Final Project Documentation

Name: Dan Imbimbo
Email: imbimbod2@montclair.edu

USERNAMES & PASSWORDS

-Customers

U: test@gmail.com P: password
U: thisisatest@gmail.com P: test123

-Employees

U: mmackey@gmail.com P: mathguru
U: jwickens@gmail.com P: ezpass

-CEO

U: fdecker@gmail.com P: bossman

DOCUMENTATION

Project Requirements

-What are you developing?

For this final project I have developed an ecommerce storefront based around a bakery that specializes in desserts. The bakery sells an assortment of sweets, under the specific categories of brownies, cakes, cookies, donuts, and puddings. This fictitious business has had their daily sales negatively impacted by the ongoing COVID-19 pandemic, and as such have been looking to expand their business beyond solely selling products in-person. Their catalogue of goods all fall under the categories listed above, each with varying flavors to fit the differing tastes of their customers.

The expected clientele for this website are those with a sweet tooth who happen to be in pursuit of freshly-baked desserts. for whatever the occasion may be. These users will be met with an inviting user interface that allows them to seamlessly sort and search through whichever products they may desire, before then placing their order with relative ease.

-What access your various users have to your product:

The website provides different access for customers, employees, as well as the CEO. The landing page opens by greeting users with a login screen, where they are able to create a customer account or login with their existing account (which will determine if they are an employee or a customer).

When customers log in they are presented with a page that contains a brief description of the business and a navigation bar that provides various functions, such as: access to the entire product list (which is into categories and allows for the ability to search through those items based on entered terms); their current cart (containing the quantities and totals of any items they have placed inside, as well as the ability to complete their order); an account settings page (where they can edit the details associated with their account, change their password, as well view their previous order history and the specifics of each individual order); and the ability to log out of their account.

When employees log in they are presented with a page that displays the list of current employees (along with their employee IDs, department IDs, whether they are the manager of their department, their names, and their phone numbers). The navigation bar provided also gives them access to the following: a page containing the list of products on the database (including their individual product IDs, category IDs, product names, prices, image locations, and their available stock); a page showcasing the orders placed by customers (including their order IDs, order dates, order totals, and

the emails identifying the customers who placed them); and the ability to log out of their account.

As mentioned above, there is also a third group of access reserved for the owner of the business. They have a view similar to that of the employees, but with the following alterations: on the employee list page they also have the option to add new employees, fire current ones, or make an employee the manager of their department; on the product page they have the option to add new products, change the price of existing products, or alter a product's stock; and on the order history page they can remove any placed orders.

-Where did you get the product from?

The products that I have chosen for each of the specified categories on the website were primarily decided based on those that I have eaten or have encountered beforehand. That said, for the categories of which I could not entirely come up with the necessary amount of 6 products, I simply searched through google images for, "*insert category* flavors," which would provide group shots of various different flavors; at which point I would simply pick those that seemed appealing. Once I settled on the various flavors for each category, I then began to search for the flavors individually so that I could find images that would properly show the differences among them. Those images for each flavor of each category were all sourced from the following:

1. **Brownies**
   Blondie:
   https://www.fifteenspatulas.com/featured-friday-brown-butter-toffee-blondies/
   Cheesecake:
   https://diycrafts9.com/a-simple-guideline-to-a-delicious-cheesecake-brownie/
   Cookies and cream:
   https://cantstayoutofthekitchen.com/2014/02/08/cookies-n-creme-brownies/
   Fudge: https://www.rockrecipes.com/fudge-brownies/
   Marshmallow: https://www.rockrecipes.com/toasted-marshmallow-brownies/
   Mint: https://www.shugarysweets.com/mint-chocolate-chip-brownies/
2. **Cakes**
   Angel food: https://www.pinterest.com/pin/329818372715096374/
   Carrot: https://www.punchfork.com/recipe/Carrot-Cake-Spoon-Fork-Bacon
   Crumb: https://gimmedelicious.com/new-york-style-crumb-cake-2/
   Pineapple upside-down:
   https://lifemadesimplebakes.com/pineapple-upside-down-cake/
   Pound: https://www.onceuponachef.com/recipes/perfect-pound-cake.html
   Tiramisu: https://bakingamoment.com/tiramisu-cake/

Fruit *for example*: https://www.rockrecipes.com/apricot-fruitcake/

3. **Cookies**
   Chocolate chip:
   https://www.yummly.com/recipe/Classic-Chocolate-Chip-Cookies-2648385
   Gingerbread: https://www.smalltownwoman.com/soft-gingerbread-cookies/
   Macaroons:
   https://feelslikehomeblog.com/coconut-macaroons-recipe-gluten-free/
   Oatmeal raisin: https://www.persnicketyplates.com/best-oatmeal-raisin-cookies/
   Peanut butter:
   https://cookbookfundraiser.com/recipe/4750654/peanut-butter-cookies.html
   Shortbread:
   https://thedomesticrebel.com/2020/12/09/buttery-whipped-shortbread-cookies/
4. **Donuts**
   Apple cider: https://www.pinterest.de/pin/584764951506395l8/
   Boston cream: https://www.julieseatsandtreats.com/boston-cream-donuts/
   Chocolate frosted: https://www.rockrecipes.com/donuts-glazed-filled/
   French cruller: https://reschbakery.com/Columbus%20Bakery/french-cruller/
   Glazed: https://natashaskitchen.com/glazed-donuts-recipe/
   Strawberry sprinkled:
   https://www.yummyhealthyeasy.com/baked-funfetti-donuts-birthday-donuts/
   Jelly-filled *for the demo*:
   https://www.tasteofhome.com/recipes/jolly-jelly-doughnuts/
5. **Puddings**
   Banana: https://www.spoonforkbacon.com/banana-pudding/
   Caramel apple:
   https://www.punchfork.com/recipe/Caramel-Apple-Pudding-Delish
   Chocolate raspberry: https://www.natalieshealth.com/raspberry-chia-pudding/
   Red velvet: https://www.tastesoflizzyt.com/homemade-red-velvet-pudding/
   Rice: https://tiger-corporation-us.com/rice-cooker-rice-pudding/
   Vanilla: https://www.melskitchencafe.com/the-best-vanilla-pudding/

I also took the following image, which I slightly photoshopped for my website, to display the inside of my imaginary bakery: https://www.pinterest.com/pin/231161393344566795/

-What resources did you model your site after?

For the most part, I modeled my website after the previous storefronts I have made, during the spring of 2018 at Sussex County Community College in my “Database Management Systems” course (which was not actually a website, but done through MS Access) and the spring of 2021 in my “Internet Computing” course. When developing

the website for Internet Computing last spring, I referred to “Murach’s PHP and mySQL” 2nd edition textbook by Joel Murach and Ray Harris, specifically the chapter 5 "guitar shop” example & the chapter 11 “cart” example; which assisted in developing my product list and cart functionalities, respectively. Beyond those references, for this website I also referred to the chapter 11 examples from the “PHP and MySQL Web Development” 4th edition textbook by Luke Welling and Laura Thomson, which assisted in developing my site’s product search functionality and for connecting to my database via the mysqli class in PHP.

I suppose I also referred to other existing ecommerce websites namely walmart.com, amazon.com, and target.com (although the influence these websites had on the final product was ultimately rather miniscule).

Problem Analysis - Assumptions

As stated in the previous section, this website is a storefront with varying views provided for different users, so it was assumed that the database necessitated tables for the Products being sold, the Categories that they are classified under, the Employees who work at the bakery, the Departments that they work for, the Customers who order the products, the Orders recording who placed the order and for how much, as well as how what and many OrderItems are bought in each order.

These assumptions were then further grouped and broken down as follows:

-**Categories**

- **Category ID (unique)**
- **Category name (unique)**
- **Has products**

Each category needs a unique category ID number for the needs of identification, they also need a unique name that broadly describes the products that it contains, and each category has products listed within it in the catalogue.

-Products

- **Product ID (unique)**
- **Product name (unique)**
- **Price**
- **Stock**
- **Product image**
- **Has a category**

Each product needs a unique product ID number to be easily identified, they also need a unique product name that describes the flavor of the product within its category, each

product also has a price that must be paid to purchase it, they also have an amount of stock available for purchase, as well as the location of the product's image file for listing, and the flavors of products will be properly sorted into categories.

-Departments

- **Department ID (unique)**
- **Department name (unique)**
- **Has employees**
- **An employee is the department manager**
- **Start date of manager**

Each department needs a unique department ID to act as its identifier, it also needs a unique name to properly describe the function of the department, each department is occupied by several employees, one of which will be the manager who oversees the department, and their start date as the manager will be kept track of for documentation purposes.

-Employees

- **Employee ID (unique)**
- **Employee email (unique)**
- **Password**
- **Employee name - first name, middle initial, last name**
- **Employee phone number**
- **Hire date**
- **Salary**
- **Works in a department**

Each employee needs a unique employee id for adequate identification & managerial purposes, as well as a unique employee email in order to log into their account, their account also needs a password to ensure its security, they need to supply a name (as well as their phone number) for employee records, their hire date is recorded for documentation purposes, their salary is recorded for payroll and accounting, and each employee works for a department that is kept track of for organizational purposes.

-Customers

- **Customer email address (unique)**
- **Password**
- **Customer name - first name, middle initial, last name**
- **Customer address - street, city, state, zip**
- **Customer phone number**
- **Has an order history**

Each customer needs a unique customer email address to identify & log into their account, their accounts need a password for security purposes, they need to supply a name for the shipping address (as well as their street, city, state and zip code), they also need to supply a phone number so the source of any contact with employees over the phone can be verified, and they will have an order history detailing the individual orders that they have placed.

-Orders

- **Order ID (unique)**
- **Order total**
- **Order date**
- **Placed by a customer**
- **Has order items**

Each order needs a unique order ID number to identify the order by, as well as an order total made up of the contents of the cart upon purchase, an order date for organizing a list of several orders by, each order is placed by an individual customer account, and they are composed of the order items that were in the cart.

-OrderItems

- **Order ID (weak entity connected to Orders)**
- **Product ID (weak entity connected to Products)**
- **Item quantity**
- **Item overall cost**

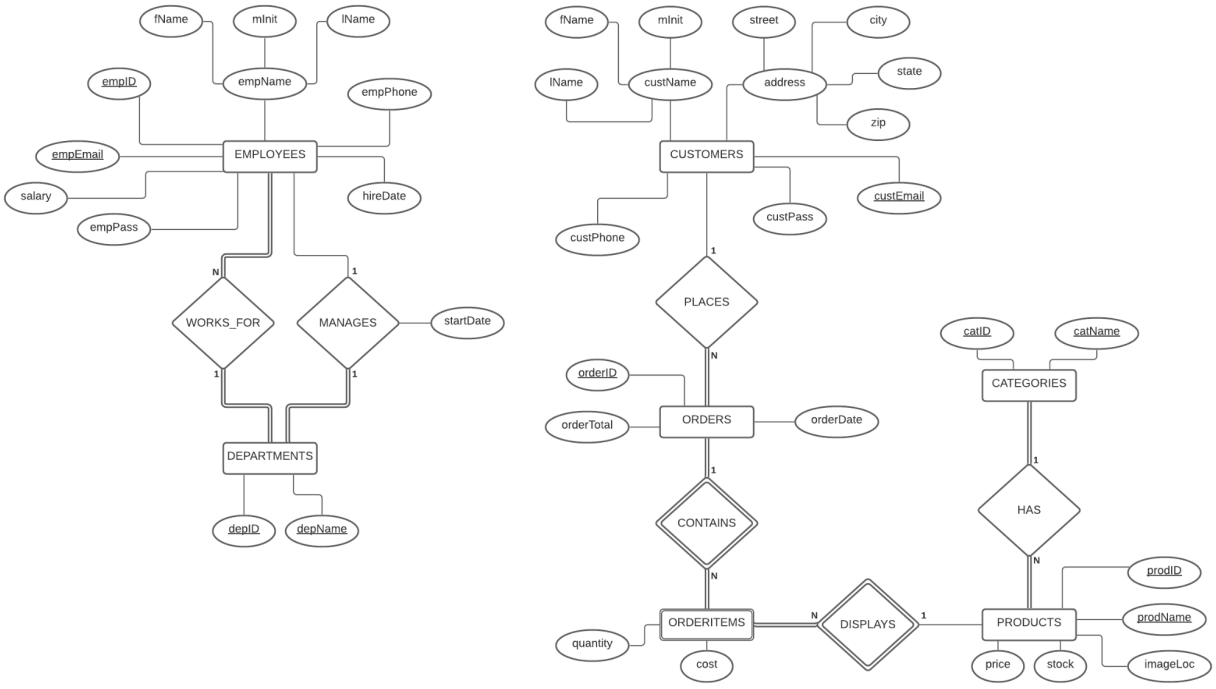
Each order item is identified by the ID of the order it is associated with and the product ID of the item it is representing, they also keep track of the quantity of the item that was in the cart upon purchase, and the total cost of the item when accounting for that quantity.

Interactions:

-Each Customer can place an Order
-Each Employee works in a Department
-Each Department has an Employee that is its manager
-Each Product has a Category
-Each Order is composed of at least one OrderItem
-Each OrderItem represents the Product that was purchased

<u>Solutions</u>

-Data Model – conceptual

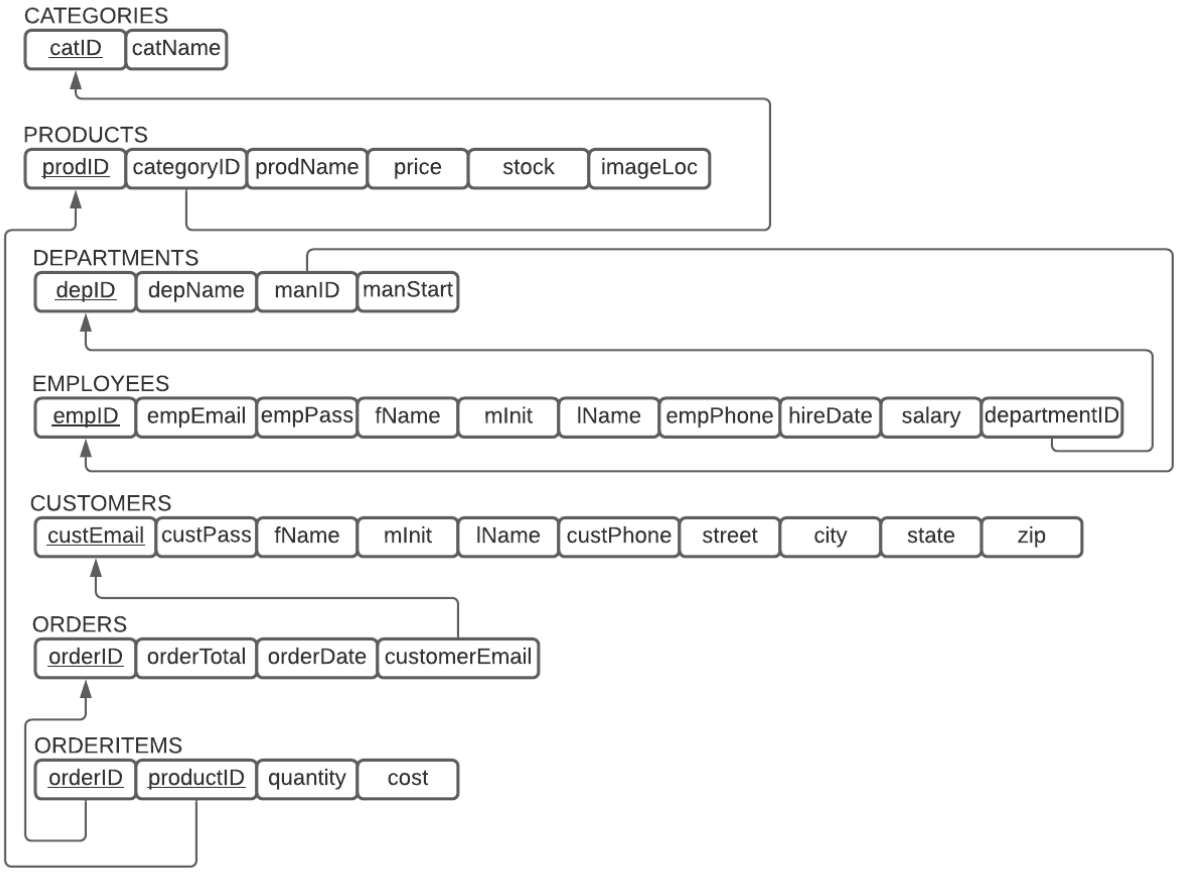


-EMPLOYEES work for DEPARTMENTS through a many-to-one relationship, which both entities must participate in.
-EMPLOYEES manage DEPARTMENTS through a one-to-one relationship, which DEPARTMENTS must participate in.
-CUSTOMERS can place ORDERS through a one-to-many relationship, which ORDERS must participate in.
-ORDERS contain ORDERITEMS through a one-to-many relationship, which both entities must participate in, with ORDERITEMS being a weak entity.
-ORDERITEMS display PRODUCTS through a many-to-one relationship, which ORDERITEMS must participate in and is a weak entity.
-PRODUCTS have CATEGORIES through a many-to-one relationship, which PRODUCTS must participate in.

Truthfully, not much of my initial ER model changed throughout the semester because I already knew how to approach a storefront design in terms of structuring the tables for SQL, which I accounted for when I made the model in the first place. The only things I changed, I believe, were: that CUSTOMERS originally had a custID to identify them by, which I determined was redundant since identification by custEmail made more sense in terms of presenting the data to the employees (basically, I didn't feel like the customer numbers were very meaningful to the employees viewing them); and that ORDERITEMS originally had an itemID as a partial key to go along with the orderID as

the composite primary key, but I removed that and made the composite primary key be composed of the orderID and productID instead.

-Data Model – implementation



-The CATEGORIES table consists of: “catID” which is its primary key of type int; and the simple attribute “catName” of type varchar.
-The PRODUCTS table consists of: “prodID” which is its primary key of type int; “categoryID” which is a foreign key referencing “catID” from CATEGORIES and is of type int, this represents the HAS relationship from the ER diagram; the simple attribute “prodName” of type varchar; the simple attribute “price” of type decimal; the simple attribute “stock” of type int; and the simple attribute “imageLoc” of type varchar.
-The DEPARTMENTS table consists of: “deptID” which is its primary key of type int; the simple attribute “deptName” of type varchar; “manID” which is a foreign key references “empID” from the EMPLOYEES and is of type int; and the simple attribute “manStart” of type datetime.
-The EMPLOYEES table consists of: “empID” which is its primary key of type int; the simple attribute “empEmail” of type varchar; the simple attribute “empPass” of type

varchar; the composite attribute “empName” from the ER diagram is broken down into the simple attributes “fName”, “mInit”, and “lName”, which are of type varchar, char, and varchar, respectively; the simple attribute “empPhone” of type varchar; the simple attribute “hireDate” of type datetime; the simple attribute “salary” of type decimal; and “departmentID” which is a foreign key referencing “deptID” from DEPARTMENTS and is of type int.
-The CUSTOMERS table consists of: “custEmail” which is its primary key of type varchar; the simple attribute “custPass” of type varchar; the composite attribute “custName” from the ER diagram is broken down into the simple attributes “fName”, “mInit”, and “lName”, which are of type varchar, char, and varchar, respectively; the simple attribute “custPhone” of type varchar; and the composite attribute “address” from the ER diagram is broken down into the simple attributes “street”, “city”, “state”, and “zip”, which are all of type varchar.
-The ORDERS table consists of: “ordID” which is its primary key of type int; the simple attribute “ordTotal” of type decimal; the simple attribute “ordDate” of type datetime; and “customerEmail” which is a foreign key referencing “custEmail” from CUSTOMERS and is of type varchar.
-The ORDERITEMS table consists of: “orderID” which is a foreign key referencing “ordID” from ORDERS and is of type int; “productID” which is a foreign key referencing “prodID” from PRODUCTS and is of type int; the simple attribute “quantity” of type int; the simple attribute “cost” of type decimal; and its primary key is a composite primary key composed of “orderID” and “productID”.

Application Development

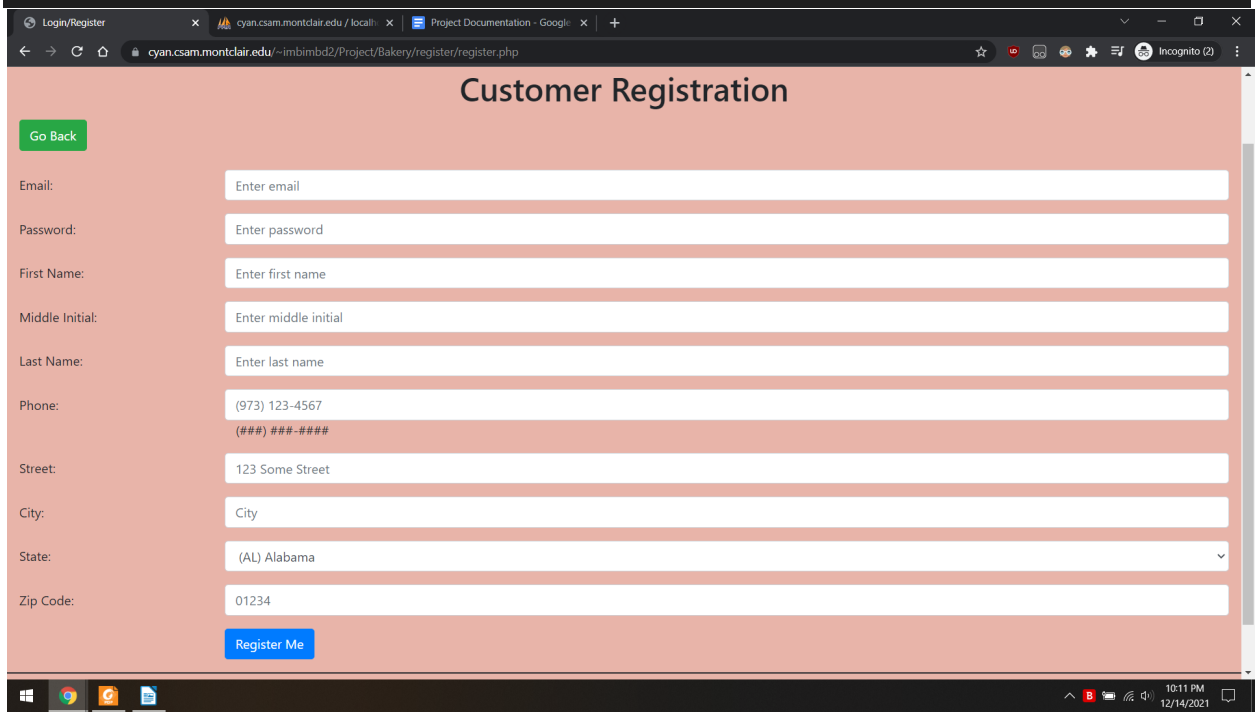
-Shopper

Creating an account:

On the login screen, the customer can click the button “Press to register” which will take them to a page containing a form that they must fill out to create an account. All of the fields in the form are required, except for the middle initial, and as such it will not allow them to submit unless they are filled in. The fields also have specific patterns that must be followed to prevent invalid inputs, and will similarly not submit otherwise. Once the fields are correctly filled and the user hits submit, the “commit.php” script will connect to the database, SELECT all of the customer and employee accounts and check if the email entered in the form is already in use or not. If the email is found to be in use, then the user will be informed of this via an error message, and will have to refill the form using a different email address. Once the user enters an email address that is not already in use, then the script will hash the entered password and INSERT all of the variables from the form into the CUSTOMERS table of the database.

-SQL Statements used in PHP

```
"select * from CUSTOMERS";
"select * from EMPLOYEES";
"insert into CUSTOMERS(custEmail, custPass, fName, mInit, lName,
custPhone, street, city, state, zip)
 values('".$email."', '".$hashedPass."', '".$fName."', '".$mInit."',
'".$lName."', '".$custPhone."', '".$street."', '".$city."', '".$state."',
'".$zip."')";
```

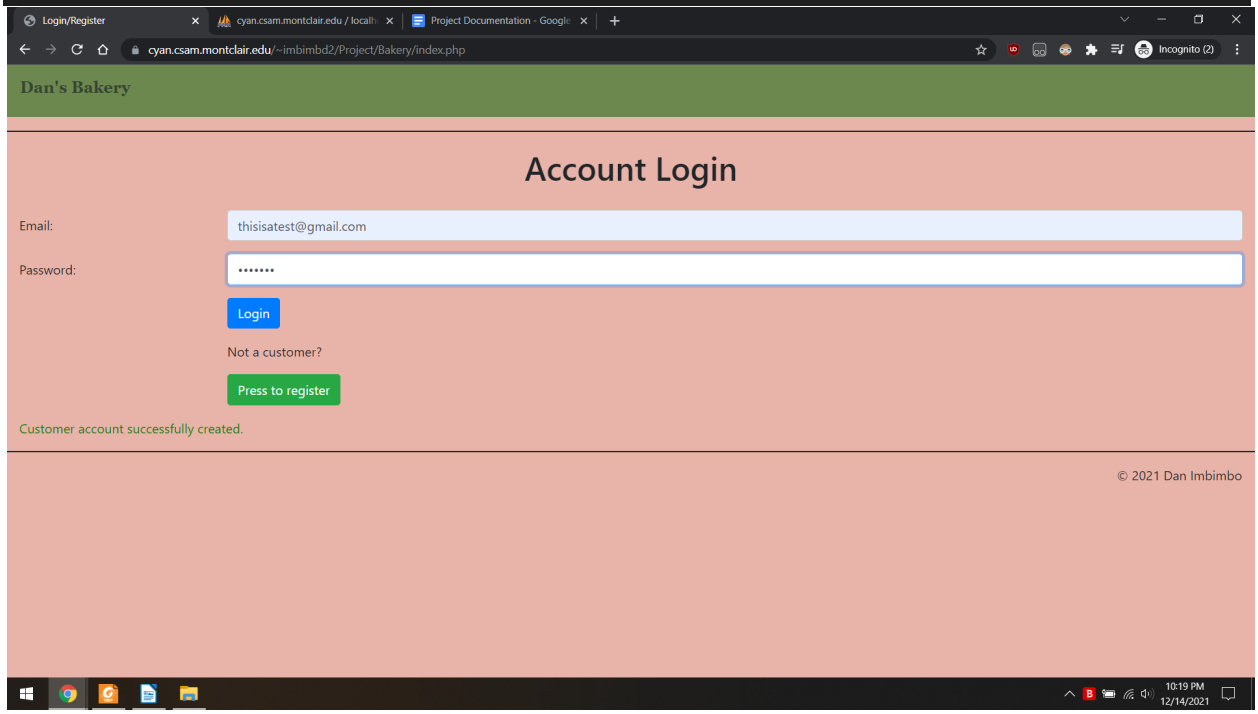


Logging in:

On the login screen, the user can enter the email and password to their existing account. The script “checkLogin.php” will connect to the database and SELECT all of the customer accounts to check if the email address matches with any of them, if it is not found to match with any of the customer accounts then it will SELECT all of the employee accounts to check if the email matches with any of those. Once the script finds a matching email address, it will verify that the entered password matches the hashed password stored with the account on the database. If the password is not found to match, then the user will be prompted with an error message and will have to try again. Similarly, if the entered email address is not found to match one of the customer or employee accounts on the database then the user will also be prompted with an error message. However, if the email was found among the customer accounts and the password was found to match, then the user will be navigated to the customer landing page.

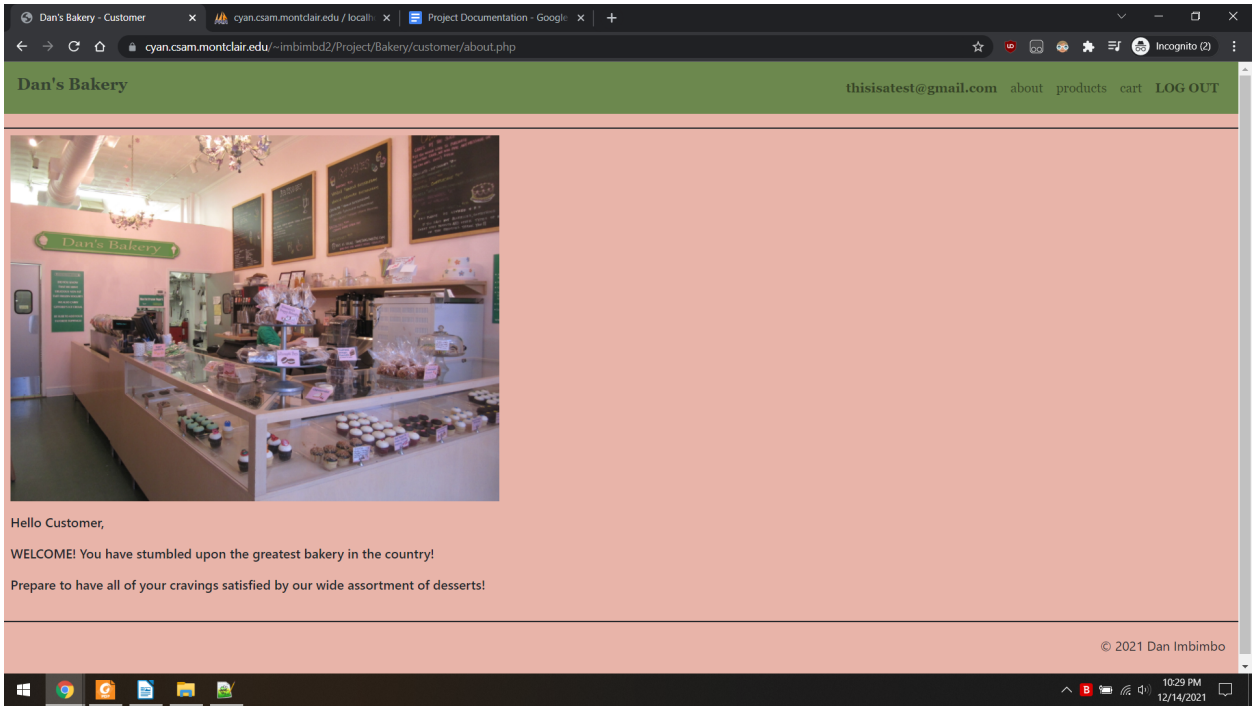
-SQL Statements used in PHP

```
"select * from CUSTOMERS";
"select * from EMPLOYEES";
```



Customer Landing Page:

On the customer landing page, the user will be presented with a brief description about the business, along with an image representing the inside of the bakery. They are also given a navigation bar that allows them to traverse through the various pages on the customer view of the site, as well as log out of their account at any point.

