

Fair Visions Website and Database Project Documentation

Marc Stern

sternm2@montclair.edu

Login Information

User view login: user@montclair.edu
password: FP_UserView

Admin view login: sternm2@montclair.edu
password: FP_AdminView

1. Executive Summary

This project seeks to create an e-commerce website that implements a database. The database will store information about the website users, items for sale, and it will keep track of each sale made. The website contents will look visually clear and concise for easy accessibility and understanding for all types of users. Additionally, the website will implement security measures to ensure user data is properly protected.

2. Business Plan and Analysis of Client Needs

The client is Fair Visions, a Brooklyn-based indie rock band formed in 2018. The trio has performed throughout New York City, but has only released its first EP as of September of 2020. As part of their continued success, they look to build out a website that can be a platform for all things Fair Visions.

The group is looking to sell their music, as well as other merchandise, including posters, stickers, clothing, and more. They have over 30 products in their inventory, each storing a unique set of information based on the category in which they belong. These products are broken down into two main categories Music and Merchandise. Music will be categorized by the format in which it sold. Merchandise will similarly be categorized by the type, either an item, such as a mug, or a blanket, or an article of clothing, like a t-shirt or a hoodie, which will also have a size. The band leader/manager needs to be able to view, insert into, delete from, and modify the inventory. He also needs to be able to review any orders that are made so that products can be shipped out to the correct customer's address. Likewise, digital products will need to be delivered to the correct email address.

The client also expects that customers will have separate access to the website, not allowing access to update the inventory, but instead allowing them to have a shopping cart where they can add and delete items. Within the shopping cart, a user needs to be able to see what they've already added, as well as the ability to modify and finalize item quantity in the cart, and make the purchase. Purchases need to display a receipt for the user that is stored for future viewing purposes. It is also key that the shopping cart does not allow customers to add quantities of items greater than what is available as to avoid any confusion between the customers and administrators. A customer should have access to only their own shopping cart and receipts, as well as viewing capability of the inventory in order to add items to their cart.

The shopping cart needs to automatically interact with the inventory section of the database in order to keep accurate track of inventory quantities. When a customer adds an item to the cart, the product inventory should decrement the quantity of that item, effectively reserving an item for a customer. On the contrary, when an item quantity is reduced from the cart, it should be added back into the product inventory for other customers to see. If an item quantity in the cart is reduced to zero, it should additionally be deleted from the cart.

Finally, the client expects any sensitive data stored in the database, such as passwords, need to be stored in a secure way. They do not want to be liable for any database injections. This will be implemented in three steps, a login page, password encryption, and SQL parameters. There will be a full discussion on security in section 7.

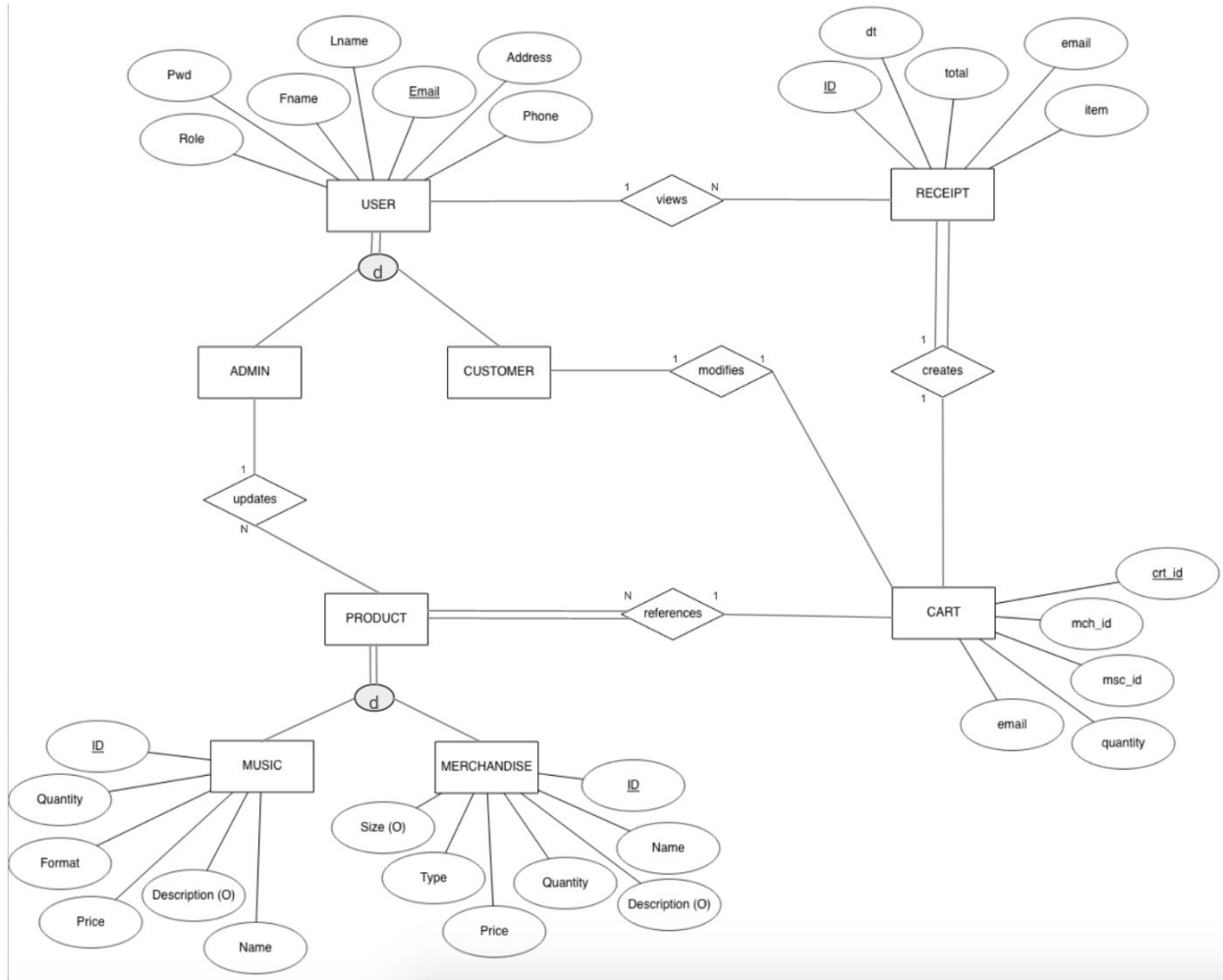
4. Conceptual Data Model

There are four main entities for which information needs to be stored. The first of which, User, has seven attributes: *Fname*, *Lname*, *Email*, *Address*, *Pwd*, *Phone*, and *Role*. *Email* is the unique identifier, and *Role* is important as it will dictate the view a user sees when logging into the website. It can be thought of as two distinct subclasses, Admin and Customer. The Product entity is also made up of two distinct subclasses, Music and Merchandise. Music stores six attributes: *ID*, *Name*, *Description*, *Format*, *Price*, and *Quantity*. Merchandise stores similar attributes, except instead of *Format*, it contains *Type*, and also stores an optional category, *Size*. In both cases, *ID* is the primary key, and *Description* is optional. Cart is another entity, storing attributes: *crt_id* (cart ID), *msc_id* (music ID), *mch_id* (merchandise ID), *quantity*, and *email*. The *crt_id* is the unique identifier, and *msc_id* and *mch_id* are both optional categories, under the condition that one is filled out, and not both. The final entity is Receipt, which has attributes *ID*, *dt* (date time), *total*, *email*, and *item*. *ID* is the unique key.

User has a relationship with every entity, but in varying ways, thus the need for the *Role* attribute to act as a subclass distinction between Admin and Customer. Admin has a relationship with Product, *updates*, where one Admin can update many Products. One customer *modifies* one Cart at a time. Both subclasses of Users can *view* many Receipts. More relationships include Cart, who *references* many Products, a one to many relationship, and *creates* Receipts, a one to one relationship. All relationships are not required except for a Receipt and Product, which must be created and referenced by a Cart, respectively. The anchor concepts for this model are as follows:

- Admin *updates* Product
- Customer *modifies* Cart
- User *views* Receipt
- Cart *references* Product
- Cart *creates* Receipt.

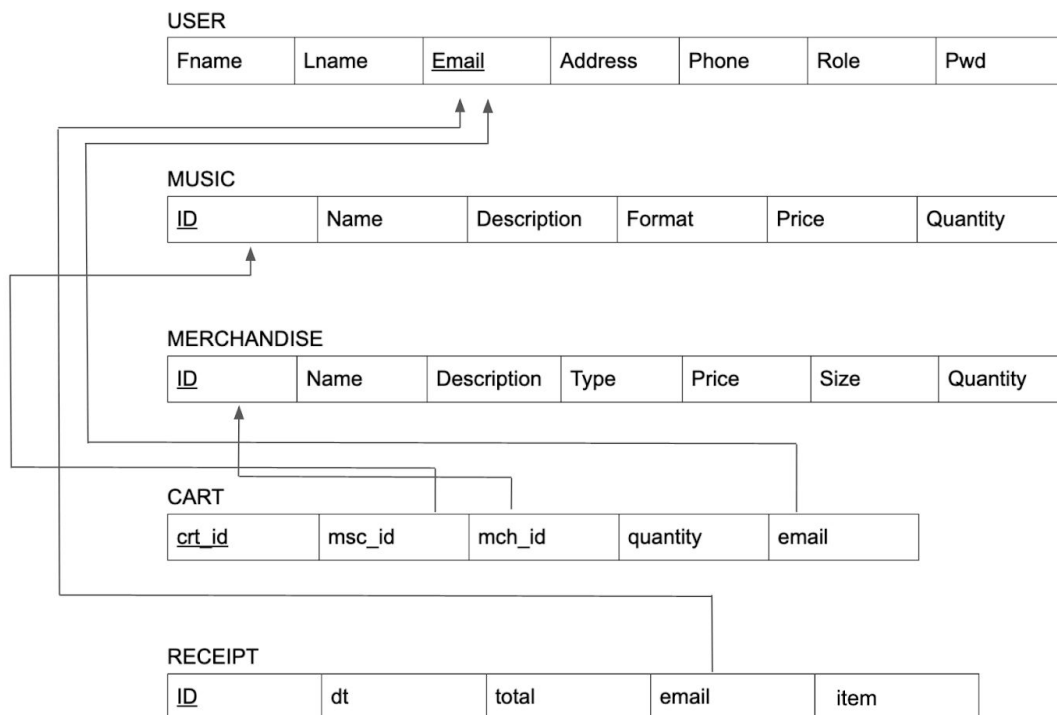
The enhanced entity relationship (EER) diagram is displayed on page 5.



5. Logical Data Model

In this section, the EER Diagram from the conceptual data model is converted into the relational model for practical use in the database construction. The relational model that has been converted (as seen on page 6) is a refined version of the traditional methods of relational model conversion, made more effective for application purposes. Here shows the relationships between entities. Email modelled in the Cart and Receipt both use the User Email as a foreign key. The Cart will also use the Merchandise ID and the Music Id as foreign keys for the mch_id and msc_id, respectively. This model describes the tables that are implemented in the database. While it did not seem necessary to create additional tables or use foreign keys for the Cart quantity, there is a relationship between the Cart and the Music and Merchandise entities that is automatically updating the quantity fields to keep track of the inventory.

Relational Model:



6. Data Views

There are two views of this database: Admin, and Customer. These are both modelled under the generalized classification User, which contains the attributes storing the information about each type of User. The User role is a defining attribute, as it will determine the view of the database seen by the current User. Upon logging in, the system checks for the user role, based on the email and password provided by the user with this SQL statement:

```
$query = "select Email, Pwd, Role
         from User
         where Email = ? and Pwd = ?";
```

The query searches the User table for a matching password and email. If a match is found, the role will also be returned, which is checked for determining which is the next php document to access. This is simply known as a view.

6.1 Customer View If a user is identified as a customer the view they see looks like this:

Buy Products	My Profile
<div>Search Music Select Format(s): <input type="checkbox"/> mp3 <input type="checkbox"/> WAV <input type="checkbox"/> CD <input type="checkbox"/> Vinyl <input type="checkbox"/> Cassette Name: <input type="text"/> <input type="button" value="Search"/></div>	<div>Update Profile Re-enter Login info to update: Email: <input type="text"/> Password: <input type="text"/> <input type="button" value="Update"/></div>
<div>Search Merch Select Type(s): <input type="checkbox"/> clothing <input type="checkbox"/> item Name: <input type="text"/> <input type="button" value="Search"/></div>	<div>My Cart <input type="button" value="View/Edit"/></div>
	<div>Past Orders <input type="button" value="View"/></div>

Here, the customer has five actionable options. The search music form allows the user to query the database by name and/or by the format. The code for this check with the subsequent query is as follows:

```

if ($format_str && $name) {
    $query = "select * from Music where Name like '%" . $name . "%' and Format in
              ('." . $format_str . "')";
}
elseif ($format_str) {
    $query = "select * from Music where Format in ('." . $format_str . "')";
}
elseif ($name) {
    $query = "select * from Music where Name like '%" . $name . "%'";
}
else {
    $query = "select * from Music";
}

```

If the Name is included, then the database selects all the columns that have match on the Name. The Name does not need to be exactly correct, thus the *Like* operator is used. Instead, it will display to the customer all possibilities that have a matching string in the Name column. If the Format is selected, all rows of the selected format are displayed to the user. Using the *in* operator allows multiple or no formats to be checked. If both are selected, then the matching conditions are more specific, including both types of queries. When nothing is selected, the user will be displayed every item in the catalog.

What the customer is displayed from their search determines which items can they currently see, enabling them to discover the corresponding item identification and later input it into the Cart, as seen here:

Music Search Results

Number of items found: 16

ID: 24
 Name: Lay Out In The Sun
 Price: \$0.99
 Quantity: 25
 Format: mp3
 Description: Single

ID: 23
 Name: A Goodbye
 Price: \$0.99
 Quantity: 25
 Format: WAV
 Description: Single

ID: 22
 Name: A Goodbye
 Price: \$0.99
 Quantity: 25
 Format: mp3
 Description: Single

ID: 21
 Name: A Way Out
 Price: \$9.99
 Quantity: 25
 Format: Cassette
 Description: EP

ID: 20
 Name: A Way Out
 Price: \$9.99
 Quantity: 25
 Format: Vinyl
 Description: EP

Add to Cart

Enter the item ID to add an item to your cart.

