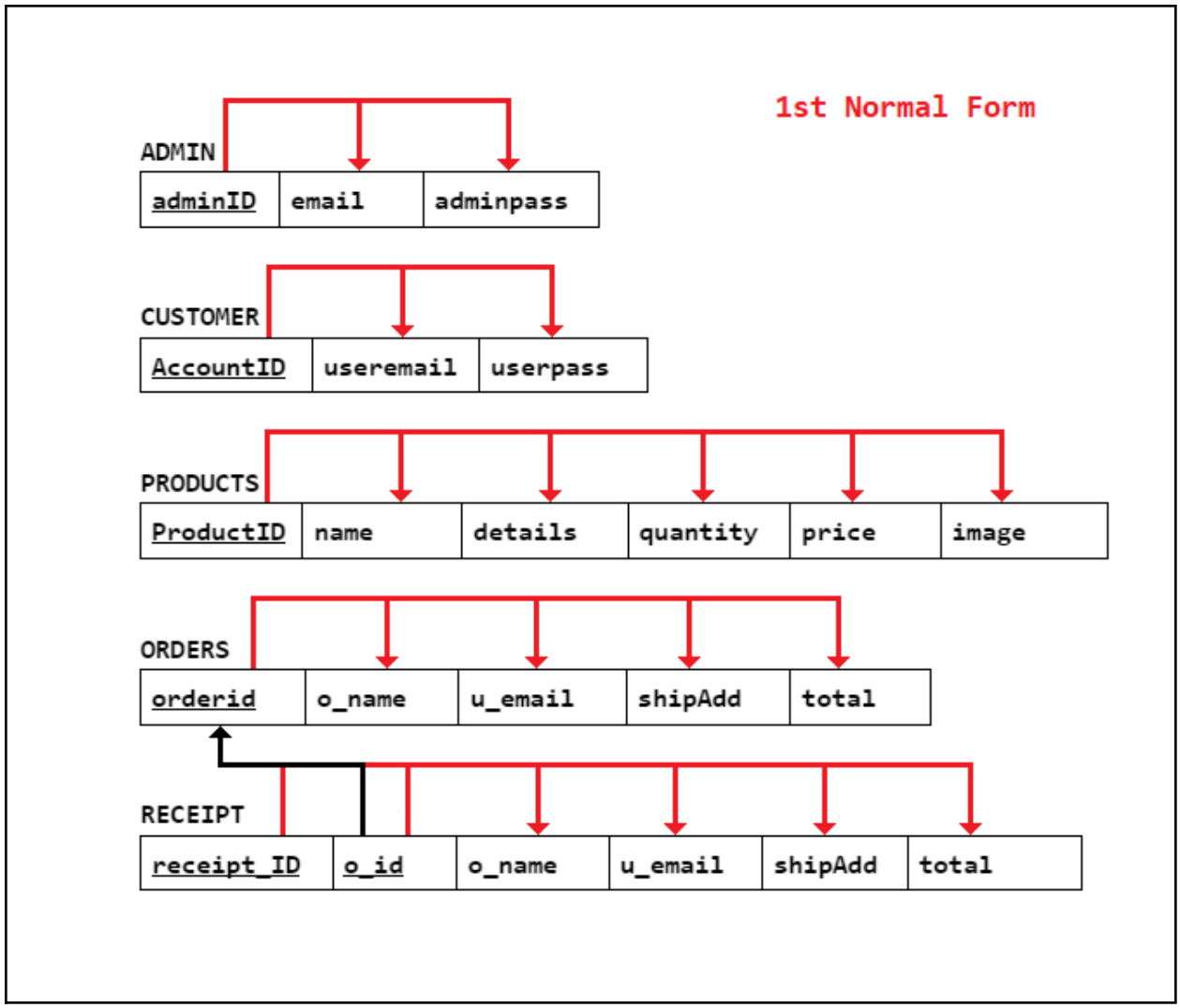
This user view page is just for seeing and tracking live changes to the user side as need be. If a new product is added or names/inventory changes, an admin can access this information directly by going into user view mode. They can’t add anything to a cart or browse normally as a user would with a search bar. This is just for visual testing purposes.

SQL:
There is not much extra SQL that is needed to be gone over except for user side order removal. The query for this is as follows:

```
$receipt = "delete from RECEIPT where ".$searchtype." = '".$searchterm."'";
```

This line, as partially explained for admin side deletions, will delete receipts pertaining to the order that a user decides to delete. The search types being email and order id. So a user does have the ability to mass cancel, or individually cancel orders they have made previously.

Logical Design Diagram with Functional Dependencies:



Database Design and Normal Form:
In my design, there is redundancy, namely in the RECEIPT table where an orderid that is a foreign element from another table is used to reference basically everything else under ORDERS inside RECEIPT - the tables can stand to be simplified a bit more than they stand now. However, for everything else, I do not see as much redundancy as I do in the lower tables as stated previously.

All of my data is reachable in my database via the primary and foreign key attributes that are present in the image above. Everything (though possibly highly redundant), is neatly referenced, and can be joined if need be.

The not null attributes in my table are mainly for everything listed in PRODUCTS - nothing can be null there, except for product details. Everything else must not be null in order for it to be identified and added to the carts and checked out later. Other attributes include user accounts and admin accounts - all of the attributes under it must not be null since it is storing user data. If one attribute is missing or null, there will be no way to login, or identify a user/admin for removal. All the not nulls are highly sensitive and dependent in my database.

If a null were to occur for a certain attribute, it would only be for product details or any table that is meant to start out as empty (RECEIPT and ORDERS) - there is nothing else within my database that relies on the description of a product, nor is database integrity threatened when there are no orders or receipts present in database. So if it goes in as empty, it will not be paid much mind to in the database.

Data enters each table based on queries called in the PHP of my database - most of these queries are selects or inserts that mainly revolve around products and orders. Data is modified by using update queries as follow up actions to certain query commands. This only really applies to edits or updates of stock. Other modifications such as inserts or deletes plug directly into the database and grab the data to do what the query requested. It is essentially a lot of inserting and selection of data for mostly display work and retrieval of records and placement of orders.

I do not believe that there is any possibility for lists to be put into an attribute cell for my design - everything is meant to be referenced 1 to 1 - if any lists are being made they are arrays in the CART class that are being specifically used for cart purposes. Even if I were to implement lists into cells, it would be too cumbersome and essentially wasted time since the information I need can be picked from tables with simple selects and joins.

PRIMARY KEY	SECONDARY KEY	Table it belongs to
adminID (AUTO)	email	**ADMINISTRATORS**
AccountID (AUTO)	useremail	**CUSTOMER**
ProductID (AUTO)	N/A	**PRODUCTS**
orderid	u_email	**ORDERS**
receipt_ID	u_email	**RECEIPT**

Test Plan

To test my project, I had made two folders in my directory - Admin Side TEST and Customer Or User Side TEST. When implementing features on each side, I would initially work on their base forms in my own environment on Visual Studio Code. I would make base changes there, and then upload them to Cyan to test them live from the Index. I would have two tabs open at a time, the actual index and the html/php file open in Cyan's editor accessible from the File Manager. Any edits I would make I would immediately refresh the index file page to see how elements on the page were corrected/changed. It was easier to test this way, rather than make blind changes and hope things would "work".

For SQL testing, I would test this on any form entry page since it was the easiest to see changes on. For user sign-up, I would make a dummy account and see what would be entered on the database in an alternate tab to test successful sign-up and hashing of sensitive information. For product entry, update and deletion, it was the same idea. I would always have alternate tabs up and see how the database would be affected. Now, during this testing phase, an "errorlog" file would be generated and constantly updated if an error was caught - this would only happen if the page would not load after data entry or crash. Any instances where pages would not crash I would go through the code

to fix any issues found during testing that were not originally caught in the log.

Other forms of testing/debugging mainly included using echo statements to print variables at certain points in the code to see if information was being properly handled or not. Security tests mainly consisted of testing usernames and passwords for the proper case returns if information was incorrect, as well.

How it was developed

1. Since I worked by myself, no work was broken up - it was tested/worked on when I had arrived at it/decided to handle it at that moment. It was based on order of importance. However if I was stuck on something, I would work on it until at a comfortable stopping point to decide to switch to another element of the project, instead. I used checklists and brainstormed how I would handle certain sections, what sections to keep, what sections to remove, and so on. Anything that was too tedious or over complicated to handle I would opt to not implement it at that moment or at all for sake of time. Not practical/logical given the specs, however I was most focused on getting a base level version of the site's views functioning at their minimal states.

2. I developed mostly in Microsoft Visual Studio Code. This is where I would handle any code I had used from past projects, the source code given to us, or anything that I had found online. It was easiest and safest to start here so I could still work offline. When I would go to work, I'd take time out of my break to work on what I was able to in VSC. Migration was a matter of uploading the files to the file directory - so there was no issue for migrating. This was only really done when the code I finished editing and felt confident a base form had been established, I would upload it, and continue to edit it in Cyan since they would become the finalized files later in the project. I would only

download the folder I was working on in Cyan for security purposes and to work offline if necessary.

I also used template code from past projects such as a project not too dissimilar to this one from Internet Computing, and Template code from online for certain elements from websites listed in the earlier section of this documentation (i.e. GeeksforGeeks, TutorialsPoint, TutorialRepublic, w3Schools, and so on). It was a matter of modifying the code as I needed to, and migrating/implementing it properly.

For queries directly coded into anything, I would just test them through the page (such as basic SELECT, UPDATE, DELETE, etc.). This was a matter of going to the Admin side and testing entries, updates, selects and deletions of anything existing in the database. For anything that required deleting anything, I would inject dummy data and then perform these actions. Anything that requires returning table results such as using JOINS, I would test that in the SQL tab on Cyan.

To test the cart, I would add random products to it to see if the contents would be carried over to the user's session cart. For emptying the cart, I would delete products from the cart and see if the action would be carried through and not ignored. Browsing to the cart, I would test the link to see if it would let me in even with nothing in the cart. Post checkout, I would test to see if the cart was emptied or not once checkout was complete and the user was given their order number.

The reason why I was being so careful with this was because this cart was actually from a previous project that I had mentioned prior (The Internet Programming Project). I was unsure if the cart would work the same way it had when it was running in XAMPP versus on Cyan. There were some

problems with some of the php in regards to checkout procedure, but the cart remained consistently stable throughout.

To test order histories, I would place several orders with any content in the cart and submit it. To know order histories worked - and wasn't just submitting to the database, I had implemented an "Order Lookup" tab for the user. They could come to this page and find their order based on Order ID or Email - one would return in batches or return a single item based on entry criteria. On the Admin side, this data is also accessed based on user emails, or order IDs directly drawn from the database. This helped me know that data was not just being submitted to the database, and that everything was actually being entered into their appropriate columns since there were certain lines that had to be returned when doing an order lookup on either side. If data was not being returned, I would know that it was likely something happening in the code, and not in the database. If it was a database issue, I would have to check the order side since receipts are directly converted from this data - no nulls can exist in the order table. This would thankfully barely happen, and most of these errors for receipt were more localized to code syntax and how I wrote queries.

Security

Security mainly rotated around making sure user data was secure, and that nothing could be seen by random internet users when browsing and finding my page. Deeper level security included password hashing of new and existing users, and SQL injection protections that are elaborated on further in the sections below.

1. Login: For both user and admin, I used a hashing system available in PHP called password_hash(), that would use a BCRYPT algorithm to hash any plaintext into 60 encrypted characters. No matter the length of the password it would become 60 characters long. Both sides had to use an email

address and existing password from their respective tables to log into their views. To validate the password, it would be taken in as plaintext, then hashed with the same BCRYPT algorithm to be compared against what is in the database.

If correct, the user would be taken to their page immediately. If incorrect, it would stay on the login page until the proper information is entered and checked to confirm the user's existence. Anyone can enter an email, so I wasn't too concerned about hashing the email since the password crosscheck was more important to me to complete.

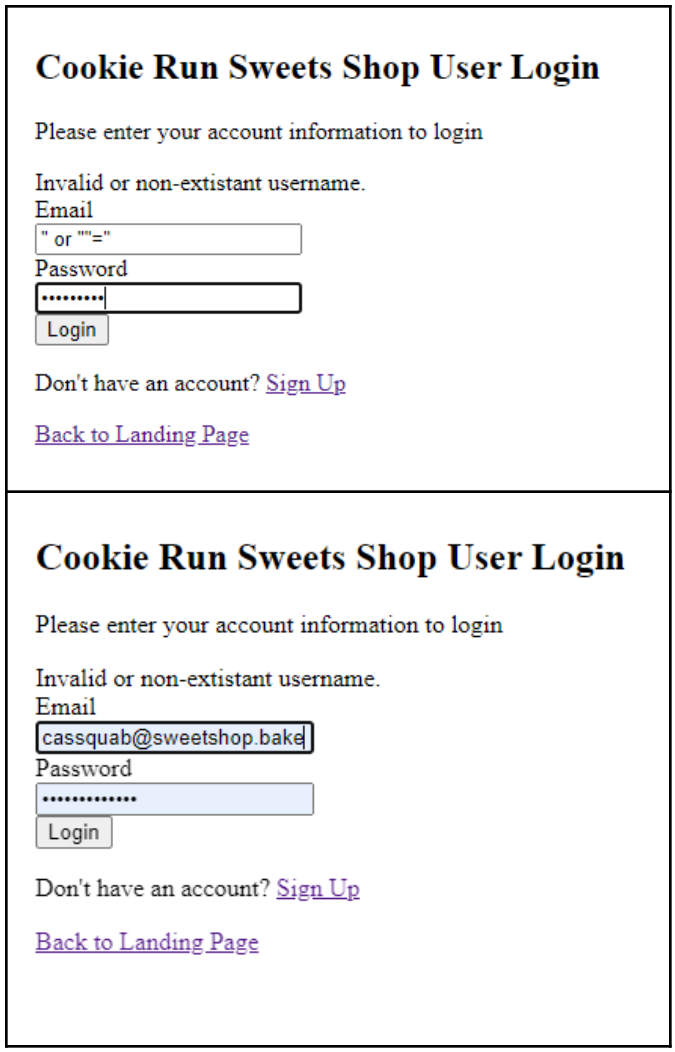
Sensitive information: On the system, any sensitive data would be encrypted as stated beforehand with password_hash() or hash(). It would be hashed, then stored into their respective tables. To compare this data for future retrieval, any entered information related to this would have to be hashed, then cross-checked with the hashed information in the database. If it returns true, the user/admin would be given access to the information they are attempting to gain access to. Anything highly sensitive must be hashed for the protection of users and admins alike.

