

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-3-2024/grade/yc73>

IT202-008-S2024 - [IT202] API Project Milestone 3 2024

Submissions:

Submission Selection

1 Submission [active] 4/22/2024 1:01:24 PM

Instructions

[^ COLLAPSE ^](#)

1. Implement the Milestone 3 features from the project's proposal document: <https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88E>
2. Make sure you add your ucid/date as code comments where code changes are done
3. All code changes should reach the Milestone3 branch
4. Create a pull request from Milestone3 to dev and keep it open until you get the output PDF from this assignment.
5. Gather the evidence of feature completion based on the below tasks.
6. Once finished, get the output PDF and copy/move it to your repository folder on your local machine.
7. Run the necessary git add, commit, and push steps to move it to GitHub
8. Complete the pull request that was opened earlier
9. Create and merge a pull request from dev to prod
10. Upload the same output PDF to Canvas

Branch name: Milestone3

Tasks: 26 Points: 10.00

 API (1 pt.)

[^ COLLAPSE ^](#)

 Task #1 - Points: 1

Text: Data Related to Users

Checklist

*The checkboxes are for your own tracking

#

Points

Details

#	Points	Details
<input checked="" type="checkbox"/> #1	1	What's the concept/association?
<input checked="" type="checkbox"/> #2	1	What sort of relationship is it (one to many, many to one, many to many, etc)
<input checked="" type="checkbox"/> #3	1	Note any other considerations

Response:

The concept/association for my project is assigning products to users within my shop. Users can "purchase" products which creates a relationship between them and that selected product. They can also "return" those products which removes that (existing) relationship. For now, the relationship is one to many as one user be associated with multiple products, but products can only be associated with one user - once a product is purchased it can't be purchased by anyone else. This is because I wanted to show how it looks when a product sells out and I plan to work on using stock and quantity after the semester (as there's not enough time right now since this Milestone 3 is due soon). So many to many is possibly but as of right now it is one to many.

Task #2 - Points: 1

Text: Updating Entities

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	When an update occurs either manually or from the API how does it affect associated data?
<input checked="" type="checkbox"/> #2	1	Do users see the old data, new data, does data need to be reassociated, etc?

Response:

When an update occurs, either manually or from the API, it will just edit the corresponding product information. Since it just edits/modifies the existing products, users and the associations won't be negatively affected. Users will just see the new version of the data.

Handle Data Association (1 pt.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Option 1: Related pages will have a button to do association (like favorites or similar), Option 2: a separate page will be used to associate entities to a user by some other user (like assignment of entities)
<input checked="" type="checkbox"/> #2	1	Include ucid/date comments for each code screenshot
<input checked="" type="checkbox"/> #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

```
order_history.php purchase_product.php M product_card.php X
partials > product_card.php > ...
10 <div class="list-group-item"><span>Price:</span> <span>$</span><span>product["price"]</span></div>
11 <div class="list-group-item"><span>Type:</span> <span>product["type"]</span></div>
12 <div class="list-group-item"><span>Available:</span> <span>product["available"]</span></div>
13 </div>
14 <div class="card-body">
15 <?php if (has_role("Admin")) : ?>
16 <?php if (!isset($product["user_id"])) || ($product["user_id"] === "NULL") : ?>
17 <a href="#"><?php echo get_url('admin/view_product.php?id=' . $product["id"]); ?> <span class="btn btn-secondary">View</a>
18 <a href="#"><?php echo get_url('api/return_product.php?id=' . $product["id"]); ?> <span class="btn btn-secondary">Return</a>
19 <a href="#"><?php echo get_url('api/purchase_product.php?id=' . $product["id"]); ?> <span class="btn btn-primary">Purchase</a>
20 </?php else : ?>
21 <!-- yc73 4/25/23 -->
22 <p class="bg-warning text-dark text-center">Product not available</p>
23 <a href="#"><?php echo get_url('admin/view_product.php?id=' . $product["id"]); ?> <span class="btn btn-secondary" style="margin-right: 10px;">View</a>
24 <a href="#"><?php echo get_url('api/return_product.php?id=' . $product["id"]); ?> <span class="btn btn-secondary" style="margin-left: 10px;">Return</a>
25 <a href="#"><?php echo get_url('profile.php?id=' . $product["user_id"]); ?> <span class="btn btn-primary">Profile</a>
26 </?php foreach ($basic_products as $user_product) : ?>
27 <?php if ($user_product['product_id'] == $product['id']) : ?>
28 <p class="card-text">Total Customers: <?php echo $user_product['total_users']; ?></p>
29 </?php endforeach; ?>
30 </div>
31 </?php endif; ?>
32 </div>
33 </?php endif; ?>
34 </div>
35 </?php endif; ?>
36 </div>
37 </?php endif; ?>
38 </div>
39 </?php endif; ?>
```

Showing product card with a purchase button (to do association)

Checklist Items (1)

#1 Option 1: Related pages will have a button to do association (like favorites or similar), Option 2: a separate page will be used to associate entities to a user by some other user (like assignment of entities)

```
order_history.php purchase_product.php M product_card.php X
purchase_product.php > Project > api > purchase_product.php > ...
1 <?php
2 // This is an internal API endpoint to receive data and do something with it
3 // Note: No view.php here because this is a temporary shop, it's not a user page
4 // Note: no view.php here because this is a temporary shop, it's not a user page
5 require __DIR__ . "/../../lib/functions.php";
6 //yc73 4/26/24
7 session_start();
8 if (isset($_GET['product_id'])) {
9     // Implement purchase logic (for now it's all free)
10     $db = getDB();
11     $query = "INSERT INTO UserProducts (user_id, product_id, product_name, price) VALUES (:user_id, :product_id, :product_name, :price)";
12     //error_log("Data: " . var_dump($query));
13     try {
14         $stmt = $db->prepare($query);
15         $stmt->execute([':user_id' => get_user_id(), ':product_id' => $_GET['product_id']]);
16         flash("Congrat your purchase was successful", "success");
17     } catch (PDOException $e) {
18         if ($e->getCode() == 1062) {
19             flash("This product isn't available", "danger");
20         } else {
21             flash("Unhandled error occurred", "danger");
22         }
23         error_log("Error purchasing product(s): " . var_export($e, true));
24     } //-- 21-24 catch (PDOException $e);
25 } //-- 25-26 if (isset($_GET['product_id'])) && is_logged_in();
26
27 //For now I'll redirect, but if I later use AJAX I need to send a reply instead
28 redirect("order_history.php");
```

Showing the purchase logic

Checklist Items (1)

#1 Option 1: Related pages will have a button to do association (like favorites or similar), Option 2: a separate page will be used to associate entities to a user by some other user (like assignment of entities)

The screenshot shows a web browser window with three tabs open. The active tab is titled 'yc73-it202-008-dev - GitHub' and displays a 'Project/home.php' page. The page has a header with 'Aeki' and navigation links for 'Home', 'Profile', 'Admin', 'Product Management', 'Order History', and 'Logout'. Below the header is a search/filter section titled 'Products' with fields for 'Product Name', 'Min Price', 'Max Price', 'Type', 'Category', 'Sort', 'Order', and 'Limit'. A 'Filter' button and a 'Clear' button are also present. The main content area shows a grid of product cards. Each card includes a thumbnail image, the product name, a brief description, and two buttons: 'View' and 'Purchase'. The first product card for 'KALLAX' is expanded, showing its details: Name: KALLAX, Price: 79.99, Type: Shelf unit, and a note: 'Product not available'. It also includes 'View' and 'Return' buttons and a link to 'yan's Profile'.

Showing purchase button on website

Checklist Items (1)

#1 Option 1: Related pages will have a button to do association (like favorites or similar), Option 2: a separate page will be used to associate entities to a user by some other user (like assignment of entities)

Task #2 - Points: 1

Text: Screenshot of the association table(s)

Checklist

*The checkboxes are for your own tracking

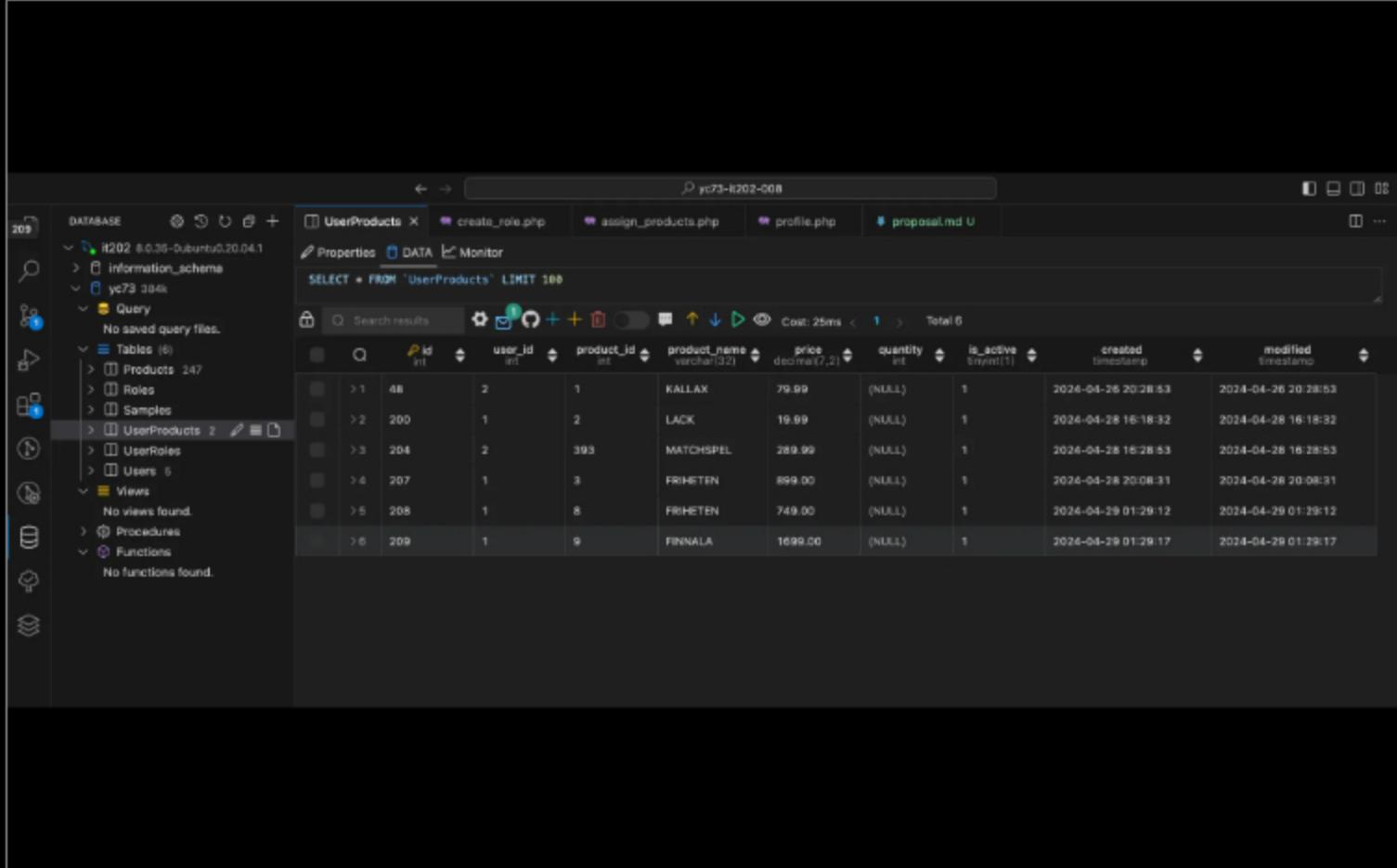
#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the table(s) you made to handle the associations
<input checked="" type="checkbox"/> #2	1	Should have some example data
<input checked="" type="checkbox"/> #3	1	Must show the associations between the tables