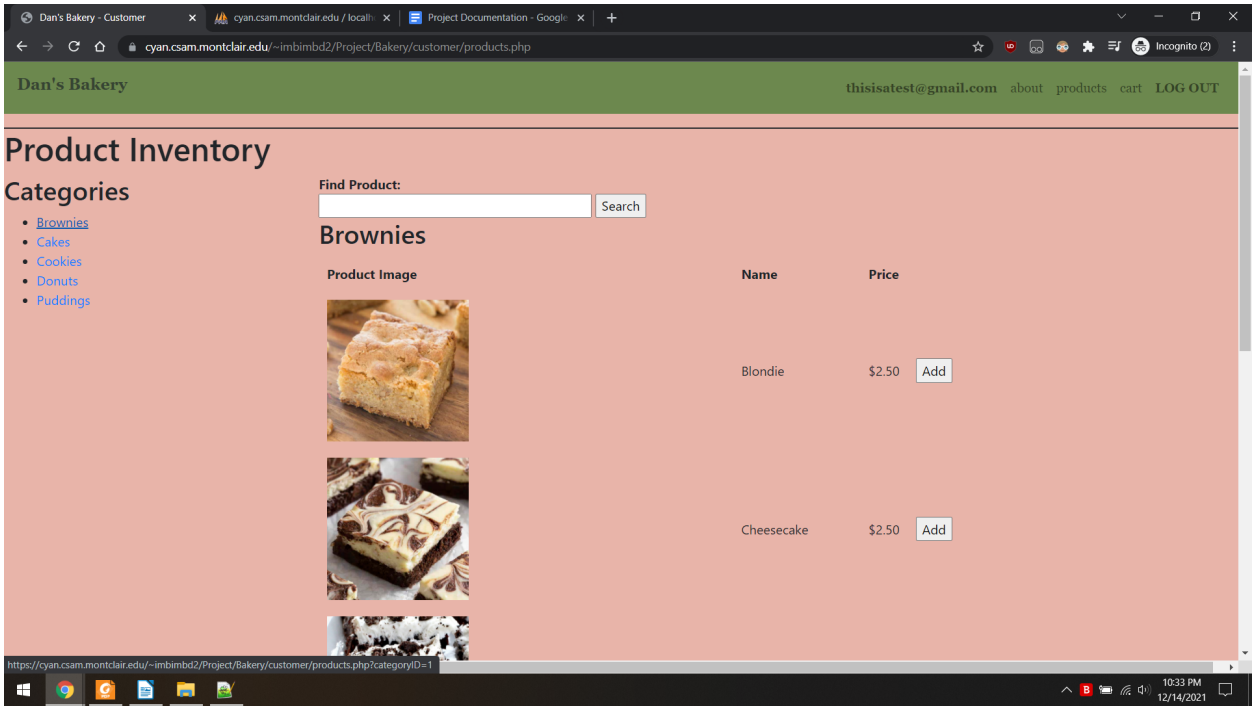


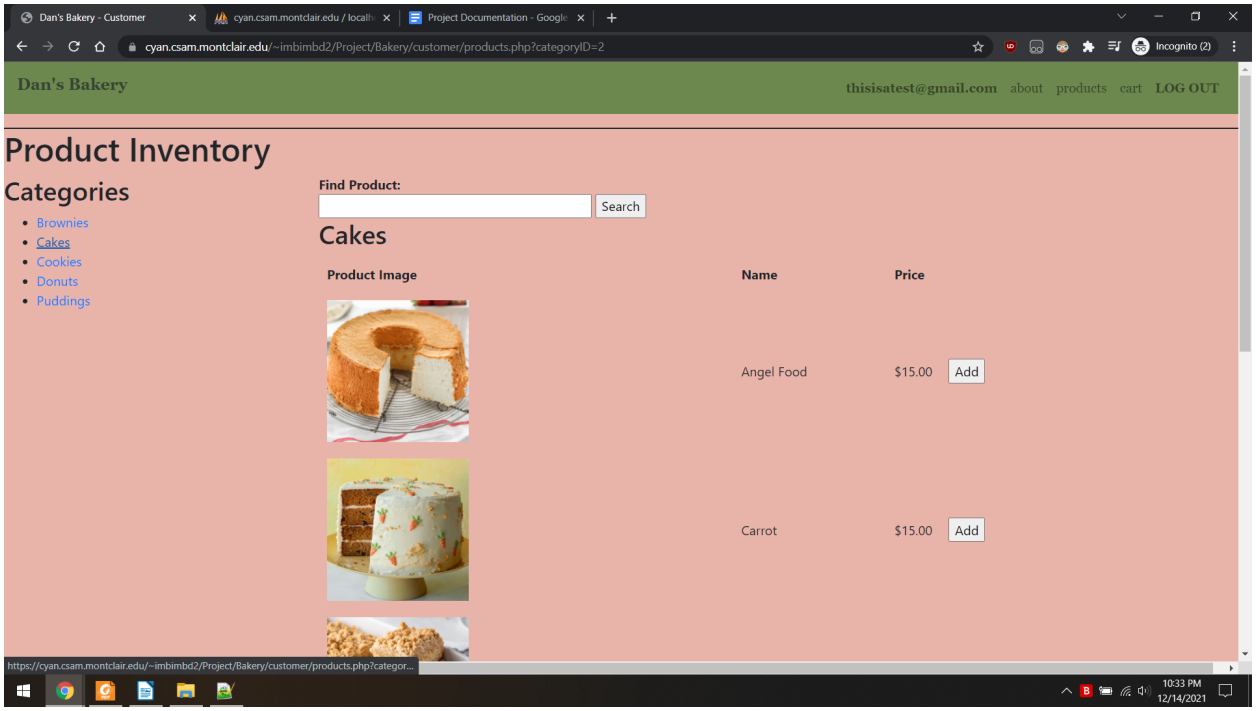
Products Page (All categories):

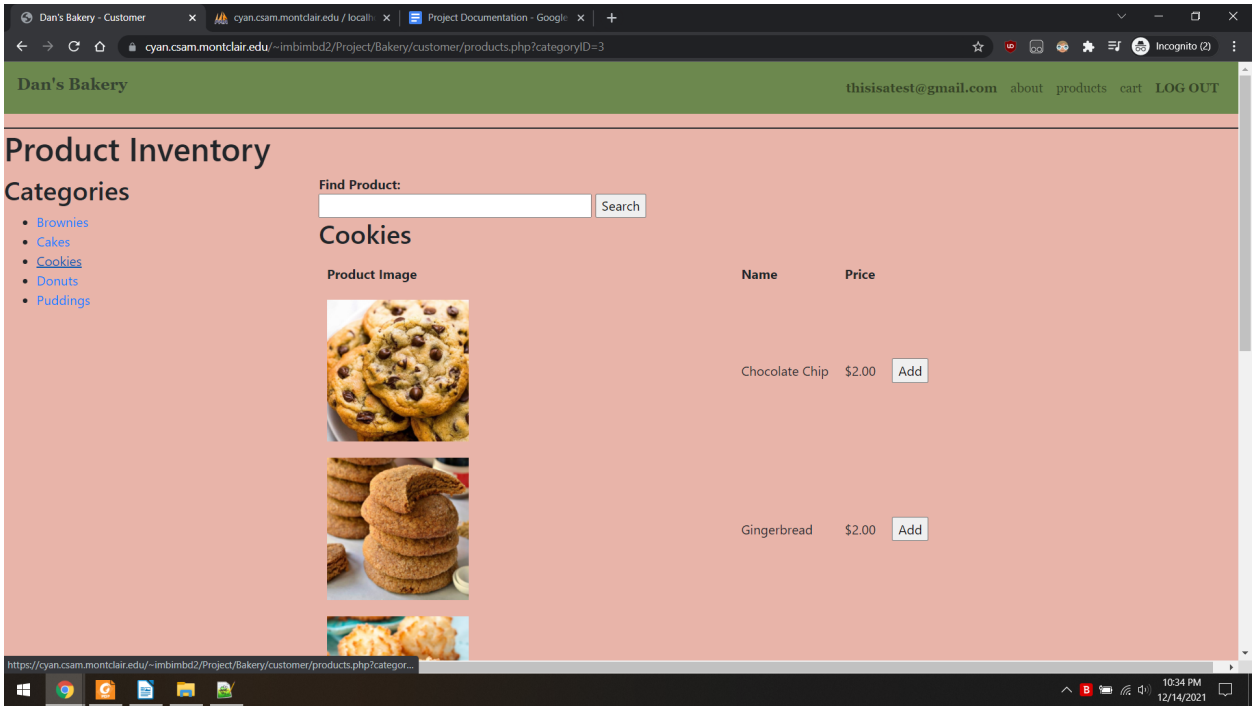
Once the products page is selected from the navigation bar, the user will be presented with the various products that can be purchased. To the right of the listed products the user can choose to view the products for a given category, and they are also able to search through all the products via the search bar. The products are retrieved by the “products.php” script, which uses functions from the “categoryFunctions.php” & “productFunctions.php” scripts to connect to the database, SELECT all of the categories from the database, SELECT the categoryName for each specified categoryID, and SELECT all of the products for the specified categoryID; before displaying them to the “productsDisplay.php” script.

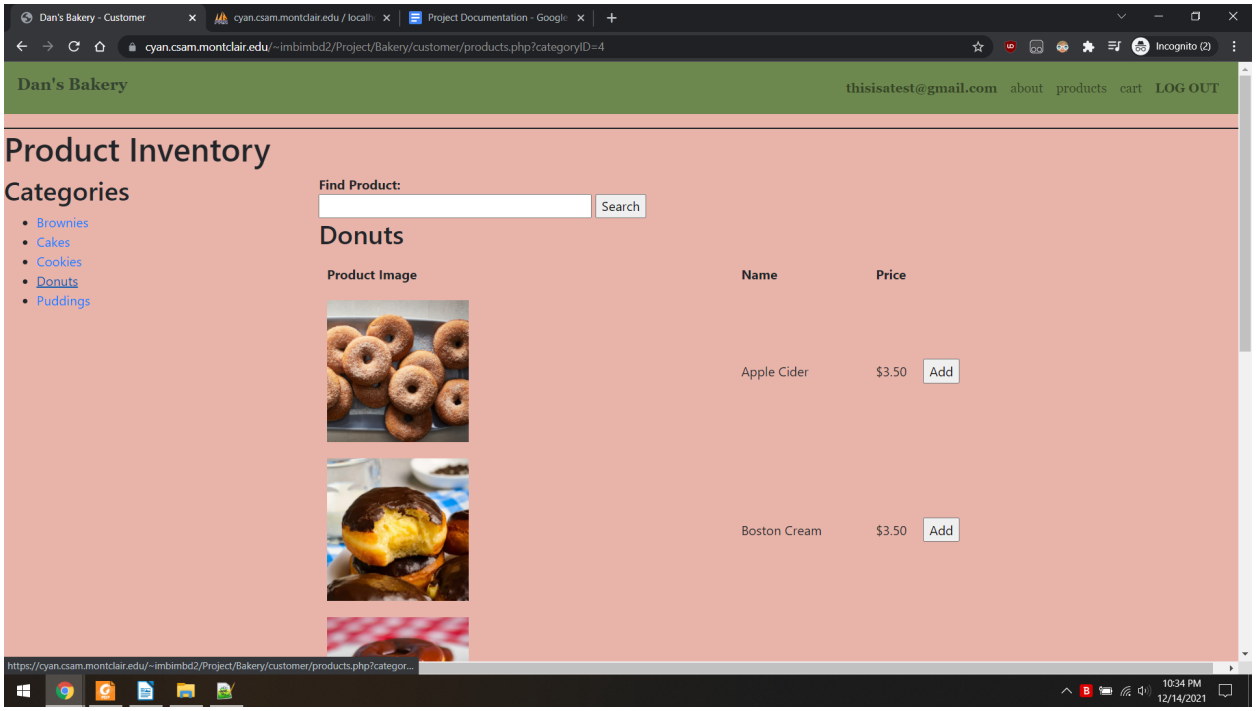
-SQL Statements used in PHP

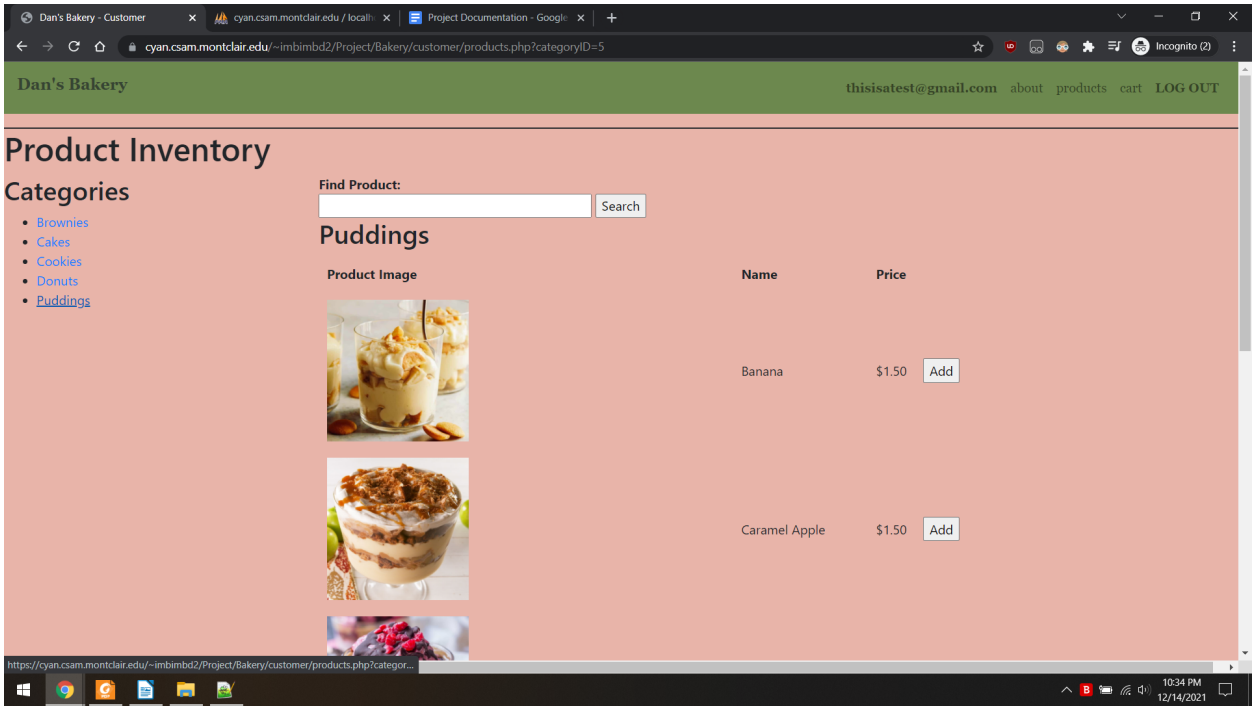
```
"select * from CATEGORIES
 order by catID";
"select * from CATEGORIES
 where catID='".$categoryID."'";
"select * from PRODUCTS
 where categoryID='".$categoryID."'
 order by prodID";
```


Dan's Bakery - Customer
cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/customer/products.php
Dan's Bakery
thisisatest@gmail.com about products cart LOG OUT
Product Inventory
Categories
Brownies
Cakes
Cookies
Donuts
Puddings
Find Product:
Search
Brownies
Product Image
Name
Price
Blondie $2.50 Add
Cheesecake $2.50 Add
https://cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/customer/products.php?categoryID=1
10:33 PM
12/14/2021


Dan's Bakery - Customer
cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/customer/products.php?categoryID=2
Dan's Bakery
thisisatest@gmail.com about products cart LOG OUT
Product Inventory
Categories
Brownies
Cakes
Cookies
Donuts
Puddings
Find Product:
Search
Cakes
Product Image
Name
Price
Angel Food $15.00 Add
Carrot $15.00 Add
https://cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/customer/products.php?categor...
10:33 PM
12/14/2021


Dan's Bakery - Customer
cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/customer/products.php?categoryID=3
Incognito (2)
Dan's Bakery
thisisatest@gmail.com about products cart LOG OUT
Product Inventory
Categories
Brownies
Cakes
Cookies
Donuts
Puddings
Find Product:
Search
Cookies
Product Image
Name
Price
Chocolate Chip $2.00 Add
Gingerbread $2.00 Add
10:34 PM
12/14/2021


Dan's Bakery - Customer
cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/customer/products.php?categoryID=4
Incognito (2)
Dan's Bakery
thisisatest@gmail.com about products cart LOG OUT
Product Inventory
Categories
Brownies
Cakes
Cookies
Donuts
Puddings
Find Product:
Search
Donuts
Product Image
Name
Price
Apple Cider $3.50 Add
Boston Cream $3.50 Add
10:34 PM
12/14/2021

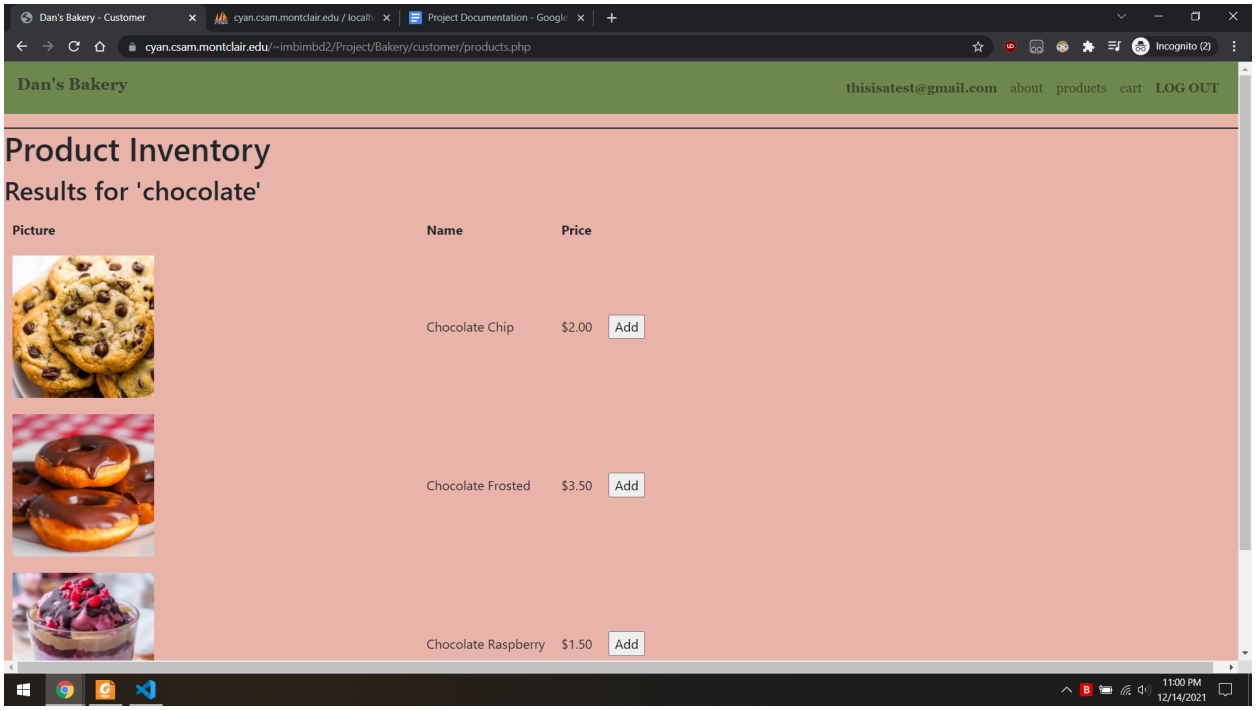


Searching For Product:

The user can enter a term in the search bar of the products page, and get a listing of the various products that contain it. This is done through the “products.php” script which uses a function from “productFunctions.php” to connect to the database & SELECT all the products from the database where the prodName is like the entered term and then displays the results via the “productsDisplay.php” script.

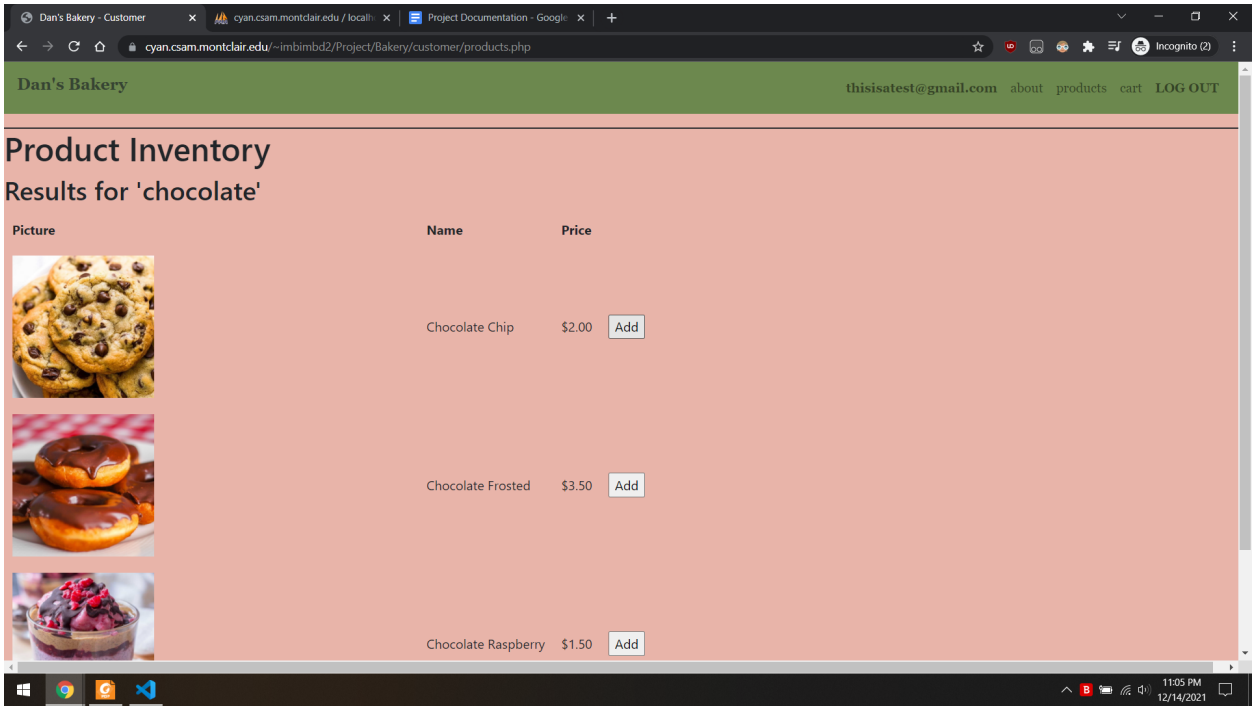
-SQL Statements used in PHP

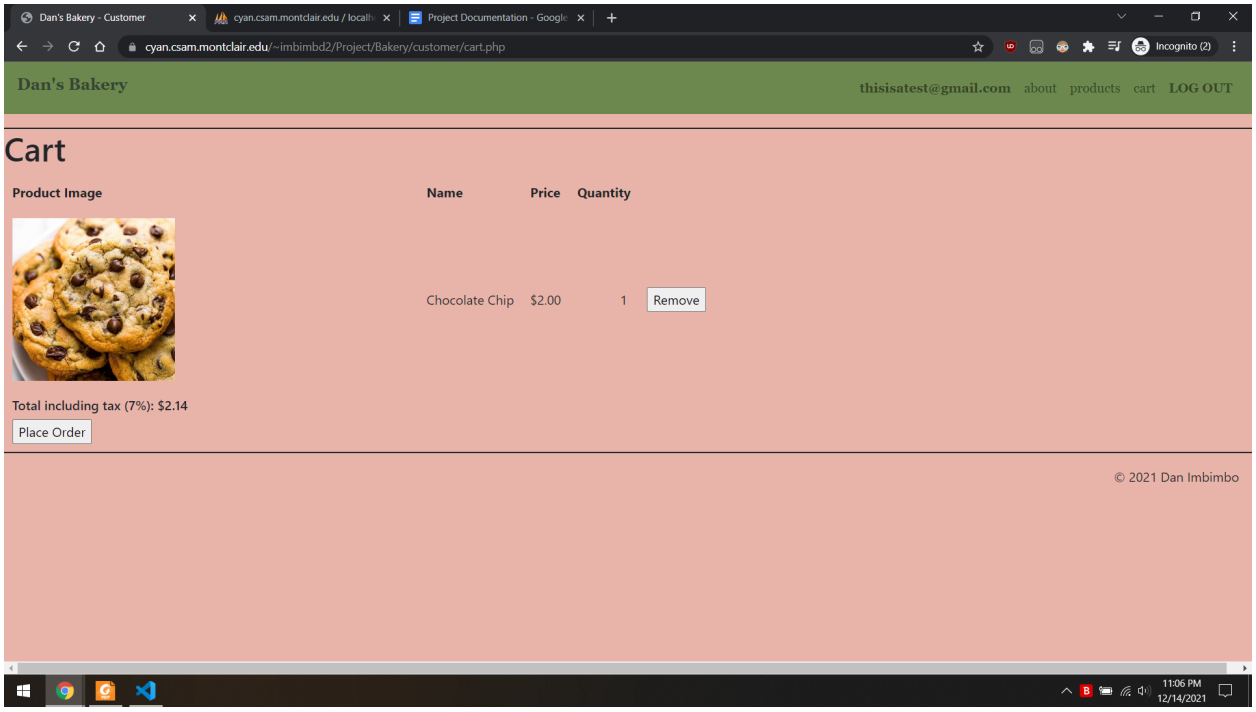
```
"select * from PRODUCTS
 where prodName like '%".$searchTerm."%'";
```



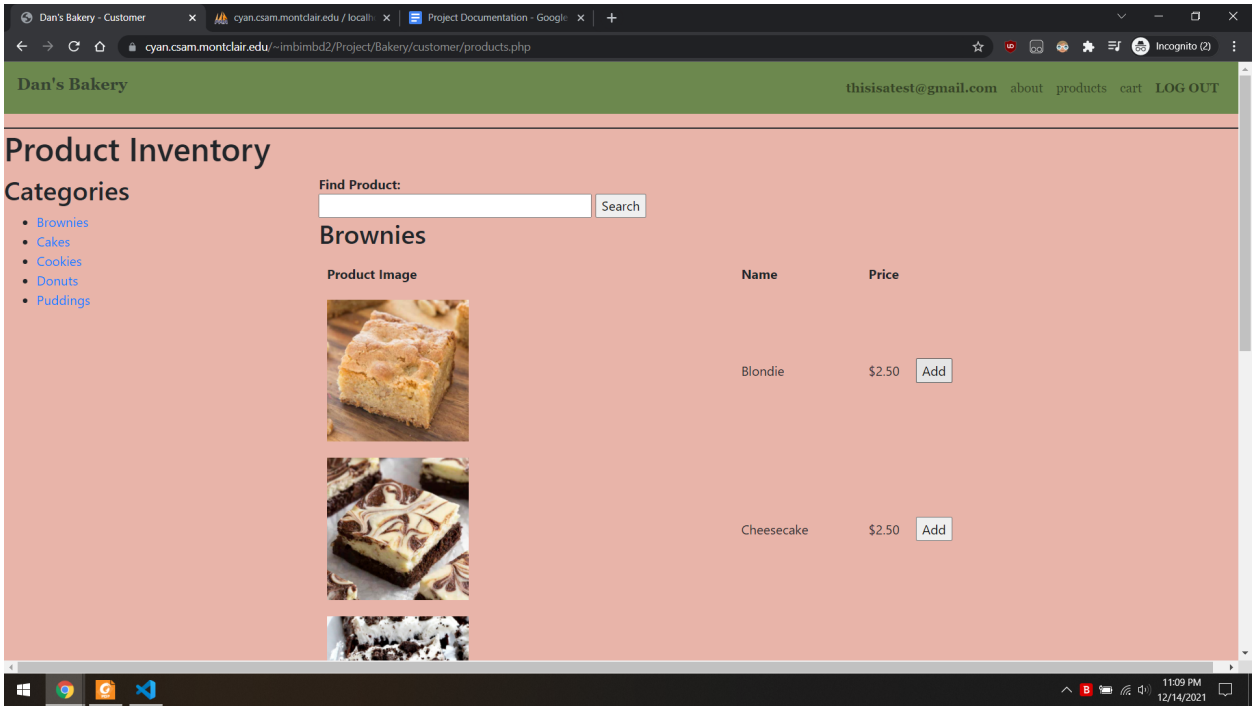
Product Added To Cart:

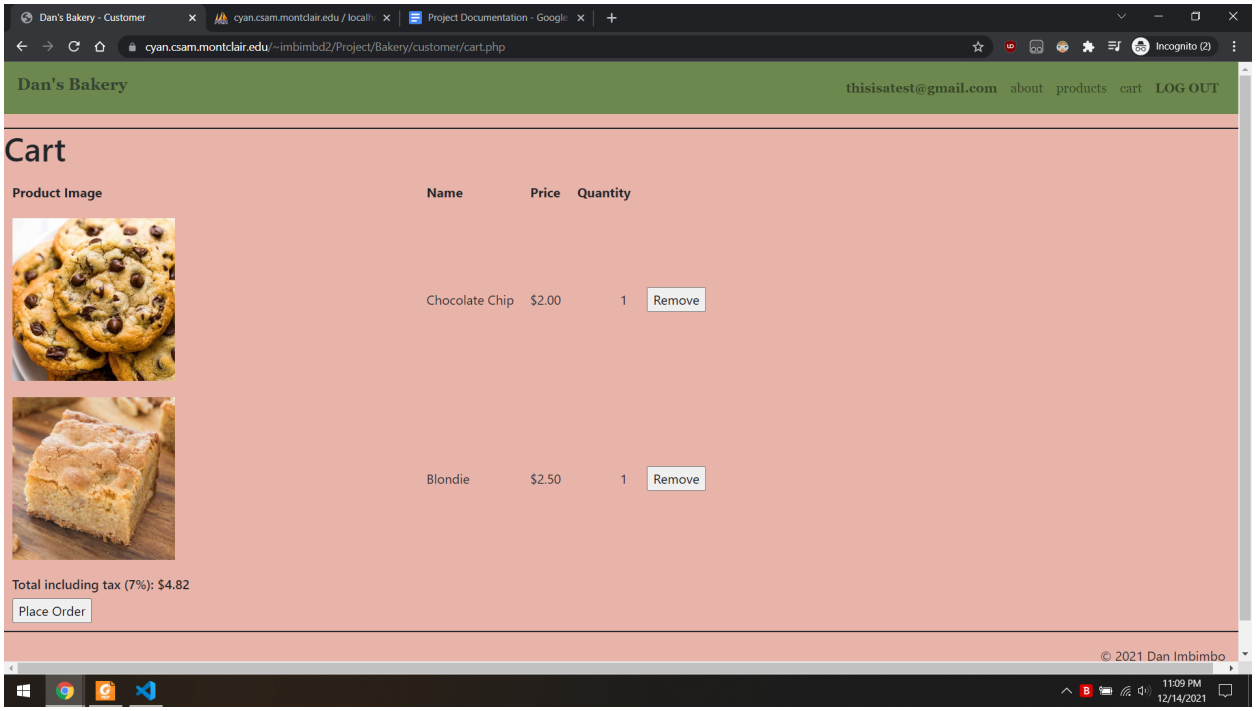
The user can hit the “Add” button next to an item on the product list, and it will store the various attributes of the item into their $_SESSION[‘cart’] array, directing them to the cart page. However, the item will only have an add button next to it if the stock for the item is currently greater than 0 and if the quantity in the cart does not currently equal the stock for the item on the database. This is determined through the “productDisplay.php” script which uses a function from “productFunctions.php” to determine if the $_SESSION[‘cart][‘quantity] amount equals the amount of available stock for the product from the array of products selected when the products page was loaded. Once an item is added, the “cart.php” script will then display the items in the session cart array via the “cartDisplay.php” script, along with the calculated total from the “orderFunctions.php” script. Products can similarly be removed from the cart if the user hits the “Remove” button next to an item, which unsets the item from their $_SESSION[‘cart’] array.