ID:

Quantity:

This screenshot shows the Search Results view, and allows them to add items to the Cart by inputting the item ID and the quantity. When an item is added to the Cart, a check for each input is done. The first being that both requirements were filled out. The second, for ID, checks to make sure that there is a corresponding item in the catalog. Considering that the item ID is

automatically created and incremented, there will never be an ID 0. So, for example, if ID 0 is added to the Cart, an error will occur. That check is seen here:

```
$query = "select Quantity
          from Music
          where ID = $id";
```

The third check is done on the quantity. This check requires both querying the database and mathematical logic. First, the quantity is returned for the matching item. If that quantity, the one currently in the catalog, is less than the requested amount, then there is not enough of that item to add to the Cart, and an error occurs. Next, a new query returns the amount of that item that already exists in the cart:

```
$query = "select quantity
          from Cart
          where msc_id = $id";
```

If the item does not already exist in the cart, then a simple insert is applied to the Cart table, and the item ID, its quantity, and the customer's email are input:

```
$query = "insert into Cart (msc_id, mch_id, quantity, email)
          values ($id, NULL, $quantity, \"$email\")";
```

If the item does exist in the cart, the quantities are added together, and query updating the cart the with new quantity is input, matching on the ID:

```
$query = "update Cart
          set quantity = $qty_sum
          where msc_id = $id";
```

The final step in this process is to adjust the quantity now available in the inventory based upon the quantity that was just added to the cart. The following code applies this:

```
$query = "update Music
          set Quantity = $qty_diff
          where ID = $id";
```

The customer can follow the same process for merchandise, but instead of searching on music format, they can search by Name and/or Type of merchandise product. First a check on the user input is performed:

```
if ($type_str && $name) {
    $query = "select * from Merchandise where Name like '%" . $name . "%' and Type in ('" . $type_str . "')";
}
elseif ($type_str) {
    $query = "select * from Merchandise where Type in ('" . $type_str . "')";
}
elseif ($name) {
    $query = "select * from Merchandise where Name like '%" . $name . "%'";
}
else {
    $query = "select * from Merchandise";
}
```

Then, the quantity of the requested item is compared with the quantity in stock, as seen with the following two queries:

```
$query = "select Quantity
        from Merchandise
        where ID = $id";
```

```
$query = "select quantity
        from Cart
        where mch_id = $id";
```

Once again, if the quantity is allowed, then a check on the item already existing in the cart is performed. If not the following query inputs the ID, quantity, and user email into the Cart table:

```
$query = "insert into Cart (mch_id, mch_id, quantity, email)
        values (NULL, $id, $quantity, \"$email\")";
```

Otherwise, only the quantity for that item matching the ID is updated in the Cart:

```
$query = "update Cart
        set quantity = $qty_sum
        where mch_id = $id";
```

And finally, again, the quantity in the appropriate catalog needs to adjust to a new total quantity based on the amount added to the cart;

```
$query = "update Merchandise
        set Quantity = $qty_diff
        where ID = $id";
```

The customer can also view their Cart. Clicking the view/edit button in the My Cart form displays the user the items in their cart. Matching on the user email, the following code:

```
$query = "select msc_id, mch_id, quantity
        from Cart
        where Email = \"$email\"";
```

displays the following view:

My Cart

Number of items found: 5

Merch
ID: 28
Name: Hoodie
Price: \$19.99
Quantity: 2
Type: clothing
Size: small
Description: Logo Design 2

Merch
ID: 35
Name: Mug
Price: \$7.99
Quantity: 5
Type: item
Size:
Description: Logo Design 2

Merch
ID: 30
Name: Beanie
Price: \$19.99
Quantity: 1
Type: clothing
Size: small
Description: Logo Design 1 (one size fits all)

Music
ID: 26
Name: Feels Right
Price: \$0.99
Quantity: 1
Format: mp3
Description: Single

Order Items

Make order!

Remove Items from Cart

Enter Change:

☐ Music ☐ Merchandise

ID:

Quantity:

make changes

From here, a customer can see the items in their cart, and the corresponding IDs. Now they can do one of two actions, either remove items from the cart by quantity, where reducing a quantity to zero removes the item from the cart, or they can make the order. The former follows a similar process to the aforementioned quantity checks for adding to the cart. Assuming the customer inputs a valid quantity to remove, the first check performed is to make sure the item already exists in the cart:

```
$query = "select quantity
         from Cart
         where email = \"\$email\" and $id_col = $id";
```

If it does, it compares the requested quantity change to the amount in the cart. If the difference equals zero, then the item is removed from the cart:

```
$query = "delete
         from Cart
         where $id_col = $id and email = \"\$email\"";
```

If the quantity is less than zero, the user is requesting to take out more items than it has in the cart, which will cause an error when adjusting the catalog product quantity, and therefore throws an error. If the quantity is greater than zero, then the requested amount is valid, and the Cart decrements the quantity of the requested item:

```
$query = "update Cart
         set quantity = $qty_diff
         where $id_col = $id and email = \"\$email\"";
```

In this case, the catalog needs to have product quantity added back in, as done with the following query:

```
$query = "select Quantity
         from $table
         where ID = $id";
```

The second action a customer can do on this page is to make the order. This is as simple as clicking the “make order!” button (which is technically a form). Upon submitting the form, the purchase is made, the cart creates a receipt row, and the receipt information is displayed to the user. Receipt creation is done through a few steps. The first is to return all of the items that are currently in the customer’s cart:

```
$query = "select *
         from Cart
         where email = \"\$email\"";
```

If no results are found, then the cart is empty for that user, and the purchase is not made. Otherwise, one of two actions is performed: either the price, name format and ID are returned from Music:

```
$query = "select Price, Name, Format, ID
         from Music
         where ID in ($msc_id_str);
```

or in the case of Merchandise, just the price, name, and ID:

```
$query = "select Price, Name, ID
         from Merchandise
         where ID in ($mch_id_str);
```

Using this information, a total price can be calculated, and the appropriate fields name, format, and product ID, will be stored in the Receipt table, and displayed to the user:

```
$query = "insert into Receipt (total, email, item)
values ($total, \"\$email\", \"\$items_str\")";
```

If the order is made successfully, then all the items in the cart for this particular user are removed:

```
$query = "delete
from Cart
where email = \"\$email\"";
```

Making an order looks like this:

RECEIPT

Number of items ordered: 5

total: **\$101.9**

items: Lay Out In The Sun [mp3] (1), Feels Right [mp3] (1), Hoodie (2), Beanie (1), Mug (5)

[return](#)

Now that the customer has a Receipt identified by their login email, they can return to the main customer screen, and view their receipts there, via Past Orders, done with a simple select query:

```
$query = "select ID, DATE(dt) as d, TIME(dt) as t, total, item
from Receipt
where email = \"\$email\"";
```

The Receipt table automatically stores a datetime data type, which is then returned by the above code as individual results for date and time. Viewing one's receipts looks like this:

PAST ORDERS

Orders made: 1

ID: 6
Date: 2020-12-06
Time: 21:28:42
Total \$: 101.9
Items: Lay Out In The Sun [mp3] (1), Feels Right [mp3] (1), Hoodie (2), Beanie (1), Mug (5)

[return](#)

The final action a customer takes is to change their user profile information. The user is required to log in a second time, to verify that the user is themselves and can update information such as the password. The login check is an identical query to the one from the login page, matching on email and password. With success, the page views as such:

Update My Profile

1. Name: Mark Styrn
 Email: user@montclair.edu
 Address: 10 Normal Ave, Montclair, NJ 07043
 Phone #: 1234567890

Enter Change(s):

First Name:

Last Name:

Email:

Address:

Phone #:

Password:

The customer can change their first or last name, respectively:

```

$query = "update User
        set FName = \"\$fname\"
        where Email = \"\$hid_email\"";

$query = "update User
        set Lname = \"\$lname\"
        where Email = \"\$hid_email\"";
    
```

Address, Phone, and Password changes are done similarly:

```

$query = "update User
        set Address = \"\$address\"
        where Email = \"\$hid_email\"";

$query = "update User
        set Phone = \"\$phone\"
        where Email = \"\$hid_email\"";

$query = "update User
        set Pwd = \"\$password\"
        where Email = \"\$hid_email\"";
    
```

Email is also available to change, but due to it being the primary key of the table it requires an extra check, which identifies if the requested email change is already used by a user:

```
$query = "select Email from User where Email = \"\$email\"";
```

If the email is not identified, then it can be updated as requested:

```
$query = "update User set Email = \"\$email\" where Email = \"\$hid_email\"";
```

6.2 Admin View If a user is identified as an admin the view they see looks like this (the image is zoomed out to capture the whole page at once, there is not actually large gap between the two columns):

Music

Insert Music

Name:

Price:

Quantity:

Description:

Format:

☐ mp3
☐ WAV
☐ CD
☐ Vinyl
☐ Cassette

Search Music

Select Format(s):

☐ mp3 ☐ WAV ☐ CD ☐ Vinyl ☐ Cassette

Name:

Delete Music

ID:

Modify Music

ID:

Merchandise

Insert Merch

Name:

Price:

Quantity:

Description:

Type:

☐ clothing
☐ item

Size (for clothing only):

☐ small
☐ large
☐ none

Search Merch

Select Type(s):

☐ clothing ☐ item

Name:

Delete Merch

ID:

Modify Merch

ID:

The admin view has many of the same capabilities as the customer, such as searching the catalog, and inserting, deleting, and modifying content, but in a different context. While the customer updates their cart, the admin will update the database directly, and does not interact with the cart. In this section, I will only be addressing the inventory updates, as the queries for searching are identical, although the resulting page view does not include the option to add the items to the cart. Rather it serves the purpose of giving the admin quick access to view the product IDs in order to input the correct deletions and/or modifications.

Music insertion checks first to make sure that all necessary fields are recorded, in this case, that is all but description. If all the categories are input, then the system checks for an item match, as only one of each item can exist:

```
$query = "select Name, Format
        from Music
        where Name = \"\$name\" and Format = \"\$format\"";
```

Assuming there is no match, and the item has not already been input, then the system simply inputs the fields to their corresponding column in the Music table:

```
$query = "INSERT INTO Music (Name, Format, Price, Quantity, Description)
        VALUES (\"\$name\", \"\$format\", \"\$price\", \"\$quantity\", \"\$description\")";
```

Merchandise input follows the same logic, but uses slightly different fields, so the requirements are different such that only the clothing type needs a corresponding size. A user can leave the

size entry blank for an item type. The following code demonstrates the check to see if the inserted item already exists in the database:

```
$query = "select Name, Type, Description
        from Music
        where Name = \"\$name\" and Format = \"\$format\" and Description = \"\$description\"";
```

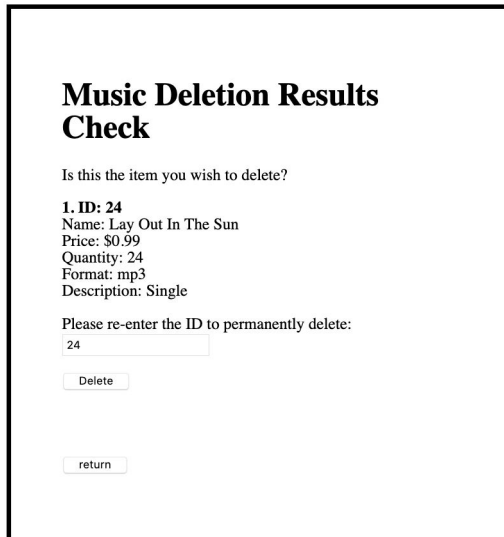
Then, if no match is found, the following insert is applied to enter the new product:

```
$query = "insert into Merchandise (Name, Description, Price, Type, Size, Quantity)
        values (\"$name\", \"$description\", \"$price\", \"$type\", \"$size\", \"$quantity\")";
```

When a product ID is entered to delete an item from the inventory, the program shows the user the item they are about to delete, and makes sure that it is the correct item using the following query:

```
$query = "select * from Music where ID = $id";
```

The admin user sees this page as:



Music Deletion Results Check

Is this the item you wish to delete?

1. ID: 24
Name: Lay Out In The Sun
Price: \$0.99
Quantity: 24
Format: mp3
Description: Single

Please re-enter the ID to permanently delete:

This gives the user a buffer to either return back to the previous page, and save the item, or to complete the delete with a simple click. That click performs this query, which double checks the database for the selected item:

```
$query = "select ID from Music where ID = $id";
```

The subsequent query performs the deletion:

```
$query = "delete from Music where ID = $id";
```

Once again, the Merchandise deletion is the same process. The first check is to inquire about the validity of the ID number:

```
$query = "select * from Merchandise where ID = $id";
```

Once identified, the user is prompted with a second form, which performs the deletion:

Merch Deletion Results Check

Is this the item you wish to delete?

1. ID: 30

Name: Beanie

Description: Logo Design 1 (one size fits all)

Price: \$19.99

Type: clothing

Size: small

Quantity: 14

Please re-enter the ID to permanently delete:

Then, the second check for ID validity:

```
$query = "select ID from Merchandise where ID = $id";
```

And finally, the deletion:

```
$query = "delete from Merchandise where ID = $id";
```

Lastly, an admin can modify the information in a given product. Looking at music first, as with the previous tasks, a check for valid ID is done:

```
$query = "select * from Music where ID = $id";
```

Once verified, a new page opens giving the user options for categories to alter, looking like:

Music Modification Results Check

Is this the item you wish to modify?