Tunneling: To protect against malicious activity, a landing page was implemented to keep the main views of the page away from a random internet user. To be able to get into the site, they'd have to either sign up, or have an existing account. This is in very simple terms, my "firewall" for the site. They would have to hand over their information to use the site - revealing who they are and if any suspicious activity arises from that account, they can be stopped if need be.

SQL Injection: for SQL injection prevention, on login pages, namely, the queries for insertion and selection are "prepared" and bound to special parameters to avoid any illegal activity that may be happening on an information

entry form. These attacks are likely to happen on the login pages for these users or admins, so prepare() statements being there was important.

The templates that I used for admin login and user login were from the same site, just slightly tweaked to access different tables and redirect to different pages when login was successful. I tested this by looking up basic SQL injection terms - the classic *" or ""="* was used in this instance on user side:



As you can see, even after testing this in both boxes, it either returns “invalid password” or “Invalid or

non-existent username”. Since I used the same template on the admin side, this is also true for it as well.

Results

This project is made for Cookie Run Fans to peruse small products to their liking. On a deeper level, it is a highly simplistic ecommerce querying site. Users are free to sign up, while admins are verified by higher powers and already exist within the system in order to prevent false admin account creation. User side is fairly simple, with admin being just as simple but with more functions. Products are added to a cart array, that is then moved to a checkout page where upon entry of credentials, it is committed to the database as an order under a user’s email with ids attached to aid a user and admin with order removal. HR functions include admin and user deletion depending on an entered reason that will remove them from the site - for admins, they are simply stripped of power and await further discipline, and user’s orders will stay on the site for record book purposes to ensure they can’t circumvent bans.

This project was definitely a challenge - taking it on solo was more cumbersome than I initially thought. The challenges mainly revolved around PHP implementation. Even though I worked on a project with a team similar to this project - this assignment was a whole other beast entirely. Database implementation was fine, PHP implementation was hard, but mapping out my idea on diagrams was the most pressing. I am still not confident in my later ER, and my logical diagram was modeled the way it was because it was the best way to visualize my logic as clearly as I possibly could. Even then, I am sure there are several parts of this project that need tweaking and cleaning. Other class responsibilities and work crunched me on time so I had to streamline as much as I possibly could manage with the time I had left and the sleep I have absolutely lost.

If I had more time - I would try to implement a live updating cart table that takes all orders from the cart class, and inserts

them into the table as a record of what exact products a user bought. As well as seeing if the inventory update is really working. Because for some unfounded reason, it looks like it works at one point, but then updates certain products with the incorrect number, then properly updates them again. It’s highly spastic and concerning. One more would be to add an admin sign up where an admin must be verified by a higher admin to permit function on the site - however this would take too much time to implement, and be too much of a security risk to just allow open enrollment. In short, there are a few things missing from the spec I should have, but don’t due to time. But with this being a solo project on my end, I would say in general I am satisfied with what I was able to at least complete.

Citations:

Group Project - Bubble Bomb Beauty from Internet Computing back in Spring 2021

Define multipart form data. GeeksforGeeks. (2022, February 1). Retrieved December 19, 2022, from https://www.geeksforgeeks.org/define-multipart-form-data/

How to - file upload button. How To Create a File Upload Button. (n.d.). Retrieved December 19, 2022, from https://www.w3schools.com/howto/howto_html_file_upload_button.asp

How to validate an email address in php ? Tutorials Point. (n.d.). Retrieved December 19, 2022, from https://www.tutorialspoint.com/how-to-validate-an-email-address-in-php

List of cookies. Cookie Run: Kingdom Wiki. (n.d.). Retrieved December 19, 2022, from https://cookierunkingdom.fandom.com/wiki/List_of_Cookies

Password_hash. php. (n.d.). Retrieved December 19, 2022, from https://www.php.net/manual/en/function.password-hash.php

PHP - hash() function. Tutorials Point. (n.d.). Retrieved December 19, 2022, from https://www.tutorialspoint.com/php/php_function_hash.htm

PHP MySQL Login System. TutorialRepublic. (n.d.). Retrieved December 19, 2022, from https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php

W3Schools online HTML editor. W3Schools Online Web Tutorials. (n.d.). Retrieved December 19, 2022, from https://www.w3schools.com/howto/tryit.asp?filename=tryhow_html_file_upload_button

CODE DUMP:

Admin Login:

💻 SNIPPET ##

```
<?php
// Initialize the session
session_start();

// Check if the user is already logged in, if yes then redirect him to
welcome page
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){

    header("location: admin_homepage.php");

    exit;
}

// Include config file
$db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details

// Define variables and initialize with empty values

$email = $adminpass = "";

$username_err = $password_err = $login_err = "";

// Processing form data when form is submitted

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

    // Check if username is empty

    if(empty(trim($_POST["email"]))){

        $username_err = "Please enter username.";

    } else{

        $email = trim($_POST["email"]);
    }

    // Check if password is empty

    if(empty(trim($_POST["adminpass"]))){

        $password_err = "Please enter your password.";
```

```
    } else{
        $adminpass = $_POST["password"];
    }
    // Validate credentials
    if(empty($username_err) && empty($password_err)){
        // Prepare a select statement
        $sql = "SELECT * FROM ADMINISTRATORS WHERE email = ?";
        if($stmt = $db->prepare($sql)){
            // Bind variables to the prepared statement as parameters
            $stmt->bind_param("s", $param_username);
            // Set parameters
            $param_username = $email;
            // Attempt to execute the prepared statement
            if($stmt->execute()){
                // Store result
                $stmt->store_result();

           // Check if username exists, if yes then verify password
                if($stmt->num_rows == 1){
                    // Bind result variables
                    $hashed_password = password_hash($_POST['adminpass'],
PASSWORD_BYCRYPT);
                    $adminpass = $_POST['adminpass'];
                    $stmt->bind_result($adminID, $email, $hashed_password);
                    if($stmt->fetch()){
                        if(password_verify($adminpass, $hashed_password)){
```

```
                            // Password is correct, so start a new session
                            session_start();
                            // Store data in session variables
                            $_SESSION["loggedin"] = true;
                            $_SESSION["adminID"] = $adminID;
                            $_SESSION["email"] = $email;
                            // Redirect user to welcome page
                            header("location: admin_homepage.php");
                        } else{
                            // Password is not valid, display a generic
error message
                            $login_err = "Invalid password.";

                        }
                    }
                } else{
                    // Username doesn't exist, display a generic error
message
                    $login_err = "Invalid or non-extistant username.";
                }
            } else{
                echo "Oops! Something went wrong. Please try again later.";
            }

            // Close statement
            $stmt->close();
        }
    }

    // Close connection
    $db->close();
}
?>

<!DOCTYPE html>
```

```
<html>
<head>
    <title>Admin Login</title>
</head>
<body>
    <h2>CR: Sweets Shop Admin Login</h2>
        <p>Please enter your account information to login</p>

        <?php
        // if there is a login error, display error message
        if(!empty($login_err)){
            echo '<div class="alert alert-danger">' . $login_err .
'</div>';

        }
        ?>

        <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);
?>" method="post">

            <div class="form-group row">
                <label class = "col-form-label col-sm-2">Email</label>
               <div class = "col-sm-10"> <input type="text" name="email"
class="form-control <?php echo (!empty($email_error)) ? 'is-invalid' : '';
?>" value="<?php echo $email; ?>">
                <span class="invalid-feedback"><?php echo $email_error;
?></span></div>


            </div>
            <div class="form-group row">
                <label class = "col-form-label col-sm-2 ">Password</label>
                <div class = "col-sm-10"><input type='password'
name='adminpass' class="form-control <?php echo (!empty($password_error)) ?
'is-invalid' : ''; ?>">
                <span class="invalid-feedback"><?php echo $password_error;
?></span></div>


            </div>
            <div class="form-group">
                <div class= "offset-sm-2 col-sm-10">
                    <input type="submit" class="btn btn-primary"
value="Login">
                </div>

                    <p><a href="landing_homepage.html">Back to Landing
Page</a></p>
            </div>
```

```
        </form>
    </div>
</body>
</html>

🔤 TITLE ########################################

admin_login.php - cPanel File Manager v3

📝 DESCRIPTION ##################################

This function initialize a single user - sequence with all of the necessary
data.

🌐 LANGUAGE #####################################

Php

🏷️ TAGS #########################################

COOKIES, FACEBOOK, SESSION, ASP.NET, LARAVEL

🔗 RELATED LINKS ################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=admin_login.php&fileop=&dir=%2Fhome%2Fihekwac1%2
Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweet
s+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=
```

Admin Homepage

```
💻 SNIPPET ######################################

<?php
// Initialize the session
session_start();

if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){
    header("location: admin_login.php");
    exit;
}
?>

<!DOCTYPE html>
<html>
    <head>
        <title>Cookie Run Sweets Shop</title>
```

```
            <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.
css">
    <style>
        body{ font: 14px sans-serif; text-align: center; }
    </style>
    </head>
    <body>
        <div class="p-3 mb-2 bg-light text-dark"><h1>Welcome to CR:SS</h1>
            <h3>Administrator Homepage</h3>
            <nav class="navbar navbar-expand-sm justify-content-center">
            <ul class="navbar-nav">
                <li class="nav-item">
                  <a class="nav-link" href="search.html">Search Product</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="new_item.html">Add Product</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="update_item.html">Modify
Product</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="delete_item.html">Delete
Product</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="manage_user.html">Manage Site
Users</a>
                </li>
                 <li class="nav-item">
                  <a class="nav-link" href="manage_admin.html">Manage
Admins</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="manage_orders.html">Manage Site
Orders</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="manage_receipts.html">See
Receipts</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="admin_index.php">User View
Mode</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="logout.php">Logout</a>
                </li>
```

```
            </ul>
          </nav>
        </div>
        </div>
    </body>
</html>

🔤 TITLE ##########################################

admin_homepage.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

<?php // Initialize the session and initialize the n - sequence sequence
sequence sequence sequence sequence sequence

🌐 LANGUAGE ######################################

Php

🏷️ TAGS ##########################################

WORDPRESS

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=admin_homepage.php&fileop=&dir=%2Fhome%2Fihekwac
1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sw
eets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=
https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
```

Add Product

```
💻 SNIPPET ########################################

<html>
<head>
  <title>Sweets Shop Product Entry</title>
</head>
<body>
<h1>Product Entry</h1>
<?php
  // create short variable names - ProductID, name, details, price,
quantity
  $product=$_POST['ProductID'];
  $name=$_POST['name'];
```

```
  $details=$_POST['details'];
  $price=$_POST['price'];
  $quant=$_POST['quantity'];

  $imagename=$_FILES["image"]["name"]; /* file upload setup in PHP -
updates
  new product with an image - first need file name*/

  $imagetmp=$_FILES["image"]["tmp_name"]; //temp name for file used for
upload

  $folder = "images/" . $imagename; //target folder in cyan directory


  if (!$product || !$name || !$details || !$price || !$quant ||
!$imagename) {
     echo "Incomplete Entry.<br />"
          ."Please enter all product information.";
     exit;
  }

  $product = addslashes($product);
  $name = addslashes($name);
  $details = addslashes($details);
  $price = doubleval($price);
  $quant = addslashes($quant);
  $image = addslashes($image);

  $servername = "localhost";
  $username = "ihekwac1_cr_ss123"; //make username for shop db - after work
  $password = "exM3KYUftdJ="; //make passwd for db

  $db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details

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

  $query = "insert into PRODUCTS values
            ('".$product."', '".$name."', '".$details."', '".$price."',
'".$quant."', '".$imagename."')"; //possibly add an image

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

  move_uploaded_file($imagetmp, $folder); //moves uploaded file into
directory, showing it in the database and website view
```

```
  if ($result) {
      echo "Product $name: Addition Successful!";
      echo "<br><a href = 'new_item.html'>Back to Add Product</a>";
  } else {
        echo "An error has occurred.  $name was not added.";
      echo "<br><a href = 'new_item.html'>Back to Add Product</a>";
  }

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

🔤 TITLE ########################################

insert_product.php - cPanel File Manager v3

📝 DESCRIPTION ###################################

This function creates a short variable entry in the products table.

🌐 LANGUAGE ######################################

Php

🏷️ TAGS ##########################################

FORMS, VALIDATION, AJAX

🔗 RELATED LINKS #################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=insert_product.php&fileop=&dir=%2Fhome%2Fihekwac
1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sw
eets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=

⛔ HIGH SEVERITY INFORMATION ####################

Generic API Key: password = "exM3KYUftdJ="
```

Add Product Form

```
💻 SNIPPET ######################################

<html>
```

```
<head>
<title>Sweets Shop - Product Entry</title>
</head>
<body>
<h1>Add Product to Inventory</h1>

<p><a href = "admin_homepage.php">Admin Homepage</a> | <!--Make needed
edits for a nav bar-->
<a href = "search.html">Search Product</a> |
<a href = "new_item.html">Add Product</a> | <a href = "delete_item.html">
Delete Product</a> | <a href = "update_item.html">Modify Product</a> | <a
href = "manage_user.html"> Manage Site Users</a> | <a href =
"manage_admin.html"> Manage Admins</a> | <a href = "manage_orders.html">
Manage Site Orders</a> | <a href = "manage_receipts.html"> See Receipts</a>
| <a href = "logout.php">Logout</a></p>


<hr> <form action="insert_product.php" method="post"
enctype="multipart/form-data"> <table border="0">

<tr> <td>Product Identification Number</td> <!--use DB names--> <td><input
type="text" name="ProductID" maxlength="13" size="13"></td> </tr>

<tr> <td>Product Name</td> <td> <input type="text" name="name"
maxlength="60" size="30"></td> </tr>

<tr> <td>Details</td> <td> <input type="text" name="details" maxlength="80"
size="30"></td> </tr>

<tr> <td>Price $</td> <td><input type="text" name="price" maxlength="7"
size="7"></td> </tr>

<tr> <td>Quantity</td> <td> <input type="text" name="quantity"
maxlength="6" size="30"></td> </tr>

<tr> <td>Image</td> <td> <input type="file" name="image" maxlength="100"
size="30"></td> </tr>

<tr> <td colspan="2"><input type="submit" value="Register"></td> </tr>
</table> </form> </body> </html>

🔤 TITLE ##########################################

new_item.html - cPanel File Manager v3

📝 DESCRIPTION ####################################

A simple example of how to add a product to a shop - product entry.
```