#3	1	Make sure the heroku dev url is visible in the address bar
#4	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



	id int	user_id int	product_id int	product_name varchar(32)	price decimal(7,2)	quantity int	is_active tinyint(1)	created_timestamp timestamp	modified_timestamp timestamp
> 1	48	2	1	KALLAX	79.99	(NULL)	1	2024-04-26 20:28:53	2024-04-26 20:28:53
> 2	200	1	2	LACK	19.99	(NULL)	1	2024-04-28 16:18:32	2024-04-28 16:18:32
> 3	204	2	393	MATCHSPEL	289.99	(NULL)	1	2024-04-28 16:28:53	2024-04-28 16:28:53
> 4	207	1	3	FRIHETEN	899.00	(NULL)	1	2024-04-28 20:08:31	2024-04-28 20:08:31
> 5	208	1	8	FRIHETEN	749.00	(NULL)	1	2024-04-29 01:29:12	2024-04-29 01:29:12
> 6	209	1	9	FINNALA	1699.00	(NULL)	1	2024-04-29 01:29:17	2024-04-29 01:29:17

Showing my `UserProducts` table which handles associations (with example data)

Checklist Items (1)

#1 Show the table(s) you made to handle the associations

Task #3 - Points: 1

Text: Explain solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Mention which option your application handles regarding association
<input checked="" type="checkbox"/> #2	1	Describe each column/association table

Response:

My application has a "order history" page to display a user's existing associations. It also uses a "purchase" button to handle the associations. For regular (non-admin) users, each product card on the home page has a purchase button. When clicked, that item will be associated with the (current) logged-in user. There is also a "return" button on each

which checked, that item will be associated with the (current) logged in user. There is also a "return" button on each product card - which handles removing an existing relationship (unassociate). Admins can also associate themselves with an item using the purchase button or use the separate association management form to assign themselves or other users with specific products.

These associations are stored in the "UserProducts" table - which has the columns id, user_id, product_id, product_name, price, quantity, is_active, created, and modified.

- id = holds a number to uniquely identify each (association) record in the table
- user_id = holds the unique id for each user (foreign key that references "id" from the Users table)
- product_id = holds the unique id for each product (foreign key that references "id" from the Products table)
- product_name = stores the name of each product
- price = stores the price of each product
- quantity = meant to hold the number of items a user wants to purchase (p.s. I was not able to use quantity at the time this is being submitted, but I do plan on working on this project later on and using quantity to work with stock)
- is_active = holds a 1 when a record is inserted into this table, I use it to check if there is an existing association
- created = holds the timestamp of when a record was created/inserted into
- modified = holds the timestamp of when a record was updated

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Project/order_history.php

URL #2

<https://github.com/yaneliii/yc73-it202-008/pull/69>

Current User's Association Page (2 pts.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the summary of the results with relevant information per entity
<input type="checkbox"/> #2	1	Show the single view buttons/links, delete button/links, and delete all button/link

<input checked="" type="checkbox"/> #3	1	Show variations of the number of shown items count and show the count of total number of associated items to the user
<input checked="" type="checkbox"/> #4	1	Show variations of the filter/sort including no results (proper message should be visible)
<input checked="" type="checkbox"/> #5	1	Make sure the heroku dev url is visible in the address bar
<input checked="" type="checkbox"/> #6	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

Congrats your purchase was successful!

My Order History

Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
Name	Min Price	Max Price	Type	Category	Created	-	10

Filter Clear Return All Products

Results: 4/4

Owned By: yan

LACK

Name: LACK
Price: 19.99
Type: TV unit

[View](#) [Return](#)

[yan's Profile](#)
Total Customers: 1

Owned By: yan

FRIHETEN

Name: FRIHETEN
Price: 899.00
Type: Sleeper sectional,3 seat w/storage

[View](#) [Return](#)

[yan's Profile](#)
Total Customers: 1

Owned By: yan

FRIHETEN

Name: FRIHETEN
Price: 749.00
Type: Sleeper sofa

[View](#) [Return](#)

[yan's Profile](#)
Total Customers: 1

Owned By: yan

FINNALA

Name: FINNALA
Price: 1899.00
Type: Sectional, 4-seat

[View](#) [Return](#)

[yan's Profile](#)
Total Customers: 1

Showing user association page - with summary, view btn, delete btn, delete ALL btn (p.s. my delete btn is called "return")

Checklist Items (2)

#1 Show the summary of the results with relevant information per entity

#2 Show the single view buttons/links, delete button/links, and delete all button/link

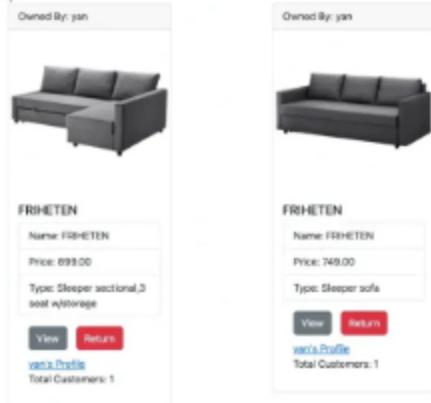
Congrats your purchase was successful!

My Order History

Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
Name	Min Price	Max Price	Type	Category	Created	-	10

Filter Clear Return All Products

Results: 2/4



Showing variation of count - number of items shown and total number of items associated items

Checklist Items (1)

#3 Show variations of the number of shown items count and show the count of total number of associated items to the user

My Order History

Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
frimeten	500	800	sofa	sofa	Created	desc	10

Filter Clear Reset All Products

Results: 1

Owned By: yan

FRIMETEN

Name: FRIMETEN

Price: 749.00

Type: Sleeper sofa

[View](#) [Return](#)

[yan's Profile](#)

Total Customers: 1

Showing variation of the filter/sort

Checklist Items (1)

#4 Show variations of the filter/sort including no results (proper message should be visible)

My Order History

Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
bob	Min Price	Max Price	Type	Category	Created	-	10

Filter **Clear** **Return All Products**

Results: 0/4
No results to show

Showing variation of the filter/sort - NO result with proper message

Checklist Items (1)

#4 Show variations of the filter/sort including no results (proper message should be visible)

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the code related to fetching the user's associations (including the query)
<input checked="" type="checkbox"/> #2	1	Show the code related to the display of the results
<input checked="" type="checkbox"/> #3	1	Each record should have a button/link for single view
<input checked="" type="checkbox"/> #4	1	Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity
<input checked="" type="checkbox"/> #5	1	Show the logic for deleting all associations for the user (this may be admin-only but should be present for the specific role)
<input checked="" type="checkbox"/> #6	1	Show the logic related to the count of all associated items to the user (even the ones not shown in the filtered results)
<input checked="" type="checkbox"/> #7	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
<input checked="" type="checkbox"/> #8	1	Show the logic related to filter/sort (limit should be constrained to 1-100 otherwise default to 10)

- #9 1 Include ucid/date comments for each code screenshot
- #10 1 Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

Showing code that fetches the user's association - with the query

Checklist Items (1)

#1 Show the code related to fetching the user's associations (including the query)

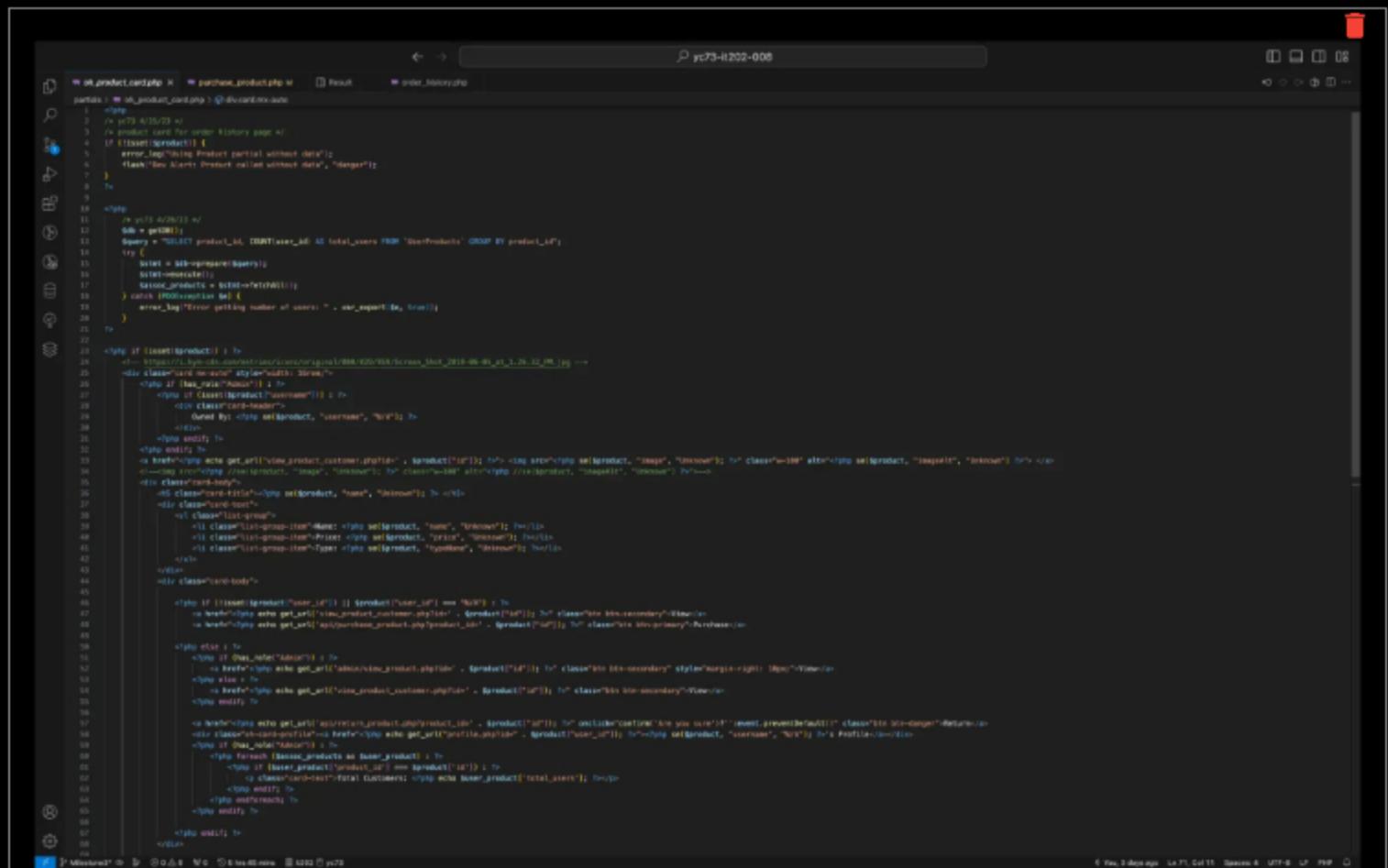
```
189 <?php if (isset($product)) : ?>
190 |   <div class="col">
191 |     |   // results to show
192 |   </div>
193 </div>
194
195 <?php require_once __DIR__ . "/../../partials/flash.php"; ?>
196
```