1. ID: 30

Name: Modern Kids

Price: \$9.99

Quantity: 25

Format: CD

Description: EP (pre-order)

Enter Change(s):

ID:

Name:

Price:

Format:

☐ mp3

☐ WAV

☐ CD

☐ Vinyl

☐ Cassette

Quantity:

Description:

Any one, or multiple, of these categories can be changed with the click of the “make changes” button (like delete, it’s actually a form). As always, a check on the identification is done first:

```
$query = "select ID from Music where ID = $id";
```

Then it checks to see if the item that is being updated is altered to an item that already exists in the database, by checking for existing format and name matches, when one of the two columns in left empty, respectively:

```
$query = "select Name, Format from Music where Name = \"\$name\" and Format = \"\$hid_format\"";
```

```
$query = "select Name, Format from Music where Name = \"\$hid_name\" and Format = \"\$format\"";
```

If no match is found, the update is made:

```
$query = "update Music set Name = \"\$name\" where ID = $id";
```

```
$query = "update Music set Format = \"\$format\" where ID = $id";
```

For other categories, in the order price, quantity, and description, the following queries demonstrate the updates:

```
$query = "update Music set Price = \"\$price\" where ID = $id";
```

```
$query = "update Music set Quantity = \"\$quantity\" where ID = $id";
```

```
$query = "update Music set Description = \"\$description\" where ID = $id";
```

The same view type for merchandise starts with an ID check:

```
$query = "select * from Merchandise where ID = $id";
```

If the ID is found, the user is displayed this new page:

Merchandise Modification Results Check

Is this the item you wish to modify?

1. ID: 30
Name: Beanie
Description: Logo Design 1 (one size fits all)
Price: \$19.99
Type: clothing
Size: small
Quantity: 14

Enter Change(s):

ID:

Name:

Price:

Type:

☐ clothing
☐ item

Size (for clothing only):

☐ small
☐ medium
☐ large

Quantity:

Description:

The update queries for name, price, and quantity look like the following:

```
$query = "update Merchandise set Name = \"\$name\" where ID = $id";
```

```
$query = "update Merchandise set Price = \"\$price\" where ID = $id";
```

```
$query = "update Merchandise set Description = \"\$description\" where ID = $id";
```

The uniqueness of modifying this table comes with the type category, which determines if size is required. After passing through the proper logic checks for this issue successfully, the update is permitted and recorded via this query if the product is changed from clothing to item:

```
$query = "update Merchandise set Type = \"\$type\", Size = null where ID = $id";
```

On the contrary, when clothing is changed from item, then the size also needs to be input. If it is, the following query updates the product information:

```
$query = "update Merchandise set Type = \"\$type\", Size = \"\$size\" where ID = $id";
```

7. Security Implementation

There are four main techniques used in this project to help protect the security of user data, and to prevent SQL injections. The first technique is the implementation of a login landing page. The user accesses the login page, via the home page login button, separately from the database. This allows the system to actually check to make sure that the user has been granted access to their appropriate view. As previously mentioned, the system uses the landing page to check for the user role upon finding a match, with which it will display the user the appropriate page, the two roles being admin and customer.

Secondly, when the system checks for a match on email and password, although the user types a password with letters, numbers, and/or special characters, the entirety of the passcode is converted into a much longer random string of numbers and letters, enabling greater security for storing user passwords. In other words, the users password is not the password stored in the database. That is accomplished with the following code:

```
$query = "select Email, Pwd, Role
          from User
          where Email = ? and Pwd = ?";
$stmt = $mysql->prepare($query);
$stmt->bind_param('ss', $email, sha1($password));
$stmt->execute();
```

What is seen above is that when the password is passed into the query that finds a match on the user, the password is passed through the *sha1()* function, which converts the password to a long random string. This is also done when a customer decides to change their password:

```
$password = sha1($password);
```

What is also noticeable about the code syntax above is that the query does not directly look for email and password values. Instead, a prepared statement is used with email and password as the two parameters. This is the third technique, and it helps to prevent against SQL injections. What the code is doing is instead of simply concatenating the passed values, email and password, and running the query based on the values and checking for a match, as

seen with queries in section 6, it uses a prepared statement to bind the values to a specific parameter. What this means is that the scope of the *where* condition is limited to each of the parameters. A hacker could input a SQL injection into the login email or password, and instead of not finding a match and still running the injected query, it will simply not find a match on each passed value. The intended malicious code will not run as part of the query.

The final technique also has to do with preventing SQL injections. Every time the database is opened to run a query in any script used for this website, the code looks like this:

```
@ $mysql = new mysqli('localhost:3306', 'sternm2', '●●●●●●', 'sternm2_FAIR VISIONS');
```

(The dots represent where a password is typed, which I've removed from this documentation for my own security.) By placing an @ symbol at the beginning of the variable that opens the database, any error messages from the system are suppressed. Typically a developer needs to see these error messages to inform them about what needs to be changed for their code to run properly. In this vein, it is important to not display those messages that could help a hacker understand why their injected SQL code is not doing the job it was intended to do. A hacker could use error messages, just like a developer, to aid them into writing code that the system understands. The system does not understand the intent of the coder's code, and therefore these security protections against SQL injections are important for database and website implementation.

8. Website Implementation

The website homepage looks like this:



A visitor can click on the right button to directly access the link to their music on Spotify. On the left, a visitor can login to the website. The css style code for these buttons, and the column layout is as follows:

```

<style>
* {
  box-sizing: border-box;
}

.column {
  float: left;
  width: 33.33%;
  padding: 10px;
}

a:link, a:visited {
  background-color: white;
  color: black;
  border: 2px solid green;
  padding: 10px 20px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: green;
  color: white;
}
</style>

```

Additionally, every page outside of the home page has its contents displayed inside a frame, as seen in several screen captures in section 6. The code looks like:

```

<style>
* {
  box-sizing: border-box;
}

.form {
  background-color: white;
  width: 500px;
  border: 5px solid black;
  padding: 50px;
  margin: 30px;
  text-align: left;
}
</style>

```

Moreover, after logging in, for both types of users, there is a form at every dead end, such as errors, that allows the user to return to their respective homepage. In the case of the admin, the form code looks like this:

```

<form method="post" action="admin_page.php">
  <p><input type="submit" name="return" value="return"></p>
</form>

```

This is straightforward, and the implementation makes it easy for users to navigate back to the main page. The customer return form is slightly more complicated, however, and is written:

```

<form method="post" action="customer_page.php">
  <p><input type="submit" name="return" value="return"></p>
  <input type="hidden" id="hid_email" name="hid_email" value="<?php echo $email?>">

```

There is an extra line of code in this form, and it passes a hidden value, which is set to the email variable passed in from the previous script. The reason for this is that the customer email is required for every script that runs a check on the database. Since the customer only inputs their

email information at the login page, and to update their profile, the rest of the pages need to pass around the email value. Every form passes the email variable as a hidden value that can be retrieved in the subsequent script.

```
if ($num_results > 0) {  
    if ($row[2] == "Admin") {  
        header("Location: admin_page.php");  
    }  
    elseif ($row[2] == "Customer") {  
        header("Location: customer_page.php?email=$email");  
    }  
}
```

The above code is the check performed on the login page to see the matched user's role. The code demonstrates that after determining the user role, it redirects the user to the corresponding page. In the case of the customer, it also passes the email variable in the redirect, just like all the following forms will too. Because there are two different cases for passing the hidden variable, there are two different syntaxes for returning the value in the subsequent script:

```
$email = $_GET['email'];  
if (empty($email)) {  
    $email = $_POST['hid_email'];  
}
```

The `$_GET[]` method is required for retrieving the value from redirect, while the `$_POST[]` method retrieves the value from a form. The program first retrieves the value from a redirect. If the value is empty, implying nothing was found, then the program resets the variable to retrieve the value from a form on the preceding page.

9. Analysis of Database

Most of the tables in the database are designed in first normal form. The main reason for this design is for simplicity. First normal form is not as restrictive or pure as second or third normal form, but for this project it was necessary to simplify the tables to work under time constraints, but also to avoid joins as much as possible. In fact, all database joins are implemented imperatively. This was made easy by creating a database in first normal form. For extra convenience, there are two tables that break normal form for practical reasons.

The first table to break first normal form is the User table. This table is mostly in first normal form, except for address attribute. Typically an address should be broken up into pieces, having a column for each element of information, such as address number, street name, city, state, and zip code. The implementation of address in this table is one long string. Now, technically the column can only support one value, which is a string, but it is arguably not in any normal form.

Secondly, the receipt table. This table certainly breaks the normal form. Instead of having a separate table for the items purchased, linking them to the receipt ID, the receipt table simply takes a string input that includes all items purchased and stores them in the item column. Again, technically the column only takes one input, and the string concatenation happens before the insert query, but this could certainly have been implemented more robustly. Given the time

constraint on the project, this was a trade-off I decided to make, as it does not affect the functionality of the system.

In general all the data is reachable, and there is little redundancy. The cart table has an odd implementation, checking storing either the music product ID or the merch product ID. This table has a null value in every row by definition, and therefore is not the best implementation. A better way to implement this would be to have a separate product table that stores product ID that matches the products in the music and merch tables respectively. This would allow the cart to only need a column for product ID, relieving the table of all null values. It could also streamline some of the website contents like having separate inserts for music and merch based on the ID. Currently, there can be two products with the same ID, and the system requires the user to tell it whether it will be found in merchandise or music. Instead it would be easier for the user if there was only one ID per product.

Overall, the system could have been implemented in second normal form by adding these extra tables. The main reason I used a combination of first normal form and breaking the normal form was for time constraint, convenience, and overall understanding of database implementation, which I have a much greater understanding of having completed this project.

11. Conclusion

Now that the project has been completed, there are still many areas that can use improvement. The first improvement I would make is the homepage. To truly meet the band's expectations for the project, the homepage needs to be a lot more lively, and include a multitude of links to every location their music can be streamed, as well as links to their social media. The group also has a link tree from their Instagram account that includes links to causes that are important to the band, such as voting registration check, and funds for saving local Brooklyn venues during the pandemic. A front page is key for this type of client, so that is the area where I would start.

Secondly, the database could be more robust. Given more time, I would create the database entirely in second normal form, as mentioned in section 9. Accomplishing this would allow the database to continue to grow and add features without having to redo the tables along the way. In essence, the tables will most likely need to be revised as new elements are added to the website, therefore I might as well start by updating the database form, and the subsequent code that queries the database.

There are additional security measures I would take. For example, the @ symbol in front of a variable is implemented every single time the database is opened. Likewise, I would add the security functionality from the login page that uses the prepared statement with the variable values as parameters every time the user inputs data. Right now the site is secure from the outside, but slightly less secure on the inside. In the future, if I were to add functionality for a user registration, it would be important to make sure that invited users are not logging in just to run malicious code while inserting items to their cart, for example.

One missing area from this website is the admin's ability to review receipts. Unfortunately, the current implementation would not allow the admin to view orders and send the products to the users who ordered them. However, this could be implemented easily by

running a query on the receipt table to select all the columns for every row. Without the capacity for real transactions to take place, and with limiting time constraints, this view was left out. It would certainly be one of the first areas, if not the first, to implement in an updated version of the website.

The final area I would keep working on is streamlining the website so there are fewer forms for both types of users to fill out. This was briefly mentioned as part of the database table designs in section 9, but in general, I think it would be possible for, in the case of the admin, to provide a toggle for insert, modify, and delete, instead of having three separate forms for each type of product type. With more time, there is likely a way to implement the admin capabilities in one or two forms. This would provide a simple view for the admin, who is not a database designer.

Appendix

FairVisions_front_page.php

```
<html>
<head>
<div><title>Fair Visions</title></div>
</head>

<style>
* {
  box-sizing: border-box;
}

.column {
  float: left;
  width: 33.33%;
  padding: 10px;
}

a:link, a:visited {
  background-color: white;
  color: black;
  border: 2px solid green;
  padding: 10px 20px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: green;
  color: white;
}
</style>

<body>
<h1 style="font-family:Arial Black"><center>Fair
Visions</center></h1>
<div class="column">
  <center><a href='data/Login_Page.php'
style="font-family:Tahoma, serif;"> Login here </a></center>
```

```
</div>

<div class="column">
  <center><img src='../data/IMG_5452.JPG'
alt='Front Page Band Photo' width='468'
height='607.8'></center>
</div>

<div class="column">
  <center><a
href='https://open.spotify.com/artist/718Ln0Su1y5A3qPk6O1t
2H?si=dG5bK3dWQeK0D6jT5CJapw'
style="font-family:Tahoma,
serif;">Spotify</a></center></body>
</div>
</html>
```

Login_Page.php

```
<style>
* {
  box-sizing: border-box;
}

.form {
  background-color: white;
  width: 500px;
  border: 5px solid black;
  padding: 50px;
  margin: 30px;
  text-align: left;
}
</style>
<center><div class="form">
  <h1>Please Log In</h1>
  <form method="post" action="Login_Check.php">
    <p>Email: <input type="text" name="email"></p>
```

```

        <p>Password: <input type="password"
name="password"></p>
        <p><input type="submit" name="submit" value="Log
In"></p>
    </form>
</div></center>

```

Login_Check.php

```

<?php
    $email = $_POST['email'];
    $password = $_POST['password'];

    if (!$email || !$password) {
        echo "Missing email and/or password."; ?>
        <form method="post" action="Login_Page.php">
            <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    @ $mysql = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');
    if ($mysql -> connect_errno) {
        echo "Failed to connect to MySQL: " . $mysql ->
connect_error; ?>
        <form method="post" action="Login_Page.php">
            <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $query = "select Email, Pwd, Role
        from User
        where Email = ? and Pwd = ?";
    $stmt = $mysql->prepare($query);
    $stmt->bind_param('ss', $email, sha1($password));
    $stmt->execute();

    if(!$stmt) {
        echo "Cannot run query."; ?>
        <form method="post" action="Login_Page.php">
            <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $result = $stmt->get_result();
    $row = $result->fetch_row();
    $num_results = $result->num_rows;

    if ($num_results > 0) {
        if ($row[2] == "Admin") {
            header("Location: admin_page.php");
        }
        elseif ($row[2] == "Customer") {
            header("Location:
customer_page.php?email=$email");

```

```

        }
    }
    else {
        echo "Unable to identify User Role"; ?>
        <form method="post" action="Login_Page.php">
            <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
}