🌐 LANGUAGE ##

Html

🔗 RELATED LINKS ###################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/filemanager/showfile.html?file=new_item.html&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=

Delete Item

💻 SNIPPET ##

```
<html>
<head>
  <title> Sweets Shop Delete Product</title>

</head>
<body>
<h1>Delete Product</h1>
  <p><a href = "admin_homepage.php">Admin Homepage</a> |
  <a href = "search.html">Search Product</a> |
  <a href = "new_item.html">Add Product</a> |
  <a href = "delete_item.html"> Delete Product</a> |
  <a href = "update_item.html">Modify Product</a> |
  <a href = "user_cart.html"> See Carts</a> | <!--make next 3-->
  <a href = "manage_user.html"> Manage Site Users</a> |
  <a href = "manage_orders.html"> Manage Site Orders</a> |
  <a href = "manage_receipts.html"> See Receipts</a> |
  <a href = "logout.php">Logout</a></p>
<?php

  // create short variable names
  $productID=$_POST['ProductID'];
  $productName=$_POST['name'];

  if (!$productID) {
     echo "You have not entered the required detail.<br />"
          ."Please go back and try again.";
          echo "<br><a href = 'delete_item.html'>Back to Delete
Product</a>";
     exit;
  }
```

```
    $productID = addslashes($productID);

$servername = "localhost";
$username = "ihekwac1_cr_ss123";
$password = "exM3KYUftdJ=";

$db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details

  $count = "select * from PRODUCTS where productID = ".$productID;
  $count_res = $db->query($count);

  $query = "delete from PRODUCTS where productID = ".$productID;
  $result = $db->query($query);

  $num = $count_res->num_rows;

    if($num === 0){
         echo "<br><h3> Record does not exist. </h3>";
    echo "<br><a href = 'delete_item.html'>Back to Delete Product</a>";
    exit;
    }

    echo "<br><h3> Records deleted: ".$num. " items. </h3>";
    echo "<br><a href = 'delete_item.html'>Back to Delete Product</a>";
    exit;

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

🔤 TITLE ########################################

delete_item.php - cPanel File Manager v3

📝 DESCRIPTION ###################################

Displays a list of short variable names for a given product.

🌐 LANGUAGE ######################################

Php

🔗 RELATED LINKS #################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=delete_item.php&fileop=&dir=%2Fhome%2Fihekwac1%2
Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweet
```

```
s+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=

⛔ HIGH SEVERITY INFORMATION ######################

Generic API Key: password = "exM3KYUftdJ="
```

Delete Item Form

```
💻 SNIPPET ########################################

<html>
<head> <title>Sweets Shop Product Deletion</title>
</head>
<body>
<h1>Remove Inventory</h1> <p><a href = "admin_homepage.php">Admin
Homepage</a> | <!--Make needed edits for a nav bar-->
<a href = "search.html">Search Product</a> |
<a href = "new_item.html">Add Product</a> |
<a href = "delete_item.html"> Delete Product</a> | <a href =
"update_item.html">Modify Product</a> | <a href = "manage_user.html">
Manage Site Users</a> | <a href = "manage_admin.html"> Manage Admins</a> |
<a href = "manage_orders.html"> Manage Site Orders</a> | <a href =
"manage_receipts.html"> See Receipts</a> | <a href =
"logout.php">Logout</a></p><hr>
<br><br>
<div class="container mt-3 w-25">

<form action="delete_item.php" method = "post"> <div class="mb-3 mt-3">

<label for="ProductID">Product ID:</label> <input type="text"
class="form-control" id="ProductID" placeholder="Enter Product ID: "
name="ProductID"> </div>

<button type="submit" class="btn btn-primary">Remove Product</button>
</form> </div> <br><br> </body> </html>

🔤 TITLE ##########################################

delete_item.html - cPanel File Manager v3

📝 DESCRIPTION ####################################

A simple template showing the logic for removing a shop product.

🌐 LANGUAGE ######################################
```

Html

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/filemanager/showfile.html?file=delete_item.html&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=

Modify Product

💻 SNIPPET ##

```
<html>
<head>
  <title>Sweets Shop Product Edit</title>
</head>
<body>
<h1>Product Edit</h1>
<?php
  // create short variable names - ProductID, name, details, price,
quantity

  $productID=$_POST['ProductID'];
  $name=$_POST['name'];
  $details=$_POST['details'];
  $price=$_POST['price'];
  $quant=$_POST['quantity'];

  if (!$productID && !$name && !$details && !$price && !$quant) {
     echo "You are not modifying the item.<br />"
          ."If this is a mistake, enter proper data or return to main
menu.";
      echo "<br><a href = 'update_item.html'>Back to Item Update</a>";
     exit;
  }

  $product = addslashes($product);
  $name = addslashes($name);
  $details = addslashes($details);
  $price = doubleval($price);
  $quant = addslashes($quant);

  $servername = "localhost";
  $username = "ihekwac1_cr_ss123"; //make username for shop db - after work
  $password = "exM3KYUftdJ="; //make passwd for db
```

```
  $db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details

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

  $query = "update PRODUCTS set
            name='".$name."' where ProductID = " .$productID;

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

  if ($result) {
      echo " $name: Edit Successful";
      echo "<br>All corresponding data under Product ID $productID have
also been updated.</br>";

      echo "<br><a href = 'update_item.html'>Back to Item Update</a>";
  } else {
        echo "An error has occurred.  The item was not updated.";
  }

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


🔤 TITLE ##########################################

update_product.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Updates shop product in database

🌐 LANGUAGE #######################################

Php

🏷️ TAGS ###########################################

FORMS, VALIDATION, AJAX

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
```

```
manager/showfile.html?file=update_product.php&fileop=&dir=%2Fhome%2Fihekwac
1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sw
eets+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=

⛔ HIGH SEVERITY INFORMATION ######################

Generic API Key: password = "exM3KYUftdJ="
```

Modify Product Form

```
💻 SNIPPET ########################################

<html> <head> <title>Sweets Shop - Product Edit</title> </head> <body>
<h1>Product Edit</h1> <p><a href = "admin_homepage.php">Admin Homepage</a>
| <!--Make needed edits for a nav bar--> <a href = "search.html">Search
Product</a> | <a href = "new_item.html">Add Product</a> | <a href =
"delete_item.html"> Delete Product</a> | <a href =
"update_item.html">Modify Product</a> | <a href = "manage_user.html">
Manage Site Users</a> | <a href = "manage_admin.html"> Manage Admins</a> |
<a href = "manage_orders.html"> Manage Site Orders</a> | <a href =
"manage_receipts.html"> See Receipts</a> | <a href =
"logout.php">Logout</a></p><hr>

<form action="update_product.php" method="post">
<table border="0">
<tr> <td>Product Identification Number</td> <!--use DB names--> <td><input
type="text" name="ProductID" maxlength="13" size="13"></td> </tr>

<tr> <td>Product Name</td>
<td> <input type="text" name="name" maxlength="60" size="30"></td> </tr>

<tr> <td colspan="2"><input type="submit" value="Register"></td> </tr>
</table> </form> </body> </html>

🔤 TITLE ##########################################

update_item.html - cPanel File Manager v3

📝 DESCRIPTION ####################################

Sweet Shop - Product Edit

🌐 LANGUAGE ######################################

Html

🔗 RELATED LINKS #################################
```

```
https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=update_item.html&fileop=&dir=%2Fhome%2Fihekwac1%
2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Swee
ts+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=
```

Product Search

💻 SNIPPET ##

```
<html>
<head>
  <title>Sweets Shop Search Results</title>
</head>
<body>
<h1>Search Results</h1>
<?php
  // create short variable names
  $searchtype=$_POST['searchtype'];
  $searchterm=trim($_POST['searchterm']);

  if (!$searchtype && !$searchterm) {
     echo 'You have not entered search details.  Please go back and try
again.';
     echo "<br><a href = 'search.html'>Back to Product Search</a>";
     exit;
  }

  if ($searchtype && $searchtype === "all") {

    $searchtype = addslashes($searchtype);

    $db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details

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

  if ($searchtype === "all"){
    $query = "select * from PRODUCTS";
  }

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

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

  echo "<h4>Number of products found: ".$num_results."</h4>";
  echo "<br><a href = 'search.html'>Back to Product Search</a>";

  for ($i=0; $i <$num_results; $i++) {
     $row = $result->fetch_assoc();
     echo "<p><strong>".($i+1).". Product ID: ";
     echo htmlspecialchars(stripslashes($row['ProductID']));
     echo "</strong><br />Product Name: ";
     echo stripslashes($row['name']);
     echo "<br />Price: ";
     echo stripslashes($row['price']);
     echo "<br />Current Quantity: ";
     echo stripslashes($row['quantity']);
     echo "</p>";
  }
  exit;
  $result->free();
  $db->close();
 }

 else{

    $searchtype = addslashes($searchtype);
    $searchterm = addslashes($searchterm);

    $db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details

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

  $query = "select * from PRODUCTS where ".$searchtype." like
'%".$searchterm."%'";

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

  $num_results = $result->num_rows;

  echo "<h4>Number of products found: ".$num_results."</h4>";
  echo "<br><a href = 'search.html'>Back to Product Search</a>";

  for ($i=0; $i <$num_results; $i++) {
     $row = $result->fetch_assoc();
     echo "<p><strong>".($i+1).". Product ID: ";
     echo htmlspecialchars(stripslashes($row['ProductID']));
```

```
    echo "</strong><br />Product Name: ";
    echo stripslashes($row['name']);
    echo "<br />Price: ";
    echo stripslashes($row['price']);
    echo "<br />Current Quantity: ";
    echo stripslashes($row['quantity']);
    echo "</p>";
  }

  exit;
  $result->free();
  $db->close();
}

?>
</body>
</html>

🔤 TITLE ##########################################

results.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Returns results of search entry

🌐 LANGUAGE #######################################

Php

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=results.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpub
lic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Sh
op&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=
```

Product Search Form

```
💻 SNIPPET ########################################

<html> <head> <title>Cookie Run Sweets Shop Inventory Search</title>
</head> <body> <h1>Sweets Shop Product Search</h1> <p><a href =
"admin_homepage.php">Admin Homepage</a> | <!--Make needed edits for a nav
bar-->
<a href = "search.html">Search Product</a> |
<a href = "new_item.html">Add Product</a> |
```

```
<a href = "delete_item.html"> Delete Product</a> | <a href =
"update_item.html">Modify Product</a> |
<a href = "manage_user.html"> Manage Site Users</a> |
<a href = "manage_admin.html"> Manage Admins</a> |
<a href = "manage_orders.html"> Manage Site Orders</a> |
<a href = "manage_receipts.html"> See Receipts</a> |
<a href = "logout.php">Logout</a></p><hr>

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

Choose Search Type:<br />
<select name="searchtype">
<option value="ProductId">Product ID
<option value="name">Product Name
<option value="price">Price
<option value="all">All
</select> <br />

<h6>Manual Search</h6> Enter Search Term:<br />
<input name="searchterm" type="text" size="40"> <br />
<input type="submit" name="submit" value="Search">
</form> </body> </html>

🔤 TITLE ########################################

search.html - cPanel File Manager v3

📝 DESCRIPTION ##################################

Sweet Shop Inventory Search

🌐 LANGUAGE #####################################

Html

🔗 RELATED LINKS ################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=search.html&fileop=&dir=%2Fhome%2Fihekwac1%2Fpub
lic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Sh
op&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=
```

Order Delete Admin Side

```
💻 SNIPPET ######################################

<html>
```

```
<head>
  <title> Sweets Shop Order Cancellation</title>

</head>
<body>
<h1>Delete Order</h1>
  <p><a href = "admin_homepage.php">Admin Homepage</a> |
  <a href = "search.html">Search Product</a> |
  <a href = "new_item.html">Add Product</a> |
  <a href = "delete_item.html"> Delete Product</a> |
  <a href = "user_cart.html"> See Carts</a> | <!--make next 3-->
  <a href = "manage_user.html"> Manage Site Users</a> |
  <a href = "manage_orders.html"> Manage Site Orders</a> |
  <a href = "manage_receipts.html"> See Receipts</a> |
  <a href = "logout.php">Logout</a></p>
<?php

  // create short variable names
  $searchtype=$_POST['searchtype'];
  $searchterm=$_POST['searchterm'];


  if (!$searchtype || !$searchterm) {
     echo "You have not entered the required detail.<br />"
          ."Please go back and try again.";
     exit;
  }

  $searchtype = addslashes($searchtype);
  $searchterm = addslashes($searchterm);

$servername = "localhost";
$username = "ihekwac1_cr_ss123";
$password = "exM3KYUftdJ=";

$db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details

  $count = "select * from ORDERS where ".$searchtype." =
'".$searchterm."'";
  $count_res = $db->query($count);

  $query = "delete from ORDERS where ".$searchtype." = '".$searchterm."'";
  $result = $db->query($query);

  $num = $count_res->num_rows;

    if($num === 0){
         echo "<br><h3> Record does not exist. </h3>";
```

```
    echo "<br><a href = 'delete_item.html'>Back to Delete Product</a>";
    }

    else {

    $receipt = "delete from RECEIPT where ".$searchtype." =
'".$searchterm."'";
    $db->query($receipt);

    echo $num. " Order(s) and receipt(s) removed: $num";
    echo "<br><a href = 'manage_orders.html'>Back to Site Orders</a>";

    }

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

🔤 TITLE ########################################

delete_order.php - cPanel File Manager v3

📝 DESCRIPTION ###################################

Displays a list of short variable names and their associated values.

🌐 LANGUAGE ######################################

Php

🔗 RELATED LINKS #################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=delete_order.php&fileop=&dir=%2Fhome%2Fihekwac1%
2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Swee
ts+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=

⛔ HIGH SEVERITY INFORMATION ####################

Generic API Key: password = "exM3KYUftdJ="
```