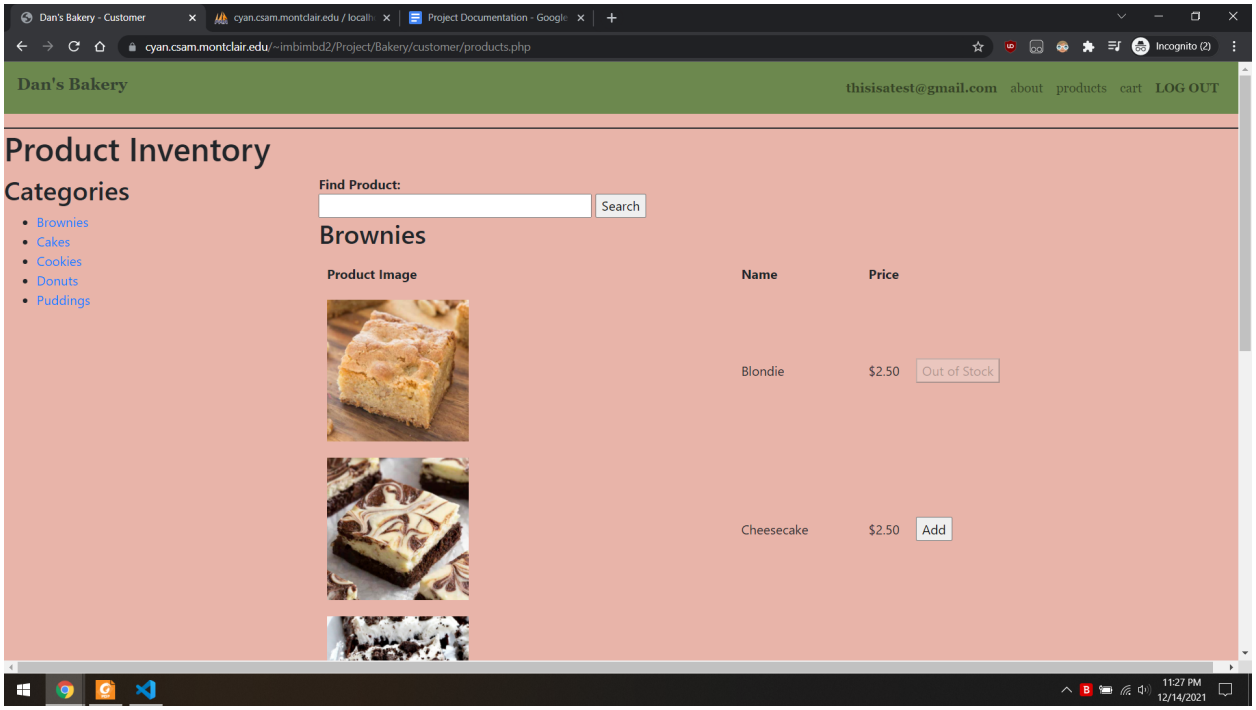


Adding 2nd Item:





Item “Out of Stock” Example:

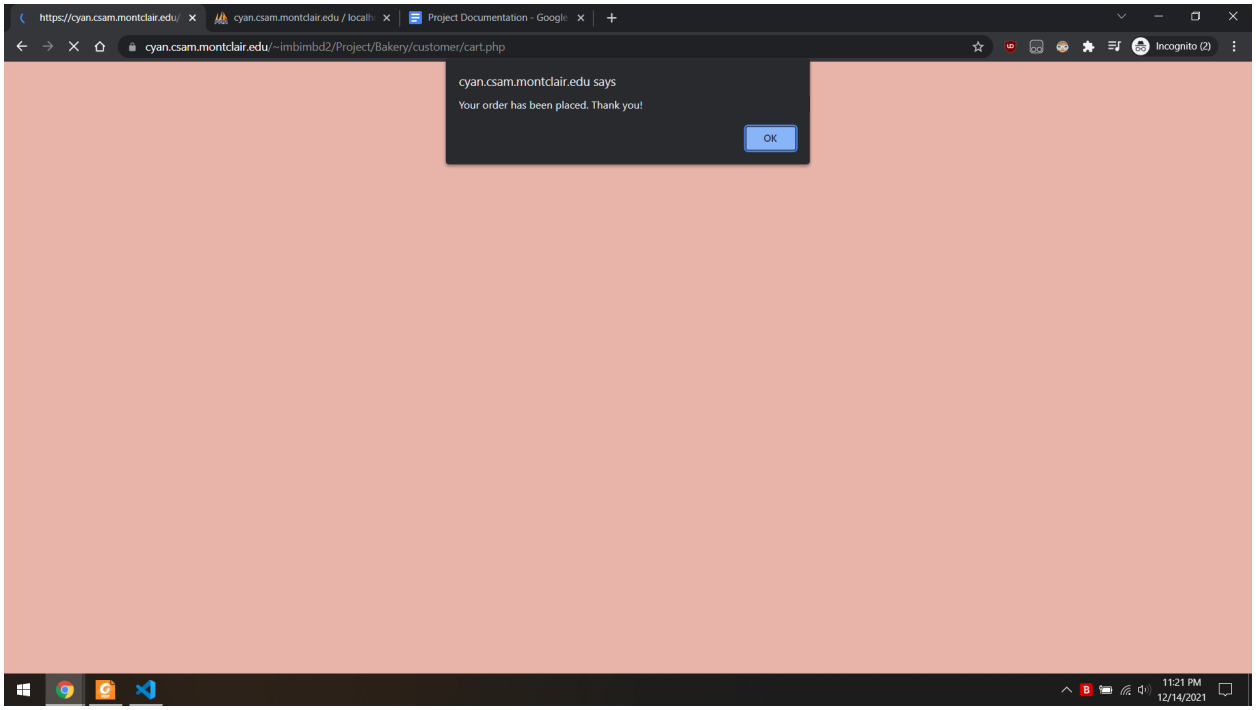


Order Placed:

The user can hit the “Place Order” button on the cart page to place their order. Then the “cart.php” script will use two functions from the “orderFunctions.php” script to connect to the database, INSERT the order into the database, and to INSERT each of the orderItems of the order into the database. The script also UPDATES the stock of each product that was purchased to reflect the quantity that is no longer available. At which point, the user will receive an alert informing them that their order has been placed and direct them back to the login page so that all session variables can be unset and the session ID can be removed.

-SQL Statements used in PHP

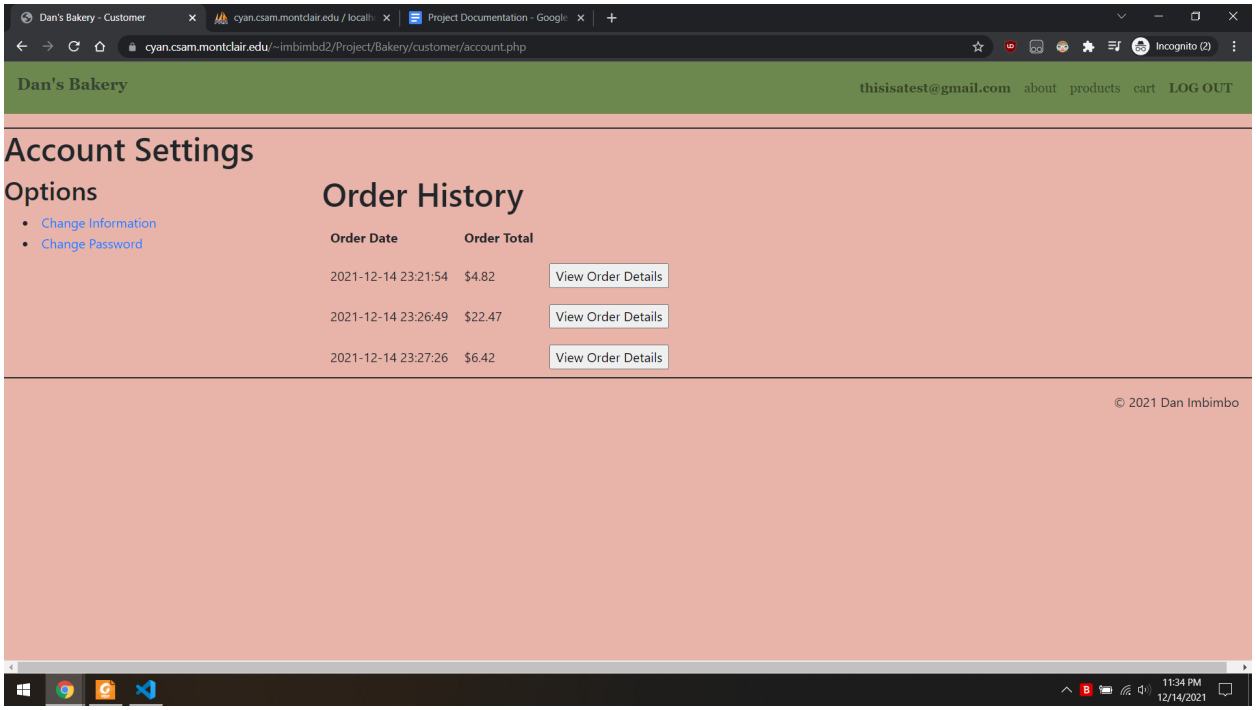
```
"insert into ORDERS(ordTotal, ordDate, customerEmail)
 values('".$orderTotal."', '".$orderDate."', '".$customerEmail."')";
"insert into ORDERITEMS(orderID, productID, quantity, cost)
 values('".$orderID."', '".$item['productID']."', '".$item['quantity']."',
'".$cost."')";
"update PRODUCTS
 set stock = stock - '".$item['quantity']."'
 where prodID = '".$item['productID']."'";
```



Account Settings/Order History:
From the navigation bar the user can also select the option containing their email address, bringing them to their account settings page, which also lists their order history. This is done through the “account.php” script which uses a function from “orderFunctions” to connect to the database & SELECT all of the orders associated with the given customer email address, and orders them by ascending dates.

-SQL Statements used in PHP

```
"select * from ORDERS
 where customerEmail='".$customerEmail."'
 order by ordDate ASC";
```

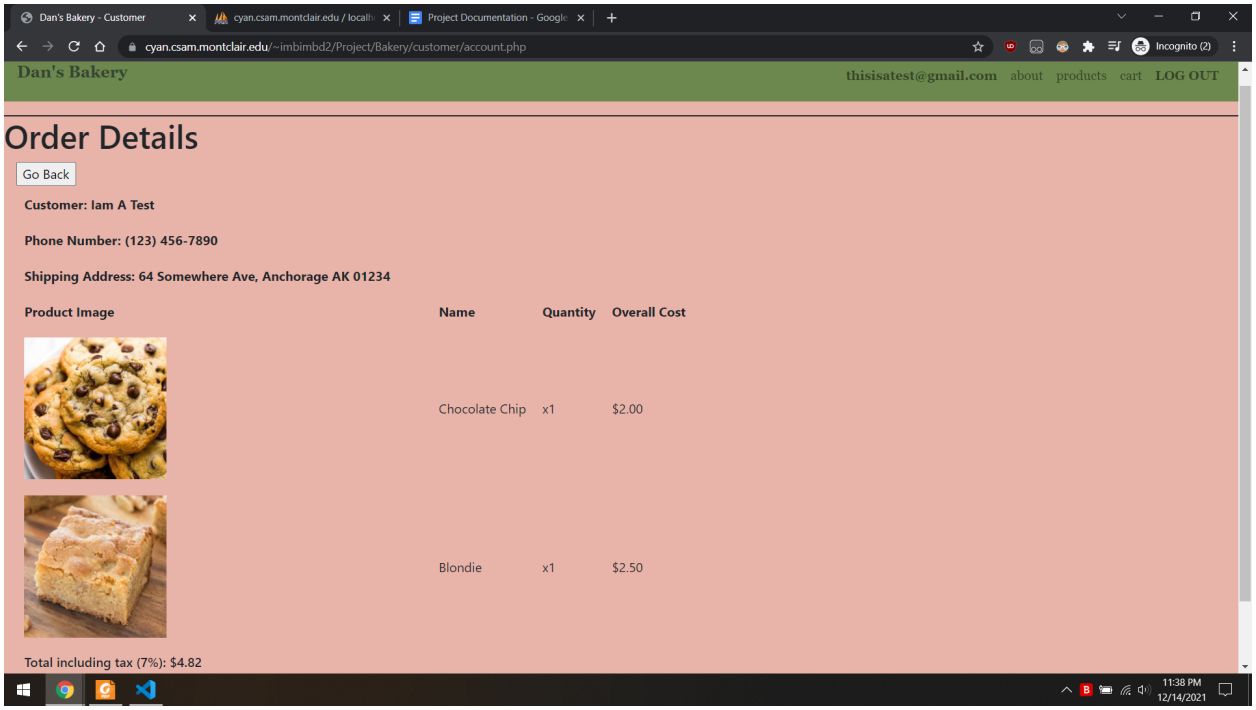


View Order Details:

The user can select the “View Order Details” button next to one of the orders listed in their order history and they will be brought to a page displaying the details of the specified order. This is done by the “account.php” script which uses two functions from the “orderFunctions.php” script to connect to the database, SELECT the order by its specified ordID, and SELECT all of the order items associated with that ordID, ordering them by cost in ascending order. Then this is displayed to the user via the “orderDetails.php” script, along with the customer’s name, phone number, and address which are retrieved from a function in the “custFunctions.php” script to connect to the database & SELECT the customer associated with the specified $_SESSION(‘email’) entered during login. The user is also presented with a button to go back to their Account Settings page.

-SQL Statements used in PHP

```
"select * from ORDERS
 where ordID='".$orderID."'";
"select * from ORDERITEMS
 where orderID='".$orderID."'
 order by cost ASC";
"select * from CUSTOMERS
 where custEmail='".$custEmail."'";
```



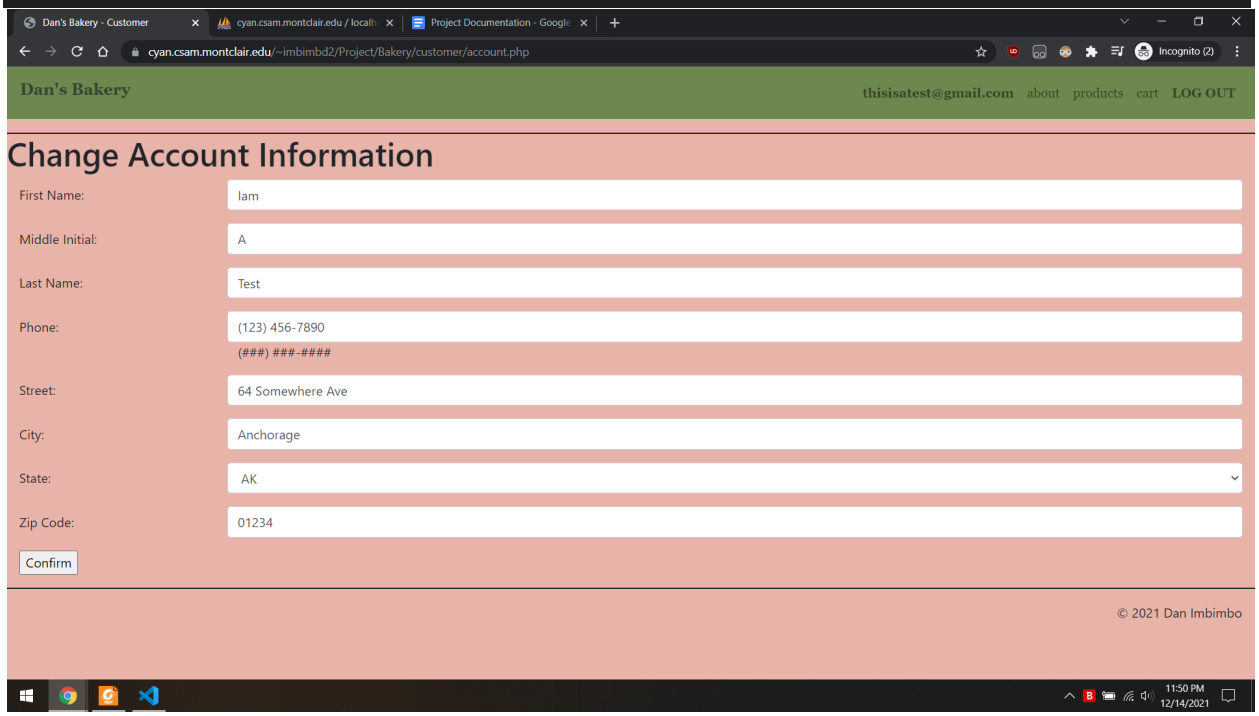
Change Account Information:

From the Account Settings page the user can select one of the options on the left-hand side of the screen to change the information associated with their account. When selected, the user will be presented with a form that contains the information currently associated with their account. This is retrieved by the “account.php” script which uses a function from the “custFunctions.php” script to connect to the database & SELECT the customer associated with the specified $_SESSION(‘email’) entered during login. The user can change any of their information through the various fields, and when they press the confirm button, the “account.php” script will use a function from the “custFunctions.php” script to connect to the database & UPDATE the information in the customer record associated with the $_SESSION(‘email’). At which point the user will be presented with an alert informing them that their account information has been successfully updated and they will be redirected back to the Account Settings page.

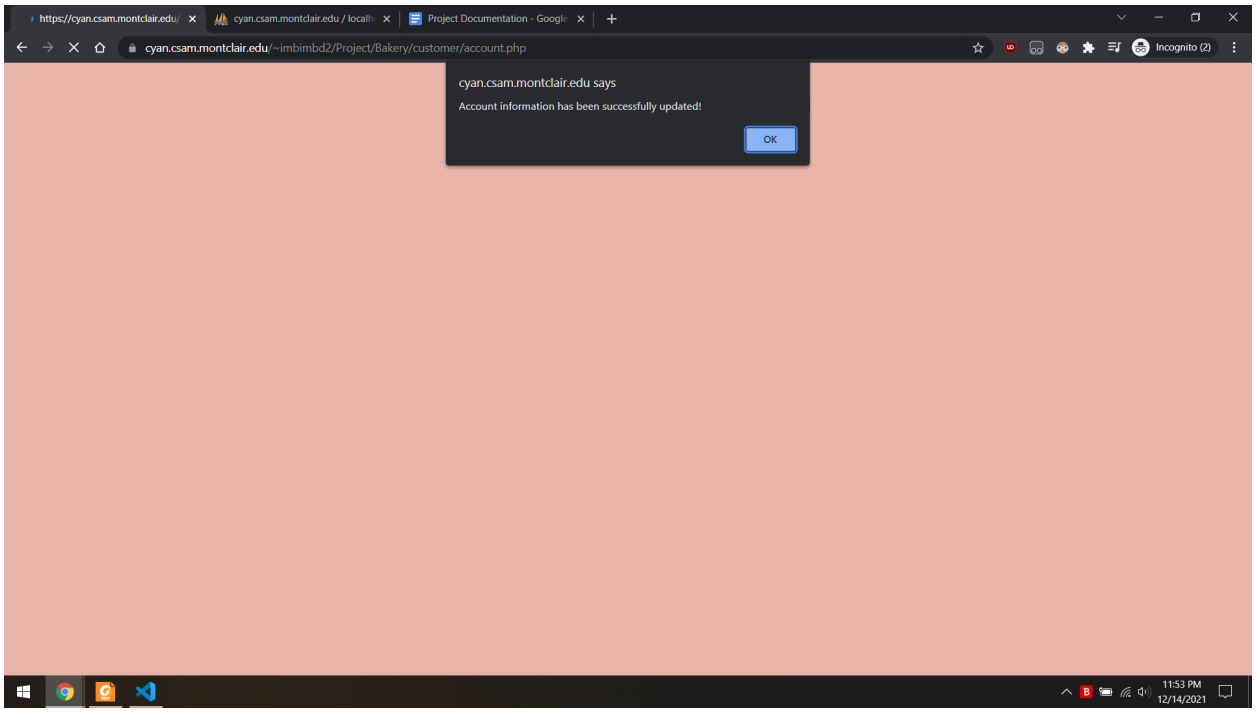
-SQL Statements used in PHP

```
"select * from CUSTOMERS
 where custEmail='".$custEmail."'";
"update CUSTOMERS
 set
fName='".$fName."',mInit='".$mInit."',lName='".$lName."',custPhone='".$pho
ne."',street='".$street."',city='".$city."',state='".$state."',zip='".$zip
."'
```

```
where custEmail='".$custEmail."'";
```



Updated:



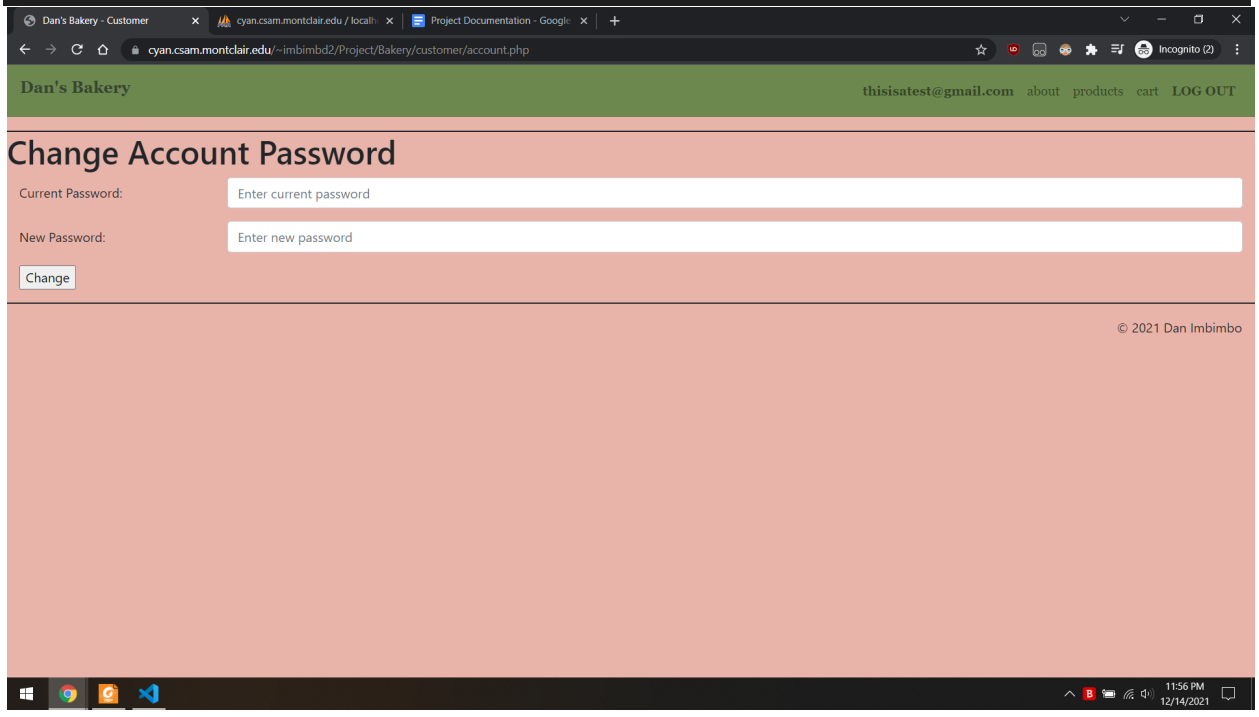
Change Account Password:

From the Account Settings page the user can select one of the options on the left-hand side of the screen to change the password associated with their account. When

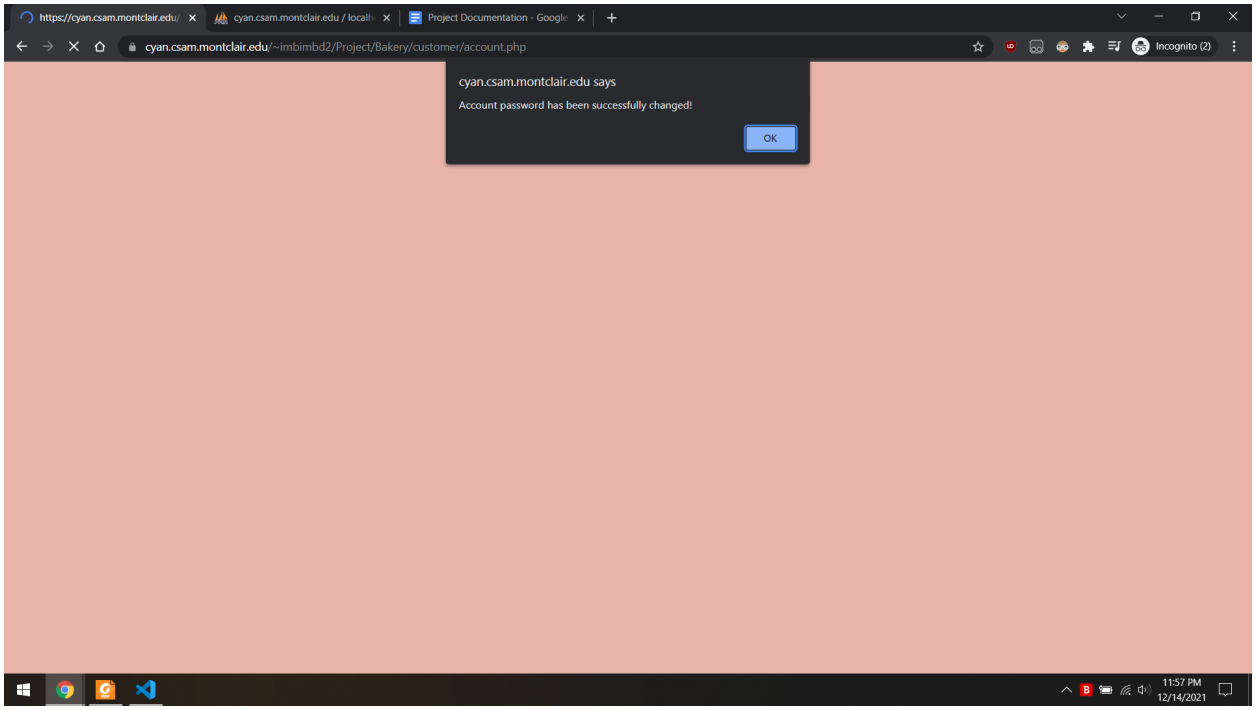
selected, the user will be presented with a form that contains a field for their current password and their new password. Once the user supplies their current password and their new password in the fields, the “account.php” script will use a function from the “custFunctions.php” script to connect to the database, SELECT the customer associated with the specified $_SESSION(‘email’) entered during login, and verify that the entered current password matches the hashed password on the database. If it does not match then the user will be presented with an error message and will have to re-enter their passwords before trying again. However, if the passwords do match, then the new password will be hashed and a function from the “custFunctions.php” script will connect to the database & UPDATE the password in the customer record associated with the $_SESSION(‘email’). At which point the user will be presented with an alert informing them that their account password has been successfully updated and they will be redirected back to the Account Settings page.

-SQL Statements used in PHP

```
"select * from CUSTOMERS
 where custEmail='".$custEmail."'";
"update CUSTOMERS
 set custPass='".$hashedPass."'
 where custEmail='".$custEmail."'";
```



Changed:



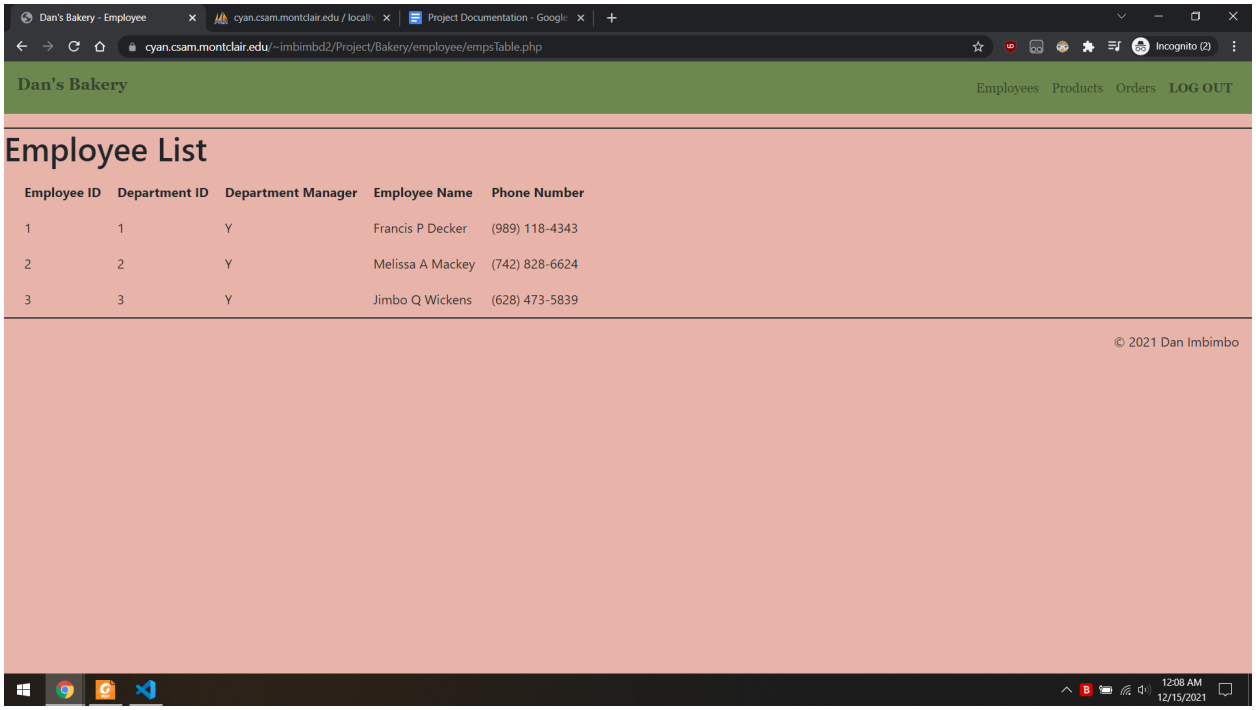
-Employees

Employee Landing Page:

On the employee landing page, the user is presented with a list of the employees who work for the business that contains their employee IDs, department IDs, managerial status (from the deptFunction mentioned below), name, and phone number. This is done by the “empTable.php” script which uses a function from the “empFunctions.php” script to connect to the database & SELECT all of the employees in the database, with some of their attributes then displayed via the “empDisplay.php” script. The “empDisplay.php” script then also uses a function from the “deptFunctions.php” script to connect to the database & SELECT all the manIDs for the departments and see if a given employee’s ID matches it, displaying answer via the “empDisplay.php” script.The user is also given a navigation bar that allows them to traverse through the various pages on the employee view of the site, as well as log out of their account at any point.