0 You, 4 days ago Lin 185, Col 25 Spaces: 4 UTF-8 LF PHP

Showing code that grabs the results to display (passing results to the product card)

Checklist Items (1)

#2 Show the code related to the display of the results



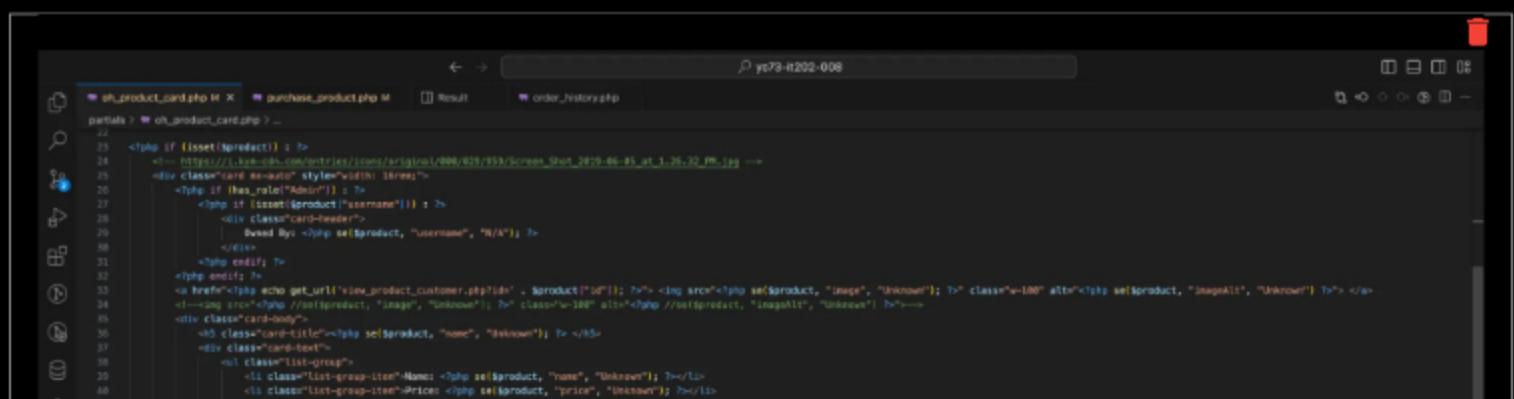
```
partials > oh_product_card.php > purchase_product.php > Result > order_history.php
1 <?php
2 |   <?php if ($product) : ?>
3 |     // product card for order history page
4 |   </?php>
5 |   <?php if (isset($product)) : ?>
6 |     error_log("Using Product partial without data");
7 |     die("Error: Product called without data", 1);
8 |   </?php>
9 |
10 <?php
11 |   // $product = $product->get();
12 |   $id = $product->id;
13 |   $query = "SELECT product_id, COUNT(user_id) AS total_users FROM `UserProducts` GROUP BY product_id";
14 |   try {
15 |     $stmt = $db->prepare($query);
16 |     $stmt->execute();
17 |     $total_products = $stmt->fetch();
18 |   } catch (PDOException $e) {
19 |     error_log("Error getting number of users: " . $e->getMessage());
20 |   }
21 |
22 <?php
23 |   if (isset($product)) : ?>
24 |     <a href="https://127.0.0.1:8080/concrete/sites/iconicorigin1/888/826/553/Screen_Shot_2019-06-01_at_1-26-32_PM.jpg" style="width: 300px;">
25 |       <img alt="Thumbnail image of the product" />
26 |     </a>
27 |     <p>If this user's name:</p>
28 |     <div class="card-header">
29 |       Based By: <?php self::product('username'); ?>
30 |     </div>
31 |     <div class="card-body">
32 |       <?php if ($this_user['id'] == $product['user_id']) : ?>
33 |         <a href="https://127.0.0.1:8080/concrete/sites/iconicorigin1/888/826/553/Screen_Shot_2019-06-01_at_1-26-32_PM.jpg" style="width: 100%; height: 100%; display: block; margin-bottom: 10px;">
34 |           <img alt="Thumbnail image of the product" />
35 |         <?php else : ?>
36 |           <img alt="Thumbnail image of the product" />
37 |         </?php endif; ?>
38 |       <?php endif; ?>
39 |       <?php if ($this_user['id'] == $product['user_id']) : ?>
40 |         <div class="list-group-item">
41 |           <?php if ($this_user['name'] == 'Unknown') : ?>
42 |             <?php echo get_url('view_product_customer_photograph', 'product[' . $product['id'] . ']'); ?>
43 |           <?php else : ?>
44 |             <?php echo get_url('view_product_customer_photograph', 'product[' . $product['id'] . ']'); ?>
45 |           </?php endif; ?>
46 |         </div>
47 |       <?php endif; ?>
48 |       <?php if ($this_user['id'] == $product['user_id']) : ?>
49 |         <div class="list-group-item">
50 |           <?php if ($this_user['name'] == 'Unknown') : ?>
51 |             <?php echo get_url('view_product_customer_photograph', 'product[' . $product['id'] . ']'); ?>
52 |           <?php else : ?>
53 |             <?php echo get_url('view_product_customer_photograph', 'product[' . $product['id'] . ']'); ?>
54 |           </?php endif; ?>
55 |         </div>
56 |       <?php endif; ?>
57 |       <?php if ($this_user['id'] == $product['user_id']) : ?>
58 |         <div class="list-group-item">
59 |           <?php if ($this_user['name'] == 'Unknown') : ?>
60 |             <?php echo get_url('view_product_customer_photograph', 'product[' . $product['id'] . ']'); ?>
61 |           <?php else : ?>
62 |             <?php echo get_url('view_product_customer_photograph', 'product[' . $product['id'] . ']'); ?>
63 |           </?php endif; ?>
64 |         </div>
65 |       <?php endif; ?>
66 |     </div>
67 |   </div>
68 | </?php>
```

0 You, 3 days ago Lin 71, Col 11 Spaces: 4 UTF-8 LF PHP

Showing product card code related to the display of the results

Checklist Items (1)

#2 Show the code related to the display of the results



```
partials > oh_product_card.php > purchase_product.php > Result > order_history.php
1 <?php if (isset($product)) : ?>
2 |   <a href="https://127.0.0.1:8080/concrete/sites/iconicorigin1/888/826/553/Screen_Shot_2019-06-01_at_1-26-32_PM.jpg" style="width: 300px;">
3 |     <img alt="Thumbnail image of the product" />
4 |   </a>
5 |   <?php if (isset($product['username'])) : ?>
6 |     <div class="card-header">
7 |       Based By: <?php self::product('username'); ?>
8 |     </div>
9 |   <?php endif; ?>
10 <?php
11 |   <?php echo get_url('view_product_customer_photograph', 'product[' . $product['id'] . ']'); ?>
12 |   <img alt="Thumbnail image of the product" />
13 |   <?php if ($product['image'] == 'Unknown') : ?>
14 |     <img alt="Thumbnail image of the product" />
15 |   <?php else : ?>
16 |     <img alt="Thumbnail image of the product" />
17 |   </?php endif; ?>
18 | </div>
19 | <div class="card-title"><?php self::product('name', 'Unknown'); ?></div>
20 | <div class="card-body">
21 |   <?php class="list-group">
22 |     <?php class="list-group-item"><?php self::product('name', 'Unknown'); ?></li>
23 |     <?php class="list-group-item"><?php self::product('price', 'Unknown'); ?></li>
24 |   </?php>
```

0 You, 3 days ago Lin 71, Col 11 Spaces: 4 UTF-8 LF PHP

Showing the view button and delete button (on the product card)

Checklist Items (2)

#3 Each record should have a button/link for single view

#4 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity

Showing the logic to delete all associations

Checklist Items (1)

#5 Show the logic for deleting all associations for the user (this may be admin-only but should be present for the specific role)

Showing the logic to delete a single association

Checklist Items (2)

#4 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity

#5 Show the logic for deleting all associations for the user (this may be admin-only but should be present for the specific role)

Showing the logic related to the count of items on the page / and the count of all items associated with the user

Checklist Items (2)

#6 Show the logic related to the count of all associated items to the user (even the ones not shown in the filtered results)

#7 Show the logic related to the count of the items on the page (this value should change based on the filter applied)

Showing the logic related to filter/sort (and limit is constrained to 1-100 & defaults to 10 otherwise)

Checklist Items (1)

#8 Show the logic related to filter/sort (limit should be constrained to 1-100 otherwise default to 10)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
■ #1	1	Mention how you determine the result list (include the association logic and filters)
■ #2	1	Mention the logic behind the two counts (total items and visible result count)
■ #3	1	Mention the logic for single delete and delete all associations for the specific user

Response.

First, when a user clicks the purchase button on a card it grabs that product's id and starts to build a query to INSERT into the "UserProducts" table. It will insert the (current logged-in) user's id, the specific product's id, the product name, and the product price.