```

admin_page.php

```

<html>
<head>
<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}

.column {
    float: left;
    width: 50%;
    padding: 10px;
}

</style>
</head>

<center><div class="column">
    <h1>Music</h1>
    <body>
        <div class="form">
            <h1>Insert Music</h1>
            <form method="post" action="insert_music.php">
                <p>Name: <input type="text" name="name"></p>
                <p>Price: <input type="text" name="price"></p>
                <p>Quantity: <input type="text"
name="quantity"></p>
                <p>Description: <textarea id="description"
name="description" rows="2" cols="40"></textarea></p>
                <p>Format: <p>
                    <input type="radio" id="mp3" name="format"
value="mp3">
                    <label for="mp3">mp3</label><br>
                    <input type="radio" id="WAV" name="format"
value="WAV">
                    <label for="WAV">WAV</label><br>
                    <input type="radio" id="CD" name="format"
value="CD">
                    <label for="CD">CD</label><br>
                    <input type="radio" id="Vinyl" name="format"
value="Vinyl">

```



```

        <label for="Vinyl">Vinyl</label><br>
        <input type="radio" id="Cassette" name="format"
value="Cassette">
        <label for="Cassette">Cassette</label><br>

        <p><input type="submit" name="submit"
value="Insert"></p>
        </form>
    </div>

    <div class="form">
    <h1>Search Music</h1>
    <form action="music_search_results.php"
method="post">
        Select Format(s):<br />
        <input type="checkbox" id="mp3" name="format[]"
value="mp3">
        <label for="mp3">mp3</label>
        <input type="checkbox" id="WAV" name="format[]"
value="WAV">
        <label for="WAV">WAV</label>
        <input type="checkbox" id="CD" name="format[]"
value="CD">
        <label for="CD">CD</label>
        <input type="checkbox" id="Vinyl" name="format[]"
value="Vinyl">
        <label for="Vinyl">Vinyl</label>
        <input type="checkbox" id="Casette"
name="format[]" value="Casette">
        <label for="Casette">Casette</label>
        <p>Name:<br />
        <input name="Name" type="text" size="40"></p>
        <input type="submit" name="submit"
value="Search">
    </form>
    </div>

    <div class="form">
    <h1>Delete Music</h1>
    <form method="post"
action="delete_music_check.php">
        <p>ID: <input type="text" name="ID"></p>
        <input type="submit" name="delete" value="Delete">
    </form>
    </div>

    <div class="form">
    <h1>Modify Music</h1>
    <form method="post"
action="modify_music_check.php">
        <p>ID: <input type="text" name="ID"></p>
        <input type="submit" name="modify" value="Modify">
    </form>
    </div>
</body>
</div></center>

<center><div class="column">
    <h1>Merchandise</h1>
    <body>
        <div class="form">

```

```

            <h1>Insert Merch</h1>
            <form method="post" action="insert_merch.php">
                <p>Name: <input type="text" name="name"></p>
                <p>Price: <input type="text" name="price"></p>
                <p>Quantity: <input type="text"
name="quantity"></p>
                <p>Description: <textarea id="description"
name="description" rows="2" cols="40"></textarea></p>
                <p>Type: </p>
                <input type="radio" id="clothing" name="type"
value="clothing">
                <label for="clothing">clothing</label><br>
                <input type="radio" id="item" name="type"
value="item">
                <label for="item">item</label><br>
                <p>Size (for clothing only): </p>
                <input type="radio" id="small" name="size"
value="small">
                <label for="small">small</label><br>
                <input type="radio" id="large" name="size"
value="large">
                <label for="large">large</label><br>
                <p><input type="submit" name="submit"
value="Insert"></p>
            </form>
        </div>

        <div class="form">
        <h1>Search Merch</h1>
        <form action="merch_search_results.php"
method="post">
            Select Type(s):<br />
            <input type="checkbox" id="clothing" name="type[]"
value="clothing">
            <label for="clothing">clothing</label>
            <input type="checkbox" id="item" name="type[]"
value="item">
            <label for="item">item</label>
            <p>Name:<br />
            <input name="name" type="text" size="40"></p>
            <input type="submit" name="search"
value="Search">
        </form>
        </div>

        <div class="form">
        <h1>Delete Merch</h1>
        <form method="post"
action="delete_merch_check.php">
            <p>ID: <input type="text" name="ID"></p>
            <input type="submit" name="delete" value="Delete">
        </form>
        </div>

        <div class="form">
        <h1>Modify Merch</h1>
        <form method="post"
action="modify_merch_check.php">
            <p>ID: <input type="text" name="ID"></p>
            <input type="submit" name="modify" value="Modify">
        </form>
        </div>
    </div>
</center>

```

```

        </div>
    </body>
</div></center>

</html>

```

insert_music.php

```

<html>
<head>
    <title>Music Product Entry Results</title>
</head>
<body>
<h1>Music Product Entry Results</h1>
<?php

$name=ucwords(strtolower($_POST['name']));
$description=$_POST['description'];
$price=$_POST['price'];
$format=$_POST['format'];
$quantity=$_POST['quantity'];

$bad=0;

if (!$name) {
    echo "Missing name";
    $bad=1;
}

if (!$format) {
    echo "Missing format";
    $bad=1;
}

if (!$price) {
    echo "Missing price";
    $bad=1;
}

if (!$quantity) {
    echo "Missing quantity";
    $bad=1;
}

if ($bad>0) {
    exit(1);
}

if (!get_magic_quotes_gpc()) {
    $name = addslashes($name);
    $description = addslashes($description);
    $format = addslashes($format);
    $price = addslashes($price);
    $quantity = doubleval($quantity);
}

@ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
    echo "Error: Could not connect to database. Please try
again later."; ?>
    <form method="post" action="admin_page.php">

```

```

        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$query = "select Name, Format
        from Music
        where Name = \"\$name\" and Format = \"\$format\"";

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

if (!$result) {
    echo "Query failed to execute"; ?>
    <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$num_results = $result->num_rows;

if ($num_results > 0) {
    echo 'Item already exists.'; ?>
    <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$query = "INSERT INTO Music (Name, Format, Price,
Quantity, Description)
        VALUES (\"$name\", \"\$format\", \"\$price\",
\"$quantity\", \"\$description\")";
$result = $db->query($query);

if ($result) {
    echo $db->affected_rows." music items were properly
inserted into database.";
} else {
    echo "An error has occurred. The item was not
added.";
}
?>
    <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
    $db->close();

?>
</body>
</html>

```

music_search_results.php

```

<html>
<head>

```

```

<title>Music Search Results</title>
</head>
<body>
<h1>Music Search Results</h1>
<?php

$format=$_POST['format'];
$name=trim($_POST['name']);
$format_str = "";

if (!get_magic_quotes_gpc()){
    if (count($format) > 0) {
        $a = [];
        foreach ($format as $f) {
            $f = "" . addslashes($f) . "" ;
            $a[] = $f;
        }
        $format_str = join(', ', $a);
    }
    $name = addslashes($name);
}

@ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

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

if ($format_str && $name) {
    $query = "select * from Music where Name like
'%" . $name . "%' and Format in
('" . $format_str . ")";
}
elseif ($format_str) {
    $query = "select * from Music where Format in
('" . $format_str . ")";
}
elseif ($name) {
    $query = "select * from Music where Name like
'%" . $name . "%'";
}
else {
    $query = "select * from Music";
}

$result = $db->query($query);
if (!$result) {
    echo "Query failed to execute";
}

$num_results = $result->num_rows;

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

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

```

```

echo "</strong><br />Name: ";
echo stripslashes($row['Name']);
echo "<br />Price: $";
echo stripslashes($row['Price']);
echo "<br />Quantity: ";
echo stripslashes($row['Quantity']);
echo "<br />Format: ";
echo stripslashes($row['Format']);
echo "<br />Description: ";
echo stripslashes($row['Description']);
echo "</p>";
}

```

```

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

```

```

?>
</body>
</html>
-----

```

delete_music_check.php

```

<html>
<head>
<title>Music Deletion Results Check</title>
</head>
<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><body>
<div class="form">
    <h1>Music Deletion Results Check</h1>
    <?php

        $id = (int)$_POST['ID'];

        @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

        if (mysqli_connect_errno()) {
            echo 'Error: Could not connect to database. Please
try again later.'; ?? <br /><br />
            <form method="post" action="admin_page.php">
                <p><input type="submit" name="return"
value="return"></p>
            </form>
            <?php exit();
        }
    }

```

```

if ($id > 0) {
    $query = "select * from Music where ID = $id";
}
else {
    echo "Music Product ID: [$id] is invalid. No deletion
will be made."; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $result = $db->query($query);
    if (!$result) {
        echo "Query failed to execute"; ?> <br /><br />
        <form method="post" action="admin_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            </form>
            <?php exit();
        }

        $num_results = $result->num_rows;

        if ($num_results == 0) {
            echo "No match was found."; ?> <br /><br />
            <form method="post" action="admin_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }

            echo "<p>Is this the item you wish to delete?<br /></p>";

            for ($i=0; $i <$num_results; $i++) {
                $row = $result->fetch_assoc();
                echo "<p><strong>".($i+1).". ID: ";
                echo htmlspecialchars($row['ID']);
                echo "</strong><br />Name: ";
                echo stripslashes($row['Name']);
                echo "<br />Price: $";
                echo stripslashes($row['Price']);
                echo "<br />Quantity: ";
                echo stripslashes($row['Quantity']);
                echo "<br />Format: ";
                echo stripslashes($row['Format']);
                echo "<br />Description: ";
                echo stripslashes($row['Description']);
                echo "</p>";
            }

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

            ?>

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

```

```

        <p>Please re-enter the ID to permanently delete: <input
type="text" name="ID" value="<?php echo $id?>"
readonly></p>
        <input type="submit" name="delete" value="Delete">
        <input type="hidden" name="hid" value="<?php $id?>">
        </form>
        <br /><br />
        <form method="post" action="admin_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            </form>
        </div>
    </body></center>
</html>
-----

```

delete_music.php

```

<html>
<head>
    <title>Music Product Deletion Results</title>
</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><div class="form"><body>
<h1>Music Product Deletion Results</h1>
<?php

    $id = $_POST['ID'];
    $hid = $_POST['hid'];

    if (!get_magic_quotes_gpc()) {
        $id = addslashes($id);
    }

    @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

    if (mysqli_connect_errno()) {
        echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
        <form method="post" action="admin_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            </form>
            <?php exit();
        }

        $query = "select ID from Music where ID = $id";

```

```

$result = $db->query($query);
if (!$result) {
    echo "Query failed to execute"; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$num_results = $result->num_rows;

if ($num_results == 0) {
    echo 'No match was found.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

if ($id) {
    $query = "delete from Music where ID = $id";
}
else {
    echo 'Music Product ID was not entered. No deletion will
be made.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$result = $db->query($query);
if (!$result) {
    echo "Deletion failed to execute"; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

echo "<p>Item successfully deleted<br /></p>";

$db->close(); ?> <br /><br />
<form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
</form>
<?php

?>

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

```

modify_music_check.php

```

<html>
<head>
    <title>Music Modification Results Check</title>
</head>
<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><div class="form">
<body>
<h1>Music Modification Results Check</h1>
<?php

    $id = (int)$_POST['ID'];

    @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

    if (mysqli_connect_errno()) {
        echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
        <form method="post" action="admin_page.php">
            <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    if ($id > 0) {
        $query = "select * from Music where ID = $id";
    }
    else {
        echo "Music Product ID: [$id] is invalid. No deletion will
be made."; ?> <br /><br />
        <form method="post" action="admin_page.php">
            <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $result = $db->query($query);
    if (!$result) {
        echo "Query failed to execute"; ?> <br /><br />
        <form method="post" action="admin_page.php">
            <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
}

```

```

$num_results = $result->num_rows;

if ($num_results == 0) {
    echo 'Item not found.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    echo "<p>Is this the item you wish to modify?<br /></p>";

    for ($i=0; $i <$num_results; $i++) {
        $row = $result->fetch_assoc();
        echo "<p><strong>".($i+1).". ID: ";
        echo htmlspecialchars(stripslashes($row['ID']));
        echo "</strong><br />Name: ";
        $name = $row['Name'];
        echo stripslashes($name);
        echo "<br />Price: $";
        $price = $row['Price'];
        echo stripslashes($price);
        echo "<br />Quantity: ";
        $quantity = $row['Quantity'];
        echo stripslashes($quantity);
        echo "<br />Format: ";
        $format = $row['Format'];
        echo stripslashes($format);
        echo "<br />Description: ";
        $description = $row['Description'];
        echo stripslashes($description);
        echo "</p>";
    }

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

    ?>

    <form method="post" action="modify_music.php">
        Enter Change(s):<br />
        <p>ID: <input type="text" name="id" value="<?php echo
$id?>" readonly></p>
        <p>Name: <input type="text" name="name"></p>
        <p>Price: <input type="text" name="price"></p>

        <p>Format: <br />
        <input type="radio" id="mp3" name="format"
value="mp3">
            <label for="mp3">mp3</label><br>
        <input type="radio" id="WAV" name="format"
value="WAV">
            <label for="WAV">WAV</label><br>
        <input type="radio" id="CD" name="format" value="CD">
            <label for="CD">CD</label><br>
        <input type="radio" id="Vinyl" name="format"
value="Vinyl">
            <label for="Vinyl">Vinyl</label><br>

```

```

        <input type="radio" id="Cassette" name="format"
value="Cassette">
            <label for="Cassette">Cassette</label><br></p>

        <p>Quantity: <input type="text" name="quantity"></p>
        <p>Description: <textarea id="description"
name="description" rows="6" cols="40"></textarea></p>
        <input type="submit" name="make changes" value="make
changes">
        <input type="hidden" id="hid_name" name="hid_name"
value="<?php echo $name?>">
        <input type="hidden" id="hid_format" name="hid_format"
value="<?php echo $format?>">
        </form>

    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>

</body></div></center>
</html>
-----

```

modify_music.php

```

<html>
<head>
    <title>Music Product Modification Results</title>
</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}

</style>

<center><div class="form">
<body>
<h1>Music Product Modification Results</h1>
<?php

    $id = (int)$_POST['id'];
    $name=ucwords(strtolower($_POST['name']));
    $price = (float)$_POST['price'];
    $quantity = (int)$_POST['quantity'];
    $format = $_POST['format'];
    $description = $_POST['description'];

    $hid_name = ucwords(strtolower($_POST['hid_name']));
    $hid_format = $_POST['hid_format'];

    if (!get_magic_quotes_gpc()) {