-SQL Statements used in PHP

```
"select * from EMPLOYEES";
"select manID from DEPARTMENTS";
```

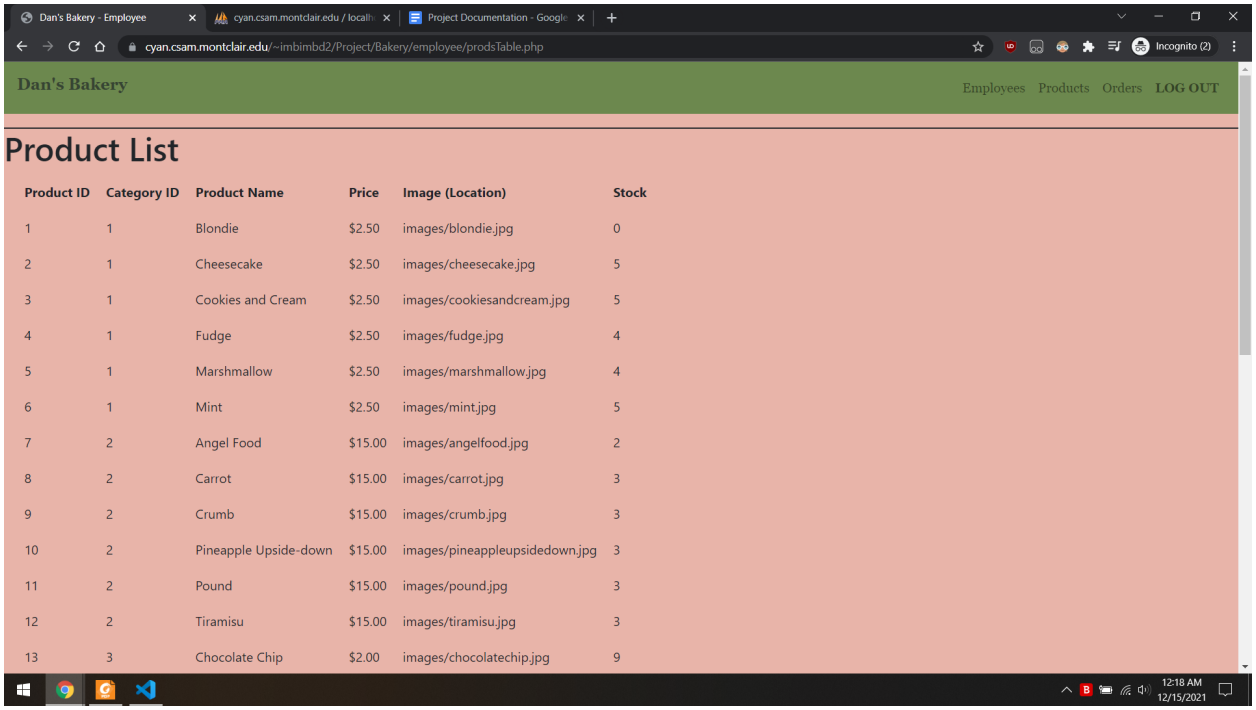


View Products:

Once the Products page is selected from the navigation bar, the user is presented with a list of the products sold by the business. This is done by the “prodsTable.php” script which uses a function from the “productFunctions.php” script to connect to the database & SELECT all of the products in the database, with all of their attributes then displayed via the “prodsDisplay.php” script.

-SQL Statements used in PHP

```
"select * from PRODUCTS";
```

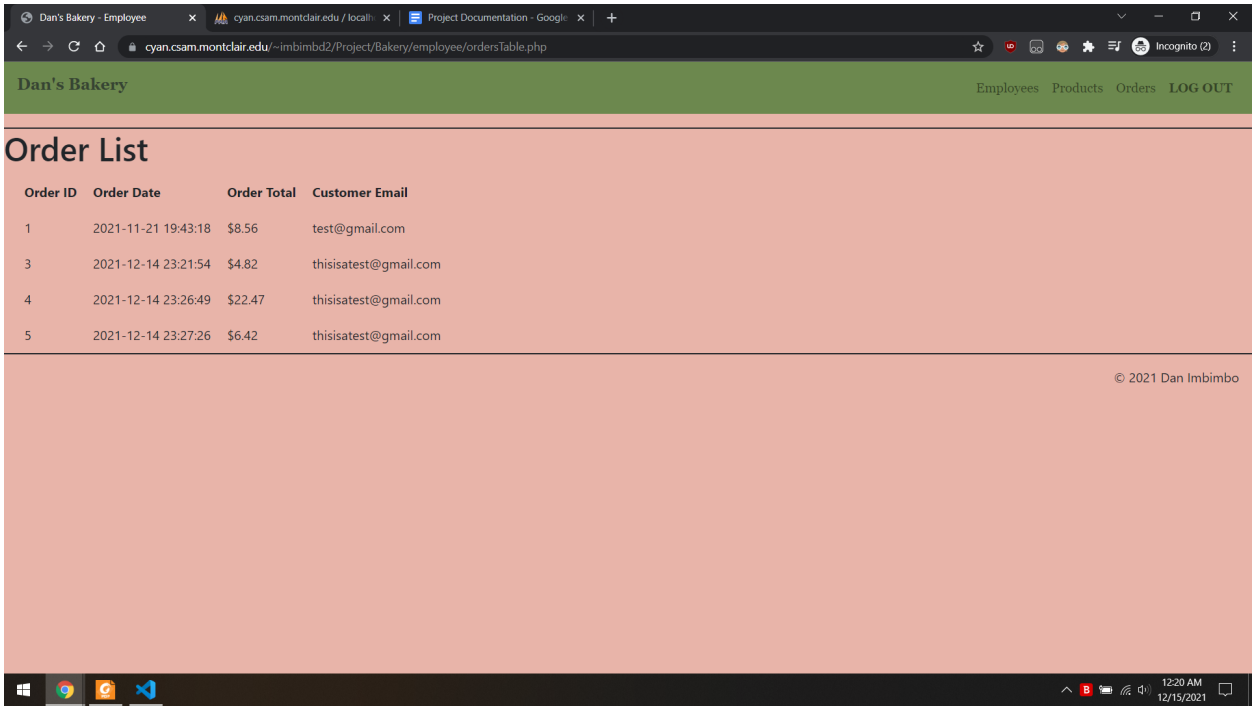


View Orders:

Once the Orders page is selected from the navigation bar, the user is presented with a list of the orders placed by customers. This is done by the “ordersTable.php” script which uses a function from the “orderFunctions.php” script to connect to the database & SELECT all of the orders in the database, with all of their attributes then displayed via the “ordersDisplay.php” script.

-SQL Statements used in PHP

```
"select * from ORDERS
 order by ordDate ASC";
```



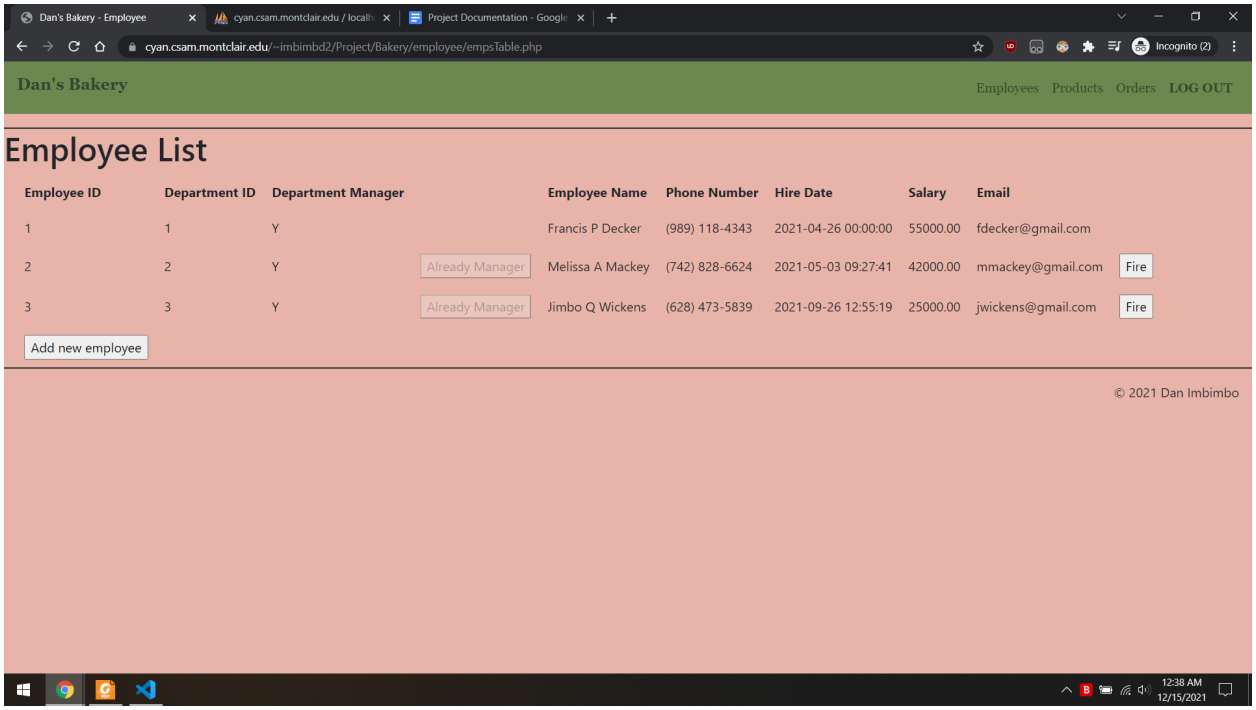
-CEO

CEO Landing Page:

On the CEO landing page, the user is presented with a similar view as the employees, however the list of the employees who work for the business also contains their hire dates, salaries, and emails. The user is also given the option to fire employees, as well as make an employee the manager of their department. This is done by the “empTable.php” script which uses a function from the “empFunctions.php” script to connect to the database & SELECT all of the employees in the database, with all of their attributes then displayed via the “empDisplay.php” script. The “empDisplay.php” script then also uses a function from the “deptFunctions.php” script to connect to the database & SELECT all the manIDs for the departments and see if a given employee’s ID matches it, displaying answer via the “empDisplay.php” script along with a button to make the employee the manager if they aren’t already. The user is also given a navigation bar that allows them to traverse through the various pages on the employee view of the site, as well as log out of their account at any point.

-SQL Statements used in PHP

```
"select * from EMPLOYEES";
"select manID from DEPARTMENTS";
```



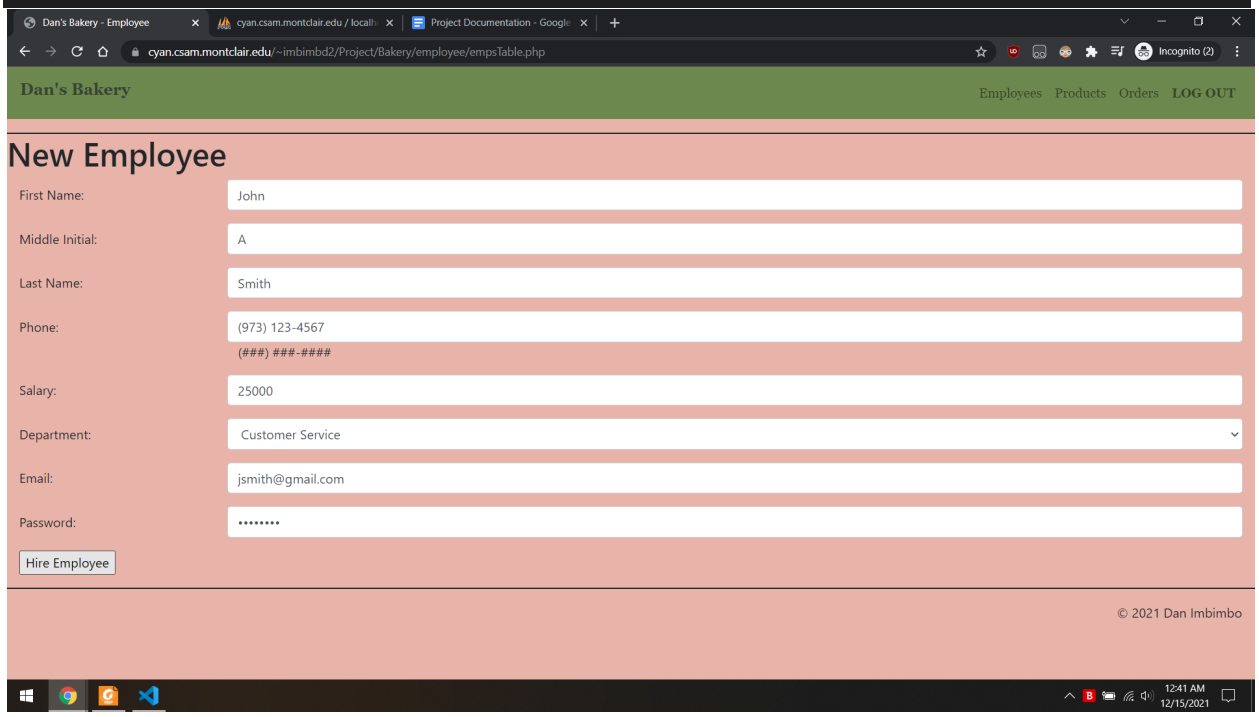
Add New Employee:

On the Employees page, the CEO can click the button “Add new employee” which will take them to a page containing a form that they must fill out to create the employee’s account. All of the fields in the form are required, except for the middle initial, and as such it will not allow them to submit unless they are filled in. The fields also have specific patterns that must be followed to prevent invalid inputs, and will similarly not submit otherwise. Once the fields are correctly filled and the user hits submit, the “empTable.php” script will use a function from the “empFunctions.php” script to connect to the database, SELECT all of the customer and employee accounts and check if the email entered in the form is already in use or not. If the email is found to be in use, then the user will be informed of this via an error message, and will have to refill the form using a different email address. Once the user enters an email address that is not already in use, then the script will hash the entered password and INSERT all of the variables from the form into the EMPLOYEES table of the database. At which point the user will be presented with an alert informing them that the employee has been successfully hired and they are redirected back to the Employees page.

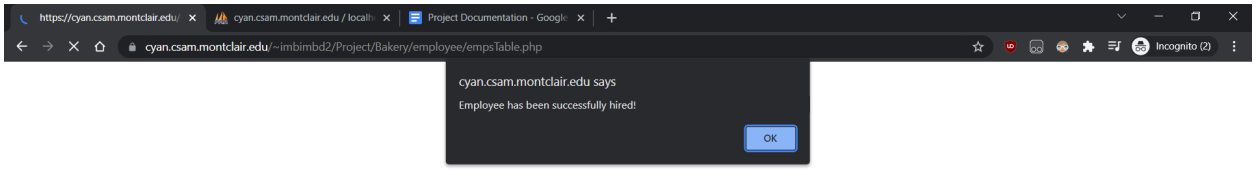
-SQL Statements used in PHP

```
"select * from CUSTOMERS";
"select * from EMPLOYEES";
```

```
"insert into
EMPLOYEES(empEmail,empPass,fName,mInit,lName,empPhone,hireDate,salary,depa
rtmentID)
 values('".$empEmail."', '".$hashedPass."', '".$fName."', '".$mInit."',
'".$lName."', '".$empPhone."', '".$hireDate."', '".$salary."',
'".$departmentID."')";
```



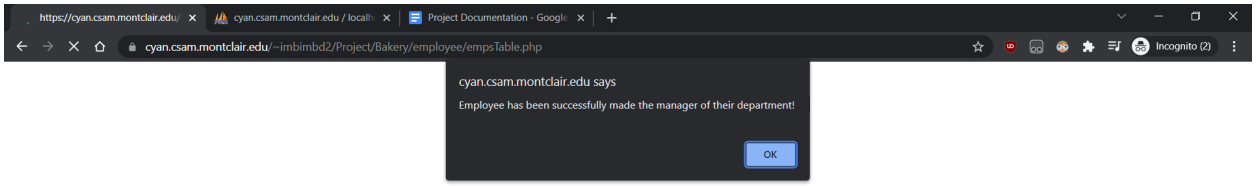
Hired:



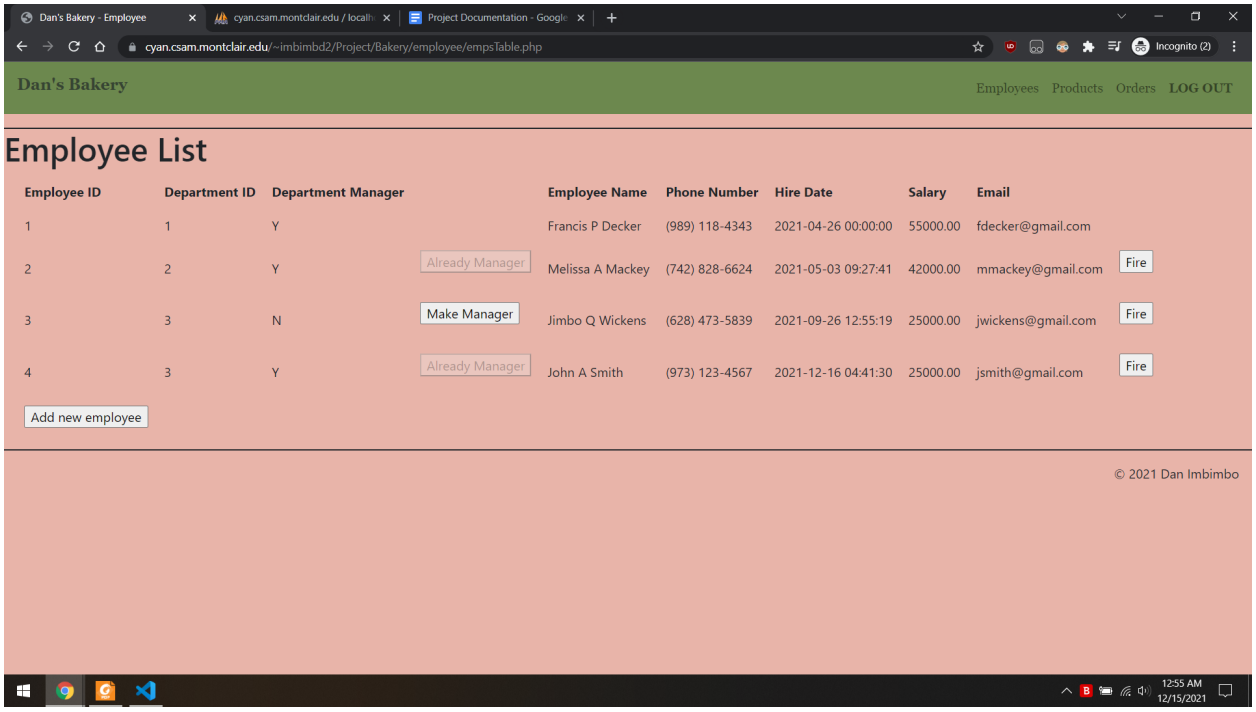
Make Manager:
On the Employees Page, the CEO can click the button “Make Manager” to make the specified employee the manager of their department. This is done through the “empTable.php” script which uses a function from the “deptFunctions.php” script to connect to the database, UPDATE the manager start date to the current time for the specified department, and UPDATE the manager ID to the given employee for the specified department. At which point the user will be presented with an alert informing them that the employee has been successfully made the manager of their department and they are redirected back to the Employees page.

-SQL Statements used in PHP

```
"update DEPARTMENTS
 set manStart='".$manStart."'
 where deptID='".$empDeptID."'";
"update DEPARTMENTS
 set manID='".$empID."'
 where deptID='".$empDeptID."'";
```



Now Manager:



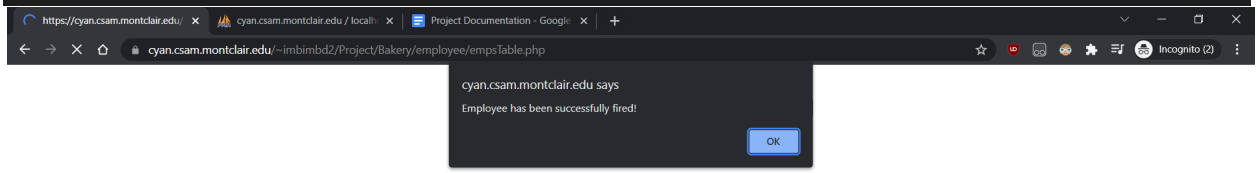
Fire Employee:

On the Employees Page, the CEO can click the button “Fire” to fire an employee from the company. This is done through the “empTable.php” script which uses a function from the “empFunctions.php” script to connect to the database & DELETE the specified

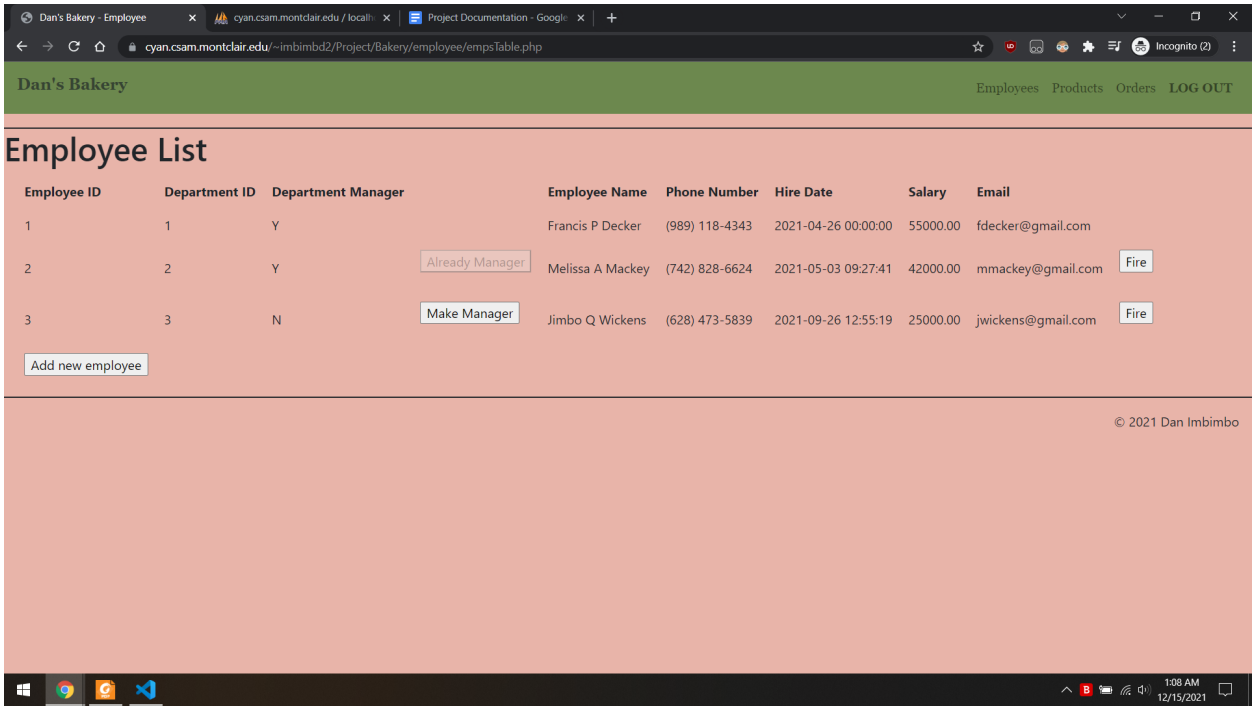
employee from the database. The function will also use two other functions from the “deptFunctions.php” script to connect to the database & SELECT all the manIDs for the departments and see if the employee’s ID matches it, and if it does then UPDATES the manager start time to the current time and sets the manager ID to the CEO’s employee ID. At which point the user will be presented with an alert informing them that the employee has been successfully fired and they are redirected back to the Employees page.

-SQL Statements used in PHP

```
"select manID from DEPARTMENTS";
"update DEPARTMENTS
 set manStart='".$manStart."'
 where manID='".$empID."'";
"update DEPARTMENTS
 set manID='1'
 where manID='".$empID."'";
"delete from EMPLOYEES
 where empID='".$empID."'";
```



Now Fired:

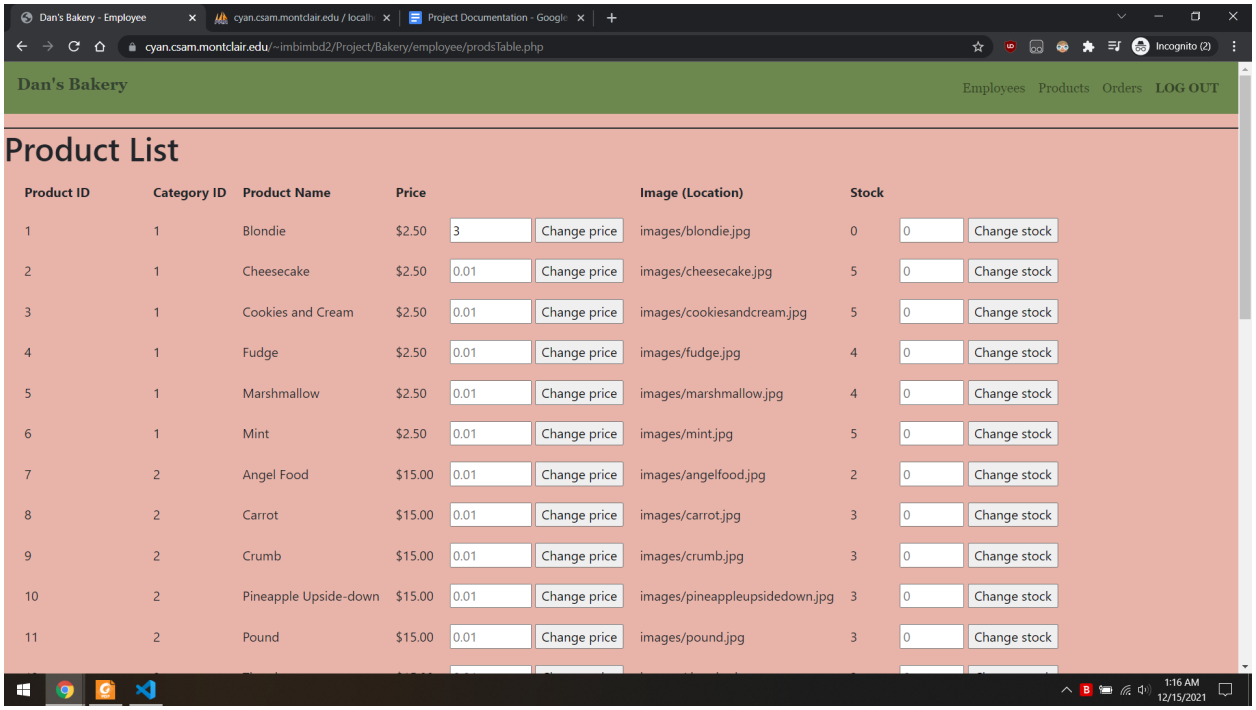


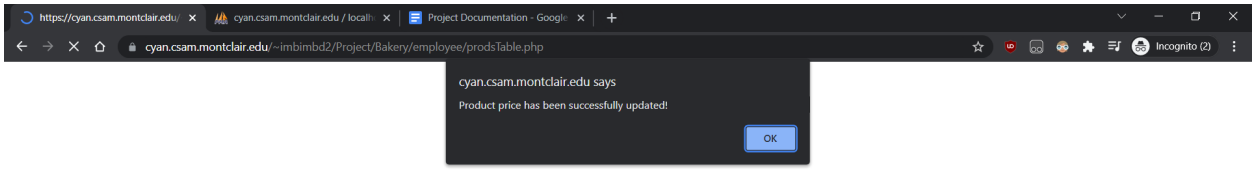
View Products/Change Price:

Once the Products page is selected from the navigation bar, the user is presented with a list of the products sold by the business. This is done by the “prodsTable.php” script which uses a function from the “productFunctions.php” script to connect to the database & SELECT all of the products in the database, with all of their attributes then displayed via the “prodsDisplay.php” script. In addition, as CEO the user is able to change the price and stock of products, as well as add new products. If the user enters an amount into the change price field and then clicks its button, the “prodsTable.php” script will use a function from “productFunctions.php” to connect to the database & UPDATE the price associated with the specified product. At which point the user will be presented with an alert informing them that the price has been successfully changed and they are redirected back to the Products page.