- (also since the product's name and price weren't passed I grabbed it with a SELECT subquery based on the product id)

The query is prepared and executed. If it is successfully inserted it will flash a success message to the user.

Now that table "UserProducts" has the association, the "Order History" page builds a query to get the necessary user and product information to display it. The query starts by SELECTing username, user id, product id, product name, product price, and other product info (api_id, measurement, typeName, image, contextualImageUrl, imageAlt, url, categoryPath, stock, and is_api). It gets the product info from the "Products" table and JOINs it with the "UserProducts" based on the (associating) product's id. It LEFT JOINs the "Users" table based on user's id to get the (associating) user's info. Then, it filters based on WHERE it matches the (associating) user's id.

If someone uses filters (name, min/max price, product type, category, sort, order, limit), it:

- grabs what they entered and assigns it to a variable
- checks if the variable is not empty & if it's a number (min/max price) it checks if it's greater than -1
- if it passes: the entered filter gets added to the query with AND as well as LIKE - to get partial matches (for product name, type, and category), the value is also assigned to the corresponding parameter with "%" - which is a wildcard that helps look for partial matches
- for sort and order, it checks if the user chose one of the options in the selection (array), and if not it defaults sort to created & order to descending and gets added to the query with ORDER BY
- for limit, it checks to see if the entered number falls within a range: if it's less than 1 or greater than 100 it fails and defaults to 10, if valid the entered limit number gets added to the query with LIMIT

The query is prepared and executed. If it successfully selects it will fetch and store the \$results in a variable. Each product in the result variable will be passed to the product card to render and display the results.

To count the visible items:

- it uses the count() function to count the number of fetched records (from the assoc table) in the \$results array

To count the total items:

- first, a query specifies that it's grabbing data from the 'Products' table and JOINs it with the "UserProducts" table based on the (associating) product's id, and filters based on WHERE it matches the (associating) user's id
- second, it assigns the value of the (current) user's id to the corresponding parameter
- it passes that query and parameter to the get_total_count function
- the get_total_count function: first sanitizes the table references, next it fixes the format of parameters that don't start with a colon(:), then it uses a SELECT query to count the number of records from the table, and executes, fetches, & returns the total count result as an integer

To render the count it calls the function render_result_counts() and passes visible items count & total items count. It first checks if the visible count & total count are set. If they are NOT it flashes a user-friendly error message and sets the value to -1. If they are set, then they are displayed.

If someone wants to delete association they can do a single delete or delete all:

- Single delete: When a user clicks the "return" button (p.s. my delete button is called return) on a card it passes the product id to the UserProducts controller's destroy method. This method finds the record in the UserProducts table with the matching product id and user id, and then deletes it.

the product's id to "return_product.php". "return_product.php" first checks if the product id is set, if it is then it specifies a query. This query will SELECT the corresponding (associated) user's id FROM the "UserProducts" table based on WHERE it matches the (associated) product's id. The query is executed, fetches, and stores the result in a variable (\$result). It proceeds to grab the associated user's id from the \$result. Another query is made to DELETE the record from the "UserProducts" table WHERE the (assoc) user id AND (assoc) product id match. It gets executed, and if it successfully deletes the association it will flash a success message to the user.

- Delete all: When a user clicks the "return all products" (p.s. my delete is called return) on the page, it uses <a> & href to add "?remove" to the URL. Also upon clicking that button, it activates an "onclick" event which displays a box asking the user to confirm if they want to delete their associations (aka return their products). By clicking ok it continues. This code notices the "?remove" and first checks if it's set. If it is, then it makes a query to DELETE all the records from the "UserProducts" table WHERE it matches the current (assoc) user's id.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input checked="" type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Project/order_history.php

URL #2

<https://github.com/yaneliii/yc73-it202-008/pull/69>

All Users Association Page (likely an admin page) (2 pts.)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the summary of the results with relevant information per entity
<input checked="" type="checkbox"/> #2	1	Show the single view buttons/links, delete button/links
<input checked="" type="checkbox"/> #3	1	Show the username related to the specific entity and that it's clickable
<input checked="" type="checkbox"/> #4	1	Show variations of the number of shown items count and show the count of total number of associated items
<input checked="" type="checkbox"/> #5	1	Show variations of the filter/sort including no results (proper message should be visible)

<input checked="" type="checkbox"/> #6	1	Make sure the heroku dev url is visible in the address bar
<input checked="" type="checkbox"/> #7	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Showing all user association page (admin) - with summary, view btn, delete btn, delete ALL btn, username/profile link
(p.s. my delete btn is called "return")

Checklist Items (1)

#1 Show the summary of the results with relevant information per entity



Showing the link of username - while hovering over it (its clickable)

Checklist Items (1)

#3 Show the username related to the specific entity and that it's clickable

The link of the username CLICKED - showing profile related to the specific user

Checklist Items (1)

#3 Show the username related to the specific entity and that it's clickable

[Filter](#)[Clear](#)[Return All Products](#)

Results:

2/6

Owned By: yan2



MATCHSPEL

Name: MATCHSPEL

Price: 289.99

Owned By: yan2



KALLAX

Name: KALLAX

Price: 79.99

Showing variation of count - number of items shown and total number of items associated items

Checklist Items (1)

#4 Show variations of the number of shown items count and show the count of total number of associated items

Associated Products

Username	Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
Username	la	Min Price	Max Price	Type	Category	Created	-	10

Filter Clear Return All Products

Results: 3/6

Owned By: yan



LACK

Name: LACK

Price: 19.99

Type: TV unit

Product not available

[View](#) [Return](#)

[yan's Profile](#)

Total Customers: 1

Owned By: yan



FINNALA

Name: FINNALA

Price: 1699.00

Type: Sectional, 4-seat

Product not available

[View](#) [Return](#)

[yan's Profile](#)

Total Customers: 1

Owned By: yan2



KALLAX

Name: KALLAX

Price: 79.99

Type: Shelf unit

Product not available

[View](#) [Return](#)

[yan2's Profile](#)

Total Customers: 1

Showing variation of count - number of items shown and total number of items associated items

Checklist Items (1)

#4 Show variations of the number of shown items count and show the count of total number of associated items

Associated Products

Username	Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
Username	la	15	200	unit	Category	Created	-	10
<input type="button" value="Filter"/> <input type="button" value="Clear"/> <input type="button" value="Return All Products"/>								
Results: 2/6								
Owned By: yan		Owned By: yan2						
 <p>LACK</p> <p>Name: LACK</p> <p>Price: 19.99</p> <p>Type: TV unit</p> <p>Product not available</p> <p><input type="button" value="View"/> <input type="button" value="Return"/></p> <p>yan's Profile</p> <p>Total Customers: 1</p>		 <p>KALLAX</p> <p>Name: KALLAX</p> <p>Price: 79.99</p> <p>Type: Shelf unit</p> <p>Product not available</p> <p><input type="button" value="View"/> <input type="button" value="Return"/></p> <p>yan2's Profile</p> <p>Total Customers: 1</p>						

Showing variation of the filter/sort

Checklist Items (1)

#5 Show variations of the filter/sort including no results (proper message should be visible)

Username	Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
yan	Name	250	800	Type	Category	Created	-	10
<input type="button" value="Filter"/> <input type="button" value="Clear"/> <input type="button" value="Return All Products"/>								
Results: 2/6								
Owned By: yan2		Owned By: yan						
 <p>MATCHSPEL</p> <p>Name: MATCHSPEL</p> <p>Price: 289.99</p> <p>Type: Gaming chair, Bomstад black</p> <p>Product not available</p> <p><input type="button" value="View"/> <input type="button" value="Return"/></p> <p>yan2's Profile</p> <p>Total Customers: 1</p>		 <p>FRIHETEN</p> <p>Name: FRIHETEN</p> <p>Price: 749.00</p> <p>Type: Sleeper sofa</p> <p>Product not available</p> <p><input type="button" value="View"/> <input type="button" value="Return"/></p> <p>yan's Profile</p> <p>Total Customers: 1</p>						

Showing variation of the filter/sort

Checklist Items (1)

Checklist Items (1)

#5 Show variations of the filter/sort including no results (proper message should be visible)

A screenshot of a web application titled "Associated Products". At the top, there is a navigation bar with links for "Home", "Profile", "Admin", "Product Management", "Order History", and "Logout". Below the navigation bar, there is a search form with fields for "Username" (containing "bob"), "Product Name", "Min Price", "Max Price", "Type", and "Category". There are also dropdowns for "Sort" (set to "Created") and "Order" (set to "-"). A limit of "10" is selected. Below the search form are three buttons: "Filter" (blue), "Clear" (gray), and "Return All Products" (red). The main content area displays the message "Results: 0/6" and "No results to show".

Showing variation of the filter/sort - NO result with proper message

Checklist Items (1)

#5 Show variations of the filter/sort including no results (proper message should be visible)

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the code related to fetching all associations (including the query)
<input checked="" type="checkbox"/> #2	1	Show the code related to the display of the results
<input checked="" type="checkbox"/> #3	1	Each record should have a button/link for single view
<input checked="" type="checkbox"/> #4	1	Each record should have a username field that is clickable to go to the user's profile page
<input checked="" type="checkbox"/> #5	1	Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity
<input checked="" type="checkbox"/> #6	1	Show the logic related to the count of all associated items (even the ones not shown)

#6	1	in the filtered results)
#7	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
#8	1	Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)
#9	1	Include ucid/date comments for each code screenshot
#10	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

Showing code that fetches all associations - with the query

Checklist Items (1)

#1 Show the code related to fetching all associations (including the query)

Show the code related to the display of the results (left side passing results to the product card & right side product card code displays results)

Checklist Items (1)