```

```

    $id = addslashes($id);
}

@ $db = new mysqli('localhost:3306', 'sterm2',
'Breakbot123', 'sterm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$query = "select ID from Music where ID = $id";

$result = $db->query($query);
if (!$result) {
    echo "Query failed to execute"; ?> <br /><br />
    <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$num_results = $result->num_rows;

if ($num_results == 0) {
    echo 'No match was found.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

if ($name && !$format) {
    $query = "select Name, Format from Music where Name
= \"$name\" and Format = \"$hid_format\"";

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

    if (!$result) {
        echo "Query failed to execute"; ?> <br /><br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $num_results = $result->num_rows;

    if ($num_results > 0) {
        echo 'Item already exists.'; ?>
        <a href="admin_page.php">Return</a>
        <?php exit;
    }
}

```

```

    $query = "update Music set Name = \"$name\" where ID
= $id";
    $result = $db->query($query);
    if (!$result) {
        echo "Name modification failed to execute"; ?> <br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
    else {
        echo "<p>Name successfully modified<br /></p>";
    }
}

if (!$name && $format) {
    $query = "select Name, Format from Music where Name
= \"$hid_name\" and Format = \"$format\"";

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

    if (!$result) {
        echo "Query failed to execute"; ?> <br /><br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $num_results = $result->num_rows;

    if ($num_results > 0) {
        echo 'Item already exists.'; ?> <br /><br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $query = "update Music set Format = \"$format\" where
ID = $id";
    $result = $db->query($query);
    if (!$result) {
        echo "Name modification failed to execute"; ?> <br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
    else {
        echo "<p>Format successfully modified<br /></p>";
    }
}

```

```

if ($name && $format) {
    $query = "select Name, Format from Music where Name
= \"\$name\" and Format = \"\$format\"";

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

    if (!$result) {
        echo "Query failed to execute"; ?> <br /><br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $num_results = $result->num_rows;

    if ($num_results > 0) {
        echo "Item already exists."; ?> <br /><br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $query = "update Music set Name = \"\$name\", Format =
\"\$format\" where ID = $id";
    $result = $db->query($query);
    if (!$result) {
        echo "Name modification failed to execute"; ?> <br
/><br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
    else {
        echo "<p>Format and Name successfully modified<br
/></p>";
    }
}

if ($price) {
    $query = "update Music set Price = \"\$price\" where ID =
$id";
    $result = $db->query($query);
    if (!$result) {
        echo "Price modification failed to execute"; ?> <br
/><br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
    else {
        echo "<p>Price successfully modified<br /></p>";
    }
}

```

```

if ($quantity) {
    $query = "update Music set Quantity = \"\$quantity\"
where ID = $id";
    $result = $db->query($query);
    if (!$result) {
        echo "Quantity modification failed to execute"; ?> <br
/><br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
    else {
        echo "<p>Quantity successfully modified<br /></p>";
    }
}

if ($description) {
    $query = "update Music set Description = \"\$description\"
where ID = $id";
    $result = $db->query($query);
    if (!$result) {
        echo "Description modification failed to execute"; ?>
<br /><br />
        <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
    else {
        echo "<p>Description successfully modified<br
/></p>";
    }
}

$db->close();

?>

<form method="post" action="admin_page.php">
<p><input type="submit" name="return"
value="return"></p>
</form>
</body></div></center>
</html>
-----

```


insert_merch.php

```
<html>
<head>
  <title>Merch Product Entry Results</title>
</head>

<style>
* {
  box-sizing: border-box;
}

.form {
  background-color: white;
  width: 500px;
  border: 5px solid black;
  padding: 50px;
  margin: 30px;
  text-align: left;
}
</style>

<body>
<div class="form">
  <h1>Merch Product Entry Results</h1>
  <?php

    $name = ucwords(strtolower($_POST['name']));
    $description = $_POST['description'];
    $price = (float)$_POST['price'];
    $type = $_POST['type'];
    $quantity = (int)$_POST['quantity'];
    $size = $_POST['size'];

    if ($type == 'clothing' && !$size) {
      echo "Clothing insert must have a size."; ?> <br /><br />
    }

    <form method="post" action="admin_page.php">
      <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
  }

  if ($type == 'item' && $size) {
    echo "Items have no size."; ?> <br /><br />
    <form method="post" action="admin_page.php">
      <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
  }

  if (!$name || !$price || !$type || !$quantity) {
    echo "Missing name, price, type, or quantity."; ?> <br />
  }

  <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
  </form>
  <?php exit();
}
```

```
@ $db = new mysqli("localhost:3306", 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
  echo "Error: Could not connect to database. Please try
again later."; ?> <br /><br />
  <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
  </form>
  <?php exit();
}

$query = "select Name, Type, Description
from Music
where Name = \"\$name\" and Format = \"\$format\"
and Description = \"\$description\"";

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

if (!$result) {
  echo "Query failed to execute"; ?>
  <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
  </form>
  <?php exit();
}

$num_results = $result->num_rows;

if ($num_results > 0) {
  echo 'Item already exists.'; ?>
  <form method="post" action="admin_page.php">
    <p><input type="submit" name="return"
value="return"></p>
  </form>
  <?php exit();
}

$query = "insert into Merchandise (Name, Description,
Price, Type, Size, Quantity)
values (\"$name\", \"$description\", \"$price\",
\"$type\", \"$size\", \"$quantity\")";
$result = $db->query($query);

if ($result) {
  echo " $size Merch was properly inserted into
database.";
} else {
  echo "An error has occurred. The item was not
added.";
}
?> <br /><br />
<form method="post" action="admin_page.php">
  <p><input type="submit" name="return"
value="return"></p>
</form>
<?php exit();
```

```

        $db->close();

    ?>
</div>
</body>
</html>
-----

merch_search_results.php
<html>
<head>
    <title>Merch Search Results</title>
</head>
<body>
<h1>Merch Search Results</h1>
<?php

$type = $_POST['type'];
$name = ucwords(strtolower($_POST['name']));
$type_str = "";

if (!get_magic_quotes_gpc()){
    if (count($type) > 0) {
        $a = [];
        foreach ($type as $f) {
            $f = "" . addslashes($f) . "";
            $a[] = $f;
        }
        $type_str = join(',', $a);
    }
    $name = addslashes($name);
}

@ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
    <a href="admin_page.php">Return</a>
    <?php exit;
}

if ($type_str && $name) {
    $query = "select * from Merchandise where Name like
'%" . $name . "%' and Type in (". $type_str . ")";
}
elseif ($type_str) {
    $query = "select * from Merchandise where Type in
('". $type_str . ")";
}
elseif ($name) {
    $query = "select * from Merchandise where Name like
'%" . $name . "%'";
}
else {
    $query = "select * from Merchandise";
}

$result = $db->query($query);
if (!$result) {

```

```

        echo "Query failed to execute"; ?> <br /><br />
        <a href="admin_page.php">Return</a>
        <?php exit;
    }

$num_results = $result->num_rows;

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

for ($i=0; $i <$num_results; $i++) {
    $row = $result->fetch_assoc();
    echo "<p><strong>".($i+1).". ID: ";
    echo htmlspecialchars(stripslashes($row['ID']));
    echo "</strong><br />Name: ";
    echo stripslashes($row['Name']);
    echo "<br />Description: ";
    echo stripslashes($row['Description']);
    echo "<br />Price: $";
    echo stripslashes($row['Price']);
    echo "<br />Type: ";
    echo stripslashes($row['Type']);
    echo "<br />Size: ";
    echo stripslashes($row['Size']);
    echo "<br />Quantity: ";
    echo stripslashes($row['Quantity']);
    echo "</p>";
}

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

?> <br /><br />
    <a href="admin_page.php">Return</a>
</body>
</html>
-----

delete_merch_check.php
<html>
<head>
    <title>Merch Deletion Results Check</title>
</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><div class="form"><body>
<h1>Merch Deletion Results Check</h1>
<?php

```

```

$id = (int)$_POST['ID'];

@ $db = new mysqli('localhost:3306', 'sterm2',
'Breakbot123', 'sterm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

if ($id > 0) {
    $query = "select * from Merchandise where ID = $id";
}
else {
    echo "Merchandise Product ID: [$id] is invalid. No
deletion will be made."; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$result = $db->query($query);
if (!$result) {
    echo "Query failed to execute"; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$num_results = $result->num_rows;

if ($num_results == 0) {
    echo 'No match was found.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

echo "<p>Is this the item you wish to delete?<br /></p>";

for ($i=0; $i <$num_results; $i++) {
    $row = $result->fetch_assoc();
    echo "<p><strong>".($i+1).". ID: ";
    echo htmlspecialchars($row['ID']);
    echo "</strong><br />Name: ";
    echo stripslashes($row['Name']);
    echo "<br />Description: ";
    echo stripslashes($row['Description']);
    echo "<br />Price: $";

```

```

        echo stripslashes($row['Price']);
        echo "<br />Type: ";
        echo stripslashes($row['Type']);
        echo "<br />Size: ";
        echo stripslashes($row['Size']);
        echo "<br />Quantity: ";
        echo stripslashes($row['Quantity']);
        echo "</p>";
    }

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

?>

    <form method="post" action="delete_merch.php">
        <p>Please re-enter the ID to permanently delete: <input
type="text" name="ID" value="<?php echo $id?>"
readonly></p>
        <input type="submit" name="delete" value="Delete">
        <input type="hidden" name="hid" value="<?php $id?>">
    </form>
    <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>

</body></div></center>
</html>
-----

delete_merch.php
<html>
<head>
    <title>Merchandise Product Deletion Results</title>
</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><div class="form">
<body>
<h1>Merchandise Product Deletion Results</h1>
<?php

    $id = $_POST['ID'];
    $hid = $POST['hid'];

```

```

if (!get_magic_quotes_gpc()) {
    $id = addslashes($id);
}

@ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$query = "select ID from Merchandise where ID = $id";

$result = $db->query($query);
if (!$result) {
    echo "Query failed to execute"; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$num_results = $result->num_rows;

if ($num_results == 0) {
    echo 'No match was found.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

if ($id) {
    $query = "delete from Merchandise where ID = $id";
}
else {
    echo 'Music Product ID was not entered. No deletion will
be made.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$result = $db->query($query);
if (!$result) {
    echo "Deletion failed to execute";
}

echo "<p>Item successfully deleted<br /></p>";

$db->close(); ?>

```

```

<br /><br />
<form method="post" action="admin_page.php">
<p><input type="submit" name="return"
value="return"></p>
</form>
<?php exit();

```

```
?>
```

```

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

```

modify_merch_check.php

```

<html>
<head>
    <title>Merchandise Modification Results Check</title>
</head>
<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><div class="form">
<body>
<h1>Merchandise Modification Results Check</h1>
<?php

    $id = (int)$_POST['ID'];

    @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

    if (mysqli_connect_errno()) {
        echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
        <form method="post" action="admin_page.php">
            <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    if ($id > 0) {
        $query = "select * from Merchandise where ID = $id";
    }
    else {
        echo "Merchandise Product ID: [$id] is invalid. No
modification will be made."; ?> <br /><br />
        <form method="post" action="admin_page.php">

```

```

        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$result = $db->query($query);
if (!$result) {
    echo "Query failed to execute"; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

$num_results = $result->num_rows;

if ($num_results == 0) {
    echo 'Item not found.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

echo "<p>Is this the item you wish to modify?<br /></p>";

for ($i=0; $i <$num_results; $i++) {
    $row = $result->fetch_assoc();
    echo "<p><strong>".($i+1).". ID: ";
    $id = $row['ID'];
    echo htmlspecialchars(stripslashes($id));
    echo "</strong><br />Name: ";
    $name = $row['Name'];
    echo stripslashes($name);
    echo "<br />Description: ";
    $description = $row['Description'];
    echo stripslashes($description);
    echo "<br />Price: $";
    $price = $row['Price'];
    echo stripslashes($price);
    echo "<br />Type: ";
    $type = $row['Type'];
    echo stripslashes($type);
    echo "<br />Size: ";
    $size = $row['Size'];
    echo stripslashes($size);
    echo "<br />Quantity: ";
    $quantity = $row['Quantity'];
    echo stripslashes($quantity);
    echo "</p>";
}

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

?>

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

```

```

    Enter Change(s):<br />
    <p>ID: <input type="text" name="id" value="<?php echo
$id?>" readonly></p>
    <p>Name: <input type="text" name="name"></p>
    <p>Price: <input type="text" name="price"></p>
    <p>Type: </p>
        <input type="radio" id="clothing" name="type"
value="clothing">
        <label for="clothing">clothing</label><br>
        <input type="radio" id="item" name="type"
value="item">
        <label for="item">item</label><br>
    <p>Size (for clothing only): </p>
        <input type="radio" id="small" name="size"
value="small">
        <label for="small">small</label><br>
        <input type="radio" id="medium" name="size"
value="medium">
        <label for="medium">medium</label><br>
        <input type="radio" id="large" name="size"
value="large">
        <label for="large">large</label><br>
    <p>Quantity: <input type="text" name="quantity"></p>
    <p>Description: <textarea id="description"
name="description" rows="6" cols="40"></textarea></p>
    <p><input type="submit" name="make changes"
value="make changes"></p>
    </form>

    <form method="post" action="admin_page.php">
    <p><input type="submit" name="return" value="return"></p>
    </form>
</body></div></center>
</html>
-----

modify_merch.php
<html>
<head>
    <title>Merchandise Product Modification Results</title>
</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><div class="form">
<body>
<h1>Merchandise Product Modification Results</h1>
<?php