-SQL Statements used in PHP

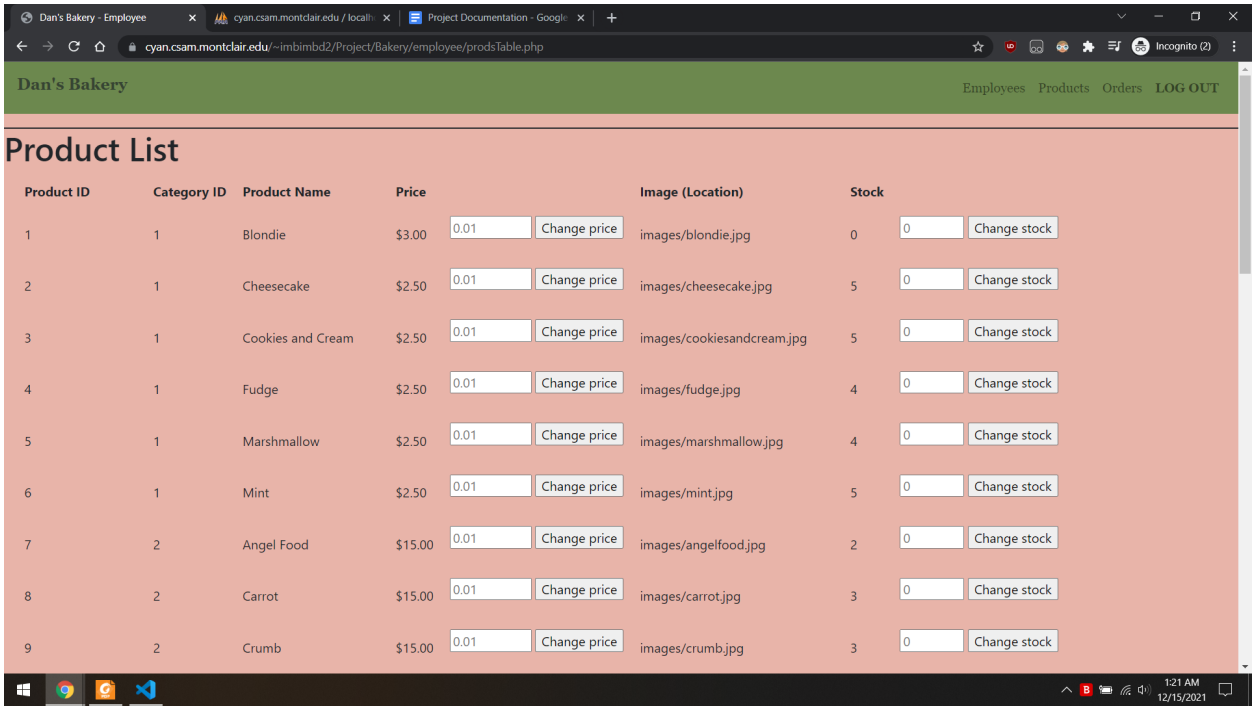
```
"select * from PRODUCTS";
"update PRODUCTS
 set price='".$price."'
 where prodID='".$prodID."'";
```


Dan's Bakery - Employee
cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/employee/prodsTable.php
Dan's Bakery
Employees Products Orders LOG OUT
Product List
Product ID Category ID Product Name Price Image (Location) Stock
1 1 Blondie $2.50 3 Change price images/blondie.jpg 0 0 Change stock
2 1 Cheesecake $2.50 0.01 Change price images/cheesecake.jpg 5 0 Change stock
3 1 Cookies and Cream $2.50 0.01 Change price images/cookiesandcream.jpg 5 0 Change stock
4 1 Fudge $2.50 0.01 Change price images/fudge.jpg 4 0 Change stock
5 1 Marshmallow $2.50 0.01 Change price images/marshmallow.jpg 4 0 Change stock
6 1 Mint $2.50 0.01 Change price images/mint.jpg 5 0 Change stock
7 2 Angel Food $15.00 0.01 Change price images/angelfood.jpg 2 0 Change stock
8 2 Carrot $15.00 0.01 Change price images/carrot.jpg 3 0 Change stock
9 2 Crumb $15.00 0.01 Change price images/crumb.jpg 3 0 Change stock
10 2 Pineapple Upside-down $15.00 0.01 Change price images/pineappleupsidedown.jpg 3 0 Change stock
11 2 Pound $15.00 0.01 Change price images/pound.jpg 3 0 Change stock
1:16 AM
12/15/2021


cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/employee/prodsTable.php
cyan.csam.montclair.edu says
Product price has been successfully updated!
OK


1:20 AM
12/15/2021

Now changed:

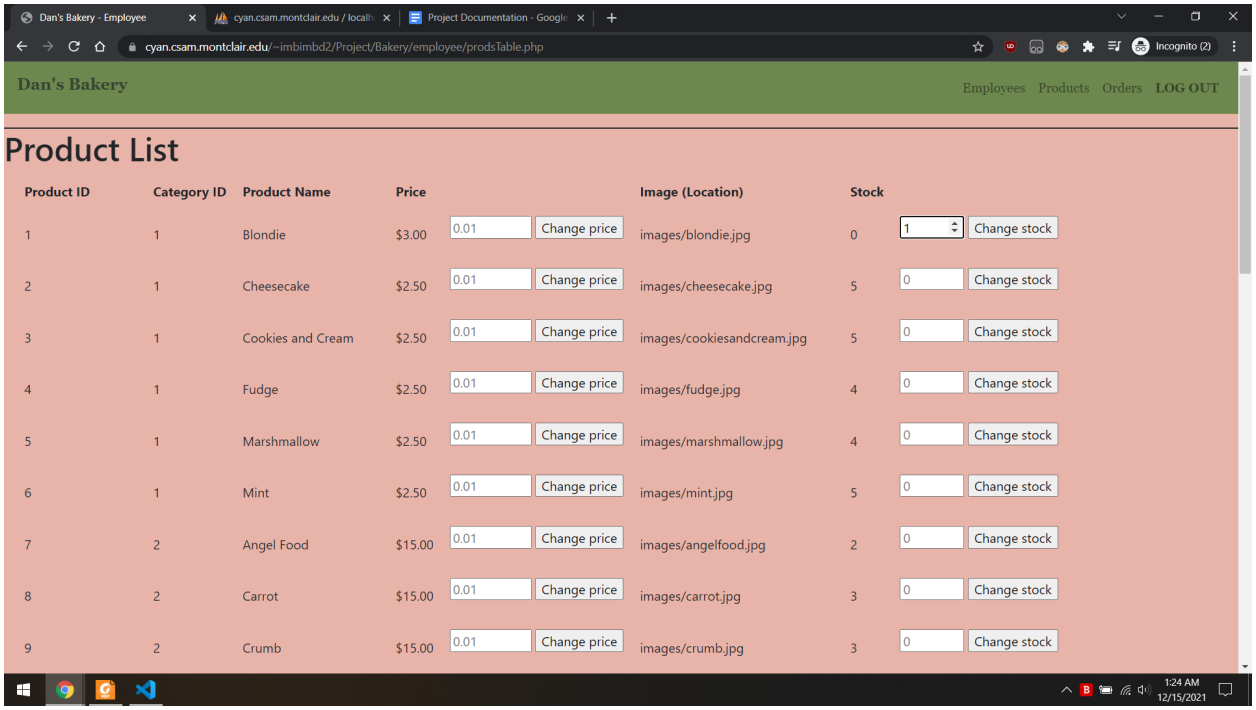


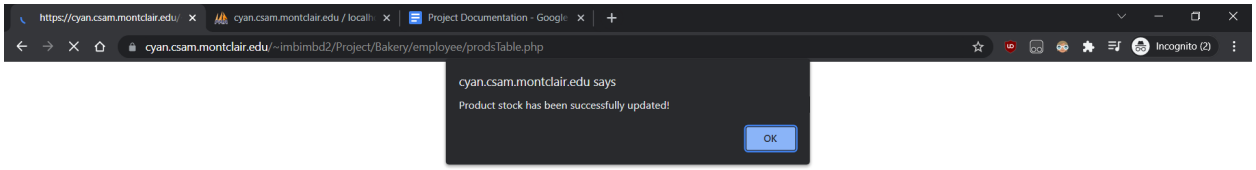
Change Stock:

On the Products page, if the CEO enters an amount into the change stock field and then clicks its button, the “prodsTable.php” script will use a function from “productFunctions.php” to connect to the database & UPDATE the stock associated with the specified product. At which point the user will be presented with an alert informing them that the stock has been successfully changed and they are redirected back to the Products page.

-SQL Statements used in PHP

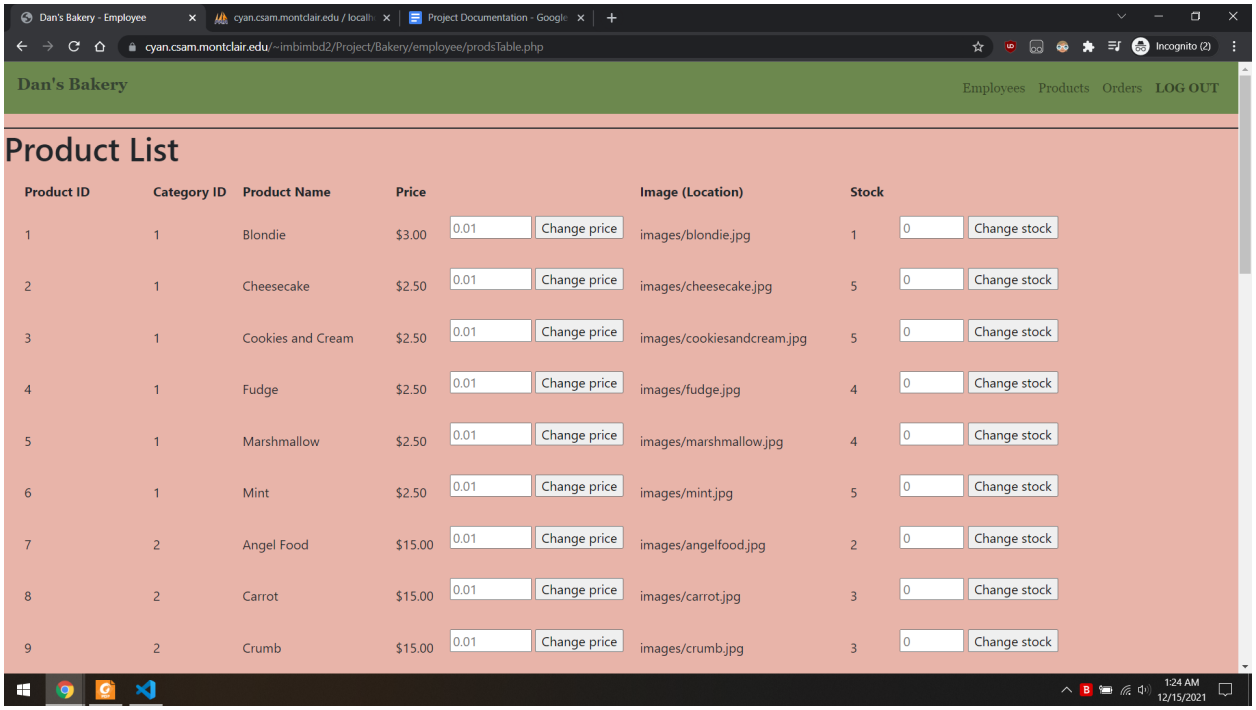
```
"update PRODUCTS
 set stock='".$stock."'
 where prodID='".$prodID."'";
```


Dan's Bakery - Employee
cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/employee/prodsTable.php
Incognito (2)
Dan's Bakery
Employees Products Orders LOG OUT
Product List
Product ID Category ID Product Name Price Image (Location) Stock
1 1 Blondie $3.00 0.01 Change price images/blondie.jpg 0 1 Change stock
2 1 Cheesecake $2.50 0.01 Change price images/cheesecake.jpg 5 0 Change stock
3 1 Cookies and Cream $2.50 0.01 Change price images/cookiesandcream.jpg 5 0 Change stock
4 1 Fudge $2.50 0.01 Change price images/fudge.jpg 4 0 Change stock
5 1 Marshmallow $2.50 0.01 Change price images/marshmallow.jpg 4 0 Change stock
6 1 Mint $2.50 0.01 Change price images/mint.jpg 5 0 Change stock
7 2 Angel Food $15.00 0.01 Change price images/angelfood.jpg 2 0 Change stock
8 2 Carrot $15.00 0.01 Change price images/carrot.jpg 3 0 Change stock
9 2 Crumb $15.00 0.01 Change price images/crumb.jpg 3 0 Change stock
1:24 AM
12/15/2021


cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/employee/prodsTable.php
Incognito (2)
cyan.csam.montclair.edu says
Product stock has been successfully updated!
OK


1:24 AM
12/15/2021

Now changed:

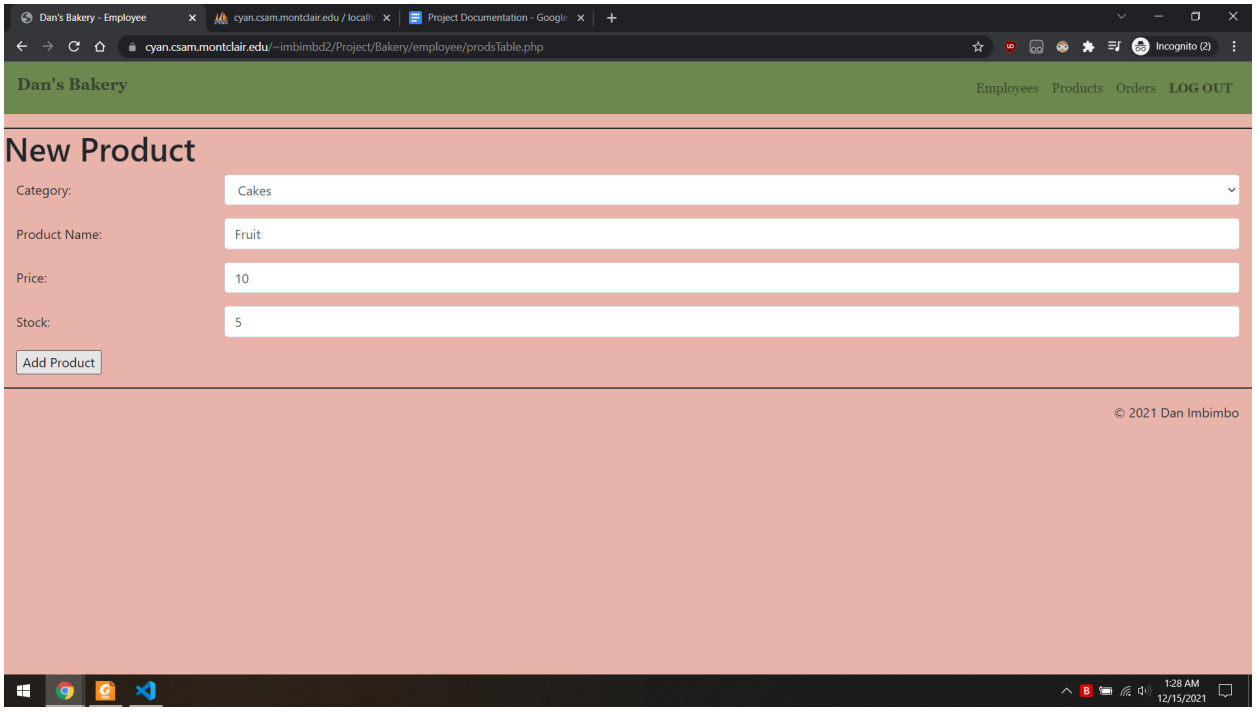


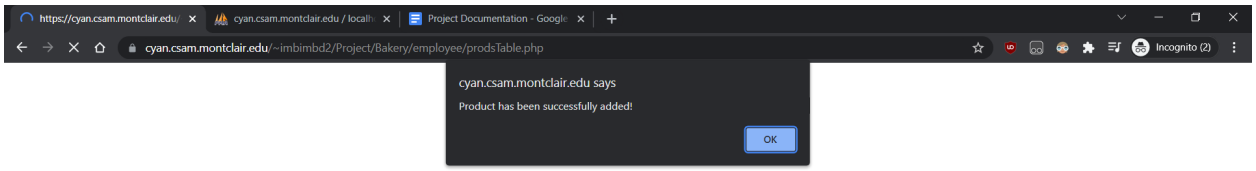
Add New Product:

At the bottom of the Products page, if the CEO clicks the “Add new product” button then they will be presented with a new page containing a form that they must fill out to create the product. All of the fields in the form are required, and as such it will not allow them to submit unless they are filled in. The fields also have specific patterns that must be followed to prevent invalid inputs, and will similarly not submit otherwise. Then, the “productsTable.php” script will use a function from the “productsFunctions.php” script to connect to the database, SELECT all of the products in the database, and check if the product name entered already exists in the same category. If the product name is found to be in use for that category, then the user will be informed of this via an error message. Once the fields are correctly filled and the user hits submit, the “prodsTable.php” script will use a function from the “productFunctions.php” script to connect to the database & INSERT all of the variables from the form (and an image location based on the product name) into the PRODUCTS table of the database. At which point the user will be presented with an alert informing them that the product has been successfully added and they are redirected back to the Products page.

-SQL Statements used in PHP

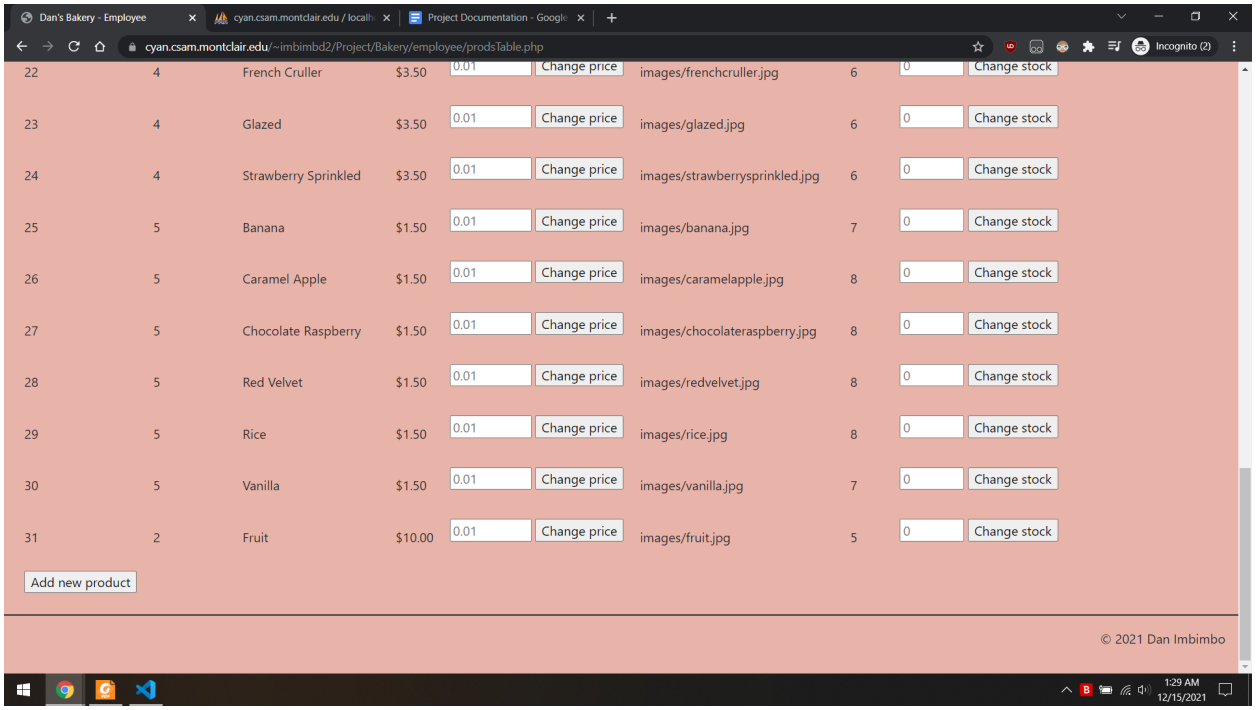
```
"select * from PRODUCTS";
"insert into PRODUCTS(categoryID,prodName,price,stock,imageLoc)
 values('".$categoryID."', '".$prodName."', '".$price."', '".$stock."',
'".$imageLoc."')";
```


Dan's Bakery - Employee
cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/employee/prodsTable.php
Incognito (2)
Dan's Bakery
Employees Products Orders LOG OUT
New Product
Category:
Cakes
Product Name:
Fruit
Price:
10
Stock:
5
Add Product
© 2021 Dan Imbimbo
1:28 AM
12/15/2021


cyan.csam.montclair.edu/~imbimbd2/Project/Bakery/employee/prodsTable.php
Incognito (2)
cyan.csam.montclair.edu says
Product has been successfully added!
OK


1:29 AM
12/15/2021

Now added:

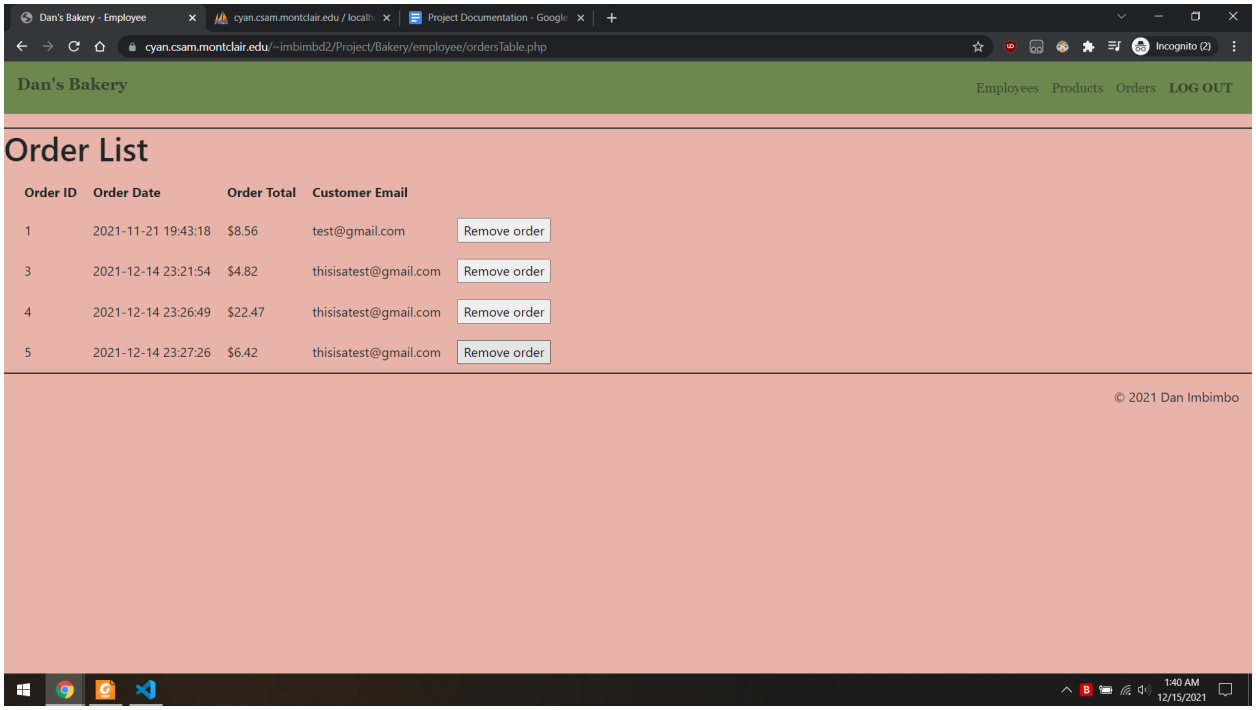


View Orders/Remove Order:

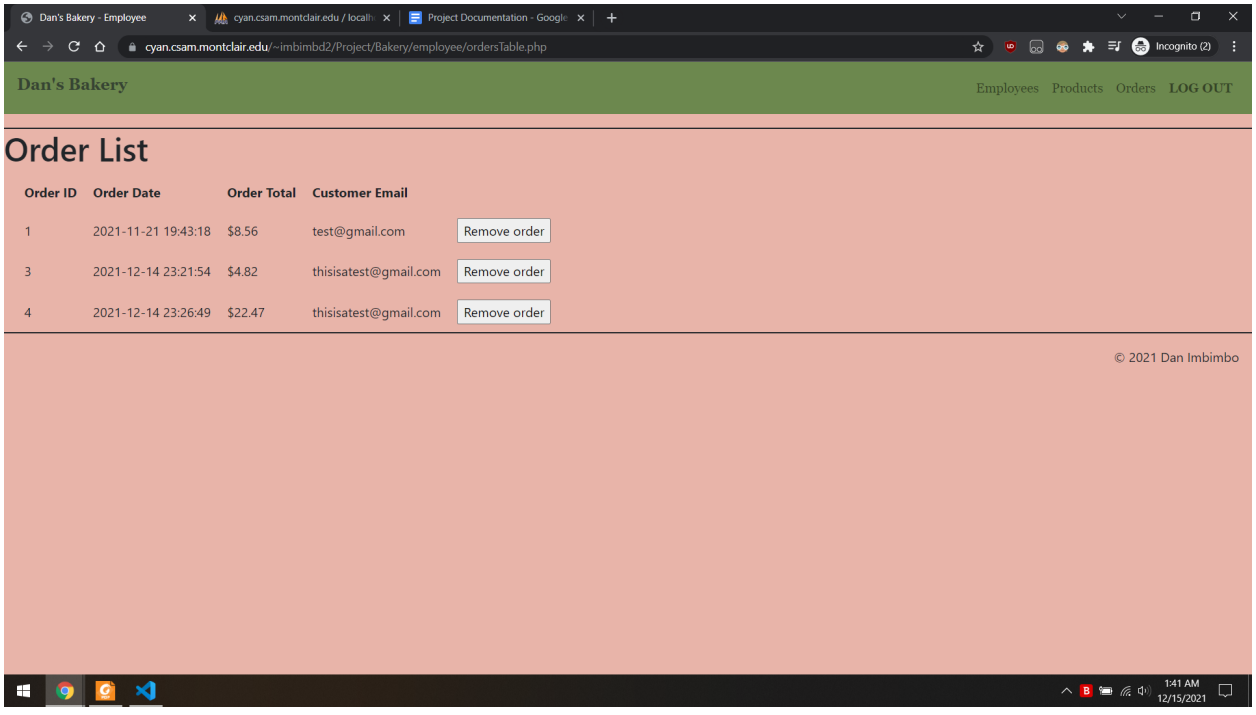
Once the Orders page is selected from the navigation bar, the user is presented with a list of the orders placed by customers. This is done by the “ordersTable.php” script which uses a function from the “orderFunctions.php” script to connect to the database & SELECT all of the orders in the database, with all of their attributes then displayed via the “ordersDisplay.php” script. In addition, as CEO the user is able to remove orders. If the user clicks the “Remove order” button, the “ordersTable.php” script will use a function from “orderFunctions.php” to connect to the database, DELETE the order items associated with the specified order ID, and also DELETE the order that is associated with that same order ID. At which point the user will be redirected to the updated order display list.

-SQL Statements used in PHP

```
"select * from ORDERS
 order by ordDate ASC";
"delete from ORDERITEMS
 where orderID='".$orderID."'";
"delete from ORDERS
 where ordID='".$orderID."'";
```



Removed:



Analysis of Database Issues

Normal Forms

My database is at least in 1st normal form because there are no multi-valued or composite attributes among the tables. In terms of it being in the 2nd normal form, I

slightly am unsure if it can, technically speaking, be defined as such, however I personally do believe it to be so. The main issue, again technically speaking, is that the ORDERITEMS table would have an “update anomaly” if the price was changed in the PRODUCTS table for an item since the cost in ORDERITEMS wouldn’t be updated to reflect it, however I don’t actually view this as an update anomaly because the cost of an already existing order should not change just because the price of an item is increased afterwards. Therefore, I do believe the database to be in 2nd normal form because aside from the ORDERITEMS table every other table has a single-attribute primary key, and so they don’t have any partial dependencies. When it comes to the 3rd normal form I am admittedly a little more unsure, I don’t believe there are any transitive dependencies among my entities, but I could be missing something.

In terms of the database design, there is no data redundancy and all of the data is reachable.

The attributes that would be NOT NULL per each table are as follows:
-CATEGORIES: catID and catName (since both are being used by the website’s frontend to display the categories and list the products).
-PRODUCTS: prodID, categoryID, prodName (since these three are used within the frontend to display the product list and list them by category, as well as place them in the session cart for orders); price (to calculate the orderTotal and the cart’s subtotal within the frontend); and stock (to determine if an item can be added to the cart or not within the front end).
-DEPARTMENTS: depID and depName (since both are used for listing and categorizing the employees within the employee view of the website’s frontend).
-EMPLOYEES: empID (for listing the employees within the employee view of the website’s frontend); empEmail (for logging into the employee account within the website’s frontend and for listing the employees within the CEO’s view of the website’s frontend); empPass (for logging into the employee account within the website’s frontend); fName, mInit, lName (for listing the employees within the employee view of the website’s frontend); empPhone (for listing the employees within the CEO’s view of the website’s frontend); hireDate (for listing the employees within the CEO’s view of the website’s frontend); salary (for listing the employees within the CEO’s view of the website’s frontend); departmentID (for categorizing the employees within the employee view of the website’s frontend).
-CUSTOMERS:custEmail (for logging into the customer account within the website’s frontend, for placing orders, and for listing a customer’s order history); custPass (for logging into the customer account within the website’s frontend); fName, mInit, lName, custPhone, street, city, state, and zip (for listing within a customer’s order history).

-ORDERS: ordID, ordTotal (for placing orders, for listing within a customer's order history, and for listing the orders within the employee view of the website's frontend); and orderDate & customerEmail (for listing & ordering within a customer's order history and for listing the orders within the employee's view of the website's frontend).
-ORDERITEMS: ordID, productID (for placing orders and for listing within a customer's order history); and quantity (for calculating the total of an order and for listing within a customer's order history).

NULLs are expected for the ordTotal, the manID, and the manStartDate, since they all default to NULL. These nulls are allowed primarily for the initial query that populates the database outside of the website's frontend because: ordTotal is calculated from the sum of the ORDERITEMS that are associated with a given orderID, and so ORDERITEMS can only be created after an orderID has been created, meaning that the order initially needs to be inserted with a null total and updated after the orderItems are then inserted (due to the inability to simply perform the calculation beforehand on the frontend as would be done every other time); manID is only added to the DEPARTMENTS table after the EMPLOYEES table has been populated with the employees who will be updated as the department managers; and manStartDate is null upon the initial query for the same reason as the manID.

Testing

I tested the website by slowly working my way through each script I was writing, starting at the login page, then working my way through the customer view scripts, and finally moving on to the employee/ceo scripts. I utilized XAMPP throughout the project's development for debugging and would tackle any errors as they came about. I tested the cart and order history in a similar manner, through XAMPP, by seeing if my intended actions would take place, one by one, and if not then trying to break the process down and find out where it was going awry.

When migrating over to CPANEL, which I did once I completed the customer side of the website, I had a similar experience of trying to figure out what was going wrong with my code in that environment by starting at the login page. Once I made my way to the customer view and through the product list, I was mostly able to figure out what the wrinkles were in my code in terms of getting it to work on CPANEL, so it was fairly easy to simply alter everything else to match a similar syntax. For example, starting a session seemingly must be on the first line of a script in CPANEL, whereas in XAMPP it did not matter where I started the session, so it was just a matter of moving these statements all to the top of their respective scripts. Another example was that any SQL statements seemingly required the tables to be capitalized in CPANEL, whereas in XAMPP the

case of the tables in the statement didn't seem to matter at all, so I just had to adjust all my functions accordingly.