Order Delete Form

```
💻 SNIPPET ######################################
```

```
<html> <head> <title>Sweets Shop User Orders</title> </head> <body>
<h1>Order Cancellation</h1> <p><a href = "admin_homepage.php">Admin
Homepage</a> | <!--Make needed edits for a nav bar--> <a href =
"search.html">Search Product</a> | <a href = "new_item.html">Add
Product</a> |
<a href = "delete_item.html"> Delete Product</a> |
<a href = "update_item.html">Modify Product</a> |
<a href = "manage_user.html"> Manage Site Users</a> |
<a href = "manage_admin.html"> Manage Admins</a> |
<a href = "manage_orders.html"> Manage Site Orders</a> |
<a href = "manage_receipts.html"> See Receipts</a> |
<a href = "logout.php">Logout</a></p><hr>
<br><br>
<div class="container mt-3 w-25">
<form action="delete_order.php" method = "post">
Choose Search Type:<br />
<select name="searchtype">
<option value="orderid">Order ID
<option value="u_email">Email
</select> <br />

<h4>ID Entry</h4>
Enter ID Term:<br />
<input name="searchterm" type="text" size="40"> <br />
<button type="submit" class="btn btn-primary">Cancel Order(s)</button>
</form> </div> <br><br> </body> </html>

🔤 TITLE ########################################

manage_orders.html - cPanel File Manager v3

📝 DESCRIPTION ###################################

Order Deletion of sweet shop user orders.

🌐 LANGUAGE #####################################

Html

🔗 RELATED LINKS #################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=manage_orders.html&fileop=&dir=%2Fhome%2Fihekwac
1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sw
eets+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=
```

Order/Receipt Lookup

💻 SNIPPET ##

```
<html>
<head>
  <title>Sweets Shop Order Receipts</title>
</head>
<body>
<h1>Receipt</h1>
<?php

$recTotal = 0.00;
  // create short variable names
  $searchtype=$_POST['searchtype'];
  $searchterm=trim($_POST['searchterm']);

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

    $searchtype = addslashes($searchtype);
    $searchterm = addslashes($searchterm);

    $db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details

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

  $query = "select * from RECEIPT where ".$searchtype." like
'%".$searchterm."%'";

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

  $num_results = $result->num_rows;

  echo "<h4>Order Receipt for Given Criteria: </h4>";

  echo "<p>Order processed at ".date('H:i, jS F Y')."</p>"; //temp value
may extract this info from user side order page


  for ($i=0; $i <$num_results; $i++) {
    $row = $result->fetch_assoc();
    echo "<p><strong>".($i+1).". Receipt ID: ";
    echo htmlspecialchars(stripslashes($row['receipt_ID']));
```

```
     echo "</strong><br />Order ID: ";
    echo stripslashes($row['o_id']);
    echo "</strong><br />Shipping Address: ";
    echo stripslashes($row['shipAdd']);
    echo "<br />Item Total: ";
    echo stripslashes($row['total']);

    $recTotal = $row['total'] += $recTotal;

    echo "</p>";
 }


        $r_id = $row['receipt_ID'];
     $o_id= $row['o_id'];
  $u_email=$row['u_email'];
  $payType=$row['payType'];
  $shipAdd=$row['shipAdd'];

   $o_id = addslashes($o_id);
   $r_id = addslashes($r_id);
    $u_email = addslashes($u_email);
    $payType = addslashes($payType);
    $shipAdd = addslashes($shipAdd);

    echo "<p>User Email: $u_email </p>";
    echo "<p>Payment Type: $payType </p>";
    echo "<p>Address: $shipAdd <p>";

 echo "Total: $recTotal";

 echo "<br><a href = 'manage_receipts.html'>Back to Receipt Search</a>";

  $result->free();
  $db->close();

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

🔤 TITLE ##

receipts.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

This function renders the content of a single node of a node that has a single receipt.

🌐 LANGUAGE ##

Php

🔗 RELATED LINKS ###################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/filemanager/showfile.html?file=receipts.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=

Order Lookup Form

💻 SNIPPET ##

```
<html>
<head>
<title>Cookie Run Sweets Shop Receipts</title>
</head>
<body> <h1>Sweets Shop Receipt Search</h1>

<p><a href = "admin_homepage.php">Admin Homepage</a> | <!--Make needed
edits for a nav bar-->
<a href = "search.html">Search Product</a> |
<a href = "new_item.html">Add Product</a> |
<a href = "delete_item.html"> Delete Product</a> |
<a href = "update_item.html">Modify Product</a> |
<a href = "manage_user.html"> Manage Site Users</a> |
<a href = "manage_admin.html"> Manage Admins</a> |
<a href = "manage_orders.html"> Manage Site Orders</a> |
<a href = "manage_receipts.html"> See Receipts</a> |
<a href = "logout.php">Logout</a></p><hr>

<form action="receipts.php" method="post"> Choose Search Type:<br />
<select name="searchtype">
<option value="o_id">Order ID
<option value="u_email">Email
</select> <br />

<h4>Manual Search</h4> Enter Receipt Information:<br />
<input name="searchterm" type="text" size="40"> <br />
<input type="submit" name="submit" value="Search">
</form> </body> </html>
```

🔤 TITLE ##

```
manage_receipts.html - cPanel File Manager v3

📝 DESCRIPTION ####################################

Sweet Shop Receipts

🌐 LANGUAGE #######################################

Html

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=manage_receipts.html&fileop=&dir=%2Fhome%2Fihekw
ac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+
Sweets+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=
```

Admin Handling (HR)

```
💻 SNIPPET ########################################

<html>
<head>
  <title> Sweets Shop Admin Removal</title>

</head>
<body>
<h1>Admin Removal</h1>
<?php

  // create short variable names
  $adminID=$_POST['adminID'];
  $Reason=$_POST['reason'];

  if (!$adminID || !$Reason) {
     echo "You have not entered the required detail.<br />"
          ."Please go back and try again.";
     exit;
  }

    $adminID = addslashes($adminID);
    $Reason = addslashes($Reason);

$servername = "localhost";
$username = "ihekwac1_cr_ss123";
$password = "exM3KYUftdJ=";
```

```
$db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details


  $count = "select * from ADMINISTRATORS where adminID = ".$adminID;
  $adm_count = $db->query($count);

  $num_results = $adm_count->num_rows;

  $query = "delete from ADMINISTRATORS where adminID = ".$adminID;
  $result = $db->query($query);

  if ($num_results === 0) {
      echo "Admin not in Database.";
      echo "<br><a href = 'manage_admin.html'>Back to Admin
Management</a>";
  }

  else {
    echo "Admin ID: $adminID removed for $Reason";
    echo "<br><a href = 'manage_admin.html'>Back to Admin Management</a>";

}
  $db->close();
?>
</body>
</html>

🔤 TITLE ##########################################

delete_admin.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Deletes an admin based on ID

🌐 LANGUAGE #######################################

Php

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=delete_admin.php&fileop=&dir=%2Fhome%2Fihekwac1%
2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Swee
ts+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=

⛔ HIGH SEVERITY INFORMATION ######################
```

```
Generic API Key: password = "exM3KYUftdJ="
```

Admin Handling Form

```
💻 SNIPPET ########################################

<html> <head> <title>Sweets Shop Admin Removal</title> </head> <body>
<h1>Admin Removal</h1> <p>
<a href = "admin_homepage.php">Admin Homepage</a> |
<a href = "search.html">Search Product</a> |
<a href = "new_item.html">Add Product</a> |
<a href = "delete_item.html"> Delete Product</a> |
<a href = "update_item.html">Modify Product</a> |
<a href = "manage_user.html"> Manage Site Users</a> |
<a href = "manage_admin.html"> Manage Admins</a> |
<a href = "manage_orders.html"> Manage Site Orders</a> |
<a href = "manage_receipts.html"> See Receipts</a> |
<a href = "logout.php">Logout</a></p>
<br><br>
<div class="container mt-3 w-25">
<form action="delete_user.php" method = "post">
<div class="mb-3 mt-3">
<p>This is an HR page - If any admins are failing at their work, enter
their ID below.</p>
<br><br/>
<label for="AccountID">Admin ID:</label>
<input type="text" class="form-control" id="AccountID" placeholder="Enter
User ID: " name="AccountID"> </div>
<br><br/>
 Why are you removing this admin:<br />
<select name="reason">
<option value="Lack of Regulation - No duties have been carried out by this
admin with proper workmanship.">Poor Work
<option value="Power abuse - kicking users for unfounded reason or
controlling stock to manipulate purchases, etc. ">Abuse of Powers </select>
<br><br/>
<button type="submit" class="btn btn-primary">Remove Admin Account</button>
</form> </div> <br><br> </body> </html>

🔤 TITLE ##########################################

manage_admin.html - cPanel File Manager v3

📝 DESCRIPTION ####################################
```

Sweet Shop Admin Removal

🌐 LANGUAGE ##

Html

🔗 RELATED LINKS ###################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/filemanager/showfile.html?file=manage_admin.html&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=

Delete User

💻 SNIPPET ##

```
<html>
<head>
  <title> Sweets Shop User Removal</title>
</head>
<body>
<h1>Account Deletion</h1>

<?php

  // create short variable names
  $AccountID=$_POST['AccountID'];
  $Reason=$_POST['reason'];

  if (!$AccountID||!$Reason) {
     echo "You have not entered the required detail.<br />"
          ."Please go back and try again.";
     exit;
  }

    $AccountID = addslashes($AccountID);
    $Reason = addslashes($Reason);

$servername = "localhost";
$username = "ihekwac1_cr_ss123";
$password = "exM3KYUftdJ=";

$db = new mysqli('localhost', 'ihekwac1_cr_ss123', 'exM3KYUftdJ=',
```

```
'ihekwac1_cookierunsweetsshopdb'); //chnage to own db details

  $count = "select * from CUSTOMER where AccountID = ".$AccountID;
  $usr_count = $db->query($count);

  $num_results = $usr_count->num_rows;

  $query = "delete from CUSTOMER where AccountID = ".$AccountID;
  $db->query($query);

  if ($num_results === 0) {
      echo "User not in Database.";
      echo "<br><a href = 'manage_user.html'>Back to Delete User</a>";
  }

  else {
    echo "User with ID: $AccountID removed for $Reason.";
    echo "<br><a href = 'manage_user.html'>Back to Delete User</a>";

}
  $db->close();
?>
</body>
</html>

🔤 TITLE ##########################################

delete_user.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

This function creates a simple list of short variable names for a given
user.

🌐 LANGUAGE #######################################

Php

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=delete_user.php&fileop=&dir=%2Fhome%2Fihekwac1%2
Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweet
s+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=

⛔ HIGH SEVERITY INFORMATION ######################

Generic API Key: password = "exM3KYUftdJ="
```

Delete User Form

💻 SNIPPET ##

```
<html> <head> <title>Sweets Shop User Removal</title> </head> <body>
<h1>Account Deletion</h1> <p>
<a href = "admin_homepage.php">Admin Homepage</a> | <!--Make needed edits
for a nav bar-->
<a href = "search.html">Search Product</a> |
<a href = "new_item.html">Add Product</a> |
<a href = "delete_item.html"> Delete Product</a> |
<a href = "update_item.html">Modify Product</a> |
<a href = "manage_user.html"> Manage Site Users</a> |
<a href = "manage_admin.html"> Manage Admins</a> |
<a href = "manage_orders.html"> Manage Site Orders</a> |
<a href = "manage_receipts.html"> See Receipts</a> |
<a href = "logout.php">Logout</a></p>

<hr>
<br><br>
<div class="container mt-3 w-25">
<form action="delete_user.php" method = "post">
<div class="mb-3 mt-3"> <p>If a user is suspicious, or breaks site rules,
enter their ID below:</p>
<br><br/>
<label for="AccountID">Account ID:</label>
<input type="text" class="form-control" id="AccountID" placeholder="Enter
User ID: " name="AccountID">
<br><br/>
Why are you removing this user:<br />
<select name="reason">
<option value="Inappropriate User Name/Fraudulent Mailing Address">Site
Misconduct (Ban Evasion)
<option value="Suspicious Purchase Activity (Possible bot purchases with
rapid order submission in small window)">Suspicious Activity (Bot
Purchasing)
<option value=" Inactive Account - (>3 years with no orders)">Inactive
Account
</select>
<br><br/>
</div>
<button type="submit" class="btn btn-primary">Remove User Account</button>
</form> </div> <br><br> </body> </html>
```

🔤 TITLE ##

manage_user.html - cPanel File Manager v3

📝 DESCRIPTION ####################################

A simple dashboard showing how to delete a user s account.

🌐 LANGUAGE ####################################

Html

🔗 RELATED LINKS ####################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/filemanager/showfile.html?file=manage_user.html&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=

Landing Page

💻 SNIPPET ####################################

```
<!DOCTYPE html> <html> <head> <title>Cookie Run Sweets Shop Landing Page</title> <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"> <style> body{ font: 14px sans-serif; text-align: center; } </style> </head> <body> <div class="p-3 mb-2 bg-light text-dark"><h1>Welcome to Cookie Run Sweets Shop!</h1> <h3>Are you an <strong>Administrator</strong>, or a <strong>User</strong></h3> <h3>Select your login page below:</h3> <nav class="navbar navbar-expand-sm justify-content-center"> <ul class="navbar-nav"> <li class="nav-item"> <h4><a class="nav-link" href="admin_login.php">I'm an Admin</a></h4> </li> <li class="nav-item"> <h4><a class="nav-link" href="user_login.php">I'm a User</a></h4> </li> </ul> </nav> </div> </body> </html>
```

🔤 TITLE ####################################

landing_homepage.html - cPanel File Manager v3

📝 DESCRIPTION ####################################

The main HTML page for the cookie run sweets shop landing page.

🌐 LANGUAGE ####################################

Html

🏷️ TAGS ####################################

ANGULARJS

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/filemanager/showfile.html?file=landing_homepage.html&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=
https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

User View on Admin Side

💻 SNIPPET ##