#2 Show the code related to the display of the results

```
partials > ● product_card.php M purchase_product.php M Result order_history.php M return_product.php db_helpers.php M product_card.php M
partials > ● product_card.php M purchase_product.php M Result order_history.php M return_product.php db_helpers.php M product_card.php M
38     <li class="list-group-item">Type: <?php se($product, "typeName", "Unknown"); ?></li>
39   </ul>
40 </div>
41 <div class="card-body">
42   <?php if ($has_role("admin")) : ?>
43     <?php if (!isset($product["user_id"])) || $product["user_id"] === "N/A" : ?>
44       <a href="#"><?php echo get_url('admin/view_product.php?id' . $product["id"]); ?>" class="btn btn-secondary">View</a>
45       <a href="#"><?php echo get_url('api/return_product.php?product_id=' . $product["id"]); ?>" class="btn btn-primary">Purchase</a>
46     <?php else : ?>
47       <!-- ycf3 4/25/23 -->
48       <p class="bg-warning text-dark text-center">Product not available.</p>
49       <a href="#"><?php echo get_url('admin/view_product.php?id' . $product["id"]); ?>" class="btn btn-secondary" style="margin-right: 10px;">View</a>
50       <a href="#"><?php echo get_url('api/return_product.php?product_id=' . $product["id"]); ?>" onclick="confirm('Are you sure?')?event.preventDefault(): class="btn btn-danger">Return</a>
51     </div>
52     <div class="oh-card-profile">
53       <a href="#"><?php echo get_url('profile.php?id' . $product["user_id"]); ?>"><?php se($product, "username", "N/A"); ?>'s Profile</a>
54       <?php foreach ($assoc_products as $user_product) : ?>
55         <?php if ($user_product['product_id'] == $product['id']) : ?>
56           <p class="card-text">Total Customers: <?php echo $user_product['total_users']; ?></p>
57         <?php endif; ?>
58       <?php endforeach; ?>
59     </div>
60   You, yesterday + added a count of the total number of users assoc.
61 <?php endif; ?>
62
63 <?php else : ?>
64   <?php if (!isset($product["user_id"])) || $product["user_id"] === "N/A" : ?>
65     <a href="#"><?php echo get_url('view_product_customer.php?id' . $product["id"]); ?>" class="btn btn-secondary">View</a>
66     <a href="#"><?php echo get_url('api/purchase_product.php?product_id=' . $product["id"]); ?>" class="btn btn-primary">Purchase</a>
67
68 <?php else : ?>
69   <p class="bg-warning text-dark text-center">Product not available.</p>
70   <a href="#"><?php echo get_url('view_product_customer.php?id' . $product["id"]); ?>" class="btn btn-secondary" style="margin-left:auto; margin-right:auto;">View</a>
71
72 <?php endif; ?>
73 <?php endif; ?>
74 </div>
75
76 >
77
78 ?>
```

Showing the code for the view button, delete button, clickable link to the user's profile

Checklist Items (3)

#3 Each record should have a button/link for single view

#4 Each record should have a username field that is clickable to go to the user's profile page

#5 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the sake of completeness).

#3 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity

The screenshot shows two code editor panes. The left pane contains `assign_products.php` and the right pane contains `return_product.php`. Both files are part of a project named `public_html > Project > admin`.

```
assign_products.php (183 lines)
return_product.php (183 lines)
```

`assign_products.php` handles the deletion of associations between users and products. It includes logic to check user roles, remove all associations for a user, and handle errors.

```
if ($user['role'] != 'Admin') {
    flash("You don't have permission to view this page", "warning");
    redirect("home.php");
}

$db = getDB();
// yc73 4/25/23 w/
//remove all associations
if (isset($_GET['remove'])) {
    $query = "DELETE FROM `UserProducts`";
    try {
        $stmt = $db->prepare($query);
        $stmt->execute();
        flash("Successfully returned all products", "success");
    } catch (PDOException $e) {
        error_log("Error removing product associations: " . var_export($e, true));
        flash("Error returning product", "danger");
    }
}

redirect("admin/product_associations.php");
} // yc73-23 if (isset($_GET['remove']))
```

`return_product.php` is an internal API endpoint to receive data and do something with it. It notes that this is not a standard page and uses session_start() to handle single associations.

```
// yc73 4/25/23 w/
//remove single association
require_once __DIR__ . "/Lib/Functions.php";
session_start();
if (isset($_GET['product_id']) && is_logged_in()) {
    $db = getDB();

//note for me this part grabs user id associated with the product from UserProducts
$query = "SELECT user_id FROM `UserProducts` WHERE product_id = :product_id";
$stmt = $db->prepare($query);
$stmt->execute(['product_id' => $_GET['product_id']]);
$assoc = $stmt->fetch(PDO::FETCH_ASSOC);
error_log("Query: " . var_export($assoc, true));
if ($assoc) {
    $user_id = $assoc['user_id'];
    //note for me this deletes using that grabbed user id
    $query = "DELETE FROM `UserProducts` WHERE user_id = :user_id AND product_id = :product_id";
    $stmt = $db->prepare($query);
    $stmt->execute(['user_id' => $user_id, 'product_id' => $_GET['product_id']]);
    error_log("Query: " . var_export($stmt, true));
}

if ($assoc) {
    $user_id = $assoc['user_id'];
    //note for me this deletes using that grabbed user id
    $query = "DELETE FROM `UserProducts` WHERE user_id = :user_id AND product_id = :product_id";
    $stmt = $db->prepare($query);
    $stmt->execute([$user_id]);
    flash("Successfully returned product", "success");
} catch (PDOException $e) {
    error_log("Error removing product associations: " . var_export($e, true));
    flash("Error returning product", "danger");
}
} // yc73-23 if (isset($_GET['product_id']) && is_logged_in())
```

Showing the logic to delete all associations and single delete

Checklist Items (1)

#5 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity

The screenshot shows two code editor panes. The left pane contains `profile.php` and the right pane contains `return_product.php`. Both files are part of a project named `public_html > Project > admin`.

```
profile.php (183 lines)
return_product.php (183 lines)
```

`profile.php` handles user profile management. It includes logic to fetch user data, update the profile, and handle file uploads for profile pictures.

```
$user = [];
if ($user['id'] > 0) {
    $db = getDB();
    $query = "SELECT email, username, created FROM users WHERE id = :user_id";
    try {
        $stmt = $db->prepare($query);
        $stmt->execute(['user_id' => $user['id']]);
        $r = $stmt->fetch();
        $user = $r;
    } catch (PDOException $e) {
        error_log("Error fetching user: " . var_export($e, true));
        flash("Error fetching user", "danger");
    }
} // yc73-123 if ($user['id'] > 0)
}

// yc73 4/26/23 w-


<div if ($is_me && $is_edit) i>
        <div class="profile-edit">
            <a class="btn btn-secondary btn-sm prof-view-btn" href="#">View</a>
            <form method="POST" encType="multipart/form-data">
                <div class="profileEdit">
                    <input type="text" name="email" value="{$user['email']}"/>
                    <input type="text" name="username" value="{$user['username']}"/>
                    <input type="password" name="currentPassword" value="{$user['password']}"/>
                    <input type="password" name="newPassword" value="{$user['newPassword']}"/>
                    <input type="password" name="confirmPassword" value="{$user['confirmPassword']}"/>
                    <input type="submit" value="Update Profile"/>
                </div>
            </form>
        </div>
    </div>
    <div class="profile-view">
        <a href="<?php echo get_url('home.php'); ?>" class="btn btn-secondary btn-sm prof-edit-btn" style="margin-right: 10px">Edit</a>
        <?php if ($is_me) : >
            | <a class="btn btn-primary btn-sm prof-edit-btn" href="edit.php">Edit</a>
        <?php endif; >
        <div class="card">
            <div class="card-body">
                <div class="mb-3" style="border-bottom: 1px solid #ccc; padding-bottom: 5px">
                    <input type="text" name="username" value="{$user['username']}"/>
                </div>
                <div class="mb-3" style="border-bottom: 1px solid #ccc; padding-bottom: 5px">
                    <input type="text" name="email" value="{$user['email']}"/>
                </div>
                <div class="mb-3" style="border-bottom: 1px solid #ccc; padding-bottom: 5px">
                    <input type="text" name="password" value="{$user['password']}"/>
                </div>
                <div class="mb-3" style="border-bottom: 1px solid #ccc; padding-bottom: 5px">
                    <input type="text" name="newPassword" value="{$user['newPassword']}"/>
                </div>
                <div class="mb-3" style="border-bottom: 1px solid #ccc; padding-bottom: 5px">
                    <input type="text" name="confirmPassword" value="{$user['confirmPassword']}"/>
                </div>
            </div>
        </div>
    </div>


```

Showing logic for profile view (and edit)

Checklist Items (1)

#4 Each record should have a username field that is clickable to go to the user's profile page

Showing the logic related to the count of items on the page / and the count of all items associated

Checklist Items (2)

#6 Show the logic related to the count of all associated items (even the ones not shown in the filtered results)

#7 Show the logic related to the count of the items on the page (this value should change based on the filter applied)

```

75 //username
76 $username = $_GET['username'];
77 if (!empty($username)) {
78     $query .= " AND username like '$username'";
79     $params['username'] = "%$username%";
80 }
81 //name
82 $name = $_GET['name'];
83 if (!empty($name)) {
84     $query .= " AND name like '%$name%'";
85     $params['name'] = "%$name%";
86 }
87
88 //price
89 $price_min = $_GET['price_min'] == '-' ? 1 : $price_min;
90 if (!empty($price_min) && $price_min >= 0) {
91     $query .= " AND pr.price >= $price_min";
92     $params['price_min'] = $price_min;
93 }
94
95 $price_max = $_GET['price_max'] == '-' ? 100 : $price_max;
96 if (!empty($price_max) && $price_max >= 0) {
97     $query .= " AND pr.price <= $price_max";
98     $params['price_max'] = $price_max;
99 }
100
101 /*+ yC73 4/25/23 */
102 //fetch products
103

```

Showing the logic related to filter/sort (and limit is constrained to 1-100 & defaults to 10 otherwise)

Checklist Items (1)

#8 Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)

```

partial > ● product_card.php > ...
You 8 hours ago | Author (this)
1 <?php
2 if (!isset($product)) {
3     error_log("Using Product partial without data");
4     flash("Dev Alert: Product called without data", "danger");
5 }
6
7
8 <?php /*+ yC73 4/25/23 */
9 $db = getDB();
10 $query = "SELECT product_id, COUNT(user_id) AS total_users FROM `UserProducts` GROUP BY product_id";
11 try {
12     $stmt = $db->prepare($query);
13     $stmt->execute();
14     $assoc_products = $stmt->fetchall();
15 } catch (PDOException $e) {
16     error_log("Error getting number of users: " . var_export($e, true));
17 }
18
19
20 <!-- yC73 4/25/23 -->
21 <?php if (isset($product)) : >
22 <-- https://yoursite.com/entries/icons/original/1000/929/953/Screen_Shot_2019-06-05_at_1.16.32_AM.jpg -->
23 <div class="card-img-out" style="width: 36rem;">
24     <p>If it has relate('Admin') : </p>
25     <p>If it's (isset($product['username'])) : </p>
26     <div class="card-header">
27         Owned By: <?php echo($product['username'], 'N/A'); ?>
28     </div>

```

```

partial > ● product_card.php > ○ div.card.m-auto > ○ div.card-body > ○ div.card-profile
60 <?php else : >
61     <!-- yC73 4/25/23 -->
62     <p class="bg-warning text-dark text-center">Product not available</p>
63     <a href=?> echo get_url('admin/view_product.php?id' . $product['id']); ?> class="btn btn-secondary" >
64     <a href=?> echo get_url('api/review_product.php?product_id=' . $product['id']); ?> onclick="confirm('Are you sure you want to review this product?'); return true;" >
65     <div class="oh-card-profile">
66         <?php echo get_url('profile.php?id' . $product['user_id']); ?>
67         <?php se($product, 'username') >
68         <?php foreach ($assoc_products as $user_product) : >
69             <?php if ($user_product['product_id'] == $product['id']) : >
70                 <p class="card-text">Total Customers: <?php echo $user_product['total_users']; ?></p>
71             </?php endif; ?>
72         <?php endforeach; ?>
73     </div>
74
75 <?php endif; ?>
76
77 <?php endif; ?>
78
79 </div>

```

Showing count of total number of users associated with one product

Checklist Items (0)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Mention how you determine the result list (include the association logic and filters)

		Mention how you determine the result list (include the association logic and filters)
#2	1	Mention the logic behind the two counts (total items and visible result count)
#3	1	Mention the logic for single delete
#4	1	Mention the logic for handling the username requirements

Response:

First, when a user clicks the purchase button on a card it grabs that product's id and starts to build a query to INSERT into the "UserProducts" table. It will insert the (current logged-in) user's id, the specific product's id, the product name, and the product price.