Security

When it comes to the security of the website I have a lot of input validation across all of the accessible fields, in the form of if statements, specified patterns, and input types, to ensure that only proper inputs can be submitted to the database. The most significant form of security I have for the website, though, is in the login page, which ensures that only user accounts on the database can access any of the views of the website. And that users cannot access views that they are not permitted to (such as customers viewing the employee side, or employees having the functionalities of the CEO). The login function also has password verification to ensure that no one can access accounts that they shouldn't have access to, and the passwords stored on the database are hashed to avoid any levels of security being viewable in plaintext.

That said, when I tried to go further and prevent SQL injection via the methods specified in chapter 11 of the Welling book, I got all sorts of errors through XAMPP using the "get_magic_quote_gpc" function. This function is seemingly deprecated, and even removed from PHP versions above 8.0, and my version of XAMPP has PHP v8.0.2, so this is almost certainly why I was receiving those errors. Therefore I wasn't able to go about preventing these injections in the way specified in class, and as such had no idea about how to prevent them at all, so unfortunately I don't have that level of security in my website.

Conclusion

I pretty much got everything done with the website that I wanted to, I even went so far as to add an additional view with further functionalities in the form of the CEO. Everything that I have incorporated has been thoroughly tested by myself and works without issue, on CPANEL and on XAMPP. However, if I had more time there are a few things that I know I would've added, such as the ability for the CEO to remove products from the database, allowing for the CEO to remove an existing department manager without simply firing them or making another employee that department's manager, more security features such as hashing of email address and other sensitive information, allowing employees to view the details of orders that have been placed like customers can do for their order history, and figuring out how to protect from SQL Injections in a different manner than what wasn't working for me.

This was certainly a challenging project, especially since I worked on it by myself, but I'm glad I was able to meet the expectations I had for myself. All in all, I don't think this documentation nearly lives up to the website I created, but hopefully it is sufficient.

Appendix

Code Dump

header.php

```
<!DOCTYPE html>
<html>
<!-- the head section -->
<head>
	<title>Login/Register</title>
	<meta charset = "utf-8">
	<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
	<!-- Bootstrap CSS -->
	<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
	<!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

	<style>
		#email:valid {
			background-color: #E8F0FE
		}
		header {
			border-bottom: 2px solid;
			padding-bottom: 1em;
		}
		footer {
			border-top: 2px solid;
			padding: 1em 1em;
			text-align: right;
		}
		body {
			background-color: #E8B4A9;
		}
		th {
			padding: 10px 10px;
		}
		td {
			padding: 10px 10px;
		}
		.picture{
			padding-right: 0px;
		}
		#option{
			background: none;
			border: none;
			color: #007BFF;
		}
		#option:hover {
			text-decoration: underline;
			color: #0056B3;
		}
	</style>
</head>

<!-- the body section -->
<body>
	<header>
		<!--Navbar START-->
		<nav class="navbar navbar-expand-sm navbar-light sticky-top" style="background-color: #6C884E; color:
#364433; font-family: Georgia, serif;">
			<a class="navbar-brand h1"><b>Dan's Bakery</b></a>
```

```
                </nav>
                <!--Navbar END-->
        </header>
```

custHeader.php

```
<!DOCTYPE html>
<html>
<!-- the head section -->
<head>
        <title>Dan's Bakery - Customer</title>
        <meta charset = "utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
        <!-- Bootstrap CSS -->
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <!-- Optional JavaScript -->
   <!-- jQuery first, then Popper.js, then Bootstrap JS -->
   <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

        <style>
                #email:valid {
                        background-color: #E8F0FE
                }
                header {
                        border-bottom: 2px solid;
                        padding-bottom: 1em;
                }
                footer {
                        border-top: 2px solid;
                        padding: 1em 1em;
                        text-align: right;
                }
                body {
                        background-color: #E8B4A9;
                }
                th {
                        padding: 10px 10px;
                }
                td {
                        padding: 10px 10px;
                }
                .picture{
                        padding-right: 0px;
                }
                #option{
                        background: none;
                        border: none;
                        color: #007BFF;
                }
                #option:hover {
                        text-decoration: underline;
                        color: #0056B3;
                }
        </style>
</head>

<!-- the body section -->
<body>
        <header>
                <!--Navbar START-->
      <nav class="navbar navbar-expand-sm navbar-light sticky-top" style="background-color: #6C884E; color: #364433; font-family:
Georgia, serif;">
          <a class="navbar-brand h1"><b>Dan's Bakery</b></a>
```

```
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsingNav">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="collapsingNav">
            <ul class="navbar-nav ml-auto">
                                        <li class="nav-item">
                    <a class="nav-link" href="account.php" style="font-weight:bold"><?php echo $_SESSION["email"] ?></a>
                </li>
                                                <li class="nav-item">
                    <a class="nav-link" href="about.php">about</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="products.php">products</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="cart.php">cart</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="../index.php" style="font-weight:bold">LOG OUT</a>
                </li>
            </ul>
        </div>
    </nav>
    <!--Navbar END-->
        </header>
```

empHeader.php

```
<!DOCTYPE html>
<html>
<!-- the head section -->
<head>
        <title>Dan's Bakery - Customer</title>
        <meta charset = "utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
        <!-- Bootstrap CSS -->
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

        <style>
                #email:valid {
                        background-color: #E8F0FE
                }
                header {
                        border-bottom: 2px solid;
                        padding-bottom: 1em;
                }
                footer {
                        border-top: 2px solid;
                        padding: 1em 1em;
                        text-align: right;
                }
                body {
                        background-color: #E8B4A9;
                }
                th {
                        padding: 10px 10px;
                }
                td {
                        padding: 10px 10px;
```

```
                        }
                        .picture{
                                padding-right: 0px;
                        }
                        #option{
                                background: none;
                                border: none;
                                color: #007BFF;
                        }
                        #option:hover {
                                text-decoration: underline;
                                color: #0056B3;
                        }
                </style>
</head>

<!-- the body section -->
<body>
        <header>
                <!--Navbar START-->
    <nav class="navbar navbar-expand-sm navbar-light sticky-top" style="background-color: #6C884E; color: #364433; font-family:
Georgia, serif;">
      <a class="navbar-brand h1"><b>Dan's Bakery</b></a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsingNav">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="collapsingNav">
        <ul class="navbar-nav ml-auto">
                                        <li class="nav-item">
          <a class="nav-link" href="account.php" style="font-weight:bold"><?php echo $_SESSION["email"] ?></a>
        </li>
                                                <li class="nav-item">
          <a class="nav-link" href="about.php">about</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="products.php">products</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="cart.php">cart</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="../index.php" style="font-weight:bold">LOG OUT</a>
        </li>
        </ul>
      </div>
    </nav>
    <!--Navbar END-->
        </header>
```

footer.php

```
  <footer>
    <p class="copyright">
      &copy; <?php echo date("Y"); ?> Dan Imbimbo
    </p>
  </footer>
</body>
</html>
```

index.php

```
<?php session_start(); //Starts the session ?>

<!--Includes the defined header-->
<?php include 'view/header.php'; ?>

<main>
        <div class="container-fluid">
                <h1 style="text-align:center; padding: .5em;">Account Login</h1>
                <form action = "checkLogin.php" method = "post">
                        <div class="form-group row">
```

```
                                        <label for="email" class="col-sm-2 col-form-label">Email:</label>
                                        <div class="col-sm-10">
                                                <input type="email" class="form-control" id="email" name="email"
placeholder="Enter email" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="pass" class="col-sm-2 col-form-label">Password:</label>
                                        <div class="col-sm-10">
                                                <input type="password" class = "form-control" id="pass" name="pass"
placeholder="Enter password" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label class="col-sm-2 col-form-label"></label>
                                        <div class="col-sm-10">
                                                <button type="submit" class="btn btn-primary">Login</button>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label class="col-sm-2 col-form-label"></label>
                                        <div class="col-sm-10">
                                                <p>Not a customer?</p>
                                                <a href="register/register.php" class="btn btn-success  text-black">Press to
register</a>
                                        </div>
                                </div>
                                <!--Checks if success variable has been set in session array,
                                        and if so displays the defined message-->
                                <?php
                                        if(isset($_SESSION["success"])){
                                                echo $_SESSION["success"];
                                        }
                                ?>
                                <!--Checks if error variable has been set in session array,
                                        and if so displays the defined message-->
                                <?php
                                        if(isset($_SESSION["error"])){
                                                echo $_SESSION["error"];
                                        }
                                ?>
                        </form>
                </div>
</main>
<!--Ends old sessions-->
<?php
        //unsets all session variables
        $_SESSION = array();
        //removes session ID
        session_destroy();
?>

<!--Includes the defined footer-->
<?php include 'view/footer.php'; ?>
```

checkLogin.php

```
<?php session_start(); //Resumes the session ?>

<?php
// Defines error message
        $error = "<p style=\"color:red\">Invalid login. Please try again.</p>";

// Gets the entered form data
   $email = $_POST['email'];
   $pass = $_POST['pass'];

// Retrieves array of customers from database
        //open database connection - host, username, password, database
        @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
        if (mysqli_connect_errno()) {
```

```
                echo 'Error: Could not connect to database.  Please try again later.';
                exit;
        }
        $query = "select * from CUSTOMERS";
        $result = $db->query($query);
        while($row = $result->fetch_assoc()) {
                $customers[] = $row;
         }
        $result->free();
        //close database connection
        $db->close();

// Compares email from form with emails from database, if found
// then compares the pass from form with the hashed password from database
        $foundCust = false;
        foreach($customers as $customer){
                if($customer['custEmail'] == $email){
                        if(password_verify($pass,$customer['custPass'])){
                                $foundCust = true;
                                break;
                        }
                }
        }

// Checks if the email/password combination was found
        if($foundCust == true){
                // Sets name variable in session array
                $_SESSION["email"] = $email;
                echo "<script> document.location='customer/about.php'; </script>";
        }
// If not, then checks if the user is an employee
        else{
                // Retrieves array of employees from database
                        //open database connection - host, username, password, database
                        @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
                        if (mysqli_connect_errno()) {
                                echo 'Error: Could not connect to database.  Please try again later.';
                                exit;
                        }
                        $query = "select * from EMPLOYEES";
                        $result = $db->query($query);
                        while($row = $result->fetch_assoc()) {
                                $employees[] = $row;
                        }
                        $result->free();
                        //close database connection
                        $db->close();

                // Compares email from form with emails from database, if found
                // then compares the pass from form with the password from database
                        $foundEmp = false;
                        foreach($employees as $employee){
                                if($employee['empEmail'] == $email){
                                        if(password_verify($pass,$employee['empPass'])){
                                                $foundEmp = true;
                                                break;
                                        }
                                }
                        }

                // Checks if the email/password combination was found
                        if($foundEmp == true){
                                // Sets name variable in session array
                                $_SESSION["email"] = $email;
                                echo "<script> document.location='employee/empsTable.php'; </script>";
                        }
                        else{
                                // Sets error variable in session array
                                $_SESSION["error"] = $error;
                                // Returns to index (login screen)
```

```
                                        echo "<script> document.location='index.php'; </script>";
                                }
        }
?>
```

register.php

```
<?php session_start(); //Resumes the session ?>

<!--Includes the defined header-->
<?php include '../view/header.php'; ?>

<main>
        <div class="container-fluid">
                <h1 style="text-align:center; padding-top: .5em;">Customer Registration</h1>
                <form action = "commit.php" method = "post">
                        <div class="form-group row">
                                <label class="col-sm-2 col-form-label"><a href="../index.php" class="btn btn-success">Go
Back</a></label>
                        </div>
                        <!--Checks if error variable has been set in session array,
                                and if so displays the defined message-->
                                <?php
                                if(isset($_SESSION["error"])){
                                        echo $_SESSION["error"];
                                }
                        ?>
                        <div class="form-group row">
                                <label for="email" class="col-sm-2 col-form-label">Email:</label>
                                <div class="col-sm-10">
                                        <input type="email" class="form-control" id="email" name="email"
placeholder="Enter email" required>
                                </div>
                        </div>
                        <div class="form-group row">
                                <label for="pass" class="col-sm-2 col-form-label">Password:</label>
                                <div class="col-sm-10">
                                        <input type="password" class = "form-control" id="pass" name="pass"
placeholder="Enter password" required>
                                </div>
                        </div>
                        <div class="form-group row">
                                <label for="fname" class="col-sm-2 col-form-label">First Name:</label>
                                <div class="col-sm-10">
                                        <input type="text" class="form-control" id="fname" name="fname"
placeholder="Enter first name" pattern="[A-Za-z]+" required>
                                </div>
                        </div>
                        <div class="form-group row">
                                <label for="zip" class="col-sm-2 col-form-label">Middle Initial:</label>
                                <div class="col-sm-10">
                                        <input type = "text" class = "form-control" id="minit" name="minit"
placeholder="Enter middle initial" maxlength=1 pattern="[A-Za-z]+">
                                </div>
                        </div>
                        <div class="form-group row">
                                <label for="lname" class="col-sm-2 col-form-label">Last Name:</label>
                                <div class="col-sm-10">
                                        <input type="text" class="form-control" id="lname" name="lname"
placeholder="Enter last name" pattern="[A-Za-z]+" required>
                                </div>
                        </div>
                        <div class="form-group row">
                                <label for="phone" class="col-sm-2 col-form-label">Phone:</label>
                                <div class="col-sm-10">
                                        <input type = "tel" class = "form-control" id = "phone" name ="phone" placeholder
= "(973) 123-4567" pattern = "\(\d{3}\) +\d{3}-\d{4}" required>    (###) ### ####
                                </div>
                        </div>
                        <div class="form-group row">
                                <label for="street" class="col-sm-2 col-form-label">Street:</label>
```

```
                                        <div class="col-sm-10">
                                                <input type = "text" class = "form-control" id = "street" name ="street" placeholder
= "123 Some Street" pattern="[A-Za-z0-9\s]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="city" class="col-sm-2 col-form-label">City:</label>
                                        <div class="col-sm-10">
                                                <input type = "text" class = "form-control" id = "city" name ="city" placeholder =
"City" pattern="[A-Za-z]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="state" class="col-sm-2 col-form-label">State:</label>
                                        <div class="col-sm-10">
                                                <select class = "form-control" id = "state" name ="state" placeholder = "AL"
required>
                                                        <option value="AL">(AL) Alabama</option>
                                                        <option value="AK">(AK) Alaska</option>
                                                        <option value="AZ">(AZ) Arizona</option>
                                                        <option value="AR">(AR) Arkansas</option>
                                                        <option value="CA">(CA) California</option>
                                                        <option value="CO">(CO) Colorado</option>
                                                        <option value="CT">(CT) Connecticut</option>
                                                        <option value="DE">(DE) Delaware</option>
                                                        <option value="DC">(DC) District Of Columbia</option>
                                                        <option value="FL">(FL) Florida</option>
                                                        <option value="GA">(GA) Georgia</option>
                                                        <option value="HI">(HI) Hawaii</option>
                                                        <option value="ID">(ID) Idaho</option>
                                                        <option value="IL">(IL) Illinois</option>
                                                        <option value="IN">(IN) Indiana</option>
                                                        <option value="IA">(IA) Iowa</option>
                                                        <option value="KS">(KS) Kansas</option>
                                                        <option value="KY">(KY) Kentucky</option>
                                                        <option value="LA">(LA) Louisiana</option>
                                                        <option value="ME">(ME) Maine</option>
                                                        <option value="MD">(MD) Maryland</option>
                                                        <option value="MA">(MA) Massachusetts</option>
                                                        <option value="MI">(MI) Michigan</option>
                                                        <option value="MN">(MN) Minnesota</option>
                                                        <option value="MS">(MS) Mississippi</option>
                                                        <option value="MO">(MO) Missouri</option>
                                                        <option value="MT">(MT) Montana</option>
                                                        <option value="NE">(NE) Nebraska</option>
                                                        <option value="NV">(NV) Nevada</option>
                                                        <option value="NH">(NH) New Hampshire</option>
                                                        <option value="NJ">(NJ) New Jersey</option>
                                                        <option value="NM">(NM) New Mexico</option>
                                                        <option value="NY">(NY) New York</option>
                                                        <option value="NC">(NC) North Carolina</option>
                                                        <option value="ND">(ND) North Dakota</option>
                                                        <option value="OH">(OH) Ohio</option>
                                                        <option value="OK">(OK) Oklahoma</option>
                                                        <option value="OR">(OR) Oregon</option>
                                                        <option value="PA">(PA) Pennsylvania</option>
                                                        <option value="RI">(RI) Rhode Island</option>
                                                        <option value="SC">(SC) South Carolina</option>
                                                        <option value="SD">(SD) South Dakota</option>
                                                        <option value="TN">(TN) Tennessee</option>
                                                        <option value="TX">(TX) Texas</option>
                                                        <option value="UT">(UT) Utah</option>
                                                        <option value="VT">(VT) Vermont</option>
                                                        <option value="VA">(VA) Virginia</option>
                                                        <option value="WA">(WA) Washington</option>
                                                        <option value="WV">(WV) West Virginia</option>
                                                        <option value="WI">(WI) Wisconsin</option>
                                                        <option value="WY">(WY) Wyoming</option>
                                                </select>
                                        </div>
```

```
                                </div>
                                <div class="form-group row">
                                        <label for="zip" class="col-sm-2 col-form-label">Zip Code:</label>
                                        <div class="col-sm-10">
                                                <input type = "text" class = "form-control" id = "zip" name ="zip" placeholder =
"01234" maxlength=5 pattern ="[0-9]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label class="col-sm-2 col-form-label"></label>
                                        <div class="col-sm-10">
                                                <button type="submit" class="btn btn-primary">Register Me</button>
                                        </div>
                                </div>
                        </form>
                        <!--Unsets error variable in session arrray-->
                        <?php
                                unset($_SESSION["error"]);
                        ?>
                </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

commit.php

```
<?php session_start(); //Resumes the session ?>

<?php
require_once('../model/empFunctions.php');
require_once('../model/custFunctions.php');

// Defines error & success messages
        $in_use = "<p style=\"color:red\">Email already in database, please choose another email.</p>";
        $success = "<p style=\"color:green\">Customer account successfully created.</p>";

// Gets the entered form data
        $email = $_POST['email'];
        $pass = $_POST['pass'];
        $fname = $_POST['fname'];
        $minit = $_POST['minit'];
        $lname = $_POST['lname'];
        $phone = $_POST['phone'];
        $street = $_POST['street'];
        $city = $_POST['city'];
        $state = $_POST['state'];
        $zip = $_POST['zip'];

//checks if email is already being used by a customer
        $custFound = checkCustEmail($email);

//checks if email is already being used by a employee
        if($custFound == false){
                $empFound = checkEmpEmail($email);
        }

// Checks if email was found to be already in use
        if($custFound == true || $empFound == true){
                // Sets error variable in session array
                $_SESSION["error"] = $in_use;
                // Returns to register screen
                echo "<script> document.location='register.php'; </script>";
        }
        else{
                // Sets success variable in session array
                $_SESSION["success"] = $success;
                // Encrypts the entered password
                $hashedPass = password_hash($pass, PASSWORD_DEFAULT);
                // Adds customer to the database
                addCust($email,$hashedPass,$fname,$minit,$lname,$phone,$street,$city,$state,$zip);
```

```
                        // Returns to index (login screen)
                        echo "<script> document.location='../index.php'; </script>";
                }
?>
```

categoryFunctions.php

```
<?php
// Retrieves all categories
function getCats() {
   //open database connection - host, username, password, database
                @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
                if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
                }
                $query = "select * from CATEGORIES
               order by catID";
                $result = $db->query($query);
                while($row = $result->fetch_assoc()) {
                        $categories[] = $row;
                }
                $result->free();
                //close database connection
                $db->close();

   return $categories;
}

// Retrieves the category associated with the given categoryID
function getCatName($categoryID) {
   //open database connection - host, username, password, database
                @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
                if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
                }
                $query = "select * from CATEGORIES
               where catID='".$categoryID."'";
                $result = $db->query($query);
                while($row = $result->fetch_assoc()) {
                        $categoryName = $row['catName'];
                }
                $result->free();
                //close database connection
                $db->close();
   return $categoryName;
}
?>
```

custFunctions.php

```
<?php
   //Resumes session (if not already resumed)
   if(!isset($_SESSION))
   {
      session_start();
   }
?>

<?php
// Retrieves all customers from database
function getCusts() {
   //open database connection - host, username, password, database
                @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
                if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
                }
                $query = "select * from CUSTOMERS";
                $result = $db->query($query);
                while($row = $result->fetch_assoc()) {
```

```
                $customers[] = $row;
            }
            $result->free();
            //close database connection
            $db->close();
    return $customers;
}

// Retrieves the customer account information associated with the given email address
function getCustByEmail($custEmail){
    //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                    echo 'Error: Could not connect to database.  Please try again later.';
                    exit;
            }
            $query = "select * from CUSTOMERS
           where custEmail='".$custEmail."'";
            $result = $db->query($query);
            $customer = $result->fetch_assoc();
            $result->free();
            //close database connection
            $db->close();
    return $customer;
}

// Checks if the entered customer email already exists in the database
function checkCustEmail($email){
    $found = false;

    $customers = getCusts();

    // compares email from form with customer emails from database,
    // if found then flags boolean variable
    foreach($customers as $customer){
        if($customer['custEmail'] == $email){
            $found = true;
            break;
        }
    }

            return $found;
}

// Inserts a new customer into the database
function addCust($email,$hashedPass,$fName,$mInit,$lName,$custPhone,$street,$city,$state,$zip){
            //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                    echo 'Error: Could not connect to database.  Please try again later.';
                    exit;
            }
            $query = "insert into CUSTOMERS(custEmail, custPass, fName, mInit, lName, custPhone, street, city, state, zip)
                                    values('".$email."', '".$hashedPass."', '".$fName."', '".$mInit."', '".$lName."', '".$custPhone."',
'".$street."', '".$city."', '".$state."', '".$zip."')";
            $result = $db->query($query);
            //close database connection
            $db->close();
}

// Updates the account information associated with the given customer email
function updateCustInfo($fName, $mInit, $lName, $phone, $street, $city, $state, $zip, $custEmail){
            //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                    echo 'Error: Could not connect to database.  Please try again later.';
                    exit;
            }
            $query = "update CUSTOMERS
```

```
                                        set
fName='".$fName."',mInit='".$mInit."',lName='".$lName."',custPhone='".$phone."',street='".$street."',city='".$city."',state='".$state."',zip
='".$zip."'
                                        where custEmail='".$custEmail."'";
                $result = $db->query($query);
                //close database connection
                $db->close();
}

// Updates the account password associated with the given customer email
function updateCustPass($hashedPass, $custEmail){
                //open database connection - host, username, password, database
                @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
                if (mysqli_connect_errno()) {
                                echo 'Error: Could not connect to database.  Please try again later.';
                                exit;
                }
                $query = "update CUSTOMERS
                                        set custPass='".$hashedPass."'
                                        where custEmail='".$custEmail."'";
                $result = $db->query($query);
                //close database connection
                $db->close();
}
?>
```

deptFunctions.php

```
<?php
require_once('../model/empFunctions.php');

// Retrieves all departments from the database
function getDepts() {
   //open database connection - host, username, password, database
                @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
                if (mysqli_connect_errno()) {
                                echo 'Error: Could not connect to database.  Please try again later.';
                                exit;
                }
                $query = "select manID from DEPARTMENTS";
                $result = $db->query($query);
                while($row = $result->fetch_assoc()) {
                                $depts[] = $row;
                 }
                $result->free();
                //close database connection
                $db->close();
   return $depts;
}

// Determines if the given employee is the manager of their department
function checkDeptMan($empID){
   $depts = getDepts();
   $deptMan = false;

   foreach($depts as $dept){
      if($dept['manID'] == $empID){
         $deptMan = true;
         break;
      }
   }
   return $deptMan;
}

// Sets the given employee as the manager of their department
function makeDeptMan($empID){
   $empDeptID = getEmpDept($empID);
   date_default_timezone_set("America/New_York");
   $manStart = date("Y-m-d H:i:s");

   //open database connection - host, username, password, database
```

```
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                    echo 'Error: Could not connect to database.  Please try again later.';
                    exit;
            }
    //sets the manager start date to the current time
    $query1 = "update DEPARTMENTS
            set manStart='".$manStart."'
            where deptID='".$empDeptID."'";
              $result = $db->query($query1);
    //sets the managerID to the given employee's ID
              $query2 = "update DEPARTMENTS
            set manID='".$empID."'
            where deptID='".$empDeptID."'";
              $result = $db->query($query2);
              //close database connection
              $db->close();
}

// Removes the given employee from being the manager of their department,
// if they are found to be a department manager
function removeDeptMan($empID){
    date_default_timezone_set("America/New_York");
    $manStart = date("Y-m-d H:i:s");

    if(checkDeptMan($empID)){
        //open database connection - host, username, password, database
        @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
        if (mysqli_connect_errno()) {
            echo 'Error: Could not connect to database.  Please try again later.';
            exit;
        }
        //sets manager start date to current time
        $query1 = "update DEPARTMENTS
                set manStart='".$manStart."'
                where manID='".$empID."'";
        $result = $db->query($query1);
        //sets manager ID to CEO
        $query2 = "update DEPARTMENTS
                set manID='1'
                where manID='".$empID."'";
        $result = $db->query($query2);
        //close database connection
        $db->close();
    }
}
?>
```

empFunctions.php

```
<?php
require_once('../model/deptFunctions.php');

// Retrieves all employees from database
function getEmps() {
    //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                    echo 'Error: Could not connect to database.  Please try again later.';
                    exit;
            }
            $query = "select * from EMPLOYEES";
            $result = $db->query($query);
            while($row = $result->fetch_assoc()) {
                    $employees[] = $row;
              }
            $result->free();
            //close database connection
            $db->close();
    return $employees;
}
```

```
// Retrieves the department ID of the given employee
function getEmpDept($empID){
   //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
            }
            $query = "select departmentID from EMPLOYEES
           where empID ='".$empID."'";
            $result = $db->query($query);
            while($row = $result->fetch_assoc()) {
                        $departmentID = $row['departmentID'];
             }
            $result->free();
            //close database connection
            $db->close();
   return $departmentID;
}

// Removes employee with specified ID from database
function fireEmp($empID) {
   removeDeptMan($empID);
   //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
            }
            $query = "delete from EMPLOYEES
           where empID='".$empID."'";
            $result = $db->query($query);
            //close database connection
            $db->close();
}

// Checks if the entered employee email already exists in the database
function checkEmpEmail($email){
   $found = false;

   $employees = getEmps();

   // compares email from form with employee emails from database,
   // if found then flags boolean variable
   foreach($employees as $employee){
      if($employee['empEmail'] == $email){
         $found = true;
         break;
      }
   }

            return $found;
}

// Inserts new employee into database
function addEmp($empEmail,$hashedPass,$fName,$mInit,$lName,$empPhone,$salary,$departmentID){
            date_default_timezone_set("America/New_York");
   $hireDate = date("Y-m-d H:i:s");

   //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
            }
   $query = "insert into EMPLOYEES(empEmail,empPass,fName,mInit,lName,empPhone,hireDate,salary,departmentID)
                                    values('".$empEmail."', '".$hashedPass."', '".$fName."', '".$mInit."', '".$lName."', '".$empPhone."',
'".$hireDate."', '".$salary."', '".$departmentID."')";
   $result = $db->query($query);
```

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

orderFunctions.php

```
<?php
   //Resumes session (if not already resumed)
   if(!isset($_SESSION))
   {
      session_start();
   }
?>

<?php
// Add an item to the cart
function addItem($item) {
   //If item already exists in cart, increment quantity
   if (isset($_SESSION['cart'][$item['productID']])) {
      $_SESSION['cart'][$item['productID']]['quantity']++;
      return;
   }
		//Else adds item array to session cart
   else{
      $_SESSION['cart'][$item['productID']] = $item;
   }
}

// Gets total of cart items
function getTotal() {
   $subtotal = 0;
   foreach ($_SESSION['cart'] as $item) {
      $subtotal += ($item['price'] * $item['quantity']);
   }
   $total = number_format($subtotal * (1 + 0.07), 2);
   return $total;
}

// Inserts order into database
function addOrder($customerEmail){
   $orderTotal = getTotal();
   date_default_timezone_set("America/New_York");
   $orderDate = date("Y-m-d H:i:s");
   //open database connection - host, username, password, database
   @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
   if (mysqli_connect_errno()) {
      echo 'Error: Could not connect to database.  Please try again later.';
      exit;
   }
   $query = "insert into ORDERS(ordTotal, ordDate, customerEmail)
         values('".$orderTotal."', '".$orderDate."', '".$customerEmail."')";
   $result = $db->query($query);
   //close database connection
   $db->close();

   //add the order items to the database using the same orderID
   $orderID = getOrderID($orderDate,$orderTotal,$customerEmail);
   addOrderItems($orderID);
}

// Retrieve orderID for use with addOrderItems function
function getOrderID($orderDate, $orderTotal, $customerEmail){
   //open database connection - host, username, password, database
		@ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
		if (mysqli_connect_errno()) {
			echo 'Error: Could not connect to database.  Please try again later.';
			exit;
		}
		$query = "select ordID from ORDERS
	      where ordDate='".$orderDate."'
```

```
            and ordTotal ='".$orderTotal."'
            and customerEmail ='".$customerEmail."'";
              $result = $db->query($query);
              $order = $result->fetch_assoc();
              $result->free();
              //close database connection
              $db->close();
    return $order['ordID'];
}

// Inserts order items into OrderItems table
function addOrderItems($orderID){
    foreach ($_SESSION['cart'] as $item){
        $cost = number_format($item['quantity'] * $item['price'], 2);

        //open database connection - host, username, password, database
        @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
        if (mysqli_connect_errno()) {
            echo 'Error: Could not connect to database.  Please try again later.';
            exit;
        }
        $query = "insert into ORDERITEMS(orderID, productID, quantity, cost)
                values('".$orderID."', '".$item['productID']."', '".$item['quantity']."', '".$cost."')";
        $result = $db->query($query);
        //close database connection
        $db->close();
    }
}

// Retrieves all orders in the database
function getOrders() {
    //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
            }
            $query = "select * from ORDERS
                    order by ordDate ASC";
            $result = $db->query($query);
            while($row = $result->fetch_assoc()) {
                        $orders[] = $row;
             }
            $result->free();
            //close database connection
            $db->close();

    //determines if there are any orders to return
    //if not returns an empty array
    if(isset($orders)){
        return $orders;
    }
    else{
        return array();
    }
}