```
<?php
      include "config.php";
      session_start();

      $data=[];

      $sql="select * from PRODUCTS";

      $res=$db->query($sql);

      if($res->num_rows>0){

            while($row=$res->fetch_assoc()){

                  $data[]=$row;
            }
      }
?>
<html>
      <head>
        <title>Products</title>
            <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css
">
    </head>
    <body>
            <?php include "admin_navbar.php"; ?>
        <div class='container mt-5 pb-5'>
                  <h2 class='text-muted mb-4 text-center'>Products</h2>
                  <div class='row'>
                        <?php foreach($data as $row): ?>
                              <div class='col-md-3 mt-2'>
```

```
                                    <div class="card">
                                      <img class="card-img-top"
src="images/<?php echo $row["image"]; ?>" >
                                      <div class="card-body">
                                          <h5 class="card-title"><?php echo
$row["name"]; ?></h5>
                                          <p class="card-text">
                                                Price &#36; <?php echo
$row["price"]; ?>
                                          </p>
                                          <a
href="admin_view_details.php?id=<?php echo $row["ProductID"]; ?>"
class='btn btn-primary  float-right' >View Details</a>
                                      </div>
                                    </div>
                                </div>
                            <?php endforeach; ?>
                        </div>
                    </div>
    </body>
</html>

🔤 TITLE ########################################

admin_index.php - cPanel File Manager v3

📝 DESCRIPTION ###################################

<?php echo >

🌐 LANGUAGE #####################################

Php

🔗 RELATED LINKS #################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=admin_index.php&fileop=&dir=%2Fhome%2Fihekwac1%2
Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweet
s+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
```

User View Admin Side View Details

```
💻 SNIPPET ######################################
```

```
<?php
	include "config.php";
	session_start();

	include "cart.class.php";
	$cart=new Cart();

	if(isset($_POST["submit"])){
		$item=[
			"id"=>$_POST["pid"],
			"name"=>$_POST["product"],
			"price"=>$_POST["price"],
			"qty"=>$_POST["qty"],
			"total"=>($_POST["qty"]*$_POST["price"]),
			"img"=>$_POST["img"],
		];
	}

	$data=[];
	$sql="select * from PRODUCTS where ProductID={$_GET["id"]}";
	$res=$db->query($sql);
	if($res->num_rows>0){
		$data=$row=$res->fetch_assoc();
	}
?>
<html>
	<head>
	  <title>Products Details</title>
		<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css
">
    </head>
    <body>
	  <?php include "admin_navbar.php"; ?>
	    <div class='container mt-5'>
			<div class='row'>
				<div class='col-md-9 mx-auto'>
					<h2 class='text-muted mb-4'>Product
Details</h2><hr>
					<div class='row mt-5'>
						<div class='col-md-4'>
							<img src="images/<?php echo
$data["image"]; ?>" class='img-thumbnail'>
						</div>
						<div class='col-md-8'>
							<h2 class='text-muted'><?php echo
$data["name"]; ?></h2>
							<p class="font-weight-bold">Price
&#36; <?php echo $data["price"]; ?></p>
```

```
                                        <p><?php echo $data["details"];
?></p>

                                        <form method='post' action='<?php
echo $_SERVER["REQUEST_URI"];?>'>
                                            <input type='hidden'
name='pid' value='<?php echo $data["ProductID"]; ?>'>
                                            <input type='hidden'
name='product' value='<?php echo $data["name"]; ?>'>
                                            <input type='hidden'
name='price' value='<?php echo $data["price"]; ?>'>
                                            <input type='hidden'
name='img' value='<?php echo $data["image"]; ?>'>
                                        </form>
                                    </div>
                                </div>
                            </div>
                        </div>
                </div>
    </body>
</html>

🔤 TITLE ########################################

admin_view_details.php - cPanel File Manager v3

📝 DESCRIPTION ##################################

Displays product details of single product on admin side.

🌐 LANGUAGE #####################################

Php

🔗 RELATED LINKS ################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=admin_view_details.php&fileop=&dir=%2Fhome%2Fihe
kwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Ru
n+Sweets+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
```

Admin User side navigation bar

```
💻 SNIPPET ######################################
```

```
<nav class="navbar navbar-expand-lg navbar-dark bg-info ">
<div class='container'>
  <a class="navbar-brand" href="index.php">CR:SS Admin - User View </a>
  <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup"
aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav ml-auto">
      <a class="nav-item nav-link" href="admin_index.php">Products</a>
      <a class="nav-item nav-link" href="admin_homepage.php">Back to Admin
View</a>
    </div>
  </div>
  </div>
</nav>

🔤 TITLE ##########################################

admin_navbar.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Displays a navbar with a list of admin nav zones.

🌐 LANGUAGE #######################################

Html

🏷️ TAGS ###########################################

BOOT, TWITTER-BOOTSTRAP, NAVBAR, ANGULARJS, ANGULAR

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess7755510628/frontend/jupiter/file
manager/showfile.html?file=admin_navbar.php&fileop=&dir=%2Fhome%2Fihekwac1%
2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Swee
ts+Shop&dirop=&charset=&file_charset=utf-8&baseurl=&basedir=
```

USER SIDE:

Sign Up:

```
💻 SNIPPET ########################################
```

```
<?php
// Include config file
require_once "config.php";

// Define variables and initialize with empty values
$useremail = $password = $confirm_password = "";
$useremail_err = $password_err = $confirm_password_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // Validate username
    if(empty(trim($_POST["useremail"]))){
        $useremail_err = "Please enter a User Email.";
    }
elseif(!preg_match("/^([a-z0-9\+_\-]+)(\.[a-z0-9\+_\-]+)*@([a-z0
-9\-]+\.)+[a-z]{2,6}$/ix", trim($_POST["useremail"]))){
        $useremail_err = "Email can only contain letters,
numbers, and underscores.";
    } else{
        // Prepare a select statement
        $sql = "SELECT AccountID FROM CUSTOMER WHERE useremail =
?";

        if($stmt = $db->prepare($sql)){
            // Bind variables to the prepared statement as
parameters
            $stmt->bind_param("s", $param_useremail);

            // Set parameters
            $param_useremail = trim($_POST["useremail"]);

            // Attempt to execute the prepared statement
            if($stmt->execute()){
                // store result
                $stmt->store_result();

                if($stmt->num_rows == 1){
                    $useremail_err = "Email is already
registered - go back and try another email.";
```

```
                } else{
                    $useremail = trim($_POST["useremail"]);
                }
            } else{
                echo "Oops! Something went wrong. Please try
again later.";
            }

            // Close statement
            $stmt->close();
        }
    }

    // Validate password
    if(empty(trim($_POST["password"]))){
        $password_err = "Please enter a password.";
    } elseif(strlen(trim($_POST["password"])) < 6){
        $password_err = "Password must have at least 6
characters.";
    } else{
        $password = trim($_POST["password"]);
    }

    // Validate confirm password
    if(empty(trim($_POST["confirm_password"]))){
        $confirm_password_err = "Please confirm password.";
    } else{
        $confirm_password = trim($_POST["confirm_password"]);
        if(empty($password_err) && ($password !=
$confirm_password)){
            $confirm_password_err = "Password did not match.";
        }
    }

    // Check input errors before inserting in database
    if(empty($useremail_err) && empty($password_err) &&
empty($confirm_password_err)){

        // Prepare an insert statement
        $sql = "INSERT INTO CUSTOMER (useremail, userpass)
VALUES (?, ?)";
```

```
        if($stmt = $db->prepare($sql)){
            // Bind variables to the prepared statement as
parameters
            $stmt->bind_param("ss", $param_useremail,
$param_password);

            // Set parameters
            $param_useremail = $useremail;
            $param_password = password_hash($password,
PASSWORD_BCRYPT); // Creates a password hash

            // Attempt to execute the prepared statement
            if($stmt->execute()){
                header("location: user_login.php");

            } else{
                echo "Oops! Something went wrong. Please try
again later.";
            }

            // Close statement
            $stmt->close();
        }
    }

    // Close connection
    $db->close();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Sign Up</title>
    <style>
        body{ font: 14px sans-serif; }
        .wrapper{ width: 360px; padding: 20px; }
    </style>
</head>
```

```
<body>
    <div class="wrapper">
        <h2>Sign Up</h2>
        <p>Please fill this form to create an account.</p>
        <form action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
            <div class="form-group">
                <label>User Email</label>
                <input type="text" name="useremail"
class="form-control <?php echo (!empty($useremail_err)) ?
'is-invalid' : ''; ?>" value="<?php echo $useremail; ?>">
                <span class="invalid-feedback"><?php echo
$useremail_err; ?></span>
            </div>
            <div class="form-group">
                <label>Password</label>
                <input type="password" name="password"
class="form-control <?php echo (!empty($password_err)) ?
'is-invalid' : ''; ?>" value="<?php echo $password; ?>">
                <span class="invalid-feedback"><?php echo
$password_err; ?></span>
            </div>
            <div class="form-group">
                <label>Confirm Password</label>
                <input type="password" name="confirm_password"
class="form-control <?php echo (!empty($confirm_password_err)) ?
'is-invalid' : ''; ?>" value="<?php echo $confirm_password; ?>">
                <span class="invalid-feedback"><?php echo
$confirm_password_err; ?></span>
            </div>
            <div class="form-group">
                <input type="submit" class="btn btn-primary"
value="Submit">
            </div>
            <p>Already have an account? <a
href="user_login.php">Login here</a>.</p>
        </form>
    </div>
</body>
</html>
```

🔤 TITLE ##

user_signup.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Generate a sequence of user sequence sequence sequences.

🌐 LANGUAGE ######################################

Php

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=user_signup.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=

Login

💻 SNIPPET ##

```
<?php
// Initialize the session
session_start();

// Check if the user is already logged in, if yes then redirect him to welcome page
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){

    header("location: user_landing.html");

    exit;
}

// Include config file
```

```
$db = new mysqli('localhost', 'ihekwac1_cr_ss123',
'exM3KYUftdJ=', 'ihekwac1_cookierunsweetsshopdb'); //chnage to
own db details

// Define variables and initialize with empty values

$email = $password = "";

$username_err = $password_err = $login_err = "";

// Processing form data when form is submitted

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

    // Check if username is empty

    if(empty(trim($_POST["email"]))){

        $username_err = "Please enter username.";

    } else{

        $email = trim($_POST["email"]);
    }

    // Check if password is empty

    if(empty(trim($_POST["password"]))){

        $password_err = "Please enter your password.";

    } else{

        $password = $_POST["password"];
    }

    // Validate credentials

    if(empty($username_err) && empty($password_err)){

        // Prepare a select statement
```

```
        $sql = "SELECT * FROM CUSTOMER WHERE useremail = ?";

        if($stmt = $db->prepare($sql)){

            // Bind variables to the prepared statement as
parameters

            $stmt->bind_param("s", $param_username);

            // Set parameters

            $param_username = $email;

            // Attempt to execute the prepared statement

            if($stmt->execute()){

                // Store result

                $stmt->store_result();


                // Check if username exists, if yes then verify
password

                if($stmt->num_rows == 1){

                    // Bind result variables

                    $hashed_password =
password_hash($_POST['password'], PASSWORD_BCRYPT);

                    $userpass = $_POST['password'];

                    $stmt->bind_result($AccountID, $email,
$hashed_password);

                    if($stmt->fetch()){

                        if(password_verify($userpass,
```

```
$hashed_password)){
                            // Password is correct, so start a
new session
                            session_start();
                            // Store data in session variables
                            $_SESSION["loggedin"] = true;
                            $_SESSION["AccountID"] = $AccountID;
                            $_SESSION["email"] = $email;
                            // Redirect user to welcome page
                            header("location:
user_landing.html");
                        } else{
                            // Password is not valid, display a
generic error message
                            $login_err = "Invalid password.";

                        }
                    }
                } else{
                    // Username doesn't exist, display a generic
error message
                    $login_err = "Invalid or non-extistant
username.";
                }
            } else{
                echo "Oops! Something went wrong. Please try
again later.";
```

```
            }

            // Close statement
            $stmt->close();
        }
    }

    // Close connection
    $db->close();
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>User Login</title>

</head>
<body>
    <h2>Cookie Run Sweets Shop User Login</h2>
        <p>Please enter your account information to login</p>

        <?php
        // if there is a login error, display error message
        if(!empty($login_err)){
            echo '<div class="alert alert-danger">' . $login_err
. '</div>';

        }
        ?>

        <form action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">

            <div class="form-group row">
                <label class = "col-form-label
col-sm-2">Email</label>
               <div class = "col-sm-10"> <input type="text"
name="email" class="form-control <?php echo
(!empty($email_error)) ? 'is-invalid' : ''; ?>" value="<?php
echo $email; ?>">
```

```
                <span class="invalid-feedback"><?php echo
$email_error; ?></span></div>


            </div>
            <div class="form-group row">
                <label class = "col-form-label col-sm-2
">Password</label>
                <div class = "col-sm-10"><input type='password'
name='password' class="form-control <?php echo
(!empty($password_error)) ? 'is-invalid' : ''; ?>">
                <span class="invalid-feedback"><?php echo
$password_error; ?></span></div>


            </div>
            <div class="form-group">
                <div class= "offset-sm-2 col-sm-10">
                    <input type="submit" class="btn btn-primary"
value="Login">

                    <p>Don't have an account? <a
href="user_signup.php">Sign Up</a></p>
                    <p><a href="landing_homepage.html">Back to
Landing Page</a></p>
                </div>
            </div>
        </form>
    </div>
</body>
</html>

🔤 TITLE ##########################################

user_login.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

This function initialize a user sequence sequence.