- (also since the product's name and price weren't passed I grabbed it with a SELECT subquery based on the product id)

The query is prepared and executed. If it is successfully inserted it will flash a success message to the user.

Now that table "UserProducts" has the association, the "Product Associations" page builds a query to get the necessary user and product information to display it. The query starts by SELECTing username, user id, product id, product name, product price, and other product info (api_id, measurement, typeName, image, contextualImageUrl, imageAlt, url, categoryPath, stock, and is_api). It gets the product info from the "Products" table and JOINs it with the "UserProducts" based on the (associating) product's id. It JOINs the "Users" table based on user's id to get the (associating) user's info.

If someone uses filters (username, product name, min/max price, product type, category, sort, order, limit), it:

- grabs what they entered and assigns it to a variable
- checks if the variable is not empty & if it's a number (min/max price) it checks if it's greater than -1
- if it passes: the entered filter gets added to the query with AND as well as LIKE - to get partial matches (for username, product name, type, and category), the value is also assigned to the corresponding parameter with "%" - which is a wildcard that helps look for partial matches
- for sort and order, it checks if the user chose one of the options in the selection (array), and if not it defaults sort to created & order to descending and gets added to the query with ORDER BY
- for limit, it checks to see if the entered number falls within a range: if it's less than 1 or greater than 100 it fails and defaults to 10, if valid the entered limit number gets added to the query with LIMIT

The query is prepared and executed. If it successfully selects it will fetch and store the \$results in a variable. Each product in the result variable will be passed to the product card to render and display the results.

To count the visible items:

- it uses the count() function to count the number of fetched records (from the assoc table) in the \$results array

To count the total items:

- first, a query specifies that it's grabbing data from the 'Products' table and JOINs it with the "UserProducts" table based on the (associating) product's id
 - (thus counting all the records in the UserProducts table rather than only certain ones depending on the logged-in user's id)
- it passes that query to the get_total_count function
- the get_total_count function: first sanitizes the table references, next it uses a SELECT query to count the number of records from the table, and then it executes, fetches, & returns the total count result as an integer

To render the count it calls the function render_result_counts() and passes visible items count & total items count. It first checks if the visible count & total count are set. If they are NOT it flashes a user-friendly error message and sets the value to -1. If they are set, then they are displayed.

If someone wants to delete a single association:

- When a user clicks the "return" button (p.s. my delete button is called return) on a card it passes the product's id to "return_product.php". "return_product.php" first checks if the product id is set, if it is then it specifies a query. This query will SELECT the corresponding (associated) user's id FROM the "UserProducts" table based on WHERE it matches the (associated) product's id. The query is executed, fetches, and stores the result in a variable (\$result). It proceeds to grab the associated user's id from the \$result. Another query is made to DELETE the record from the "UserProducts" table WHERE the (assoc) user id AND (assoc) product id match. It gets executed, and if it successfully deletes the association it will flash a success message to the user.

Regarding the username requirements:

- To show the username an item is associated with:
 - In the product card code, it grabs the (assoc) user's username from the passed (assoc) product results and (safe) echoes it. Safe echo basically takes the passed value and uses htmlspecialchars() to convert special characters to HTML entities before using echo to display the username on the screen.
- To count the total number of users associated with an item:
 - It makes a query that SELECTs the product's id & uses COUNT(user_id) to basically count the number of times a user is associated with a specific product from the "UserProducts" table & groups by the product's id so each count corresponds to that specific product. Then the total count is displayed with echo.
- Link to redirect to the respective user's profile
 - To get the link to show the corresponding user's profile it simply grabs their username from the passed (associated) product results.
 - To generate the user's profile link, it grabs the corresponding user's id from the passed (associated) product results and (using href) adds it to the URL. By adding the id to the link, the profile.php redirects to that specific user profile when clicked.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input checked="" type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Project/admin/product_associations.php

URL #2

<https://github.com/yaneliii/yc73-it202-008/pull/70>

Unassociated Page (2 pts.)

[^COLLAPSE ^](#)

[COLLAPSE](#)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

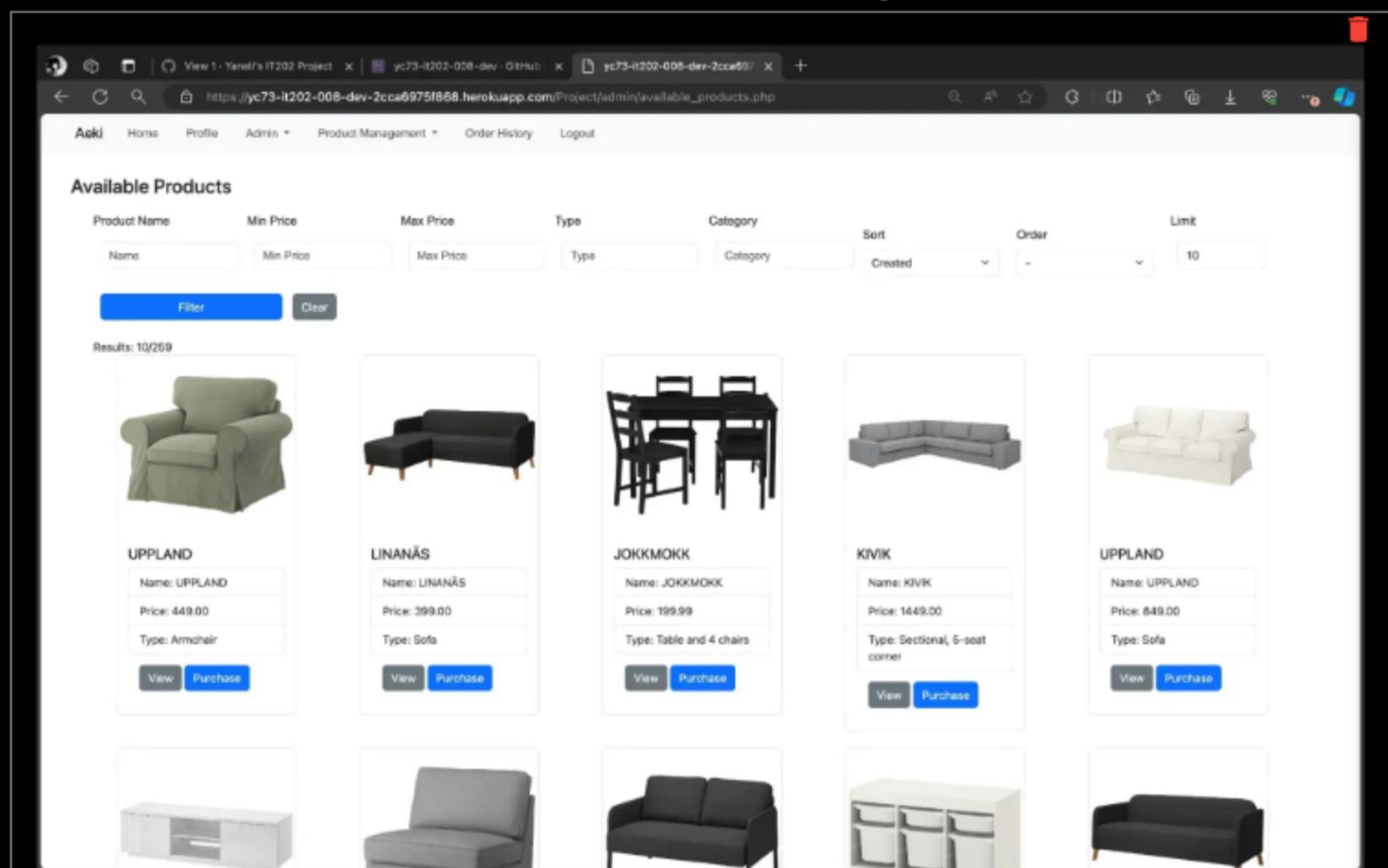
*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the summary of the results with relevant information per entity
<input type="checkbox"/> #2	1	Show the single view buttons/links
<input type="checkbox"/> #3	1	Show variations of the number of shown items count and show the count of total number of associated items
<input type="checkbox"/> #4	1	Show variations of the filter/sort including no results (proper message should be visible)
<input type="checkbox"/> #5	1	Make sure the heroku dev url is visible in the address bar
<input type="checkbox"/> #6	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Showing UNassociation page (admin) - with summary and view button

Checklist Items (2)

#1 Show the summary of the results with relevant information per entity

#2 Show the single view buttons/links

Available Products

Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
<input type="text" value="malm"/>	<input type="text" value="Min Price"/>	<input type="text" value="Max Price"/>	<input type="text" value="Type"/>	<input type="text" value="Category"/>	<input type="button" value="Created"/>	<input type="button" value="-"/>	<input type="text" value="10"/>
<input type="button" value="Filter"/> <input type="button" value="Clear"/>							
Results: 5/259							
MALM	MALM	MALM	MALM	MALM			
Name: MALM							
Price: 349.00	Price: 129.99	Price: 89.99	Price: 179.99	Price: 199.99			
Type: High bed frame/2 storage boxes	Type: Dressing table	Type: 2-drawer chest	Type: Desk with pull-out panel	Type: Desk			
<input type="button" value="View"/> <input type="button" value="Purchase"/>							

Showing variation of count - number of items shown and total number of items unassociated items