// Retrieves all orders by the given customer in the database
function getOrdersByEmail($customerEmail) {
    //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
            }
            $query = "select * from ORDERS
            where customerEmail='".$customerEmail."'
            order by ordDate ASC";
              $result = $db->query($query);
              while($row = $result->fetch_assoc()) {
```

```
                $orders[] = $row;
            }
            $result->free();
            //close database connection
            $db->close();

    //determines if there are any orders to return
    //if not returns an empty array
    if(isset($orders)){
        return $orders;
    }
    else{
        return array();
    }
}

// Retrieves the order specified by the given ID
function getOrderByID($orderID) {
    //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                    echo 'Error: Could not connect to database.  Please try again later.';
                    exit;
            }
            $query = "select * from ORDERS
          where ordID='".$orderID."'";
            $result = $db->query($query);
            $order = $result->fetch_assoc();
            $result->free();
            //close database connection
            $db->close();
    return $order;
}

// Retrieves order items for the given order ID
function getOrderItems($orderID){
    //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                    echo 'Error: Could not connect to database.  Please try again later.';
                    exit;
            }
            $query = "select * from ORDERITEMS
          where orderID='".$orderID."'
          order by cost ASC";
            $result = $db->query($query);
            while($row = $result->fetch_assoc()) {
                    $orderItems[] = $row;
            }
            $result->free();
            //close database connection
            $db->close();

    return $orderItems;
}

// Removes order with specified ID from database
function removeOrder($orderID) {
    //first, calls function to remove order items associated with order
    removeOrderItems($orderID);

    //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                    echo 'Error: Could not connect to database.  Please try again later.';
                    exit;
            }
            $query = "delete from ORDERS
          where ordID='".$orderID."'";
            $result = $db->query($query);
```

```
		//close database connection
		$db->close();
}

// Removes orderItems with specified orderID from database
function removeOrderItems($orderID) {
   //open database connection - host, username, password, database
		@ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
		if (mysqli_connect_errno()) {
			echo 'Error: Could not connect to database.  Please try again later.';
			exit;
		}
		$query = "delete from ORDERITEMS
		where orderID='".$orderID."'";
		$result = $db->query($query);
		//close database connection
		$db->close();
}
?>
```

productFunctions.php

```
<?php
   //Resumes session (if not already resumed)
   if(!isset($_SESSION))
   {
      session_start();
   }
?>

<?php
// Retrieves all products associated with the given categoryID
function getProdsByCat($categoryID) {
   //open database connection - host, username, password, database
		@ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
		if (mysqli_connect_errno()) {
			echo 'Error: Could not connect to database.  Please try again later.';
			exit;
		}
		$query = "select * from PRODUCTS
		where categoryID='".$categoryID."'
		order by prodID";
		$result = $db->query($query);
		while($row = $result->fetch_assoc()) {
			$products[] = $row;
		 }
		$result->free();
		//close database connection
		$db->close();
   return $products;
}

// Retrieves all products in the database
function getProds() {
   //open database connection - host, username, password, database
		@ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
		if (mysqli_connect_errno()) {
			echo 'Error: Could not connect to database.  Please try again later.';
			exit;
		}
		$query = "select * from PRODUCTS";
		$result = $db->query($query);
		while($row = $result->fetch_assoc()) {
			$products[] = $row;
		 }
		$result->free();
		//close database connection
		$db->close();
   return $products;
}
```

```
// Retrieves the product associated with the given product ID
function getProductByID($productID) {
   //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
            }
            $query = "select * from PRODUCTS
          where prodID='".$productID."'";
            $result = $db->query($query);
   $product = $result->fetch_assoc();
            $result->free();
            //close database connection
            $db->close();
   return $product;
}

// Determines if the product's stock has been depleted
// or if the product's stock is already accounted for by the quantity in the cart
function checkStock($product){
   $available = true;
   if($product['stock'] == 0){
      $available = false;
   }
   else if($available && isset($_SESSION['cart'])){
      foreach($_SESSION['cart'] as $item){
         if($item['productName'] == $product['prodName']){
            if($item['quantity'] == $product['stock']){
               $available = false;
            }
            break;
         }
      }
   }
   return $available;
}

// Decrease the stock of an item by its purchased quantity
function decreaseStock() {
   foreach($_SESSION['cart'] as $item){
      //open database connection - host, username, password, database
      @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
      if (mysqli_connect_errno()) {
         echo 'Error: Could not connect to database.  Please try again later.';
         exit;
      }
      $query = "update PRODUCTS
            set stock = stock - '".$item['quantity']."'
            where prodID = '".$item['productID']."'";
      $result = $db->query($query);
      //close database connection
      $db->close();
   }
}

// Retrieves the products containing the search term
function searchProds($searchTerm){
   //open database connection - host, username, password, database
            @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
            if (mysqli_connect_errno()) {
                        echo 'Error: Could not connect to database.  Please try again later.';
                        exit;
            }
            $query = "select * from PRODUCTS
          where prodName like '%".$searchTerm."%'";
            $result = $db->query($query);
            while($row = $result->fetch_assoc()) {
                        $products[] = $row;
             }
```

```
            $result->free();
            //close database connection
            $db->close();
    return $products;
}

// Inserts a new product into the database
function addProd($categoryID, $prodName, $price, $stock, $imageLoc) {
    //open database connection - host, username, password, database
    @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
    if (mysqli_connect_errno()) {
        echo 'Error: Could not connect to database.  Please try again later.';
        exit;
    }
    $query = "insert into PRODUCTS(categoryID,prodName,price,stock,imageLoc)
            values('".$categoryID."', '".$prodName."', '".$price."', '".$stock."', '".$imageLoc."')";
    $result = $db->query($query);
    //close database connection
    $db->close();
}

// Checks if the provided product name already exists within the given category
function checkProdName($categoryID, $prodName){
    $found = false;

    $products = getProds();

    // compares product name from form with product names in the same category,
    // if found then flags boolean variable
    foreach($products as $product){
        if($product['categoryID'] == $categoryID){
            if(strtoupper($product['prodName']) == strtoupper($prodName)){
                $found = true;
                break;
            }
        }
    }

            return $found;
}

// Change the stock of the given product to the number provided
function changeProdStock($prodID, $stock){
    //open database connection - host, username, password, database
    @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
    if (mysqli_connect_errno()) {
        echo 'Error: Could not connect to database.  Please try again later.';
        exit;
    }
    //updates the stock for the product ID
    $query1 = "update PRODUCTS
            set stock='".$stock."'
            where prodID='".$prodID."'";
    $result = $db->query($query1);
    //close database connection
    $db->close();
}

// Change the price of the given product to the number provided
function changeProdPrice($prodID, $price){
    //open database connection - host, username, password, database
    @ $db = new mysqli('localhost', 'imbimbd2_user1', 'passwordtest123', 'imbimbd2_bakeryDatabase');
    if (mysqli_connect_errno()) {
        echo 'Error: Could not connect to database.  Please try again later.';
        exit;
    }
    //updates the price for the product ID
    $query1 = "update PRODUCTS
            set price='".$price."'
            where prodID='".$prodID."'";
```

```
    $result = $db->query($query1);
    //close database connection
    $db->close();
}
?>
```

about.php

```
<?php
    //Resumes session (if not already resumed)
    if(!isset($_SESSION))
    {
        session_start();
    }
?>

<!--Includes the defined header-->
<?php include '../view/custHeader.php'; ?>

<main>
        <div class="container-fluid" style="padding:.5em">
                <p><img src="../images/about.jpg" alt="About Image" width="40%" height="auto"></p>
                <h6>
                        Hello Customer,<br><br>
                        WELCOME! You have stumbled upon the greatest bakery in the country!<br><br>
                        Prepare to have all of your cravings satisfied by our wide assortment of desserts!<br><br>
                </h6>
        </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

account.php

```
<?php
    //Resumes session (if not already resumed)
    if(!isset($_SESSION))
    {
        session_start();
    }
?>

<?php
require_once('../model/custFunctions.php');
require_once('../model/orderFunctions.php');

//retrieves the chosen action
$action = filter_input(INPUT_POST, 'action');

//when no action was chosen
if ($action == NULL)
{
        //sets action to showOptions
    $action = 'showOptions';
}

//selects chosen action
switch($action){
        //displays the account options, including order history
    case 'showOptions':
        $customer = getCustByEmail($_SESSION['email']);
        $orders = getOrdersByEmail($_SESSION['email']);
        include('accountOptions.php');
        break;
    //displays the details for the given order
    case 'viewDetails':
        $orderID = filter_input(INPUT_POST, 'orderID');
        $order = getOrderByID($orderID);
        $orderItems = getOrderItems($orderID);

        $customer = getCustByEmail($_SESSION['email']);
```

```
        $customerName = $customer['fName']." ".$customer["mInit"]." ".$customer['lName'];
        $customerAddress = $customer['street'].", ".$customer['city'].", ".$customer['state']." ".$customer['zip'];;
                        $customerPhone = $customer['custPhone'];

        include('orderDetails.php');
        break;
    //dispays a form containing the information of the account, which can be changed
    case 'showChangeInfo':
                        $customer = getCustByEmail($_SESSION['email']);
        include('changeInfo.php');
        break;
    //allows the user to change the name, phone number, & address associated with their account
    case 'changeInfo':
                        $fName = filter_input(INPUT_POST, 'fName');
                $mInit = filter_input(INPUT_POST, 'mInit');
                $lName = filter_input(INPUT_POST, 'lName');
                $phone = filter_input(INPUT_POST, 'phone');
                $street = filter_input(INPUT_POST, 'street');
                $city = filter_input(INPUT_POST, 'city');
                $state = filter_input(INPUT_POST, 'state');
                $zip = filter_input(INPUT_POST, 'zip');

        updateCustInfo($fName, $mInit, $lName, $phone, $street, $city, $state, $zip, $_SESSION['email']);
        echo "<script>
                alert('Account information has been successfully updated!');
            </script>";

        $customer = getCustByEmail($_SESSION['email']);
        $orders = getOrdersByEmail($_SESSION['email']);
        include('accountOptions.php');
        break;
    //dispays a form for the user to enter their old password and a new one
    case 'showChangePass':
        include('changePass.php');
        break;
    //allows the user to change their account password
    case 'changePass':
        $currentPass = filter_input(INPUT_POST, 'currentPass');
        $newPass = filter_input(INPUT_POST, 'newPass');

        $customer = getCustByEmail($_SESSION['email']);

        //checks if the entered current password matches the one in the database
        if(password_verify($currentPass, $customer['custPass'])){
            //encrypts the new password
                            $hashedPass = password_hash($newPass, PASSWORD_DEFAULT);

            updateCustPass($hashedPass, $_SESSION['email']);
            echo "<script>
                    alert('Account password has been successfully changed!');
                </script>";

            $orders = getOrdersByEmail($_SESSION['email']);
            include('accountOptions.php');
            break;
        }
        else{
            //defines error message
                    $_SESSION["passError"] = "<p style=\"color:red\">The current password entered is not correct. Please try
again.</p>";
            include('changePass.php');
            break;
        }
}
?>
```

accountOptions.php

```
<!--Includes the defined header-->
<?php include '../view/custHeader.php'; ?>
<?php require_once('../model/custFunctions.php'); ?>
```

```
<main>
   <h1>Account Settings</h1>

   <!--Displays account options-->
   <div class="row">
      <div class="col-sm-3">
         <h2>Options</h2>
         <nav>
            <ul>
               <li>
                  <form action="account.php" method="post">
                     <input type="hidden" name="action" value="showChangeInfo">
                     <input type="submit" name="submit" id="option" value="Change Information">
                  </form>
               </li>
               <li>
                  <form action="account.php" method="post">
                     <input type="hidden" name="action" value="showChangePass">
                     <input type="submit" name="submit" id="option" value="Change Password">
                  </form>
               </li>
            </ul>
         </nav>
      </div>
      <!-- displays all of the customer's orders -->
      <div class="col-sm">
         <h1>Order History</h1>
         <table>
            <tr>
               <th>Order Date</th>
               <th>Order Total</th>
            </tr>
            <!--Displays order dates & totals from the database, if there are any-->
            <?php if(!empty($orders)) :
               foreach ($orders as $order) : ?>
                  <tr>
                     <td><?php echo $order['ordDate']; ?></td>
                     <td><?php echo "$".$order['ordTotal']; ?></td>
                     <td>
                        <form action="account.php" method="post">
                           <input type="hidden" name="action" value="viewDetails">
                           <input type="hidden" name="orderID" value="<?php echo $order['ordID']; ?>">
                           <input type="submit" value="View Order Details">
                        </form>
                     </td>
                  </tr>
               <?php
               endforeach;
            endif;
            ?>
         </table>
      </div>
   </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

cart.php

```
<?php
   //Resumes session (if not already resumed)
   if(!isset($_SESSION))
   {
      session_start();
   }
?>

<?php
require_once('../model/orderFunctions.php');
```

```
require_once('../model/productFunctions.php');

//retrieves the chosen action
$action = filter_input(INPUT_POST, 'action');

//when no action was chosen
if ($action == NULL)
{
        //sets action to showCart
   $action = 'showCart';
}

//selects chosen action
switch($action){
        //displays the items in the session cart
   case 'showCart':
      include('cartDisplay.php');
      break;
   //removes specified item from session cart
   case 'remove':
      $productID = filter_input(INPUT_POST, 'productID');
                //decreases the quantity of the item in the cart by 1,
                //but if quantity is 1 then removes the item entirely
                if($_SESSION['cart'][$productID]['quantity'] > 1){
                        $_SESSION['cart'][$productID]['quantity'] -= 1;
                }
                else{
                        unset($_SESSION['cart'][$productID]);
                }
      include('cartDisplay.php');
      break;
        //retrieves variables from user account
        //and places an order via the addOrder function
        case 'placeOrder':
                //calls to create a new order in the database
                addOrder($_SESSION['email']);
                //update stock totals for purchased items
                decreaseStock();

                //displays order complete message and redirects back to welcome page
                echo "<script>
                                alert('Your order has been placed. Thank you!');
                                document.location='../index.php';
                        </script>";
                break;
}
?>
```

cartDisplay.php

```
<!--Includes the defined header-->
<?php include '../view/custHeader.php'; ?>
<?php require_once('../model/orderFunctions.php'); ?>

<?php
        //creates cart (if needed)
        if(!isset($_SESSION['cart'])){
                $_SESSION['cart'] = array();
        }
        //Appends with item quantity for better alignment on cart display
        $quantitySpacing = "             ";
?>
<main>
   <h1>Cart</h1>

   <div class="row">
      <div class="col-xl">
         <table>
            <tr>
               <th>Product Image</th>
               <th>Name</th>
```

```
            <th>Price</th>
                                                <th>Quantity</th>
            </tr>
                                        <!--Displays items in the session cart, if the cart isn't empty-->
            <?php if(!empty($_SESSION['cart'])):
               foreach ($_SESSION['cart'] as $item) : ?>
                  <tr>
                     <td class="picture"><img src=<?php echo "../".$item['imageLocation']; ?> alt="Product Image" max-width="500px"
width="40%"; height="auto"></td>
                     <td><?php echo $item['productName']; ?></td>
                     <td><?php echo "$".$item['price']; ?></td>
                     <td><?php echo $quantitySpacing.$item['quantity']; ?></td>
                     <td>
                        <form action="cart.php" method="post">
                           <input type="hidden" name="action"
                              value="remove">
                           <input type="hidden" name="productID"
                              value="<?php echo $item['productID']; ?>">
                           <input type="submit" value="Remove">
                        </form>
                     </td>
                  </tr>
               <?php
               endforeach;
            endif;
            ?>
                                        <tr>
                                                <td>
                                                        <!--Calls to function to display the total of the items in the cart-->
                                                        <?php echo "<h6>Total including tax (7%): $".getTotal()."</h6>"; ?>
                                                        <!--Only shows checkout button if the cart isn't empty-->
                                                        <?php if(!empty($_SESSION['cart'])): ?>
                                                                <form action="cart.php" method="post">
                                                                        <input type="hidden" name="action"
value="placeOrder">
                                                                        <input type="submit" value="Place Order">
                                                                </form>
                                                        <?php endif;?>
                                                </td>
                                        </tr>
         </table>
      </div>
   </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

changeInfo.php

```
<!--Includes the defined header-->
<?php include '../view/custHeader.php'; ?>

<main>
   <h1>Change Account Information</h1>

   <div class="container-fluid">
                        <form action = "account.php" method = "post">
                                <div class="form-group row">
                                        <label for="fname" class="col-sm-2 col-form-label">First Name:</label>
                                        <div class="col-sm-10">
                                                <input type="text" class="form-control" name="fName" value=<?php echo
"'".$customer['fName']."'"; ?> pattern="[A-Za-z]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="zip" class="col-sm-2 col-form-label">Middle Initial:</label>
                                        <div class="col-sm-10">
                                                <input type = "text" class = "form-control" name="mInit" value=<?php echo
"'".$customer['mInit']."'"; ?> maxlength=1 pattern="[A-Za-z]+">
                                        </div>
```

```
                                </div>
                                <div class="form-group row">
                                        <label for="lname" class="col-sm-2 col-form-label">Last Name:</label>
                                        <div class="col-sm-10">
                                                <input type="text" class="form-control" name="lName" value=<?php echo
"".$customer['lName']."""; ?> pattern="[A-Za-z]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="phone" class="col-sm-2 col-form-label">Phone:</label>
                                        <div class="col-sm-10">
                                                <input type = "tel" class = "form-control" name ="phone" value=<?php echo
"".$customer['custPhone']."""; ?> pattern = "\(\d{3}\) +\d{3}-\d{4}" required>    (###) ### ####
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="street" class="col-sm-2 col-form-label">Street:</label>
                                        <div class="col-sm-10">
                                                <input type = "text" class = "form-control" name ="street" value=<?php echo
"".$customer['street']."""; ?> pattern="[A-Za-z0-9\s]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="city" class="col-sm-2 col-form-label">City:</label>
                                        <div class="col-sm-10">
                                                <input type = "text" class = "form-control" name ="city" value=<?php echo
"".$customer['city']."""; ?> pattern="[A-Za-z]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="state" class="col-sm-2 col-form-label">State:</label>
                                        <div class="col-sm-10">
                                                <select class = "form-control" name ="state" required>
                                                        <option value=<?php echo """.$customer['state']."""; ?> selected
hidden><?php echo $customer['state']; ?></option>
                                                        <option value="AL">(AL) Alabama</option>
                                                        <option value="AK">(AK) Alaska</option>
                                                        <option value="AZ">(AZ) Arizona</option>
                                                        <option value="AR">(AR) Arkansas</option>
                                                        <option value="CA">(CA) California</option>
                                                        <option value="CO">(CO) Colorado</option>
                                                        <option value="CT">(CT) Connecticut</option>
                                                        <option value="DE">(DE) Delaware</option>
                                                        <option value="DC">(DC) District Of Columbia</option>
                                                        <option value="FL">(FL) Florida</option>
                                                        <option value="GA">(GA) Georgia</option>
                                                        <option value="HI">(HI) Hawaii</option>
                                                        <option value="ID">(ID) Idaho</option>
                                                        <option value="IL">(IL) Illinois</option>
                                                        <option value="IN">(IN) Indiana</option>
                                                        <option value="IA">(IA) Iowa</option>
                                                        <option value="KS">(KS) Kansas</option>
                                                        <option value="KY">(KY) Kentucky</option>
                                                        <option value="LA">(LA) Louisiana</option>
                                                        <option value="ME">(ME) Maine</option>
                                                        <option value="MD">(MD) Maryland</option>
                                                        <option value="MA">(MA) Massachusetts</option>
                                                        <option value="MI">(MI) Michigan</option>
                                                        <option value="MN">(MN) Minnesota</option>
                                                        <option value="MS">(MS) Mississippi</option>
                                                        <option value="MO">(MO) Missouri</option>
                                                        <option value="MT">(MT) Montana</option>
                                                        <option value="NE">(NE) Nebraska</option>
                                                        <option value="NV">(NV) Nevada</option>
                                                        <option value="NH">(NH) New Hampshire</option>
                                                        <option value="NJ">(NJ) New Jersey</option>
                                                        <option value="NM">(NM) New Mexico</option>
                                                        <option value="NY">(NY) New York</option>
                                                        <option value="NC">(NC) North Carolina</option>
                                                        <option value="ND">(ND) North Dakota</option>
```

```
                                                <option value="OH">(OH) Ohio</option>
                                                <option value="OK">(OK) Oklahoma</option>
                                                <option value="OR">(OR) Oregon</option>
                                                <option value="PA">(PA) Pennsylvania</option>
                                                <option value="RI">(RI) Rhode Island</option>
                                                <option value="SC">(SC) South Carolina</option>
                                                <option value="SD">(SD) South Dakota</option>
                                                <option value="TN">(TN) Tennessee</option>
                                                <option value="TX">(TX) Texas</option>
                                                <option value="UT">(UT) Utah</option>
                                                <option value="VT">(VT) Vermont</option>
                                                <option value="VA">(VA) Virginia</option>
                                                <option value="WA">(WA) Washington</option>
                                                <option value="WV">(WV) West Virginia</option>
                                                <option value="WI">(WI) Wisconsin</option>
                                                <option value="WY">(WY) Wyoming</option>
                                        </select>
                                </div>
                        </div>
                        <div class="form-group row">
                                <label for="zip" class="col-sm-2 col-form-label">Zip Code:</label>
                                <div class="col-sm-10">
                                        <input type = "text" class = "form-control" name ="zip" value=<?php echo
""".$customer['zip']."""; ?> maxlength=5 pattern ="[0-9]+" required>
                                </div>
                        </div>
                        <div class="form-group">
        <input type="hidden" name="action" value="changeInfo">
                                <input type="submit" value="Confirm">
                        </div>
                </form>
   </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

changePass.php

```
<?php
   //Resumes session (if not already resumed)
   if(!isset($_SESSION))
   {
      session_start();
   }
?>

<!--Includes the defined header-->
<?php include '../view/custHeader.php'; ?>

<main>
        <h1>Change Account Password</h1>

        <div class="container-fluid">
                <form action = "account.php" method = "post">
                        <!--Checks if error variable has been set in session array,
                                and if so displays the defined message-->
                                <?php
                                if(isset($_SESSION["passError"])){
                                        echo $_SESSION["passError"];
                                }
                        ?>
                        <div class="form-group row">
                                <label for="pass" class="col-sm-2 col-form-label">Current Password:</label>
                                <div class="col-sm-10">
                                        <input type="password" class = "form-control" name="currentPass"
placeholder="Enter current password" required>
                                </div>
                        </div>
                        <div class="form-group row">
                                <label for="pass" class="col-sm-2 col-form-label">New Password:</label>
```

```
                                        <div class="col-sm-10">
                                                <input type="password" class = "form-control" name="newPass"
placeholder="Enter new password" required>
                                        </div>
                                </div>
                                <div class="form-group">
        <input type="hidden" name="action" value="changePass">
                                        <input type="submit" value="Change">
                                </div>
                        </form>
                        <!--Unsets error variable in session arrray-->
                        <?php
                                unset($_SESSION["passError"]);
                        ?>
        </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

orderDetails.php

```
<!--Includes the defined header-->
<?php include '../view/custHeader.php'; ?>
<?php require_once('../model/productFunctions.php'); ?>

<main>
   <h1>Order Details</h1>
      <div class="col-sm">
         <form action="account.php" method="post">
            <input type="hidden" name="action" value="showOptions">
            <input type="submit" name="submit" value="Go Back">
         </form>
         <table>
            <tr>
               <td><b><?php echo "Customer: ".$customerName ?></b></td>
            </tr>
                                        <tr>
               <td><b><?php echo "Phone Number: ".$customerPhone ?></b></td>
            </tr>
                                        <tr>
               <td><b><?php echo "Shipping Address: ".$customerAddress ?></b></td>
            </tr>
         </table>
         <table>
            <tr>
               <th>Product Image</th>
               <th>Name</th>
               <th>Quantity</th>
               <th>Overall Cost</th>
            </tr>
            <!--Displays each item from the order-->
            <?php foreach ($orderItems as $orderItem) : ?>
               <?php $product = getProductByID($orderItem['productID']); ?>
               <tr>
                  <td class="picture"><img src=<?php echo "../".$product['imageLoc']; ?> alt="Product Image" max-width="500px"
width="35%"; height="auto"></td>
                  <td><?php echo $product['prodName']?></td>
                  <td><?php echo "x".$orderItem['quantity'] ?></td>
                  <td><?php echo "$".$orderItem['cost']; ?></td>
               </tr>
            <?php endforeach; ?>
            <tr>
                                                <td>
                                                        <?php echo "<h6>Total including tax (7%): $".$order['ordTotal']."</h6>";
?>
                                                </td>
                                        </tr>
         </table>
      </div>
   </div>
```

```
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

products.php

```
<?php
    //Resumes session (if not already resumed)
    if(!isset($_SESSION))
    {
        session_start();
    }
?>

<?php
require_once('../model/categoryFunctions.php');
require_once('../model/productFunctions.php');
require_once('../model/orderFunctions.php');

//retrieves the chosen action
$action = filter_input(INPUT_POST, 'action');

//when no action was chosen
if ($action == NULL)
{
    //checks if a category has been selected
    $action = filter_input(INPUT_GET, 'action');
            //sets action to productList
    if ($action == NULL)
            {
        $action = 'productList';
    }
}

//selects chosen action
switch($action){
    //lists products by category
    case 'productList':
        //retrieves selected category ID from the super global GET array
        $categoryID = filter_input(INPUT_GET, 'categoryID',
            FILTER_VALIDATE_INT);
        //if none was selected, sets categoryID to 1 by default
        if ($categoryID == NULL)
        {
            $categoryID = 1;
        }
        $categoryName = getCatName($categoryID);
        $categories = getCats();
        $products = getProdsByCat($categoryID);
        include('productsDisplay.php');
        break;
    //adds specified item to session cart
    case 'addToCart':
        $productID = filter_input(INPUT_POST, 'productID');
        $productName = filter_input(INPUT_POST, 'productName');
        $price = filter_input(INPUT_POST, 'price');
        $imageLocation = filter_input(INPUT_POST, 'imageLocation');
        $quantity = 1;
        $item = array(
            'productID' => $productID,
            'productName' => $productName,
            'price'  => $price,
            'quantity' => $quantity,
            'imageLocation' => $imageLocation
        );
        addItem($item);
        echo "<script> document.location='cart.php'; </script>";
        break;
    //displays the searched products
    case 'search':
```

```
        $searchTerm = filter_input(INPUT_POST, 'searchTerm');
        $searchResults = searchProds($searchTerm);
        include('searchDisplay.php');
        break;
}
?>
```

productsDisplay.php

```
<!--Includes the defined header-->
<?php include '../view/custHeader.php'; ?>
<?php require_once('../model/productFunctions.php'); ?>

<main>
    <h1>Product Inventory</h1>

    <div class="row">
        <div class="col-sm-3">
        <!-- lists all categories -->
            <h2>Categories</h2>
            <nav>
                <ul>
                    <?php foreach ($categories as $category) : ?>
                        <li>
                            <a href="?categoryID=<?php echo $category['catID']; ?>">
                                <?php echo $category['catName']; ?>
                            </a>
                        </li>
                    <?php endforeach; ?>
                </ul>
            </nav>
        </div>
        <div class="col-sm">
            <form action="products.php" method="post">
                <b>Find Product:</b><br/>
                <input type="hidden" name="action" value="search">
                <input name="searchTerm" type="text" size="40">
                <input type="submit" name="submit" value="Search">
            </form>
            <!-- displays all products -->
            <h2><?php echo $categoryName; ?></h2>
            <table>
                <tr>
                    <th>Product Image</th>
                    <th>Name</th>
                    <th>Price</th>
                </tr>
                <?php foreach ($products as $product) : ?>
                    <tr>
                        <td class="picture"><img src=<?php echo "../".$product['imageLoc']; ?> alt="Product Image" max-width="500px"
width="35%"; height="auto"></td>
                        <td><?php echo $product['prodName']; ?></td>
                        <td><?php echo "$".$product['price']; ?></td>
                        <td>
                            <form action="products.php" method="post">
                                <input type="hidden" name="action"
                                    value="addToCart">
                                <input type="hidden" name="productID"
                                    value="<?php echo $product['prodID']; ?>">
                                <input type="hidden" name="productName"
                                    value="<?php echo $product['prodName']; ?>">
                                <input type="hidden" name="price"
                                    value="<?php echo $product['price']; ?>">
                                <input type="hidden" name="productStock"
                                    value="<?php echo $product['stock']; ?>">
                                <input type="hidden" name="imageLocation"
                                    value="<?php echo $product['imageLoc']; ?>">
                                <?php //Checks if the product still has available stock
                                    if(checkStock($product)){
                                        echo "<input type=\"submit\" value=\"Add\">";
                                    }
```