```

```

$id = (int)$_POST['id'];
$name=ucwords(strtolower($_POST['name']));
$price = (float)$_POST['price'];
$quantity = (int)$_POST['quantity'];
$type = $_POST['type'];
$description = $_POST['description'];
$size = $_POST['size'];

@ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
    <form method="post" action="admin_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }

    $query = "select ID from Merchandise where ID = $id";

    $result = $db->query($query);
    if (!$result) {
        echo "Query failed to execute"; ?> <br /><br />
        <form method="post" action="admin_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            </form>
            <?php exit();
        }

        $num_results = $result->num_rows;

        if ($num_results == 0) {
            echo 'No match was found.'; ?> <br /><br />
            <form method="post" action="admin_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }

            if ($name) {
                $query = "update Merchandise set Name = \"\$name\"
where ID = $id";
                $result = $db->query($query);
                if (!$result) {
                    echo "Name modification failed to execute"; ?> <br
/><br />
                    <form method="post" action="admin_page.php">
                        <p><input type="submit" name="return"
value="return"></p>
                        </form>
                        <?php exit();
                    }
                }
                else {
                    echo "<p>Name successfully modified<br /></p>";
                }
            }
        }
    }
}

```

```

    }

    if ($price) {
        $query = "update Merchandise set Price = \"\$price\"
where ID = $id";
        $result = $db->query($query);
        if (!$result) {
            echo "Price modification failed to execute"; ?> <br
/><br />
            <form method="post" action="admin_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }
        }
        else {
            echo "<p>Price successfully modified<br /></p>";
        }
    }

    if ($quantity) {
        $query = "update Merchandise set Quantity =
\"$quantity\" where ID = $id";
        $result = $db->query($query);
        if (!$result) {
            echo "Quantity modification failed to execute"; ?> <br
/><br />
            <form method="post" action="admin_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }
        }
        else {
            echo "<p>Quantity successfully modified<br /></p>";
        }
    }

    if ($description) {
        $query = "update Merchandise set Description =
\"$description\" where ID = $id";
        $result = $db->query($query);
        if (!$result) {
            echo "Description modification failed to execute"; ?>
<br /><br />
            <form method="post" action="admin_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }
        }
        else {
            echo "<p>Description successfully modified<br
/></p>";
        }
    }

    if ($type == 'item' && $size) {
        echo "Items have no size"; ?> <br /><br />
        <form method="post" action="admin_page.php">

```

```

        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

if ($type == 'item' && !$size) {
    $query = "update Merchandise set Type = \"\$type\", Size
= null where ID = $id";
    $result = $db->query($query);
    if (!$result) {
        echo "Type modification failed to execute"; ?> <br
/><br />
        <form method="post" action="admin_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            </form>
            <?php exit();
        }
        else {
            echo "<p>Type successfully modified (Size set to
null).<br /></p>";
        }
    }

    if ($type == 'clothing' && $size) {
        $query = "update Merchandise set Type = \"\$type\", Size
= \"\$size\" where ID = $id";
        $result = $db->query($query);
        if (!$result) {
            echo "Type modification failed to execute"; ?> <br
/><br />
            <form method="post" action="admin_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }
            else {
                echo "<p>Type successfully modified<br /></p>";
            }
        }

        if ($type == 'clothing' && !$size) {
            echo "Clothing requires a size."; ?> <br /><br />
            <form method="post" action="admin_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }
        }

        if ($size && !$type) {
            $query = "select Type from Merchandise where ID = $id";
            $result = $db->query($query);
            $row = $result -> fetch_assoc();

            if (!$result) {
                echo "Size modification failed to execute"; ?> <br
/><br />
                <form method="post" action="admin_page.php">

```

```

        <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}
    if ($row['Type'] == 'item') {
        echo "Type = Item. Cannot include size.";
    }
    else {
        $query = "update Merchandise set Size = \"\$size\"
where ID = $id";
        $result = $db->query($query);
        if (!$result) {
            echo "Size modification failed to execute"; ?> <br
/><br />
            <form method="post" action="admin_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }
            else {
                echo "<p>Size successfully modified<br /></p>";
            }
        }
    }
    $db->close()

?>

<form method="post" action="admin_page.php">
<p><input type="submit" name="return" value="return"></p>
</form>
</body></div></center>
</html>
-----

```

customer_page.php

```

<?php
    $email = $_GET['email'];
    if (empty($email)) {
        $email = $_POST['hid_email'];
    }
    ?>

<html>
<head>
<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}

```

```

.column {
    float: left;
    width: 50%;
    padding: 10px;
}

</style>
</head>

<center><div class="column">
    <h1>Buy Products</h1>
    <body>

        <div class="form">
            <h1>Search Music</h1>
            <form action="music_cart.php" method="post">
                Select Format(s):<br />
                <input type="checkbox" id="mp3" name="format[]"
value="mp3">
                    <label for="mp3">mp3</label>
                <input type="checkbox" id="WAV" name="format[]"
value="WAV">
                    <label for="WAV">WAV</label>
                <input type="checkbox" id="CD" name="format[]"
value="CD">
                    <label for="CD">CD</label>
                <input type="checkbox" id="Vinyl" name="format[]"
value="Vinyl">
                    <label for="Vinyl">Vinyl</label>
                <input type="checkbox" id="Casette"
name="format[]" value="Casette">
                    <label for="Casette">Casette</label>
                <p>Name:<br />
                <input name="Name" type="text" size="40"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                <input type="submit" name="submit"
value="Search">
            </form>
        </div>

        <div class="form">
            <h1>Search Merch</h1>
            <form action="merch_cart.php" method="post">
                Select Type(s):<br />
                <input type="checkbox" id="clothing" name="type[]"
value="clothing">
                    <label for="clothing">clothing</label>
                <input type="checkbox" id="item" name="type[]"
value="item">
                    <label for="item">item</label>
                <p>Name:<br />
                <input name="name" type="text" size="40"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                <input type="submit" name="search"
value="Search">
            </form>
        </div>

    </body>

</div></center>

<div class="column">
    <h1>My Profile</h1>
    <body>

        <div class="form">
            <h1>Update Profile</h1>
            <form method="post"
action="modify_customer_profile_check.php">
                <p>Re-enter Login info to update: </p>
                <p>Email: <input type="text" name="email"></p>
                <p>Password: <input type="password"
name="password"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                <input type="submit"
name="modify_customer_profile" value="Update">
            </form>
        </div>

        <div class="form">
            <h1>My Cart</h1>
            <form method="post" action="modify_cart.php">
                <p><input type="submit" name="view"
value="View/Edit"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
        </div>

        <div class="form">
            <h1>Past Orders</h1>
            <form method="post" action="view_receipts.php">
                <p><input type="submit" name="view"
value="View"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
        </div>

    </body>
</div></center>

</html>
-----

```


music_cart.php

```
<html>
<head>
  <title>Music Search Results</title>
</head>

<style>
* {
  box-sizing: border-box;
}

.form {
  background-color: white;
  width: 500px;
  border: 5px solid black;
  padding: 50px;
  margin: 30px;
  text-align: left;
}

.column {
  float: left;
  width: 50%;
  padding: 10px;
}

</style>

<div class="column">
  <div class="form">
    <body>
      <h1>Music Search Results</h1>
      <?php

        $format = $_POST['format'];
        $name = trim($_POST['name']);
        $format_str = "";
        $email = $_POST['hid_email'];

        if (!get_magic_quotes_gpc()){
          if (count($format) > 0) {
            $a = [];
            foreach ($format as $f) {
              $f = "" . addslashes($f) . "";
              $a[] = $f;
            }
            $format_str = join(',', $a);
          }
          $name = addslashes($name);
        }

        @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

        if (mysqli_connect_errno()) {
          echo 'Error: Could not connect to database. Please try
again later.'; ?>
          <form method="post" action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>
```

```
      <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
      </form>
      <?php exit;
    }

    if ($format_str && $name) {
      $query = "select * from Music where Name like
'%" . $name . "%' and Format in
('" . $format_str . ")";
    }
    elseif ($format_str) {
      $query = "select * from Music where Format in
('" . $format_str . ")";
    }
    elseif ($name) {
      $query = "select * from Music where Name like
'%" . $name . "%'";
    }
    else {
      $query = "select * from Music";
    }

    $result = $db->query($query);
    if (!$result) {
      echo "Query failed to execute"; ?>
      <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
        </form>
        <?php exit;
      }

      $num_results = $result->num_rows;

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

      for ($i=0; $i < $num_results; $i++) {
        $row = $result->fetch_assoc();
        echo "<p><strong>ID: ";
        echo htmlspecialchars(stripslashes($row['ID']));
        echo "</strong><br />Name: ";
        echo stripslashes($row['Name']);
        echo "<br />Price: $";
        echo stripslashes($row['Price']);
        echo "<br />Quantity: ";
        echo stripslashes($row['Quantity']);
        echo "<br />Format: ";
        echo stripslashes($row['Format']);
        echo "<br />Description: ";
        echo stripslashes($row['Description']);
        echo "</p>";
      }

      $result->free();
      $db->close();
      ?>
      <form method="post" action="customer_page.php">
```

```

        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
        </form>
    </body>
</div>
</div>

<div class="column">
<div class="form">
    <h1>Add to Cart</h1>
    <form method="post" action="insert_music_cart.php">
        <p>Enter the item ID to add an item to your cart.</p>
        <p>ID: <input type="text" name="id"></p>
        <p>Quantity: <input type="text" name="quantity"></p>
        <input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
        <input type="submit" name="insert_music_cart"
value="Add to Cart">
    </form>
</div>
</div>
</html>
-----

```

insert_music_cart.php

```

<html>
<head>
    <title>Cart Results</title>
</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}

.column {
    float: left;
    width: 50%;
    padding: 10px;
}

</style>

<div class="column">
<div class="form">
    <body>
    <h1>Cart Results</h1>
    <?php

        $id = (int)$_POST['id'];

```

```

        $quantity = (int)$_POST['quantity'];
        $email = $_POST['hid_email'];

        if (!$id) {
            echo "Missing ID"; ?>
            <form method="post" action="customer_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php exit();
        }

        if (!$quantity) {
            echo "Missing quantity"; ?>
            <form method="post" action="customer_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php exit();
        }

        @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

        if (mysqli_connect_errno()) {
            echo "Error: Could not connect to database. Please try
again later."; ?>
            <form method="post" action="customer_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php exit();
        }

        $query = "select Quantity
                    from Music
                    where ID = $id";

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

        if (!$result) {
            echo "Query failed to execute"; ?>
            <form method="post" action="customer_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php exit;
        }
        else {
            $row = $result->fetch_row();
            $result_val = (int)$row[0];
            $qty_diff = $result_val - $quantity;
        }

```

```

        if ($qty_diff < 0) {
            echo "There are only $result_val in stock, and you
requested $quantity"; ?>
            <form method="post" action="customer_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                </form>
                <?php exit();
            }
            else {
                $query = "select quantity
                        from Cart
                        where msc_id = $id";

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

                if (!$result) {
                    echo "Query failed to execute"; ?>
                    <form method="post"
action="customer_page.php">
                        <p><input type="submit" name="return"
value="return"></p>
                        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                        </form>
                        <?php exit();
                    }
                    else {
                        $row = $result->fetch_row();
                        if (empty($row)) {
                            $query = "insert into Cart (msc_id, mch_id,
quantity, email)
                                values ($id, NULL, $quantity, \"$email\")";

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

                            if (!$result) {
                                echo "Cart failed to insert item(s)"; ?>
                                <form method="post"
action="customer_page.php">
                                    <p><input type="submit" name="return"
value="return"></p>
                                    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                                    </form>
                                    <?php exit();
                                }
                                else {
                                    echo "Item successfully added to cart."; ?>
                                    <form method="post"
action="customer_page.php">
                                        <p><input type="submit" name="return"
value="return"></p>
                                        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                                        </form>
                                        <?php
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    else {
        $result_val = (int)$row[0];
        $qty_sum = $result_val + $quantity;
        $query = "update Cart
                set quantity = $qty_sum
                where msc_id = $id";

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

        if (!$result) {
            echo "Cart failed to insert item(s)"; ?>
            <form method="post"
action="customer_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                </form>
                <?php exit();
            }
            else {
                echo "Item successfully added to cart."; ?>
                <form method="post"
action="customer_page.php">
                    <p><input type="submit" name="return"
value="return"></p>
                    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                    </form>
                    <?php
                }
            }
        }
    }

    $query = "update Music
            set Quantity = $qty_diff
            where ID = $id";

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

    if (!$result) {
        echo "Music Product Quantity could not update.";
?>

        <form method="post"
action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php exit();
        }
    }

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

```

```
<html>
<head>
  <title>Merch Search Results</title>
</head>

<style>
* {
  box-sizing: border-box;
}

.form {
  background-color: white;
  width: 500px;
  border: 5px solid black;
  padding: 50px;
  margin: 30px;
  text-align: left;
}

.column {
  float: left;
  width: 50%;
  padding: 10px;
}
</style>

<div class="column">
  <div class="form">
    <body>
      <h1>Merch Search Results</h1>
      <?php

        $format = $_POST['format'];
        $name = trim($_POST['name']);
        $format_str = "";
        $email = $_POST['hid_email'];

        if (!get_magic_quotes_gpc()){
          if (count($type) > 0) {
            $a = [];
            foreach ($type as $f) {
              $f = "" . addslashes($f) . "" ;
              $a[] = $f;
            }
            $type_str = join(',', $a);
          }
          $name = addslashes($name);
        }

        @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

        if (mysqli_connect_errno()) {
          echo 'Error: Could not connect to database. Please try
again later.'; ?? <br /><br />
          <form method="post" action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>

```

```
<input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
</form>
<?php exit;
}

if ($type_str && $name) {
    $query = "select * from Merchandise where Name like
'%" . $name . "%' and Type in (" . $type_str . ")";
}
elseif ($type_str) {
    $query = "select * from Merchandise where Type in
(" . $type_str . ")";
}
elseif ($name) {
    $query = "select * from Merchandise where Name like
'%" . $name . "%'";
}
else {
    $query = "select * from Merchandise";
}

$result = $db->query($query);
if (!$result) {
    echo "Query failed to execute"; ?> <br /><br />
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
    </form>
    <?php exit;
}

$num_results = $result->num_rows;

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

for ($i=0; $i <$num_results; $i++) {
    $row = $result->fetch_assoc();
    echo "<p><strong>ID: ";
    echo htmlspecialchars(stripslashes($row['ID']));
    echo "</strong><br />Name: ";
    echo stripslashes($row['Name']);
    echo "<br />Description: ";
    echo stripslashes($row['Description']);
    echo "<br />Price: $";
    echo stripslashes($row['Price']);
    echo "<br />Type: ";
    echo stripslashes($row['Type']);
    echo "<br />Size: ";
    echo stripslashes($row['Size']);
    echo "<br />Quantity: ";
    echo stripslashes($row['Quantity']);
    echo "</p>";
}

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