🌐 LANGUAGE ######################################
```

Php

🏷️ TAGS ##

COOKIES, SESSION, ASP.NET, LARAVEL

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=user_login.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=

Logout

💻 SNIPPET ##

```
<?php
// Initialize the session
session_start();

// Unset all of the session variables
$_SESSION = array();

// Destroy the session.
session_destroy();

// Redirect to login page
header("location: landing_homepage.html");
exit;
?>
```

🔤 TITLE ##

user_logout.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

This function is called to initialize the session and destroy the session if a neccesary

🌐 LANGUAGE ##

Php

🏷️ TAGS ##

SESSION, ASP.NET, LARAVEL

🔗 RELATED LINKS ##

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=user_logout.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=

User Receipts

💻 SNIPPET ##

```
<html>
<head>
  <title>Orders and Receipts</title>
</head>
<body>
<h1>Receipt</h1>
<?php

  // create short variable names
  $searchterm=trim($_POST['searchterm']);

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

```
  }
    $searchterm = addslashes($searchterm);

    $db = new mysqli('localhost', 'ihekwac1_cr_ss123',
'exM3KYUftdJ=', 'ihekwac1_cookierunsweetsshopdb'); //chnage to
own db details

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

  $query = "select * from RECEIPT where u_email =
'".$searchterm."'";

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

  $num_results = $result->num_rows;

  echo "<h4>Orders with Receipts for Given Criteria:
$num_results </h4>";
  echo "These are you order records/receipts - if none exists,
no orders have been placed <br> Or have been cancelled by user
or site admin.";


  echo "<p>Orders processed below: </p>"; //temp value may
extract this info from user side order page

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

    $r_id= $row['receipt_ID'];
         $o_id= $row['o_id'];
  $u_email=$row['u_email'];
  $o_name=$row['o_name'];
  $shipAdd=$row['shipAdd'];
  $total=$row['total'];

   $o_id = addslashes($o_id);
```

```
   $r_id = addslashes($r_id);
    $u_email = addslashes($u_email);
    $o_name = addslashes($o_name);
    $shipAdd = addslashes($shipAdd);
    $total = addslashes($total);

    $num = ($i+1);
    echo "<br></br>";
    echo "<p><strong>$num.</strong> Order ID:
<strong>$o_id</strong> </p>";
    echo "<p>Receipt ID: $r_id </p>";
    echo "<p>User Email: $u_email </p>";
    echo "<p>Name: $o_name </p>";
    echo "<p>Address: $shipAdd <p>";
    echo "Total: $total";
    echo "<br></br>";

    echo "</p>";
 }


 echo "<br><a href = 'your_receipts.html'>Back to Receipt
Search</a>";

  $result->free();
  $db->close();

?>
</body>
</html>

🔤 TITLE ##########################################

user_receipts.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

This function renders the content of a single node node which is
a sequence of records that have

🌐 LANGUAGE ######################################
```

Php

🏷️ TAGS ##

WORDPRESS

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=user_receipts.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=

View Cart

💻 SNIPPET ##

```
<?php
     include "config.php";
     session_start();

     include "cart.class.php";
     $cart=new Cart();
?>
<html>
     <head>
        <title>Cart</title>
          <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
          <script
src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    </head>
    <body>
          <?php include "navbar.php"; ?>
        <div class='container mt-3'>
               <div class='row'>
```

```
                    <div class='col-md-12'>
                         <h2 class='text-muted mb-4'>Cart
Items</h2>
                         <?php if($cart->get_cart_total()>0):
?>
                         <table class='table table-striped
table-bordered'>
                              <thead>
                                   <tr>
                                        <th colspan='2'
class='text-center'>Product</th>
                                        <th>Price</th>
                                        <th>Qty</th>
                                        <th>Total</th>
                                        <th>Action</th>
                                   </tr>
                              </thead>
                              <tbody>
                              <?php
$items=$cart->get_all_items(); ?>
                              <?php foreach($items as $item):
?>
                                   <tr>
                                        <td><img
src='images/<?php echo $item["img"];?>' style='height:80px;'
></td>
                                        <td><?php echo
$item["name"];?></td>
                                        <td>&#36; <?php echo
$item["price"];?></td>
                                        <td><input
type='number' value='<?php echo $item["qty"];?>' class='qty'
pid='<?php echo $item["id"]; ?>' min='1'></td>
                                        <td>&#36; <span
class='row_total'><?php echo $item["total"];?></span></td>
                                        <td><a
href='remove.php?id=<?php echo $item["id"]; ?>' onclick="return
confirm('Are You Sure?')">Remove</a></td>
                                   </tr>
                              <?php endforeach; ?>
                              </tbody>
```

```
                              <tfoot>
                                   <tr>
                                        <td colspan='3'><a
href='index.php' class="btn btn-success">Continue
Shopping</a></td>
                                        <th>Total</th>
                                        <th>&amp;#36;  <span
id='total'><?php echo $cart->get_cart_total();?></span></th>
                                        <td><a
href='checkout.php' class='btn btn-info'>Checkout</a></td>
                                   </tr>
                              </tfoot>
                         </table>
                         <?php else: ?>
                              <div class='alert
alert-warning'>Cart is empty...</div>
                         <?php endif; ?>
                    </div>
               </div>
          </div>
          <script>
               $(document).ready(function(){
                    $(".qty").change(function(){
                         update_cart($(this));
                    });
                    $(".qty").keyup(function(){
                         update_cart($(this));
                    });

                    function update_cart(cls){
                         var pid=$(cls).attr("pid");
                         var q=$(cls).val();

                         $.ajax({
                              url:"ajax_update_cart.php",
                              type:"post",
                              data:{id:pid,qty:q},
                              success:function(res){
                                   console.log(res);

                                   var a=JSON.parse(res);
```

```
                                    $("#total").text(a.total);

$(cls).closest("tr").find(".row_total").text(a.row_total);
                                }
                            });
                        }
                });
            </script>
    </body>
</html>

🔤 TITLE ##########################################

view_cart.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Displays a list of all items in the cart.

🌐 LANGUAGE ######################################

Php

🏷️ TAGS ##########################################

CODEIGNITER, AJAX

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/j
upiter/filemanager/showfile.html?file=view_cart.php&fileop=&dir=
%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cass
ie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_chars
et=&baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.mi
n.css">
https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

View Details

💻 SNIPPET ##

```
<?php
     include "config.php";
     session_start();

     include "cart.class.php";
     $cart=new Cart();

     if(isset($_POST["submit"])){
          $item=[
               "id"=>$_POST["pid"],
               "name"=>$_POST["product"],
               "price"=>$_POST["price"],
               "qty"=>$_POST["qty"],
               "total"=>($_POST["qty"]*$_POST["price"]),
               "img"=>$_POST["img"],
          ];
          $cart->add_to_cart($item);
          header("location:view_cart.php");
     }

     $data=[];
     $sql="select * from PRODUCTS where
ProductID={$_GET["id"]}";
     $res=$db->query($sql);
     if($res->num_rows>0){
          $data=$row=$res->fetch_assoc();
     }
?>
<html>
     <head>
        <title>Products Details</title>
          <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootst
rap.min.css">
    </head>
    <body>
     <?php include "navbar.php"; ?>
        <div class='container mt-5'>
               <div class='row'>
```

```
                    <div class='col-md-9 mx-auto'>
                         <h2 class='text-muted mb-4'>Product
Details</h2><hr>
                         <div class='row mt-5'>
                              <div class='col-md-4'>
                                   <img src="images/<?php
echo $data["image"]; ?>" class='img-thumbnail'>
                              </div>
                              <div class='col-md-8'>
                                   <h2 class='text-muted'><?php
echo $data["name"]; ?></h2>
                                   <p
class="font-weight-bold">Price &#36; <?php echo $data["price"];
?></p>
                                   <p
class="font-weight-bold">In Stock: <?php echo $data["quantity"];
?></p>
                                   <p><?php echo
$data["details"]; ?></p>

                                   <form method='post'
action='<?php echo $_SERVER["REQUEST_URI"];?>'>
                                        <input type='hidden'
name='pid' value='<?php echo $data["ProductID"]; ?>'>
                                        <input type='hidden'
name='product' value='<?php echo $data["name"]; ?>'>
                                        <input type='hidden'
name='price' value='<?php echo $data["price"]; ?>'>
                                        <input type='hidden'
name='qty' value='<?php echo $data["quantity"]; ?>'>
                                        <input type='hidden'
name='img' value='<?php echo $data["image"]; ?>'>
                                             <p><input
type='number' min='1' value='1' name='qty' required
class='form-control col-md-5'></p>
                                        <input type='submit'
name='submit' value='Add To Cart' class='btn btn-primary'>
                                   </form>
                              </div>
                         </div>
                    </div>
```

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

🔤 TITLE ##########################################

view_details.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Displays a list of all products in the system.

🌐 LANGUAGE #######################################

Php

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=view_details.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
```

Remove Cart Item

```
💻 SNIPPET ########################################

<?php
  session_start();
  $id=$_GET["id"];

  include "cart.class.php";
  $cart=new Cart();
  $cart->remove($id);
  header("location:view_cart.php");
?>
```

🔤 TITLE ##

remove.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

View cart.

🌐 LANGUAGE ######################################

Php

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=remove.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=

Index

💻 SNIPPET ##

```
<?php
     include "config.php";
     session_start();

if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !==
true){
    header("location: user_login.php");
    exit;
}

     include "cart.class.php";
     $cart=new Cart();

     $data=[];
```

```
     $sql="select * from PRODUCTS";

     $res=$db->query($sql);

     if($res->num_rows>0){

          while($row=$res->fetch_assoc()){

               $data[]=$row;
          }
     }
?>
<html>
     <head>
        <title>Products</title>
          <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootst
rap.min.css">
    </head>
    <body>
          <?php include "navbar.php"; ?>
        <div class='container mt-5 pb-5'>
               <h2 class='text-muted mb-4
text-center'>Products</h2>
               <div class='row'>
                    <?php foreach($data as $row): ?>
                         <div class='col-md-3 mt-2'>
                              <div class="card">
                                <img class="card-img-top"
src="images/<?php echo $row["image"]; ?>" >
                                <div class="card-body">
                                   <h5 class="card-title"><?php
echo $row["name"]; ?></h5>
                                   <p class="card-text">
                                        Price &#36; <?php echo
$row["price"]; ?>
                                   </p>
                                   <a
href="view_details.php?id=<?php echo $row["ProductID"]; ?>"
class='btn btn-primary  float-right' >View Details</a>
                                </div>
```

```
                                </div>
                            </div>
                        <?php endforeach; ?>
                    </div>
                </div>
    </body>
</html>

🔤 TITLE ##########################################

index.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Displays a page that displays all of the products that are not
part of a product group.

🌐 LANGUAGE #######################################

Php

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/j
upiter/filemanager/showfile.html?file=index.php&fileop=&dir=%2Fh
ome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+I
hekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&
baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.mi
n.css">
```

Index Searched

```
💻 SNIPPET ########################################

<?php
     include "config.php";
     session_start();

if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !==
```

```
true){
    header("location: user_login.php");
    exit;
}

     include "cart.class.php";
     $cart=new Cart();

     $data=[];

     $searchterm=trim($_POST["searchterm"]);

     $sql="select * from PRODUCTS where name like
'%".$searchterm."%'";

     $res=$db->query($sql);

     $count = $res->num_rows;

     if($res->num_rows>0){

          while($row=$res->fetch_assoc()){

               $data[]=$row;
          }
     }
?>
<html>
     <head>
        <title>Products</title>
          <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootst
rap.min.css">
    </head>
    <body>
          <?php include "navbar.php"; ?>
        <div class='container mt-5 pb-5'>
               <h2 class='text-muted mb-4
text-center'>Products</h2>
               <div class='row'>
                    <?php foreach($data as $row): ?>
```

```
                        <div class='col-md-3 mt-2'>
                             <div class="card">
                               <img class="card-img-top"
src="images/<?php echo $row["image"]; ?>" >
                               <div class="card-body">
                                  <h5 class="card-title"><?php
echo $row["name"]; ?></h5>
                                  <p class="card-text">
                                       Price &#36; <?php echo
$row["price"]; ?>
                                  </p>
                                  <a
href="view_details.php?id=<?php echo $row["ProductID"]; ?>"
class='btn btn-primary  float-right' >View Details</a>
                               </div>
                             </div>
                        </div>
                    <?php endforeach; ?>
                </div>
            </div>
    </body>
</html>
}

🔤 TITLE ########################################

index_search.php - cPanel File Manager v3

📝 DESCRIPTION ##################################

Displays a list of all products that match the given searchterm.

🌐 LANGUAGE #####################################

Php

🔗 RELATED LINKS ################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/j
upiter/filemanager/showfile.html?file=index_search.php&fileop=&d
ir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+C
```

```
assie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
```

Navbar

```
💻 SNIPPET ##########################################

<nav class="navbar navbar-expand-lg navbar-dark bg-info ">
<div class='container'>
  <a class="navbar-brand" href="index.php">Cookie Run Sweets Shop</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav ml-auto">

      <form action="index_search.php" method="post">
      <input name="searchterm" type="text" size="40">
      <input type="submit" name="submit" value="Search">
      </form>

      <a class="nav-item nav-link" href="index.php">Products</a>
      <a class="nav-item nav-link" href="view_cart.php">Your Cart (<?php echo $cart->get_cart_count();?>)</a>
      <a class="nav-item nav-link" href="your_orders.html">Cancel Order</a>
      <a class="nav-item nav-link" href="your_receipts.html">Order Lookup</a>
      <a class="nav-item nav-link" href="user_logout.php">Logout</a>
    </div>
  </div>
  </div>
</nav>
```

🔤 TITLE ##

navbar.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Displays a navbar with a list of all products in the cart.