Checklist Items (1)

#3 Show variations of the number of shown items count and show the count of total number of associated items

Available Products

Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
<input type="text" value="upp"/>	<input type="text" value="Min Price"/>	<input type="text" value="Max Price"/>	<input type="text" value="Type"/>	<input type="text" value="Category"/>	<input type="button" value="Created"/>	<input type="button" value="-"/>	<input type="text" value="10"/>
<input type="button" value="Filter"/> <input type="button" value="Clear"/>							
Results: 8/259							
UPPLAND	UPPLAND	UPPLAND	UPPLAND	UPPLAND			
Name: UPPLAND							
Price: 449.00	Price: 849.00	Price: 139.00	Price: 1599.00	Price: 749.00			
Type: Armchair	Type: Sofa	Type: Cover for sofa	Type: Sectional, 4-seat corner	Type: Loveseat			
<input type="button" value="View"/> <input type="button" value="Purchase"/>							

Showing variation of count - number of items shown and total number of items unassociated items

Checklist Items (1)

#3 Show variations of the number of shown items count and show the count of total number of associated items

Available Products

Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
Name	500	1000	Type	chair	Created	-	10

Results: 4/259

KIVIK
Name: KIVIK
Price: 549.00
Type: 1-seat sleeper sofa
[View](#) [Purchase](#)

INGATORP / EKEDALEN
Name: INGATORP / EKEDALEN
Price: 739.99
Type: Table and 4 chairs
[View](#) [Purchase](#)

EKEDALEN / EKEDALEN
Name: EKEDALEN / EKEDALEN
Price: 659.99
Type: Table and 6 chairs
[View](#) [Purchase](#)

DANDERYD / DANDERYD
Name: DANDERYD / DANDERYD
Price: 609.00
Type: Table and 4 chairs
[View](#) [Purchase](#)

Showing variation of the filter/sort (min price, max price, category, sort/order)

Checklist Items (1)

#4 Show variations of the filter/sort including no results (proper message should be visible)

Available Products

Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit
ss	200	700	Type	bed	Created	-	10

Results: 3/259

SONGESAND

SAGESUND

SAGESUND

Name: SONGESAND	Name: SAGESUND	Name: SAGESUND
Price: 369.00	Price: 549.00	Price: 689.00
Type: Bed frame with 2 storage boxes	Type: Upholstered bed frame	Type: Upholstered bed frame
View Purchase	View Purchase	View Purchase

Showing variation of the filter/sort (product name, min/max price, category, sort/order)

Checklist Items (1)

#4 Show variations of the filter/sort including no results (proper message should be visible)

A screenshot of a web application interface titled "Available Products". The page includes a search form with fields for Product Name, Min Price, Max Price, Type, Category, Sort, Order, and Limit. The "Sort" dropdown is set to "Created" and the "Order" dropdown is set to "=" (asc). The "Limit" field is set to 10. The search results area displays the message "Results: 0/259" and "No results to show".

Showing variation of the filter/sort - NO result with proper message

Checklist Items (1)

#4 Show variations of the filter/sort including no results (proper message should be visible)

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show the code related to fetching all unassociated entities (including the query)

#2	1	Show the code related to the display of the results
#3	1	Each record should have a button/link for single view
#4	1	Show the logic related to the count of all unassociated items (even the ones not shown in the filtered results)
#5	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
#6	1	Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)
#7	1	Include ucid/date comments for each code screenshot
#8	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```
available_products.php (line 115) public_html > Project > admin > available_products.php > $db

115 if ($limit < 1 || $limit > 500) {
116     $limit = 10;
117 }
118 //IMPORTANT make sure you fully validate/trust $limit (sql injection possibility)
119 $query .= " LIMIT $limit";
120 } <- #119-120 if (count($GET) > 0)
121
122
123 else {
124     try {
125         $limit = (int)$GET["Limit"] == 10? false;
126     } catch (Exception $e) {
127         $limit = 10;
128     }
129     if ($limit < 1 || $limit > 500) {
130         $limit = 10;
131     }
132     $query .= " LIMIT $limit";
133 } <- #120-129 else
134
135 $db = getDB();
136 $stmt = $db->prepare($query);
137 $results = [];
138 try {
139     $stmt->execute($params);
140     $r = $stmt->fetchAll();
141     if ($r) {
142         $results = $r;
143     }
144 } catch (PDOException $e) {
145     error_log("Error fetching products " . var_export($e, true));
146     die("Unhandled error occurred", "danger");
147 }
148 foreach ($results as $index => $product) {
149     foreach ($product as $key => $value) {
150         if ($key == 'value') {
151             $results[$index][$key] = $value;
152         }
153     }
154 }
155
156 $table = [
157     "data" => $results, "title" => "Products", "ignored_columns" => ["id"],
158     "view_url" => get_url("view_product.php"),
159 ];
160
161 </div> class="container">
162     <div class="ad-products-header">
163         <h1 class="ad-title">Available Products</h1>
164     </div>
165     <div class="all-products-container">
166         <form method="get">
```

available_products.php (line 1) public_html > Project > admin > available_products.php > \$db

```
available_products.php (line 25) public_html > Project > admin > available_products.php > $db

25 static_records = get_static_count("Products" pr WHERE pr.id NOT IN (SELECT product_id FROM `UserProducts`)");
26
27 // yc73 4/26/23 n/
28 $query = "SELECT pr.id, pr.name, pr.price, measurement, typeName, image, contextualImageUrl, imageAlt,
29     pr.categoryPath, stock FROM `Products` pr
30 WHERE pr.id NOT IN (SELECT product_id FROM `UserProducts`)";
31
32 $params = [];
33
34 session_key = $_SERVER["SCREPE_NAME"];
35 $is_clear = isset($_GET["clear"]);
36
37 if ($is_clear) {
38     session_delete(session_key);
39     unset($_SESSION["clear"]);
40     redirect(session_key);
41 } else {
42     $session_data = session_load(session_key);
43 }
44
45 if (count($GET) == 0 && !isset($session_data) && count($session_data) > 0) {
46     if (!isset($session_data)) {
47         $GET = $session_data;
48     }
49 }
50
51 <- #119-120 if (count($GET) == 0 && !isset($session_data) && count($session_data) > 0)
52
53 if (count($GET) > 0) {
54     session_save(session_key, $GET);
55     $keys = array_keys($GET);
56
57     foreach ($form as $k => $v) {
58         if (!in_array($k, $keys)) {
59             $form[$k]['value'] = $GET[$k['name']];
60         }
61     }
62 }
63
64 //price
65 $price_min = $GET["price_min", "", false];
66 if (!empty($price_min)) && $price_min > -1 {
67     $query .= " AND pr.price >= $price_min";
68     $params[":price_min"] = $price_min;
69 }
70
71 $price_max = $GET["price_max", "", false];
72 if (!empty($price_max)) && $price_max > -1 {
73     $query .= " AND pr.price <= $price_max";
74     $params[":price_max"] = $price_max;
75 }
76
77 //product type
78 $typeName = $GET["typeName", "", false];
79 if (!empty($typeName)) {
80     $query .= " AND typeName Like :typeName";
81     $params[":typeName"] = "%$typeName%";
82 }
```

Showing code that fetches all UNassociations - with the query

Checklist Items (0)

```

177     <a href="#">Clear</a>
178   </form>
179   <div class="row w-100 row-cards-align-m-3 row-cards-gap-0">
180     <!-- yC73 4/25/23 -->
181     <?php foreach ($records as $product) : ?>
182       <div class="col">
183         <?php render_product_card($product); ?>
184       </div>
185     </?php endif; ?>
186   </div>
187   <?php if (count($records) == 0) : ?>
188     <div class="col">
189       No results to show
190     </div>
191   </?php endif; ?>
192 </div>
193 </div>
194 </div>
195 <?php if ($records == null) : ?>
196   <div class="col">
197     <?php echo __DIR__ . "/../../partials/partials/_flash.php"; ?>
198   </div>
199 </?php endif; ?>

```

Show the code related to the display of the results (left side passing results to the product card & right side product card code displays results)

Checklist Items (1)

#2 Show the code related to the display of the results