```

        <form method="post" action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            <input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
        </form>
    </body>
</div>
</div>

<div class="column">
    <div class="form">
        <h1>Add to Cart</h1>
        <form method="post" action="insert_merch_cart.php">
            <p>Enter the item ID to add an item to your cart.</p>
            <p>ID: <input type="text" name="id"></p>
            <p>Quantity: <input type="text"
name="quantity"></p>
            <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            <input type="submit" name="insert_merch_cart"
value="Add to Cart">
        </form>
    </div>
</div>
</html>

```

insert_merch_cart.php

```

<html>
<head>
    <title>Cart Results</title>
</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}

.column {
    float: left;
    width: 50%;
    padding: 10px;
}
</style>

<div class="column">
    <div class="form">
        <body>
            <h1>Cart Results</h1>
            <?php

```

```

$Id = (int)$_POST['id'];
$quantity = (int)$_POST['quantity'];
$email = $_POST['hid_email'];

if (!$Id) {
    echo "Missing ID"; ?>
    <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

if (!$quantity) {
    echo "Missing quantity"; ?>
    <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

@ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
    echo "Error: Could not connect to database. Please try
again later."; ?>
    <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

$query = "select Quantity
        from Merchandise
        where ID = $Id";

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

if (!$result) {
    echo "Query failed to execute"; ?>
    <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

else {
    $row = $result->fetch_row();
    $result_val = (int)$row[0];
    $qty_diff = $result_val - $quantity;

```

```

    }

    if ($qty_diff < 0) {
        echo "There are only $result_val in stock, and you
requested $quantity"; ?>
        <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
        </form>
        <?php exit();
    }
    else {
        $query = "select quantity
        from Cart
        where mch_id = $id";

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

        if (!$result) {
            echo "Query failed to execute"; ?>
            <form method="post"
action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php exit();
        }
        else {
            $row = $result->fetch_row();
            if (empty($row)) {
                $query = "insert into Cart (mch_id, mch_id,
quantity, email)
                values (NULL, $id, $quantity, \"$email\")";

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

                if (!$result) {
                    echo "Cart failed to insert item(s)"; ?>
                    <form method="post"
action="customer_page.php">
                    <p><input type="submit" name="return"
value="return"></p>
                    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                    </form>
                    <?php exit();
                }
                else {
                    echo "Item successfully added to cart."; ?>
                    <form method="post"
action="customer_page.php">
                    <p><input type="submit" name="return"
value="return"></p>
                    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                    </form>
                    <?php

```

```

        }
    }
    else {
        $result_val = (int)$row[0];
        $qty_sum = $result_val + $quantity;
        $query = "update Cart
        set quantity = $qty_sum
        where mch_id = $id";

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

        if (!$result) {
            echo "Cart failed to insert item(s)"; ?>
            <form method="post"
action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php exit();
        }
        else {
            echo "Item successfully added to cart."; ?>
            <form method="post"
action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php
        }
    }
}

$query = "update Merchandise
set Quantity = $qty_diff
where ID = $id";

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

if (!$result) {
    echo "Merchandise Product Quantity could not
update."; ?>
    <form method="post"
action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

$db->close();

?>
</body>
</div>

```

```

</div>
</html>
-----

modify_customer_profile_check.php
<html>
<head>
  <title>Update My Profile</title>
</head>

<style>
* {
  box-sizing: border-box;
}

.form {
  background-color: white;
  width: 500px;
  border: 5px solid black;
  padding: 50px;
  margin: 30px;
  text-align: left;
}
</style>

<center><div class="form">
<body>
<h1>Update My Profile</h1>

<?php
$email = $_POST['email'];
$password = sha1($_POST['password']);

if ((!isset($email)) || (!isset($password))) {
  echo "Email and/or password are missing."; ?> <br /><br />
}

  <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
  </form>
  <?php exit();
}

  @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

  if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
    <form method="post" action="customer_page.php">
      <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
  }

  $query = "select * from User where Email = \"\$email\" and
Pwd = \"\$password\"";

  $result = $db->query($query);
  if (!$result) {

```

```

    echo "No match found."; ?> <br /><br />
    <form method="post" action="customer_page.php">
      <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
  }

  $num_results = $result->num_rows;

  for ($i=0; $i <$num_results; $i++) {
    $row = $result->fetch_assoc();
    echo "<p><strong>".($i+1).". Name: ";
    $fname = $row['Fname'];
    echo htmlspecialchars(stripslashes($fname));
    $lname = $row['Lname'];
    echo " $lname</strong>";
    echo "<br />Email: ";
    $email = $row['Email'];
    echo stripslashes($email);
    echo "<br />Address: ";
    $address = $row['Address'];
    echo stripslashes($address);
    echo "<br />Phone #: ";
    $phone = $row['Phone'];
    echo stripslashes($phone);
    echo "</p>";
  }

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

  ?>

  <form method="post"
action="modify_customer_profile.php">
    Enter Change(s):<br />
    <p>First Name: <input type="text" name="fname"></p>
    <p>Last Name: <input type="text" name="lname"></p>
    <p>Email: <input type="text" name="email" ></p>
    <p>Address: <input type="text" name="address" ></p>
    <p>Phone #: <input type="text" name="phone" ></p>
    <p>Password: <input type="password" name="password"
></p>
    <input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
    <p><input type="submit" name="make changes"
value="make changes"></p>
  </form>

  <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
  </form>
</body></div></center>
</html>
-----

```

modify_customer_profile.php

```
<html>
<head>
<title>Profile Update Results</title>
</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><div class="form">
<body>
<h1>Profile Update Results</h1>
<?php

$email = $_POST['email'];
$name = ucwords(strtolower($_POST['fname']));
$lname = ucwords(strtolower($_POST['lname']));
$phone = $_POST['phone'];
$address = ucwords(strtolower($_POST['address']));
$password = $_POST['password'];

$hid_email = $_POST['hid_email'];

@ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database. Please try
again later.'; ?> <br /><br />
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}

if ($email) {
    $query = "select Email from User where Email =
\"$email\"";
    $result = $db->query($query);

    if (!$result) {
        echo "Email modification failed to execute"; ?> <br
/><br />
        <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
}
```

```

}

$num_results = $result->num_rows;

if ($num_results != 0) {
    echo 'Email is already being used.'; ?> <br /><br />
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    </form>
    <?php exit();
}
else {
    $query = "update User set Email = \"$email\" where
Email = \"$hid_email\"";
    $result = $db->query($query);

    if (!$result) {
        echo "Email modification failed to execute"; ?> <br
/><br />
        <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
    else {
        echo "<p>Email successfully modified<br /></p>";
    }
}

if ($fname) {
    $query = "update User
set Fname = \"$fname\"
where Email = \"$hid_email\"";
    $result = $db->query($query);

    if (!$result) {
        echo "First Name modification failed to execute"; ?>
<br /><br />
        <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        </form>
        <?php exit();
    }
    else {
        echo "<p>First Name successfully modified<br
/></p>";
    }
}

if ($lname) {
    $query = "update User
set Lname = \"$lname\"
where Email = \"$hid_email\"";
    $result = $db->query($query);

    if (!$result) {
```



```

        echo "Last Name modification failed to execute"; ?>
<br /><br />
        <form method="post" action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            </form>
            <?php exit();
        }
        else {
            echo "<p>Last Name successfully modified<br
/></p>";
        }
    }

    if ($address) {
        $query = "update User
            set Address = \"\$address\"
            where Email = \"\$hid_email\"";
        $result = $db->query($query);

        if (!$result) {
            echo "Address modification failed to execute"; ?> <br
/><br />
            <form method="post" action="customer_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }
            else {
                echo "<p>Address successfully modified<br
/></p>";
            }
        }

        if ($phone) {
            $query = "update User
                set Phone = \"\$phone\"
                where Email = \"\$hid_email\"";
            $result = $db->query($query);

            if (!$result) {
                echo "Phone # modification failed to execute"; ?> <br
/><br />
                <form method="post" action="customer_page.php">
                    <p><input type="submit" name="return"
value="return"></p>
                    </form>
                    <?php exit();
                }
                else {
                    echo "<p>Phone # successfully modified<br
/></p>";
                }
            }

            if ($password) {
                $password = sha1($password);
                $query = "update User
                    set Pwd = \"\$password\"
                    where Email = \"\$hid_email\"";

```

```

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

        if (!$result) {
            echo "Password modification failed to execute"; ?> <br
/><br />
            <form method="post" action="customer_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                </form>
                <?php exit();
            }
            else {
                echo "<p>Password successfully modified<br
/></p>";
            }
        }

        $db->close();

        ?>

        <form method="post" action="customer_page.php">
            <p><input type="submit" name="return" value="return"></p>
        </form>
    </body></div></center>
</html>
-----

```

modify_cart.php

```

<html>
<head>
    <title>My Cart</title>
</head>
<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}

.column {
    float: left;
    width: 50%;
    padding: 10px;
}

</style>

<div class="column">
    <div class="form">
        <body>
            <h1>My Cart</h1>

            <?php

```

```

$email = $_POST['hid_email'];

@ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database. Please
try again later.'; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit;
}

$query = "select msc_id, mch_id, quantity
from Cart
where Email = \"\$email\"";

$result = $db->query($query);
if (!$result) {
    echo "Query failed to execute"; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit;
}

$num_results = $result->num_rows;
echo "<p>Number of items found:
". $num_results."</p>";

for ($i=0; $i <$num_results; $i++) {
    $row = $result->fetch_assoc();
    if (empty($row['mch_id'])) {
        $msc_id = (int)$row['msc_id'];
        $query_msc = "select *
from Music
where $msc_id = id";
        $result_msc = $db->query($query_msc);
        if (!$result_msc) {
            echo "Query failed to execute"; ?>
            <form method="post"
action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php exit;
        }
        else {
            $row_msc = $result_msc->fetch_assoc();
            echo "<p><strong>Music</strong>";
            echo "<br />ID: ";
            echo ($row_msc['ID']);

```

```

            echo "<br />Name: ";
            echo ($row_msc['Name']);
            echo "<br />Price: $";
            echo ($row_msc['Price']);
            echo "<br />Quantity: ";
            echo ($row['quantity']);
            echo "<br />Format: ";
            echo ($row_msc['Format']);
            echo "<br />Description: ";
            echo ($row_msc['Description']);
            echo "</p>";
        }
    }
    elseif (empty($row['mch_id'])) {
        $mch_id = (int)$row['mch_id'];
        $query_mch = "select *
from Merchandise
where $mch_id = id";
        $result_mch = $db->query($query_mch);
        if (!$result_mch) {
            echo "Query failed to execute"; ?>
            <form method="post"
action="customer_page.php">
            <p><input type="submit" name="return"
value="return"></p>
            <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            </form>
            <?php exit;
        }
        else {
            $row_mch = $result_mch->fetch_assoc();
            echo "<p><strong>Merch</strong>";
            echo "<br />ID: ";
            echo ($row_mch['ID']);
            echo "<br />Name: ";
            echo ($row_mch['Name']);
            echo "<br />Price: $";
            echo ($row_mch['Price']);
            echo "<br />Quantity: ";
            echo ($row['quantity']);
            echo "<br />Type: ";
            echo ($row_mch['Type']);
            echo "<br />Size: ";
            echo ($row_mch['Size']);
            echo "<br />Description: ";
            echo ($row_mch['Description']);
            echo "</p>";
        }
    }
}

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

?>
<form method="post" action="customer_page.php">
<p><input type="submit" name="return"
value="return"></p>
<input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">

```

```

        </form>
    </body>
</div>
</div>

<div class="column">
    <div class="form">
        <center><form method="post"
action="order_made.php">
            <h1>Order Items</h1>
            <p><input type="submit" name="make_order"
value="Make order!"></p>
            <input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
        </form></center>
    </div>

    <div class="form">
        <form method="post" action="update_cart.php">
            <h1>Remove Items from Cart</h1><br />
            Enter Change:<br /><br />
            <input type="radio" id="Music" name="table"
value="Music">
                <label for="Music">Music</label>
            <input type="radio" id="Merchandise" name="table"
value="Merchandise">
                <label
for="Merchandise">Merchandise</label><br>
            <p>ID: <input type="text" name="id"></p>
            <p>Quantity: <input type="text"
name="quantity"></p>
            <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
            <p><input type="submit" name="make changes"
value="make changes"></p>
        </form>
    </div>
</div>

</html>
-----

```

update_cart.php

```

<html>
<head>
    <title>Cart Results</title>
</head>

```

```

<style>
* {
    box-sizing: border-box;
}

```

```

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}

```

```

.column {
    float: left;
    width: 50%;
    padding: 10px;
}
</style>

```

```

<div class="column">
    <div class="form">
        <body>
            <h1>Cart Results</h1>
            <?php

```

```

                $table = $_POST['table'];
                $id = (int)$_POST['id'];
                $quantity = (int)$_POST['quantity'];
                $email = $_POST['hid_email'];

```

```

                if ($table == 'Music') {
                    $id_col = 'msc_id';
                }
                else {
                    $id_col = 'mch_id';
                }

```

```

                if (!$id || !$table || !$quantity) {
                    echo "Missing required fields"; ?>
                    <form method="post" action="customer_page.php">
                        <p><input type="submit" name="return"
value="return"></p>
                        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                    </form>
                    <?php exit();
                }

```

```

                if ($quantity == 0) {
                    echo "Cart was not changed."; ?>
                    <form method="post" action="customer_page.php">
                        <p><input type="submit" name="return"
value="return"></p>
                        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                    </form>
                    <?php exit();
                }

```

```

                if ($quantity < 0) {
                    echo "Invalid quantity."; ?>
                    <form method="post" action="customer_page.php">
                        <p><input type="submit" name="return"
value="return"></p>
                        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                    </form>
                    <?php exit();
                }

```