🌐 LANGUAGE ######################################

Php

🏷️ TAGS ##

BOOT, TWITTER-BOOTSTRAP, NAVBAR, ANGULARJS, ANGULAR

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=navbar.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=

Delete Order

💻 SNIPPET ##

```
<?php

  // create short variable names
  $searchtype=$_POST['searchtype'];
  $searchterm=$_POST['searchterm'];


  if (!$searchtype || !$searchterm) {
     echo "You have not entered the required detail.<br />"
          ."Please go back and try again.";
          echo "<br><a href = 'your_orders.html'>Back to Order
```

```
Delete</a>";
     exit;
  }

  $searchtype = addslashes($searchtype);
  $searchterm = addslashes($searchterm);

$servername = "localhost";
$username = "ihekwac1_cr_ss123";
$password = "exM3KYUftdJ=";

$db = new mysqli('localhost', 'ihekwac1_cr_ss123',
'exM3KYUftdJ=', 'ihekwac1_cookierunsweetsshopdb'); //chnage to
own db details


  $count = "select * from ORDERS where ".$searchtype." =
'".$searchterm."'";
  $count_res = $db->query($count);

  $query = "delete from ORDERS where $searchtype =
'".$searchterm."'";
  $result = $db->query($query);

  $num = $count_res->num_rows;

  $row = $count_res->fetch_assoc();


  if ($num === 0) {
      echo "No order(s) found...";
      echo "<br><a href = 'your_orders.html'>Back to Order
Delete</a>";
      exit;
  }

    else {

    $receipt = "delete from RECEIPT where ".$searchtype." =
'".$searchterm."'";
    $db->query($receipt);
```

```
    echo "Order(s) with Receipt(s) removed: $num.";
    echo "<br><a href = 'your_orders.html'>Back to Order
Delete</a>";
    }
$db->close();
?>

</body>
</html>

🔤 TITLE ##########################################

delete_your_order.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

<?php create a single node of type


🌐 LANGUAGE ######################################

Php

🔗 RELATED LINKS #################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/j
upiter/filemanager/showfile.html?file=delete_your_order.php&file
op=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Proj
ect+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&fi
le_charset=&baseurl=&basedir=

⛔ HIGH SEVERITY INFORMATION ######################

Generic API Key: password = "exM3KYUftdJ="
```

Order Complete

```
💻 SNIPPET ########################################
```

```
<?php
     include "config.php";
     session_start();

     include "cart.class.php";
     $cart=new Cart();

?>
<html>
     <head>
        <title>Checkout</title>
          <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootst
rap.min.css">
    </head>
    <body>
          <?php include "navbar.php"; ?>
        <div class='container mt-5'>
               <h2 class='text-muted mb-4'>Order Placed
Successfully</h2>
               <div class='row'>
                    <div class='col-md-12'>
                              <div class='alert
alert-success'>Your Order no is #<?php echo
$_GET["order_no"];?></div>

                              <br><br/>
                         </div>

                    </div>
               </div>
          </div>
    </body>
</html>

🔤 TITLE ##########################################

complete.php - cPanel File Manager v3

📝 DESCRIPTION ####################################
```

```
<?php Generate a page that displays a single unknown exception.

🌐 LANGUAGE ########################################

Php

🏷️ TAGS ############################################

CODEIGNITER, AJAX

🔗 RELATED LINKS ###################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=complete.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
```

Order Checkout

```
💻 SNIPPET ##########################################

<?php
     include "config.php";
     session_start();

     include "cart.class.php";
     $cart=new Cart();

     if(isset($_POST["submit"])){
          //from HTML
          $name=mysqli_real_escape_string($db,$_POST["name"]);

$email=mysqli_real_escape_string($db,$_POST["email"]);

$address=mysqli_real_escape_string($db,$_POST["address"]);
```

```
          #insert Order information
               $order_no=rand(10000,100000);
               $rec_no=rand(10000,100000);
               $cartid=rand(10000,100000);
               $total_amt=$cart->get_cart_total();


               $sql="insert into ORDERS (orderid, o_name,
u_email, shipAdd, total) values
('{$order_no}','{$name}','{$email}','{$address}',
'{$total_amt}')";

               $inv = "select * from PRODUCTS";
               $res = $db->query($inv);
               $row = $res->fetch_assoc();

               if($db->query($sql)){

                    $rows=[];
                    foreach($cart->get_all_items() as $item){

$rows[]="('{$oid}','{$item["id"]}','{$item["name"]}','{$item["pr
ice"]}','{$item["qty"]}','{$item["total"]}'
                              )";

                              $stock = $row['quantity'];
                              $quant = $item["qty"];

                                   //updates inverntory
                                   $upd_qty = $stock - $quant;

                                   $invupd = "update PRODUCTS set
                              quantity='".$upd_qty."' where ProductID
= " .$item["id"];
                              $db->query($invupd);
                         }


                         $sql.=implode(",",$rows);
                         $db->query($sql);
                        $cart->destroy();
```

```
header("location:complete.php?order_no={$order_no}");

                        #insert receipt Item Details
                    $sql = "insert into RECEIPT (receipt_ID,
o_id, o_name, u_email, shipAdd, total) values
('{$rec_no}','{$order_no}','{$name}','{$email}','{$address}',
'{$total_amt}')";

                    $db->query($sql);

                }

           }

?>
<html>
     <head>
        <title>Checkout</title>
          <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootst
rap.min.css">
    </head>
    <body>
          <?php include "navbar.php"; ?>
        <div class='container mt-5'>
               <h2 class='text-muted mb-4'>Delivery
Details</h2>
               <div class='row'>
                    <div class='col-md-6 mx-auto'>
                         <form method='post' action='<?php echo
$_SERVER["REQUEST_URI"];?>' autocomplete="off">
                              <div class='form-group'>
                                   <label>Name (First and
Last)</label>
                                   <input type='text'
name='name' class='form-control' required placeholder='Name'>
                              </div>
                              <div class='form-group'>
                                   <label>Email</label>
```

```
                                    <input type='email'
name='email' pattern= ".+@[a-z0-9.-]+\.[a-z]{4}$"
class='form-control' required placeholder='User Email'>
                                </div>

                                <div class='form-group'>
                                    <label>Address</label>
                                    <input type='text'
name='address' pattern="[0-9 ]{9999}+[A-Za-z ]"
class='form-control' required placeholder='Full Address'>
                                </div>
                                <input type='submit'
name='submit' value='Checkout' class='btn btn-primary'>
                            </form>
                        </div>
                    </div>
                </div>
    </body>
</html>

🔤 TITLE ##########################################

checkout.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

This function is used to insert a new order or a new receipt
item in the database.

🌐 LANGUAGE #######################################

Php

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/j
upiter/filemanager/showfile.html?file=checkout.php&fileop=&dir=%
2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassi
e+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charse
t=&baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.mi
```

```
n.css">
```

Cart Class

```
💻 SNIPPET ##########################################

<?php
if(!session_id()){
     session_start();
}


     class Cart{

          protected $cart_items=array();

          public function __construct(){

               if(!isset($_SESSION["cart"])){
                    $_SESSION["cart"]=[];
               }

               $this->cart_items=$_SESSION["cart"];
          }

          #get all ids of items
          public function get_ids(){
               $ids=[];
               foreach($this->cart_items as $item){
                    $ids[]=$item["id"];
               }
               return $ids;
          }


          #add item to cart
          public function add_to_cart($item=[]){

if(isset($item["id"],$item["name"],$item["price"],$item["qty"],$
item["total"])){
```

```
                    #Check Product already exists
                    $ids=$this->get_ids();
                    if(in_array($item["id"],$ids)){

                         #update qty and total

$this->cart_items[$item["id"]]["qty"]+=$item["qty"];

$this->cart_items[$item["id"]]["total"]=$this->cart_items[$item[
"id"]]["qty"]*$item["price"];
                    }else{

                         #add item to cart
                         $this->cart_items[$item["id"]]=$item;
                    }

                    $this->commit();
                    return true;
               }else{
                    return false;
               }
          }

          #update cart qty
          public function update_cart($item=[]){

$this->cart_items[$item["id"]]["qty"]=$item["qty"];

$this->cart_items[$item["id"]]["total"]=$this->cart_items[$item[
"id"]]["qty"]*$this->cart_items[$item["id"]]["price"];
               $this->commit();
               return true;
          }

          #remove item from cart
          public function remove($id){
               unset($this->cart_items[$id]);
               $this->commit();
          }
```

```
          #get cart total
          public function get_cart_total(){
               #update cart total
               $sum=0;
               foreach($this->cart_items as $item){
                    $sum+=$item["total"];
               }
               return $sum;
          }

          #get cart item count
          public function get_cart_count(){
               return count($this->cart_items);
          }

          #update values to session
          public function commit(){
               $_SESSION["cart"]=$this->cart_items;
          }

          #destroy cart
          public function destroy(){
               $this->cart_contents = array('cart_total' => 0,
'cart_items_count' => 0);
               unset($_SESSION["cart"]);
          }

          #get single item from cart
          public function get_item($id){
               return $this->cart_items[$id];
          }

          #get single item from cart
          public function get_item_qty(){
               $sum=0;
               foreach($this->cart_items[$id] as $itemqty){
                    $sum+=$item["qty"];
               }
               return $sum;
          }
```

```
          #get all items from cart
          public function get_all_items(){
               return $this->cart_items;
          }
     }

/*
Functions in Cart Library:

     add_to_cart()  - Add/Update Item to Cart.
     remove()       - Remove Specific item from Cart.
     get_item()     - Get Single items from Cart.
     get_all_items()- Get all items from Cart.
     get_cart_total() - Get Cart total Amount.
     get_cart_count() - Get Cart items count.
     destroy()       - Remove all items from Cart.
 */
?>

🔤 TITLE ##########################################

cart.class.php - cPanel File Manager v3

📝 DESCRIPTION ####################################

Constructor for Cart object

🌐 LANGUAGE #######################################

Php

🏷️ TAGS ###########################################

#ADD, #ADD

🔗 RELATED LINKS ##################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=cart.class.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cas
```

```
sie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_char
set=&baseurl=&basedir=
```

Update Cart

```
💻 SNIPPET ##########################################

<?php
  session_start();

  $id=$_POST["id"];
  $qty=$_POST["qty"];

  include "cart.class.php";
  $cart=new Cart();

  $cart->update_cart(["id"=>$id,"qty"=>$qty]);

  $result=[
     "row_total"=>$cart->get_item($id)["total"],
     "total"=>$cart->get_cart_total(),
  ];
  echo json_encode($result);

?>

🔤 TITLE ############################################

ajax_update_cart.php - cPanel File Manager v3

📝 DESCRIPTION ######################################

<?php show a list of items in the cart

🌐 LANGUAGE #########################################

Php

🔗 RELATED LINKS ####################################
```

```
https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=ajax_update_cart.php&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=
```

User Order Receipt Search

```
💻 SNIPPET ########################################

<html> <head> <title>Orders and Receipts</title> <link
rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"> </head> <body> <h1>Order Search</h1>
<p><a href = "index.php">Back to Main Page</a> |
<a href = "your_orders.html">Cancel Order</a> |
<a href = "user_logout.php">Logout</a></p><hr>

<form action="user_receipts.php" method="post"> Enter an Email
to find your <strong>Order no. and receipt</strong> to delete an
<strong>individual order</strong>:
<br /> <br />
Or to delete <strong>multiple orders</strong>, enter
<strong>email</strong> associated with order(s) <br><br/> on
<strong>"<a href = "your_orders.html">Cancel
Order</a>"</strong>: Page:
<br><br/>
<h4>Manual Search</h4> Enter Email Infomation:
<br />
<input name="searchterm" type="text" size="40"> <br /> <input
type="submit" name="submit" value="Search"> </form> </body>
</html>

🔤 TITLE ##########################################

your_receipts.html - cPanel File Manager v3

📝 DESCRIPTION ####################################

A simple example of how to search for orders and receipts.
```

🌐 LANGUAGE ##

Html

🔗 RELATED LINKS ####################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/jupiter/filemanager/showfile.html?file=your_receipts.html&fileop=&dir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+Cassie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_charset=&baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

User Order Cancel Form

💻 SNIPPET ##

```
<html> <head> <title>Order Cancel</title> <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootst
rap.min.css"> </head> <body> <h1>Order Cancellation</h1>

<p><a href = "index.php">Back to Main Page</a> |
<a href = "your_receipts.html">Order Lookup</a> |
<a href = "user_logout.php">Logout</a></p><hr>
<br><br>
<div class="container mt-3 w-25">
<form action="delete_your_order.php" method = "post">
Note to delete any order - you will need the <strong>Order
ID</strong> from your <strong>
<a href = "your_receipts.html">Order Lookup</a></strong> page.
If you choose <strong>Email</strong>, you will be deleting
<strong>every order under that Email.</strong> <br><br/> If you
choose <strong>Order ID</strong>, you will be deleting
<strong>individual Orders.</strong>
<br><br/>
<h5>Choose Search Type:</h5>
<select name="searchtype">
<option value="none" selected disabled hidden>Select an
```

```
Option</option>
<option value="u_email">Email
<option value="o_id">Order ID </select>
<br><br/>
<h5>Manual Search</h5> Enter ID Term:<br /> <input
name="searchterm" type="text" size="40"> <br /> <button
type="submit" class="btn btn-primary">Cancel Order(s)</button>
</form> </div> <br><br> </body> </html>

🔤 TITLE ##########################################

your_orders.html - cPanel File Manager v3

📝 DESCRIPTION ####################################

This page shows a form to cancel an order.

🌐 LANGUAGE ######################################

Html

🔗 RELATED LINKS #################################

https://cyan.csam.montclair.edu:2083/cpsess0125053364/frontend/j
upiter/filemanager/showfile.html?file=your_orders.html&fileop=&d
ir=%2Fhome%2Fihekwac1%2Fpublic_html%2FDatabase+Systems+Project+C
assie+Ihekwaba%3A+Cookie+Run+Sweets+Shop&dirop=&charset=&file_ch
arset=&baseurl=&basedir=
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.mi
n.css">
```