```
                        else{
                            echo "<button type=\"button\" disabled>Out of Stock</button>";
                        }
                    ?>
                </form>
            </td>
        </tr>
        <?php endforeach; ?>
    </table>
</div>
</div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

searchDisplay.php

```
<!--Includes the defined header-->
<?php include '../view/custHeader.php'; ?>
<?php require_once('../model/productFunctions.php'); ?>

<main>
    <h1>Product Inventory</h1>

    <div class="row">
        <div class="col-sm">
            <!-- displays found products -->
            <h2><?php echo "Results for '$searchTerm'"; ?></h2>
            <table>
                <tr>
                    <th>Picture</th>
                    <th>Name</th>
                    <th>Price</th>
                </tr>
                <?php foreach ($searchResults as $searchResult) : ?>
                    <tr>
                        <td class="picture"><img src=<?php echo "../".$searchResult['imageLoc']; ?> alt="Product Image"
max-width="500px" width="35%"; height="auto"></td>
                        <td><?php echo $searchResult['prodName']; ?></td>
                        <td><?php echo "$".$searchResult['price']; ?></td>
                        <td>
                            <form action="products.php" method="post">
                                <input type="hidden" name="action"
                                    value="addToCart">
                                <input type="hidden" name="productID"
                                    value="<?php echo $searchResult['prodID']; ?>">
                                <input type="hidden" name="productName"
                                    value="<?php echo $searchResult['prodName']; ?>">
                                <input type="hidden" name="price"
                                    value="<?php echo $searchResult['price']; ?>">
                                <input type="hidden" name="productStock"
                                    value="<?php echo $searchResult['stock']; ?>">
                                <input type="hidden" name="imageLocation"
                                    value="<?php echo $searchResult['imageLoc']; ?>">
                                <?php //Checks if the product still has available stock
                                    if(checkStock($searchResult)){
                                        echo "<input type=\"submit\" value=\"Add\">";
                                    }
                                    else{
                                        echo "<button type=\"button\" disabled>Out of Stock</button>";
                                    }
                                ?>
                            </form>
                        </td>
                    </tr>
                <?php endforeach; ?>
            </table>
        </div>
    </div>
</main>
```

```
<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>

empsDisplay.php
<?php
   //Resumes session (if not already resumed)
   if(!isset($_SESSION))
   {
      session_start();
   }
?>

<!--Includes the defined header-->
<?php include '../view/empHeader.php'; ?>
<?php require_once('../model/deptFunctions.php'); ?>

<main>
   <h1>Employee List</h1>
      <div class="col-sm">
         <!-- displays all employees -->
         <table>
            <tr>
               <th>Employee ID</th>
               <th>Department ID</th>
                                                <th>Department Manager</th>
                                                <!-- spacing in table for make manager button if CEO is logged in -->
               <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
                     <th></th>
               <?php endif;?>
               <th>Employee Name</th>
                                                <th>Phone Number</th>
                                                <!-- also displays salary, email, and password if CEO is logged in -->
                                                <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
                                                      <th>Hire Date</th>
                                                      <th>Salary</th>
                                                      <th>Email</th>
                                                <?php endif;?>
            </tr>
            <?php foreach ($employees as $employee) : ?>
               <tr>
                  <td><?php echo $employee['empID']; ?></td>
                  <td><?php echo $employee['departmentID']; ?></td>
                                                <?php
                                                      //checks if the employee is currently the manager of their
department
                                                      if(checkDeptMan($employee['empID']))
                                                            echo "<td>Y</td>";
                                                      else
                                                            echo "<td>N</td>";
                                                ?>
                                                <!-- displays option to make employee the manager of the their
department if CEO is logged in -->
                                                <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
                                                      <td>
                                                            <form action="empsTable.php" method="post">
                                                                  <input type="hidden" name="action"
value="makeDeptMan">
                                                                  <input type="hidden" name="empID"
value="<?php echo $employee['empID']; ?>">
                                                                  <?php
                                                                        //dont show manager button
next to CEO
                                                                        if($employee['empEmail'] ==
"fdecker@gmail.com"){
                                                                              echo "";
                                                                        }
                                                                        //checks if the employee is
currently the manager of their department
```

```
                                                                                    else
if(!checkDeptMan($employee['empID'])){
                                                                                        echo "<input
type=\"submit\" value=\"Make Manager\">";
                                                                                    }
                                                                                    else{
                                                                                        echo "<button
type=\"button\" disabled>Already Manager</button>";
                                                                                    }
                        ?>
                                                                            </form>
                                                                        </td>
                                                                <?php endif;?>
                                                                <td><?php echo $employee['fName']." ".$employee['mInit']."
".$employee['lName']; ?></td>
                                                                <td><?php echo $employee['empPhone']; ?></td>
                                                                <!-- also displays salary, email, and password if CEO is logged in -->
                                                                <!-- as well as the option to to fire an employee -->
                                                                <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
                                                                    <td><?php echo $employee['hireDate']; ?></td>
                                                                    <td><?php echo $employee['salary']; ?></td>
                                                                    <td><?php echo $employee['empEmail']; ?></td>
                                                                    <!-- don't show fire next to CEO -->
                                                                    <?php if($employee['empEmail'] != "fdecker@gmail.com"):
?>
                                                                        <td>
                                                                            <form action="empsTable.php"
method="post">
                                                                                <input type="hidden"
name="action" value="fireEmp">
                                                                                <input type="hidden"
name="empID"
                                                                                    value="<?php echo
$employee['empID']; ?>">
                                                                                <input type="submit"
value="Fire">
                                                                            </form>
                                                                        </td>
                                                                    <?php endif;?>
                                                                <?php endif;?>
                    </tr>
                <?php endforeach; ?>
                                                <!-- displays option to add a new employee if CEO is logged in -->
                                                <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
                                                    <tr>
                                                        <td>
                                                            <form action="empsTable.php" method="post">
                                                                <input type="hidden" name="action"
value="newEmp">
                                                                <input type="submit" value="Add new employee">
                                                            </form>
                                                        </td>
                                                    </tr>
                                                <?php endif;?>
            </table>
        </div>
    </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

empsTable.php

```
<?php
    //Resumes session (if not already resumed)
    if(!isset($_SESSION))
    {
        session_start();
    }
?>
```

```
<?php
require_once('../model/empFunctions.php');
require_once('../model/deptFunctions.php');
require_once('../model/custFunctions.php');

//retrieves the chosen action
$action = filter_input(INPUT_POST, 'action');

//when no action was chosen
if ($action == NULL)
{
		//sets action to empList
    $action = 'empList';
}

//selects chosen action
switch($action){
    //retrieves and displays current employee list
    case 'empList':
        $employees = getEmps();
        include('empsDisplay.php');
        break;
    //makes the specified employeeID the manager of their department
    case 'makeDeptMan':
        $empID = filter_input(INPUT_POST, 'empID');

        makeDeptMan($empID);
        echo "<script>
                                        alert('Employee has been successfully made the manager of their department!');
                                    </script>";

        //retrieves and displays updated employee list
        $employees = getEmps();
        include('empsDisplay.php');
        break;
    //fires the employee specified employeeID
    case 'fireEmp':
        $empID = filter_input(INPUT_POST, 'empID');

        fireEmp($empID);
        echo "<script>
                                        alert('Employee has been successfully fired!');
                                    </script>";

        //retrieves and displays updated employee list
        $employees = getEmps();
        include('empsDisplay.php');
        break;
    //opens new employee form
    case 'newEmp':
        include('newEmp.php');
        break;
    //retrieves variables from new employee form and hires new employee via the addEmp function
    //then retrieves and displays updated employee list
    case 'hireEmp':
        $fName = filter_input(INPUT_POST, 'fName');
        $mInit = filter_input(INPUT_POST, 'mInit');
                        $lName = filter_input(INPUT_POST, 'lName');
        $empPhone = filter_input(INPUT_POST, 'empPhone');
                        $salary = filter_input(INPUT_POST, 'salary');
                        $departmentID = filter_input(INPUT_POST, 'dept');
                        $empEmail = filter_input(INPUT_POST, 'empEmail');
        $empPass = filter_input(INPUT_POST, 'empPass');

        //checks if email is already being used by a customer
                $custFound = checkCustEmail($empEmail);

        //checks if email is already being used by a employee
        if($custFound == false){
```

```
        $empFound = checkEmpEmail($empEmail);
    }

                    if($custFound == true || $empFound == true){
        // Sets error variable in session array
                        $_SESSION["empError"] = "<p style=\"color:red\">Email already in database, please choose another
email.</p>";
        include('newEmp.php');
        break;
    }
    else{
        // Encrypts the entered password
                        $hashedPass = password_hash($empPass, PASSWORD_DEFAULT);

        //calls to add new employee to database
        addEmp($empEmail,$hashedPass,$fName,$mInit,$lName,$empPhone,$salary,$departmentID);
        echo "<script>
                                alert('Employee has been successfully hired!');
                            </script>";

        //retrieves and displays updated employee list
        $employees = getEmps();
        include('empsDisplay.php');
        break;
    }
}
?>
```

newEmp.php

```
<?php
   //Resumes session (if not already resumed)
   if(!isset($_SESSION))
   {
      session_start();
   }
?>

<!--Includes the defined header-->
<?php include '../view/empHeader.php'; ?>

<main>
   <h1>New Employee</h1>

   <div class="container-fluid">
                    <form action = "empsTable.php" method = "post">
                        <!--Checks if error variable has been set in session array,
                            and if so displays the defined message-->
                        <?php
                            if(isset($_SESSION["empError"])){
                                echo $_SESSION["empError"];
                            }
                        ?>
                        <div class="form-group row">
                            <label for="fname" class="col-sm-2 col-form-label">First Name:</label>
                            <div class="col-sm-10">
                                <input type="text" class="form-control" id="fname" name="fName"
placeholder="John" pattern="[A-Za-z]+" required>
                            </div>
                        </div>
                        <div class="form-group row">
                            <label for="zip" class="col-sm-2 col-form-label">Middle Initial:</label>
                            <div class="col-sm-10">
                                <input type = "text" class = "form-control" id="minit" name="mInit"
placeholder="A" maxlength=1 pattern="[A-Za-z]+">
                            </div>
                        </div>
                        <div class="form-group row">
                            <label for="lname" class="col-sm-2 col-form-label">Last Name:</label>
                            <div class="col-sm-10">
```

```
                                                <input type="text" class="form-control" id="lname" name="lName"
placeholder="Smith" pattern="[A-Za-z]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="phone" class="col-sm-2 col-form-label">Phone:</label>
                                        <div class="col-sm-10">
                                                <input type = "tel" class = "form-control" id = "empPhone" name ="empPhone"
placeholder = "(973) 123-4567" pattern = "\(\d{3}\) +\d{3}-\d{4}" required>    (###) ###-####
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="street" class="col-sm-2 col-form-label">Salary:</label>
                                        <div class="col-sm-10">
                                                <input type = "number" class = "form-control" id = "salary" name ="salary"
placeholder = "25000" min=0 pattern ="[0-9]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="state" class="col-sm-2 col-form-label">Department:</label>
                                        <div class="col-sm-10">
                                                <select class = "form-control" id = "dept" name ="dept" placeholder = "2"
required>
                                                        <option value="2">Accounting</option>
                                                        <option value="3">Customer Service</option>
                                                </select>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="city" class="col-sm-2 col-form-label">Email:</label>
                                        <div class="col-sm-10">
                                                <input type="email" class="form-control" id="empEmail" name="empEmail"
placeholder="example@gmail.com" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="pass" class="col-sm-2 col-form-label">Password:</label>
                                        <div class="col-sm-10">
                                                <input type="password" class = "form-control" id="empPass" name="empPass"
placeholder="password" required>
                                        </div>
                                </div>
                                <div class="form-group">
                <input type="hidden" name="action" value="hireEmp">
                                                <input type="submit" value="Hire Employee">
                                </div>
                        </form>
                        <!--Unsets error variable in session arrray-->
                        <?php
                                unset($_SESSION["empError"]);
                        ?>
   </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

newProd.php

```
<?php
   //Resumes session (if not already resumed)
   if(!isset($_SESSION))
   {
      session_start();
   }
?>

<!--Includes the defined header-->
<?php include '../view/empHeader.php'; ?>

<main>
```

```
    <h1>New Product</h1>

    <div class="container-fluid">
                        <form action = "prodsTable.php" method = "post">
                                <!--Checks if error variable has been set in session array,
                                        and if so displays the defined message-->
                                <?php
                                        if(isset($_SESSION["nameError"])){
                                                echo $_SESSION["nameError"];
                                        }
                                ?>
                                <div class="form-group row">
                                        <label for="state" class="col-sm-2 col-form-label">Category:</label>
                                        <div class="col-sm-10">
                                                <select class = "form-control" id = "categoryID" name ="categoryID" placeholder
= "1" required>
                                                        <option value="1">Brownies</option>
                                                        <option value="2">Cakes</option>
                                                        <option value="3">Cookies</option>
                                                        <option value="4">Donuts</option>
                                                        <option value="5">Puddings</option>
                                                </select>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="first" class="col-sm-2 col-form-label">Product Name:</label>
                                        <div class="col-sm-10">
                                                <input type = "text" class = "form-control" id = "prodName" name ="prodName"
placeholder = "Product" pattern="[A-Z][A-Za-z -]+" required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="last" class="col-sm-2 col-form-label">Price:</label>
                                        <div class="col-sm-10">
                                                <input type = "number" class = "form-control" id = "price" name ="price"
placeholder = "9.99" step="0.01" min="0.01" max=100 required>
                                        </div>
                                </div>
                                <div class="form-group row">
                                        <label for="last" class="col-sm-2 col-form-label">Stock:</label>
                                        <div class="col-sm-10">
                                                <input type = "number" class = "form-control" id = "stock" name ="stock"
placeholder="0" min=1 max=100 required>
                                        </div>
                                </div>
                                <div class="form-group">
                <input type="hidden" name="action" value="addProd">
                                        <input type="submit" value="Add Product">
                                </div>
                        </form>
                        <!--Unsets error variable in session arrray-->
                        <?php
                                unset($_SESSION["nameError"]);
                        ?>
    </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

ordersDisplay.php

```
<?php
    //Resumes session (if not already resumed)
    if(!isset($_SESSION))
    {
        session_start();
    }
?>

<!--Includes the defined header-->
```

```
<?php include '../view/empHeader.php'; ?>

<main>
    <h1>Order List</h1>
        <div class="col-sm">
            <!-- displays all orders -->
            <table>
                <tr>
                    <th>Order ID</th>
                    <th>Order Date</th>
                    <th>Order Total</th>
                    <th>Customer Email</th>
                </tr>
                <!--Displays orders from the database, if there are any-->
                <?php if(!empty($orders)) :
                    foreach ($orders as $order) : ?>
                        <tr>
                            <td><?php echo $order['ordID']; ?></td>
                            <td><?php echo $order['ordDate']; ?></td>
                            <td><?php echo "$".$order['ordTotal']; ?></td>
                            <td><?php echo $order['customerEmail']; ?></td>
                            <!-- displays option to remove order if CEO is logged in -->
                            <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
                                <td>
                                    <form action="ordersTable.php" method="post">
                                        <input type="hidden" name="action" value="removeOrder">
                                        <input type="hidden" name="ordID" value="<?php echo $order['ordID']; ?>">
                                        <input type="submit" value="Remove order">
                                    </form>
                                </td>
                            <?php endif;?>
                        </tr>
                    <?php
                    endforeach;
                endif;
                ?>
            </table>
        </div>
    </div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

ordersTable.php

```
<?php
require_once('../model/orderFunctions.php');

//retrieves the chosen action
$action = filter_input(INPUT_POST, 'action');

//when no action was chosen
if ($action == NULL)
{
            //sets action to orderList
    $action = 'orderList';
}

//selects chosen action
switch($action){
    //retrieves and displays current order list
    case 'orderList':
        $orders = getOrders();
        include('ordersDisplay.php');
        break;
    //removes the order with the specified orderID
    case 'removeOrder':
        $orderID = filter_input(INPUT_POST, 'ordID');
        removeOrder($orderID);
```

```
        //retrieves and displays updated order list
                        $orders = getOrders();
        include('ordersDisplay.php');
        break;
}
?>
```

prodsDisplay.php

```
<?php
   //Resumes session (if not already resumed)
   if(!isset($_SESSION))
   {
      session_start();
   }
?>

<!--Includes the defined header-->
<?php include '../view/empHeader.php'; ?>

<main>
   <h1>Product List</h1>
      <div class="col-sm">
         <!-- displays all products -->
         <table>
            <tr>
               <th>Product ID</th>
               <th>Category ID</th>
               <th>Product Name</th>
                                                      <th>Price</th>
               <!-- spacing in table for new price button if CEO is logged in -->
               <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
                     <th></th>
               <?php endif;?>
               <th>Image (Location)</th>
               <th>Stock</th>
            </tr>
            <?php foreach ($products as $product) : ?>
               <tr>
                  <td><?php echo $product['prodID']; ?></td>
                  <td><?php echo $product['categoryID']; ?></td>
                                                            <td><?php echo $product['prodName']; ?></td>
                                                            <td><?php echo "$".$product['price']; ?></td>
                  <!-- displays option to change product price if CEO is logged in -->
                  <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
                     <td>
                        <form action="prodsTable.php" method="post">
                           <input type="hidden" name="action" value="changePrice">
                           <input type="hidden" name="prodID" value="<?php echo $product['prodID']; ?>">
                           <input type="number" name="newPrice" placeholder="0.01" step="0.01" min="0.01" max="100" required>
                           <input type="submit" value="Change price">
                        </form>
                     </td>
                  <?php endif;?>
                                                            <td><?php echo $product['imageLoc']; ?></td>
                  <td><?php echo $product['stock']; ?></td>
                  <!-- displays option to change product stock if CEO is logged in -->
                  <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
                     <td>
                        <form action="prodsTable.php" method="post">
                           <input type="hidden" name="action" value="changeStock">
                           <input type="hidden" name="prodID" value="<?php echo $product['prodID']; ?>">
                           <input type="number" name="newStock" placeholder="0" min=0 max=100 required>
                           <input type="submit" value="Change stock">
                        </form>
                     </td>
                  <?php endif;?>
               </tr>
            <?php endforeach; ?>
            <!-- displays option to add a new employee if CEO is logged in -->
                                          <?php if($_SESSION['email'] == "fdecker@gmail.com"): ?>
```

```
                <tr>
                    <td>
                        <form action="prodsTable.php" method="post">
                            <input type="hidden" name="action" value="newProd">
                            <input type="submit" value="Add new product">
                        </form>
                    </td>
                </tr>
            <?php endif;?>
        </table>
    </div>
</div>
</main>

<!--Includes the defined footer-->
<?php include '../view/footer.php'; ?>
```

prodsTable.php

```
<?php
    //Resumes session (if not already resumed)
    if(!isset($_SESSION))
    {
        session_start();
    }
?>

<?php
require_once('../model/productFunctions.php');

//retrieves the chosen action
$action = filter_input(INPUT_POST, 'action');

//when no action was chosen
if ($action == NULL)
{
            //sets action to prodList
    $action = 'prodList';
}

//selects chosen action
switch($action){
    //retrieves and displays current product list
    case 'prodList':
        $products = getProds();
        include('prodsDisplay.php');
        break;
    //opens new product form
    case 'newProd':
        include('newProd.php');
        break;
    //retrieves variables from new product form and adds new product via the addProd function
    //then retrieves and displays updated product list
    case 'addProd':
                        $categoryID = filter_input(INPUT_POST, 'categoryID');
                        $prodName = filter_input(INPUT_POST, 'prodName');
                        $price = filter_input(INPUT_POST, 'price');
        $stock = filter_input(INPUT_POST, 'stock');

        //creates a lower-case image location string from productName, while removing any hyphens and spaces
        $imageLoc = "images/".str_replace(" ", "", str_replace("-", " ", strtolower($prodName))).".jpg";

        //checks if product name already exists in the same category
        $nameFound = checkProdName($categoryID, $prodName);

        if($nameFound == true){
            // Sets error variable in session array
                        $_SESSION["nameError"] = "<p style=\"color:red\">Product flavor already exists in that category!</p>";
            include('newProd.php');
            break;
        }
```

```
        else{
            //calls to add new product to database
            addProd($categoryID, $prodName, $price, $stock, $imageLoc);
            echo "<script>
                    alert('Product has been successfully added!');
                </script>";

            //retrieves and displays updated product list
            $products = getProds();
            include('prodsDisplay.php');
            break;
        }
    case 'changeStock':
        $prodID = filter_input(INPUT_POST, 'prodID');
        $stock = filter_input(INPUT_POST, 'newStock');

        changeProdStock($prodID, $stock);
        echo "<script>
                                        alert('Product stock has been successfully updated!');
                                </script>";

        //retrieves and displays updated product list
        $products = getProds();
        include('prodsDisplay.php');
        break;
    case 'changePrice':
        $prodID = filter_input(INPUT_POST, 'prodID');
        $price = filter_input(INPUT_POST, 'newPrice');

        changeProdPrice($prodID, $price);
        echo "<script>
                                        alert('Product price has been successfully updated!');
                                </script>";

        //retrieves and displays updated product list
        $products = getProds();
        include('prodsDisplay.php');
        break;
}
?>
```

Database Dump

```
-- phpMyAdmin SQL Dump
-- version 4.9.7
-- https://www.phpmyadmin.net/
--
-- Host: localhost:3306
-- Generation Time: Dec 15, 2021 at 01:46 AM
-- Server version: 5.7.36
-- PHP Version: 7.3.33

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: `imbimbd2_bakeryDatabase`
--

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

--
```

```
-- Table structure for table `CATEGORIES`
--

CREATE TABLE `CATEGORIES` (
  `catID` int(11) NOT NULL,
  `catName` varchar(255) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `CATEGORIES`
--

INSERT INTO `CATEGORIES` (`catID`, `catName`) VALUES
(1, 'Brownies'),
(2, 'Cakes'),
(3, 'Cookies'),
(4, 'Donuts'),
(5, 'Puddings');

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

--
-- Table structure for table `CUSTOMERS`
--

CREATE TABLE `CUSTOMERS` (
  `custEmail` varchar(255) NOT NULL,
  `custPass` varchar(255) NOT NULL,
  `fName` varchar(255) NOT NULL,
  `mInit` char(1) NOT NULL,
  `lName` varchar(255) NOT NULL,
  `custPhone` varchar(255) NOT NULL,
  `street` varchar(255) NOT NULL,
  `city` varchar(255) NOT NULL,
  `state` varchar(2) NOT NULL,
  `zip` varchar(5) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `CUSTOMERS`
--

INSERT INTO `CUSTOMERS` (`custEmail`, `custPass`, `fName`, `mInit`, `lName`, `custPhone`, `street`, `city`, `state`, `zip`)
VALUES
('test@gmail.com', '$2y$10$J8g.KOemwVEbl4/v77mrzezT.iwx15zI7IsiSWsnGeuSNIx9GpVLC', 'Nigel', 'C', 'Finster', '(973) 123-4567',
'63 Easy Street', 'Nowhere', 'NJ', '07123'),
('thisisatest@gmail.com', '$2y$10$iC3IDXjyg03lsKV5TyiItOxENFubooJAPhBtQydyZyaHBG9DBTBl2', 'Iam', 'A', 'Test', '(123)
456-7890', '64 Somewhere Ave', 'Anchorage', 'AK', '01234');

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

--
-- Table structure for table `DEPARTMENTS`
--

CREATE TABLE `DEPARTMENTS` (
  `deptID` int(11) NOT NULL,
  `deptName` varchar(255) NOT NULL,
  `manID` int(11) DEFAULT NULL,
  `manStart` datetime DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `DEPARTMENTS`
--

INSERT INTO `DEPARTMENTS` (`deptID`, `deptName`, `manID`, `manStart`) VALUES
(1, 'CEO', 1, '2021-04-26 00:00:00'),
(2, 'Accounting', 2, '2021-05-03 09:27:41'),
(3, 'Customer Service', 1, '2021-12-16 05:08:26');
```

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

--
-- Table structure for table `EMPLOYEES`
--

CREATE TABLE `EMPLOYEES` (
  `empID` int(11) NOT NULL,
  `empEmail` varchar(255) NOT NULL,
  `empPass` varchar(255) NOT NULL,
  `fName` varchar(255) NOT NULL,
  `mInit` char(1) NOT NULL,
  `lName` varchar(255) NOT NULL,
  `empPhone` varchar(255) NOT NULL,
  `hireDate` datetime NOT NULL,
  `salary` decimal(10,2) NOT NULL,
  `departmentID` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `EMPLOYEES`
--

INSERT INTO `EMPLOYEES` (`empID`, `empEmail`, `empPass`, `fName`, `mInit`, `lName`, `empPhone`, `hireDate`, `salary`,
`departmentID`) VALUES
(1, 'fdecker@gmail.com', '$2y$10$6mOFLcCnwOc7CElNMv838eFwr6zfmxu8CcveEUVkY1sYDbV4ySta', 'Francis', 'P', 'Decker',
'(989) 118-4343', '2021-04-26 00:00:00', 55000.00, 1),
(2, 'mmackey@gmail.com', '$2y$10$iEHGyhifwdiFsJKpuOzZ/e7T1b4ZGaJOXTySZpsHTXUA3OICL3awe', 'Melissa', 'A', 'Mackey',
'(742) 828-6624', '2021-05-03 09:27:41', 42000.00, 2),
(3, 'jwickens@gmail.com', '$2y$10$VZlPRT1GC.s03WgCRA0yFenn4PChYBon4cAROvv0bzbh8fN0MdDu', 'Jimbo', 'Q', 'Wickens',
'(628) 473-5839', '2021-09-26 12:55:19', 25000.00, 3);

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

--
-- Table structure for table `ORDERITEMS`
--

CREATE TABLE `ORDERITEMS` (
  `orderID` int(11) NOT NULL,
  `productID` int(11) NOT NULL,
  `quantity` int(11) NOT NULL,
  `cost` decimal(10,2) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `ORDERITEMS`
--

INSERT INTO `ORDERITEMS` (`orderID`, `productID`, `quantity`, `cost`) VALUES
(1, 5, 1, 2.50),
(1, 13, 2, 4.00),
(1, 30, 1, 1.50),
(4, 7, 1, 15.00),
(4, 1, 1, 2.50),
(3, 13, 1, 2.00),
(3, 1, 1, 2.50),
(4, 19, 1, 3.50);

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

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

CREATE TABLE `ORDERS` (
  `ordID` int(11) NOT NULL,
  `ordTotal` decimal(10,2) DEFAULT NULL,
  `ordDate` datetime NOT NULL,
```

```
  `customerEmail` varchar(255) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

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

INSERT INTO `ORDERS` (`ordID`, `ordTotal`, `ordDate`, `customerEmail`) VALUES
(1, 8.56, '2021-11-21 19:43:18', 'test@gmail.com'),
(4, 22.47, '2021-12-14 23:26:49', 'thisisatest@gmail.com'),
(3, 4.82, '2021-12-14 23:21:54', 'thisisatest@gmail.com');

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

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

CREATE TABLE `PRODUCTS` (
  `prodID` int(11) NOT NULL,
  `categoryID` int(11) NOT NULL,
  `prodName` varchar(255) NOT NULL,
  `price` decimal(10,2) NOT NULL,
  `stock` int(11) NOT NULL,
  `imageLoc` varchar(255) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

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

INSERT INTO `PRODUCTS` (`prodID`, `categoryID`, `prodName`, `price`, `stock`, `imageLoc`) VALUES
(1, 1, 'Blondie', 3.00, 1, 'images/blondie.jpg'),
(2, 1, 'Cheesecake', 2.50, 5, 'images/cheesecake.jpg'),
(3, 1, 'Cookies and Cream', 2.50, 5, 'images/cookiesandcream.jpg'),
(4, 1, 'Fudge', 2.50, 4, 'images/fudge.jpg'),
(5, 1, 'Marshmallow', 2.50, 4, 'images/marshmallow.jpg'),
(6, 1, 'Mint', 2.50, 5, 'images/mint.jpg'),
(7, 2, 'Angel Food', 15.00, 2, 'images/angelfood.jpg'),
(8, 2, 'Carrot', 15.00, 3, 'images/carrot.jpg'),
(9, 2, 'Crumb', 15.00, 3, 'images/crumb.jpg'),
(10, 2, 'Pineapple Upside-down', 15.00, 3, 'images/pineappleupsidedown.jpg'),
(11, 2, 'Pound', 15.00, 3, 'images/pound.jpg'),
(12, 2, 'Tiramisu', 15.00, 3, 'images/tiramisu.jpg'),
(13, 3, 'Chocolate Chip', 2.00, 9, 'images/chocolatechip.jpg'),
(14, 3, 'Gingerbread', 2.00, 12, 'images/gingerbread.jpg'),
(15, 3, 'Macaroon', 2.00, 12, 'images/macaroon.jpg'),
(16, 3, 'Oatmeal Raisin', 2.00, 12, 'images/oatmealraisin.jpg'),
(17, 3, 'Peanut Butter', 2.00, 11, 'images/peanutbutter.jpg'),
(18, 3, 'Shortbread', 2.00, 12, 'images/shortbread.jpg'),
(19, 4, 'Apple Cider', 3.50, 5, 'images/applecider.jpg'),
(20, 4, 'Boston Cream', 3.50, 6, 'images/bostoncream.jpg'),
(21, 4, 'Chocolate Frosted', 3.50, 6, 'images/chocolatefrosted.jpg'),
(22, 4, 'French Cruller', 3.50, 6, 'images/frenchcruller.jpg'),
(23, 4, 'Glazed', 3.50, 6, 'images/glazed.jpg'),
(24, 4, 'Strawberry Sprinkled', 3.50, 6, 'images/strawberrysprinkled.jpg'),
(25, 5, 'Banana', 1.50, 7, 'images/banana.jpg'),
(26, 5, 'Caramel Apple', 1.50, 8, 'images/caramelapple.jpg'),
(27, 5, 'Chocolate Raspberry', 1.50, 8, 'images/chocolateraspberry.jpg'),
(28, 5, 'Red Velvet', 1.50, 8, 'images/redvelvet.jpg'),
(29, 5, 'Rice', 1.50, 8, 'images/rice.jpg'),
(30, 5, 'Vanilla', 1.50, 7, 'images/vanilla.jpg'),
(31, 2, 'Fruit', 10.00, 5, 'images/fruit.jpg');

--
-- Indexes for dumped tables
--

--
-- Indexes for table `CATEGORIES`
```

```
--
ALTER TABLE `CATEGORIES`
  ADD PRIMARY KEY (`catID`);

--
-- Indexes for table `CUSTOMERS`
--
ALTER TABLE `CUSTOMERS`
  ADD PRIMARY KEY (`custEmail`),
  ADD UNIQUE KEY `custEmail` (`custEmail`);

--
-- Indexes for table `DEPARTMENTS`
--
ALTER TABLE `DEPARTMENTS`
  ADD PRIMARY KEY (`deptID`),
  ADD KEY `manID` (`manID`);

--
-- Indexes for table `EMPLOYEES`
--
ALTER TABLE `EMPLOYEES`
  ADD PRIMARY KEY (`empID`),
  ADD KEY `departmentID` (`departmentID`);

--
-- Indexes for table `ORDERITEMS`
--
ALTER TABLE `ORDERITEMS`
  ADD PRIMARY KEY (`orderID`,`productID`),
  ADD KEY `productID` (`productID`);

--
-- Indexes for table `ORDERS`
--
ALTER TABLE `ORDERS`
  ADD PRIMARY KEY (`ordID`),
  ADD KEY `customerEmail` (`customerEmail`);

--
-- Indexes for table `PRODUCTS`
--
ALTER TABLE `PRODUCTS`
  ADD PRIMARY KEY (`prodID`),
  ADD KEY `categoryID` (`categoryID`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `CATEGORIES`
--
ALTER TABLE `CATEGORIES`
  MODIFY `catID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `DEPARTMENTS`
--
ALTER TABLE `DEPARTMENTS`
  MODIFY `deptID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `EMPLOYEES`
--
ALTER TABLE `EMPLOYEES`
  MODIFY `empID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT for table `ORDERS`
```

```
--
ALTER TABLE `ORDERS`
  MODIFY `ordID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `PRODUCTS`
--
ALTER TABLE `PRODUCTS`
  MODIFY `prodID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=32;
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 */;
```

Citations

J. Murach and R. Harris, Murach's PHP and mysql: Training & reference. Fresno, CA: Mike Murach & Associates, Inc., 2017.

L. Welling and L. Thomson, PHP and MySQL: Web Development. Upper Saddle River, NJ: Addison-Wesley, 2017.