```

                @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

```

```

if (mysqli_connect_errno()) {
    echo "Error: Could not connect to database. Please try
again later."; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

$query = "select quantity
        from Cart
        where email = \"\$email\" and $id_col = $id";

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

if (!$result) {
    echo "1. Query failed to execute"; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

$row = $result->fetch_row();

if (empty($row)) {
    echo "It doesn't look like that item is in your cart."; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
    </form>
    <?php exit();
}
else {
    $result_val = (int)$row[0];
    $qty_diff = $result_val - $quantity;
}

if ($qty_diff < 0) {
    echo "You only have $result_val in your cart, and you
requested to remove $quantity"; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}
elseif ($qty_diff == 0) {
    $query = "delete
            from Cart
            where $id_col = $id and email = \"\$email\"";

```

```

$result = $db->query($query);
if (!$result) {
    echo "2. Query failed to execute"; ?>
    <form method="post"
action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}
}
else {
    $query = "update Cart
            set quantity = $qty_diff
            where $id_col = $id and email = \"\$email\"";
    $result = $db->query($query);
    if (!$result) {
        echo "3. Query failed to execute"; ?>
        <form method="post"
action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
        </form>
        <?php exit();
    }
}

$query = "select Quantity
        from $table
        where ID = $id";

$result = $db->query($query);
if (!$result) {
    echo "4. Query failed to execute"; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

$row = $result->fetch_row();
$result = (int)$row[0];
$qty_sum = $result + $quantity;

$query = "update $table
        set Quantity = $qty_sum
        where ID = $id";

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

if (!$result) {
    echo "5. Query failed to execute"; ?>
    <form method="post" action="customer_page.php">

```

```

        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
        </form>
        <?php exit();
    }

    echo "Item quantity successfully modified."; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php

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

```

order_made.php

```

<html>
<head>
    <title>Cart Results</title>
</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><div class="form">
    <body>
        <center><h1>RECEIPT</h1></center>

        <?php

            $email = $_POST['hid_email'];

            @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

            if (mysqli_connect_errno()) {
                echo "Error: Could not connect to database. Please
try again later."; ?>
                <form method="post" action="customer_page.php">

```

```

        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
        </form>
        <?php exit();
    }

    $query = "select *
        from Cart
        where email = \"\$email\"";

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

    if (!$result) {
        echo "1. Query failed to execute"; ?>
        <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
        </form>
        <?php exit();
    }

    $num_results = $result->num_rows;

    if ($num_results == 0) {
        echo "No items were in the cart"; ?>
        <form method="post" action="customer_page.php">
        <p><input type="submit" name="return"
value="return"></p>
        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
        </form>
        <?php exit();
    }

    echo "<p>Number of items ordered:
\".$num_results.\"</p>";

    $msc_qty = [];
    $mch_qty = [];

    for ($i=0; $i <$num_results; $i++) {
        $row = $result->fetch_assoc();
        $msc = $row['msc_id'];
        $mch = $row['mch_id'];
        $qty = (int)$row['quantity'];

        if (!empty($msc)) {
            $msc_id[] = $msc;
            $msc_qty[$msc] = $qty;
        }
        if (!empty($mch)) {
            $mch_id[] = $mch;
            $mch_qty[$mch] = $qty;
        }
    }

    $msc_id_str = join(",", $msc_id);

```

```

$mch_id_str = join(",", $mch_id);

if (!empty($msc_id)) {
    $query = "select Price, Name, Format, ID
              from Music
              where ID in ($msc_id_str)";
}

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

if (!$result_msc) {
    echo "1. Query failed to execute"; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

if (!empty($mch_id)) {
    $query = "select Price, Name, ID
              from Merchandise
              where ID in ($mch_id_str)";
}

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

if (!$result_mch) {
    echo "1. Query failed to execute"; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

$total = 0.0;
$items = [];

$num_results_msc = $result_msc->num_rows;

for ($i=0; $i <$num_results_msc; $i++) {
    $row = $result_msc->fetch_assoc();
    $msc_id = (int)$row['ID'];
    $msc_price = (float)$row['Price'];
    $quantity = $msc_qty[$msc_id];
    $total += ($msc_price * $quantity);
    $msc_name = $row['Name'];
    $msc_format = $row['Format'];
    $items[] = "$msc_name [$msc_format] ($quantity)";
}

$num_results_mch = $result_mch->num_rows;

for ($i=0; $i <$num_results_mch; $i++) {
    $row = $result_mch->fetch_assoc();
    $mch_id = (int)$row['ID'];

```

```

    $mch_price = (float)$row['Price'];
    $quantity = $mch_qty[$mch_id];
    $total += ($mch_price * $quantity);
    $mch_name = $row['Name'];
    $items[] = "$mch_name ($quantity)";
}

$items_str = join(" ", $items);
echo "total: <strong>$$total </strong><br /><br />items:
$items_str";

$query = "insert into Receipt (total, email, item)
          values ($total, \"$email\", \"$items_str\")";

$result = $db->query($query);
if (!$result) {
    echo "1. Query failed to execute"; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

$query = "delete
          from Cart
          where email = \"$email\"";

$result = $db->query($query);
if (!$result) {
    echo "1. Query failed to execute"; ?>
    <form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
    </form>
    <?php exit();
}

$db->close();

?>
</body>
</div></center>

<center><form method="post" action="customer_page.php">
    <p><input type="submit" name="return"
value="return"></p>
    <input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
    </form></center>
</html>
-----

view_receipts.php
<html>
<head>
    <title>Cart Results</title>

```

```

</head>

<style>
* {
    box-sizing: border-box;
}

.form {
    background-color: white;
    width: 500px;
    border: 5px solid black;
    padding: 50px;
    margin: 30px;
    text-align: left;
}
</style>

<center><div class="form">
    <body>
        <center><h1>PAST ORDERS</h1></center>

        <?php

            $email = $_POST['hid_email'];

            @ $db = new mysqli('localhost:3306', 'sternm2',
'Breakbot123', 'sternm2_FAIR VISIONS');

            if (mysqli_connect_errno()) {
                echo "Error: Could not connect to database. Please
try again later."; ?>
                <form method="post" action="customer_page.php">
                    <p><input type="submit" name="return"
value="return"></p>
                    <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                    </form>
                    <?php exit();
                }

                $query = "select ID, DATE(dt) as d, TIME(dt) as t, total,
item
                    from Receipt
                    where email = \"\$email\"";

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

                if (!$result) {

```

```

                    echo "1. Query failed to execute"; ?>
                    <form method="post" action="customer_page.php">
                        <p><input type="submit" name="return"
value="return"></p>
                        <input type="hidden" id="hid_email"
name="hid_email" value="<?php echo $email?>">
                        </form>
                        <?php exit();
                    }

                    $num_results = $result->num_rows;

                    echo "<p>Orders made: ".$num_results."</p>";

                    for ($i=0; $i <$num_results; $i++) {
                        $row = $result->fetch_assoc();
                        echo "<p><strong>ID: ";
                        echo ($row['ID']);
                        echo "</strong><br />Date: ";
                        echo ($row['d']);
                        echo "</strong><br />Time: ";
                        echo ($row['t']);
                        echo "<br />Total $: ";
                        echo ($row['total']);
                        echo "<br />Items: ";
                        echo ($row['item']);
                        echo "</p>";
                    }

                    ?>
                </body>
            </div></center>

            <center><form method="post" action="customer_page.php">
                <p><input type="submit" name="return"
value="return"></p>
                <input type="hidden" id="hid_email" name="hid_email"
value="<?php echo $email?>">
                </form></center>
            </html>

```

Database Dump (screenshots)

Database structure:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> Cart	★ Browse Structure Search Insert Empty Drop	4	MyISAM	latin1_swedish_ci	2.3 KiB	124 B
<input type="checkbox"/> Merchandise	★ Browse Structure Search Insert Empty Drop	25	MyISAM	latin1_swedish_ci	5.8 KiB	-
<input type="checkbox"/> Music	★ Browse Structure Search Insert Empty Drop	16	MyISAM	latin1_swedish_ci	3.9 KiB	-
<input type="checkbox"/> Receipt	★ Browse Structure Search Insert Empty Drop	1	MyISAM	latin1_swedish_ci	3.6 KiB	524 B
<input type="checkbox"/> User	★ Browse Structure Search Insert Empty Drop	2	MyISAM	latin1_swedish_ci	2.3 KiB	20 B
5 tables	Sum	48	MyISAM	latin1_swedish_ci	17.9 KiB	668 B












































































User table:

+ Options								
		Fname	Lname	Email	Address	Phone	Role	Pwd
<input type="checkbox"/>	Edit Copy Delete	Marc	Stern	sternm2@montclair.edu	1 Normal Ave, Montclair, Nj 07043	0987654321	Admin	a237483971274b63790435d57c48a554b1c9e87d
<input type="checkbox"/>	Edit Copy Delete	Mark	Styrn	user@montclair.edu	10 Normal Ave, Montclair, Nj 07043	1234567890	Customer	ccb287a0f8bb69063952992eaa20339592d6fc3d













Music table:

	ID	Name	Description	Price	Format	Quantity
<input type="checkbox"/> Edit Copy Delete	24	Lay Out In The Sun	Single	0.99	mp3	24
<input type="checkbox"/> Edit Copy Delete	23	A Goodbye	Single	0.99	WAV	25
<input type="checkbox"/> Edit Copy Delete	22	A Goodbye	Single	0.99	mp3	25
<input type="checkbox"/> Edit Copy Delete	21	A Way Out	EP	9.99	Cassette	25
<input type="checkbox"/> Edit Copy Delete	20	A Way Out	EP	9.99	Vinyl	24
<input type="checkbox"/> Edit Copy Delete	19	A Way Out	EP	9.99	CD	25
<input type="checkbox"/> Edit Copy Delete	17	A Way Out	EP	9.99	mp3	25
<input type="checkbox"/> Edit Copy Delete	18	A Way Out	EP	9.99	WAV	25
<input type="checkbox"/> Edit Copy Delete	25	Lay Out In The Sun	Single	0.99	WAV	25
<input type="checkbox"/> Edit Copy Delete	26	Feels Right	Single	0.99	mp3	24
<input type="checkbox"/> Edit Copy Delete	27	Feels Right	Single	0.99	WAV	25
<input type="checkbox"/> Edit Copy Delete	28	Modern Kids	EP (pre-order)	9.99	mp3	25
<input type="checkbox"/> Edit Copy Delete	29	Modern Kids	EP (pre-order)	9.99	WAV	25
<input type="checkbox"/> Edit Copy Delete	30	Modern Kids	EP (pre-order)	9.99	CD	25
<input type="checkbox"/> Edit Copy Delete	31	Modern Kids	EP (pre-order)	9.99	Vinyl	25
<input type="checkbox"/> Edit Copy Delete	32	Modern Kids	EP (pre-order)	9.99	Cassette	25

Merchandise table:

		ID	Name	Description	Price	Type	Size	Quantity
<input type="checkbox"/>	  	33	Baseball Hat	Logo Design 2, snap-back style (one size fits many...	19.99	clothing	small	15
<input type="checkbox"/>	  	32	Baseball Hat	Logo Design 1, snap-back style (one size fits many...	19.99	clothing	small	15
<input type="checkbox"/>	  	31	Beanie	Logo Design 2 (one size fits all)	19.99	clothing	small	13
<input type="checkbox"/>	  	30	Beanie	Logo Design 1 (one size fits all)	19.99	clothing	small	14
<input type="checkbox"/>	  	29	Hoodie	Logo Design 2	19.99	clothing	large	16
<input type="checkbox"/>	  	28	Hoodie	Logo Design 2	19.99	clothing	small	18
<input type="checkbox"/>	  	27	Hoodie	Logo Design 1	19.99	clothing	large	20
<input type="checkbox"/>	  	26	Hoodie	Logo Design 1	19.99	clothing	small	20
<input type="checkbox"/>	  	25	T-shirt	Logo Design 2	14.99	clothing	large	20
<input type="checkbox"/>	  	24	T-shirt	Logo Design 2	14.99	clothing	small	20
<input type="checkbox"/>	  	23	T-shirt	Logo Design 1	14.99	clothing	large	20
<input type="checkbox"/>	  	22	T-shirt	Logo Design 1	14.99	clothing	small	20
<input type="checkbox"/>	  	34	Mug	Logo Design 1	7.99	item		25
<input type="checkbox"/>	  	35	Mug	Logo Design 2	7.99	item		20
<input type="checkbox"/>	  	36	Frisbee	Logo Design 1	7.99	item		25
<input type="checkbox"/>	  	37	Frisbee	Logo Design 2	7.99	item		25
<input type="checkbox"/>	  	38	Poster	Logo Design 1	7.99	item		29
<input type="checkbox"/>	  	39	Poster	Logo Design 2	7.99	item		30
<input type="checkbox"/>	  	40	Sticker	Logo Design 1	0.99	item		250
<input type="checkbox"/>	  	41	Sticker	Logo Design 2	0.99	item		250
<input type="checkbox"/>	  	42	Umbrella	limited edition	24.99	item		10
<input type="checkbox"/>	  	43	Umbrella	limited edition	24.99	item		10
<input type="checkbox"/>	  	44	Umbrella	limited edition	24.99	item		10
<input type="checkbox"/>	  	45	Umbrella	limited edition	24.99	item		10
<input type="checkbox"/>	  	46	Umbrella	limited edition	24.99	item		10

Cart table:

		crt_id	msc_id	mch_id	quantity	email
<input type="checkbox"/>	  	28	NULL	38	1	user@montclair.edu
<input type="checkbox"/>	  	27	NULL	29	4	user@montclair.edu
<input type="checkbox"/>	  	26	NULL	31	2	user@montclair.edu
<input type="checkbox"/>	  	25	20	NULL	1	user@montclair.edu

Receipt table:

		ID	dt	total	email	item
<input type="checkbox"/>	  	6	2020-12-06 21:28:42	101.9	user@montclair.edu	Lay Out In The Sun [mp3] (1), Feels Right [mp3] (1...

Citations

[1] Luke Welling, Laura Thomson. *PHP and MySQL Web Development*, 4th ed. (e-book): Addison-Wesley, 2009, ch. 1, 3, 11.