Database Dump:

```
-- phpMyAdmin SQL Dump
-- version 4.9.7
-- https://www.phpmyadmin.net/
--
-- Host: localhost:3306
-- Generation Time: Dec 19, 2022 at 08:55 AM
-- Server version: 5.7.40
```

```
-- PHP Version: 7.4.33

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
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: `ihekwac1_cookierunsweetsshopdb`
--

-- --------------------------------------------------------

--
-- Table structure for table `ADMINISTRATORS`
--

CREATE TABLE `ADMINISTRATORS` (
  `adminID` int(7) NOT NULL,
  `email` char(255) NOT NULL,
  `adminpass` varchar(255) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `ADMINISTRATORS`
--

INSERT INTO `ADMINISTRATORS` (`adminID`, `email`, `adminpass`) VALUES
(1, 'admin01@sweetshop.admin',
'$2a$10$LooGMzBXtMUP3QrnrPnxkO1EwzHP1kaC21j/qtmWESX1sLbXtn2Ty'),
(2, 'admin02@sweetshop.admin',
'$2a$10$YPrgeTLXvizVnxvVF/j4F.Um2SHAqZIqYi9Y5uuUVLtefojrJGxnO');

-- --------------------------------------------------------

--
-- Table structure for table `CUSTOMER`
```

```
--

CREATE TABLE `CUSTOMER` (
  `AccountID` int(10) NOT NULL,
  `useremail` varchar(255) NOT NULL,
  `userpass` varchar(255) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `CUSTOMER`
--

INSERT INTO `CUSTOMER` (`AccountID`, `useremail`, `userpass`) VALUES
(1, 'nayohminty@sweetshop.bake',
'$2y$10$WBxVegtQahff0Q375sMkgeDDL7ItQmsPgSTq52BQeYSc.tT0UF.gC'),
(17, 'cquaba@sweetshop.bake',
'$2y$10$4zfbZ1nb5FYyUuRn0c5byOSOguexlELqDwkvFzHNODEwQIcz4WhUq'),
(16, 'cassquab@sweetshop.bake',
'$2y$10$6fAC/I3ZiJsgnaQMkr6cquBg5bL4.JoysNZDmcS0j6tIFsgR.2LqO');

-- --------------------------------------------------------

--
-- Table structure for table `ORDERS`
--

CREATE TABLE `ORDERS` (
  `orderid` int(6) NOT NULL,
  `o_name` varchar(255) DEFAULT NULL,
  `u_email` varchar(255) DEFAULT NULL,
  `shipAdd` varchar(255) DEFAULT NULL,
  `total` double(10,2) DEFAULT '0.00'
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `ORDERS`
--

INSERT INTO `ORDERS` (`orderid`, `o_name`, `u_email`, `shipAdd`, `total`)
VALUES
(63976, 'Naomi Mintii', 'nayohminty@sweetshop.bake', '2 Abnormal Avenue,
Schmanton Hall, SH-8876, Schmontair University, Schmontair, NJ, 99086',
15.99),
```

```
(22779, 'Naomi Mintii', 'nayohminty@sweetshop.bake', '2 Abnormal Avenue,
Schmanton Hall, SH-8876, Schmontair University, Schmontair, NJ, 99086',
17.97),
(41632, 'Cassie Ihekwaba', 'cassquab@sweetshop.bake', '2 Abnormal Avenue,
Schmanton Hall, SH-4567, Schmontair University, Schmontair, NJ, 99086',
61.98),
(63425, 'Naomi Mintii', 'nayohminty@sweetshop.bake', '2 Abnormal Avenue,
Schmanton Hall, SH-8876, Schmontair University, Schmontair, NJ, 99086',
17.97),
(29615, 'Cassie Ihekwaba', 'cassquab@sweetshop.bake', '2 Abnormal Avenue,
Schmanton Hall, SH-4567, Schmontair University, Schmontair, NJ, 99086',
15.99),
(13661, 'Cassie Ihekwaba', 'cassquab@sweetshop.bake', '2 Abnormal Avenue,
Schmanton Hall, SH-4567, Schmontair University, Schmontair, NJ, 99086',
33.96);

-- --------------------------------------------------------

--
-- Table structure for table `PRODUCTS`
--

CREATE TABLE `PRODUCTS` (
  `ProductID` int(6) NOT NULL,
  `name` varchar(255) NOT NULL DEFAULT '',
  `details` text,
  `price` double(10,2) NOT NULL DEFAULT '0.00',
  `quantity` int(10) NOT NULL DEFAULT '0',
  `image` varchar(100) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `PRODUCTS`
--

INSERT INTO `PRODUCTS` (`ProductID`, `name`, `details`, `price`, `quantity`,
`image`) VALUES
(1, 'Ginger Brave Cookie Brave Enamel Keychain', 'Have the bravest cookie in
your kingdom guard your belongings with this snazzy acrylic matte finish
charm!', 5.99, 987, '01.jpg'),
(2, 'Strawberry Crepe Cookie Acrylic Keychain', 'Smart AND cute?! How could
anyone resist? Be the talk of the oven with this keycharm!', 5.99, 999,
'12.jpg'),
```

(3, 'Strawberry Cookie Acrylic Keychain', 'A bit shy...but she will appreciate the purchase!', 5.99, 984, '04.jpg'),
(4, 'Wizard Cookie Acrylic Keychain', 'A cookie that can do magic?! Wonder if thats why you love this keychain enough to (hopefully) buy it!', 5.99, 977, '05.jpg'),
(5, 'Princess Cookie Acrylic Keychain', 'A cookie with a big heart (and stomach)! Royalty like her should always be on display with a charm like this!', 5.99, 999, '28.jpg'),
(6, 'Gumball Cookie Acrylic Keychain', 'Do not disturb an artist that is at work! With his eyes, perharps you can become as inspired as him!', 5.99, 984, '10.jpg'),
(7, 'Alchemist Cookie Acrylic Keychain', 'HANDLE WITH CARE! Those chemicals she has are incredibly volatile! But some stickers should be fine...right?', 5.99, 984, '15.jpg'),
(8, 'Cherry Cookie Acrylic Keychain', 'Haha KaBOOM! A little pyrotechnic with more than a few big surprises up her sleeve - now as a keycharm!', 5.99, 1000, '21.jpg'),
(9, 'Avocado Cookie Acrylic Keychain', 'A Punny Cookie with jokes to crack! Wear this and people will really get your CHARM - get it? Funny right?', 5.99, 1000, '17.jpg'),
(10, 'Clover Cookie Acrylic Keychain', 'A bard with beautiful music to play to the cookies and forest creatures - the melody they play can soothe any soul.', 5.99, 1000, '08.jpg'),
(11, 'Black Pearl Cookie T-Shirt', 'Whatever you do - avoid wearing this to the beach...the sea gets a bit restless for some reason...', 15.99, 986, '14.jpg'),
(12, 'Afogatto Cookie Color-Block Long Sleeve Shirt', 'A cookie born from two desserts, now on a color-blocked T-shirt! How clever - and kinda cool, too!', 15.99, 998, '06.jpg'),
(13, 'Raspberry Cookie T-Shirt', 'A skilled swordsman with pride for her House - one with such pride must dress as such, no?', 15.99, 1000, '26.jpg'),
(14, 'Tiger Lily Cookie T-Shirt', 'A cookie of the forest - illusive, but deadly! Having a shirt of her is an interesting way to be scared though...', 15.99, 1000, '24.jpg'),
(15, 'Kumiho Cookie T-Shirt', 'A Fox that wants to be a Cookie - her wish granted, though only for a little while...a rare sight to see, to be on a shirt? An interesting choice to immortalize a sighting...', 15.99, 1000, '23.jpg'),
(16, 'Chili Pepper Cookie Tank Top', 'Quite the spicy cookie with a shirt to vent the hea - wait, WHERE DID MY WALLET GO?!', 15.99, 986, '13.jpg'),
(17, 'Rye Cookie T-Shirt', 'YEEEEHAWWWW!!! BANG BANG - a gunslinging bounty hunter needs some style for her fans, right? Just try to avoid letting Chili Pepper know about this, okay?', 15.99, 995, '25.jpg'),

```
(18, 'Mont Blanc Cookie Designer Mellow Tone Sweater', 'A seamstress of high regard making clothes for OUR store?! I might faint - but this is limited stock so I gotta grab one before they are all gone!', 30.99, 985, '30.jpg'),
(19, 'Ninja Cookie T-Shirt', 'Transcendence! Focus! Enter your age of discipline with shirt to motivate your drive!', 15.99, 1000, '03.jpg'),
(20, 'Macaron Cookie Holiday Sweater', 'Christmas time is here! What is some caroling with Carol Cookie without some snare drum pep in your step?', 15.99, 1000, '19.jpg'),
(21, 'Latte Cookie Plushie', 'There is no right way of doing things - just go with the flow! Everyone needs encouragement, so have a Latte Cookie plush to help!', 25.99, 1000, '16.jpg'),
(22, 'Tea Knight Cookie Plushie', 'A cookie to keep those nasty cake hounds away - with a Halbred and...a faint tea scent as well...Grab this plushie while you can!', 25.99, 984, '11.jpg'),
(23, 'Blackberry Cookie Plushie', 'A cookie with a...strange set of etehreal helpers...', 25.99, 1000, '18.jpg'),
(24, 'Angel Cookie Plushie', 'Heaven is real! Especially if a plushie as soft and as cute as Angel Cookie exists!', 25.99, 1000, '02.jpg'),
(25, 'Adventurer Cookie Plushie', 'Always on the move for the next lead to ancient treasures of fallen civilizations - Have your own adventures with him at your side!', 25.99, 1000, '07.jpg'),
(26, 'Onion Cookie Plushie', 'A cookie that always seems to be crying - but around Blackberry Cookie she finds calm...', 25.99, 1000, '29.jpg'),
(27, 'Espresso Cookie Plushie', 'Precision is key - especially with an wizard of his calibur...and roast type!', 25.99, 986, '09.jpg'),
(28, 'Hollyberry Cookie Plushie', 'Protecting her kingdom until it falls! The heart of Hollyberry Cookie is noting to sneeze at!', 25.99, 1000, '22.jpg'),
(29, 'Carrot Cookie Plushie', 'Work for your share! This farmer accepts no slacking in the field - especially when it comes to dinner!', 25.99, 1000, '20.jpg'),
(30, 'Pumpkin Pie Cookie Plushie', 'Out of season? Possibly, but a Halloween Cookie like herself should not have to wait to shine! Strange...but cute!', 25.99, 1000, '27.jpg');

-- --------------------------------------------------------

--
-- Table structure for table `RECEIPT`
--

CREATE TABLE `RECEIPT` (
  `receipt_ID` int(10) NOT NULL,
  `o_id` int(10) DEFAULT NULL,
```

```
  `o_name` varchar(100) NOT NULL,
  `u_email` varchar(255) DEFAULT NULL,
  `shipAdd` varchar(255) DEFAULT NULL,
  `total` double(10,2) DEFAULT '0.00'
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `RECEIPT`
--

INSERT INTO `RECEIPT` (`receipt_ID`, `o_id`, `o_name`, `u_email`, `shipAdd`,
`total`) VALUES
(19372, 89311, 'Jassandra B Bhekwaqa', 'cquaba@sweetshop.bake', '2 Abnormal
Avenue, Schmanton Hall, SH-2456, Schmontair University, Schmontair, NJ,
99086', 25.99),
(63378, 63976, 'Naomi Mintii', 'nayohminty@sweetshop.bake', '2 Abnormal
Avenue, Schmanton Hall, SH-8876, Schmontair University, Schmontair, NJ,
99086', 15.99),
(65325, 22779, 'Naomi Mintii', 'nayohminty@sweetshop.bake', '2 Abnormal
Avenue, Schmanton Hall, SH-8876, Schmontair University, Schmontair, NJ,
99086', 17.97),
(28411, 41632, 'Cassie Ihekwaba', 'cassquab@sweetshop.bake', '2 Abnormal
Avenue, Schmanton Hall, SH-4567, Schmontair University, Schmontair, NJ,
99086', 61.98),
(70598, 63425, 'Naomi Mintii', 'nayohminty@sweetshop.bake', '2 Abnormal
Avenue, Schmanton Hall, SH-8876, Schmontair University, Schmontair, NJ,
99086', 17.97),
(65114, 29615, 'Cassie Ihekwaba', 'cassquab@sweetshop.bake', '2 Abnormal
Avenue, Schmanton Hall, SH-4567, Schmontair University, Schmontair, NJ,
99086', 15.99),
(43807, 13661, 'Cassie Ihekwaba', 'cassquab@sweetshop.bake', '2 Abnormal
Avenue, Schmanton Hall, SH-4567, Schmontair University, Schmontair, NJ,
99086', 33.96);

--
-- Indexes for dumped tables
--

--
-- Indexes for table `ADMINISTRATORS`
--
ALTER TABLE `ADMINISTRATORS`
  ADD PRIMARY KEY (`adminID`);
```

```
--
-- Indexes for table `CUSTOMER`
--
ALTER TABLE `CUSTOMER`
  ADD PRIMARY KEY (`AccountID`),
  ADD KEY `useremail` (`useremail`);

--
-- Indexes for table `ORDERS`
--
ALTER TABLE `ORDERS`
  ADD PRIMARY KEY (`orderid`);

--
-- Indexes for table `PRODUCTS`
--
ALTER TABLE `PRODUCTS`
  ADD PRIMARY KEY (`ProductID`);

--
-- Indexes for table `RECEIPT`
--
ALTER TABLE `RECEIPT`
  ADD PRIMARY KEY (`receipt_ID`),
  ADD KEY `o_id` (`o_id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `ADMINISTRATORS`
--
ALTER TABLE `ADMINISTRATORS`
  MODIFY `adminID` int(7) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--
-- AUTO_INCREMENT for table `CUSTOMER`
--
ALTER TABLE `CUSTOMER`
  MODIFY `AccountID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=18;
```

```
--
-- AUTO_INCREMENT for table `ORDERS`
--
ALTER TABLE `ORDERS`
  MODIFY `orderid` int(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=99500;

--
-- AUTO_INCREMENT for table `PRODUCTS`
--
ALTER TABLE `PRODUCTS`
  MODIFY `ProductID` int(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=70;

--
-- AUTO_INCREMENT for table `RECEIPT`
--
ALTER TABLE `RECEIPT`
  MODIFY `receipt_ID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=99020;
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 */;
```