```

1  .php M  ● purchase_product.php M  ● Result  ● order_history.php M  ● return_product.php  ● db_helpers.php M  ● product_card.php M  ● product_association.php M  ● ...
2  partials > ● product_card.php > ● div.card.ms-auto > ● div.card-body > ● div.card-body > ● a.btn.btn-secondary
3    <?php if (!isset($product['user_id'])) || $product['user_id'] == "N/A" : ?>
4      <?php if ($has_role("Admin")) : ?>
5        <?php if (!isset($product['user_id'])) || $product['user_id'] == "N/A" : ?>
6          <?php $id = isset($product['id']) ? $product['id'] : (isset($_GET['id']) ? $_GET['id'] : -1); ?>
7          <!-- yC73 4/25/23 -->
8          <a href="#">?php echo get_url('admin/view_product.php?id' . $product['id']); ?>" class="btn btn-secondary">View</a>
9          <a href="#">?php echo get_url('api/purchase_product.php?product_id' . $product['id']); ?>" class="btn btn-primary">Purchase</a>
10
11        <?php else : ?>
12          <!-- yC73 4/25/23 -->
13          <p class="bg-warning text-dark text-center">Product not available</p>
14          <a href="#">?php echo get_url('admin/view_product.php?id' . $product['id']); ?>" class="btn btn-secondary" style="margin-right: 10px;">View</a>
15          <a href="#">?php echo get_url('api/purchase_product.php?product_id' . $product['id']); ?>" class="btn btn-primary">Purchase</a>
16
17        <?php else : ?>
18          <!-- yC73 4/25/23 -->
19          <p class="bg-warning text-dark text-center">Product not available</p>
20          <a href="#">?php echo get_url('view_product_customer.php?id' . $product['id']); ?>" class="btn btn-secondary" style="margin-left:auto; margin-right:auto;">View</a>
21
22        <?php endif; ?>
23
24      <?php foreach ($assoc_products as $user_product) : ?>
25        <?php if ($user_product['product_id'] == $product['id']) : ?>
26          <?php echo get_url("profile.php?id" . $product['user_id']); ?>" class="btn btn-secondary">'s Profile</a>
27
28        <?php endforeach; ?>
29
30      <?php endif; ?>
31
32      <?php else : ?>
33        <!-- yC73 4/25/23 -->
34        <p class="bg-warning text-dark text-center">Product not available</p>
35        <a href="#">?php echo get_url('admin/view_product.php?id' . $product['id']); ?>" class="btn btn-secondary" style="margin-right: 10px;">View</a>
36        <a href="#">?php echo get_url('api/return_product.php?product_id' . $product['id']); ?>" onclick="confirm('Are you sure?')>event.preventDefault()>Return</a>
37
38      <?php else : ?>
39        <!-- yC73 4/25/23 -->
40        <p class="bg-warning text-dark text-center">Product not available</p>
41        <a href="#">?php echo get_url('view_product_customer.php?id' . $product['id']); ?>" class="btn btn-secondary" style="margin-left:auto; margin-right:auto;">View</a>
42
43      <?php endif; ?>
44
45      <?php else : ?>
46        <!-- yC73 4/25/23 -->
47        <p class="bg-warning text-dark text-center">Product not available</p>
48        <a href="#">?php echo get_url('admin/view_product.php?id' . $product['id']); ?>" class="btn btn-secondary">View</a>
49        <a href="#">?php echo get_url('api/purchase_product.php?product_id' . $product['id']); ?>" class="btn btn-primary">Purchase</a>
50
51      <?php else : ?>
52        <!-- yC73 4/25/23 -->
53        <p class="bg-warning text-dark text-center">Product not available</p>
54        <a href="#">?php echo get_url('admin/view_product.php?id' . $product['id']); ?>" class="btn btn-secondary" style="margin-right: 10px;">View</a>
55        <a href="#">?php echo get_url('api/return_product.php?product_id' . $product['id']); ?>" onclick="confirm('Are you sure?')>event.preventDefault()>Return</a>
56
57      <?php else : ?>
58        <!-- yC73 4/25/23 -->
59        <p class="bg-warning text-dark text-center">Product not available</p>
60        <a href="#">?php echo get_url('view_product_customer.php?id' . $product['id']); ?>" class="btn btn-secondary" style="margin-left:auto; margin-right:auto;">View</a>
61
62      <?php endif; ?>
63
64      <?php else : ?>
65        <!-- yC73 4/25/23 -->
66        <p class="bg-warning text-dark text-center">Product not available</p>
67        <a href="#">?php echo get_url('view_product_customer.php?id' . $product['id']); ?>" class="btn btn-secondary">View</a>
68        <a href="#">?php echo get_url('api/purchase_product.php?product_id' . $product['id']); ?>" class="btn btn-primary">Purchase</a>
69
70      <?php else : ?>
71        <!-- yC73 4/25/23 -->
72        <p class="bg-warning text-dark text-center">Product not available</p>
73        <a href="#">?php echo get_url('view_product_customer.php?id' . $product['id']); ?>" class="btn btn-secondary" style="margin-left:auto; margin-right:auto;">View</a>
74
75      <?php endif; ?>
76
77      <?php endif; ?>

```

Showing the code for the view button (since this is unassociated I also added a purchase button)

Checklist Items (1)

#3 Each record should have a button/link for single view

Showing the logic related to the count of items on the page / and the count of all items UNassociated

Checklist Items (2)

#4 Show the logic related to the count of all unassociated items (even the ones not shown in the filtered results)

#5 Show the logic related to the count of the items on the page (this value should change based on the filter applied)

```
public_html>Project>admin>available_products.php > ..
```

```
available_products.php M
```

```
public_html>Project>admin>available_products.php > ..
```

```
available_products.php M
```

```
77 //<--> y73 4/26/23 </>
78 //Query = "SELECT pr_id, pr_name, pr_price, measurement, typolink, image, contextualImageurl, imageAlt,
79 //url, categoryPath, stock FROM `Products` pr
80 //WHERE pr_id IN (SELECT product_id FROM `UserProducts`)" ;
81 $params = [];
82
83 session_start();
84 $session_key = $_SERVER["SCRIPT_NAME"];
85 $is_clean = isset($_GET["clear"]);
86 if ($is_clean) {
87     session_unset();
88     session_destroy();
89     redirect($session_key);
90 } else {
91     $session_data = session_load($session_key);
92 }
93
94 if (count($_GET) == 0 && !isset($session_data) && count($session_data) > 0) {
95     if ($session_data) {
96         $_GET = $session_data;
97     }
98 } else if (count($_GET) == 0 && !isset($session_data) && count($session_data) > 0) {
99     if (count($_GET) > 0) {
100     session_save($session_key, $_GET);
101     $keys = array_keys($_GET);
102
103     foreach ($keys as $key) {
104         if (in_array($key, $keys)) {
105             $form[$key]['value'] = $_GET[$key];
106         }
107     }
108 }
109
110 $start = $_GET['start'];
111 $limit = $_GET['limit'];
112 $sort = $_GET['sort'];
113 $order = $_GET['order'];
114
115 if (!empty($start)) {
116     $query .= " AND pr_id >= " . $start;
117 }
118
119 if (!empty($limit)) {
120     $query .= " LIMIT " . $limit;
121 }
122
123 if (!empty($sort)) {
124     $sort = $sort . ", " . $order;
125 }
126
127 if (!empty($order)) {
128     $order = $order;
129 }
130
131 //Product type
132 $typolink = $_GET['typolink'];
133 if (!empty($typolink)) {
134     $query .= " AND typolink LIKE '%{$typolink}%'";
135     $params["typolink"] = "%{$typolink}%";
136 }
137
138 //Category
139 $categoryPath = $_GET['categoryPath'];
140 if (!empty($categoryPath)) {
141     $query .= " AND categoryPath LIKE '%{$categoryPath}%'";
142     $params["categoryPath"] = "%{$categoryPath}%";
143 }
144
145 //Sort and order
146 $sort = $_GET['sort'];
147 if (!in_array($sort, ['name', 'price', 'typolink', 'categoryPath', 'created', 'modified'])) {
148     $sort = "created";
149 }
150
151 //MySQL I care about the data from table "pr"
152 if ($sort == "created" || $sort == "modified") {
153     $sort = "pr." . $sort;
154 }
155
156 $border = $_GET['border'];
157 if (!in_array($border, ['first', 'last'])) {
158     $border = "first";
159 }
160
161 //IMPORTANT make sure you fully validate/trust $sort and $border (SQL injection possibility)
162 $query .= " ORDER BY $sort $border";
163
164 try {
165     $stmt = $this->listTables($limit, "10", false);
166 } catch (Exception $e) {
167     $stmt = 10;
168 }
169
170 if ($stmt > 1 || $stmt > 100) {
171     $stmt = 10;
172 }
173
174 //IMPORTANT make sure you fully validate/trust $limit (SQL injection possibility)
175 $query .= " LIMIT $stmt";
176
177 //<--> y73 4/27/23 </>
178
179 else {
180     try {
181         $stmt = $this->listTables($limit, "10", false);
182     } catch (Exception $e) {
183         $stmt = 10;
184     }
185
186     if ($stmt < 1 || $stmt > 100) {
187         $stmt = 10;
188     }
189
190     $query .= " LIMIT $stmt";
191 }
192
193 //<--> y73 4/26/23 </>
194 //Category
195 //yesterday + added as admin page showing data NOT associated
196 $categoryPath = $_GET['categoryPath'];
197 if (!empty($categoryPath)) {
198     $query .= " AND categoryPath LIKE '%{$categoryPath}%'";
199     $params["categoryPath"] = "%{$categoryPath}%";
200 }
201
202 //Sort and order
203 $sort = $_GET['sort'];
204 if (!in_array($sort, ['name', 'price', 'typolink', 'categoryPath', 'created', 'modified'])) {
205     $sort = "created";
206 }
207
208 //MySQL I care about the data from table "pr"
209 if ($sort == "created" || $sort == "modified") {
210     $sort = "pr." . $sort;
211 }
212
213 $border = $_GET['border'];
214 if (!in_array($border, ['first', 'last'])) {
215     $border = "first";
216 }
217
218 //IMPORTANT make sure you fully validate/trust $sort and $border (SQL injection possibility)
219 $query .= " ORDER BY $sort $border";
220
221 //<--> y73 4/27/23 </>
222
223 try {
224     $stmt = $this->execute($query);
225     $results = [];
226
227     try {
228         $stmt->fetch();
229         $r = $stmt->fetch();
230         $results[] = $r;
231     } while ($stmt->fetch());
232
233     $stmt->close();
234 }
235 catch (Exception $e) {
236     $results = [];
237 }
238
239 return $results;
240 }
```

Checklist Items (1)

#6 Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Mention how you determine the result list (include the unassociated logic and filters)
<input checked="" type="checkbox"/> #2	1	Mention the logic behind the two counts (total items and visible result count)

Response:

It builds a query that starts by SELECTing product id, product name, product price, and other product info (api_id, measurement, typeName, image, contextualImageUrl, imageAlt, url, categoryPath, stock, and is_api) from the "Products" table. BUT, it makes sure that the product's id does not appear in the "UserProducts" table. This is because if a product is in "UserProducts" that means someone is associated with it. But, we want UNassociated products, so by checking that it's not in "UserProducts" that means NO ONE is associated with it.

If someone uses filters (product name, min/max price, product type, category, sort, order, limit), it:

- grabs what they entered and assigns it to a variable
- checks if the variable is not empty & if it's a number (min/max price) it checks if it's greater than -1
- if it passes: the entered filter gets added to the query with AND as well as LIKE - to get partial matches (for product name, type, and category), the value is also assigned to the corresponding parameter with "%" - which is a wildcard that helps look for partial matches
- for sort and order, it checks if the user chose one of the options in the selection (array), and if not it defaults sort to created & order to descending and gets added to the query with ORDER BY
- for limit, it checks to see if the entered number falls within a range: if it's less than 1 or greater than 100 it fails and defaults to 10, if valid the entered limit number gets added to the query with LIMIT

The query is prepared and executed. If it successfully selects it will fetch and store the \$results in a variable. Each product in the result variable will be passed to the product card to render and display the results.

To count the visible items:

- it uses the count() function to count the number of fetched records (from the products table) in the \$results array

To count the total items:

- first, a query specifies that it's grabbing data from the 'Products' table WHERE the products id does NOT appear "UserProducts" (basically making sure it's not grabbing any products that have an existing association)
- it passes that query to the get_total_count function
- the get_total_count function: first sanitizes the table references, next it uses a SELECT query to count the number of records from the table, and then it executes, fetches, & returns the total count result as an integer

To render the count it calls the function `render_result_counts()` and passes visible items count & total items count. It first checks if the visible count & total count are set. If they are NOT it flashes a user-friendly error message and sets the value to -1. If they are set, then they are displayed.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Project/admin/available_products.php

URL #2

<https://github.com/yaneliii/yc73-it202-008/pull/71>

Admin Association Management (like UserRoles) (1 pt.)

Task #1 - Points: 1

Text: Screenshots of the page

Checklist

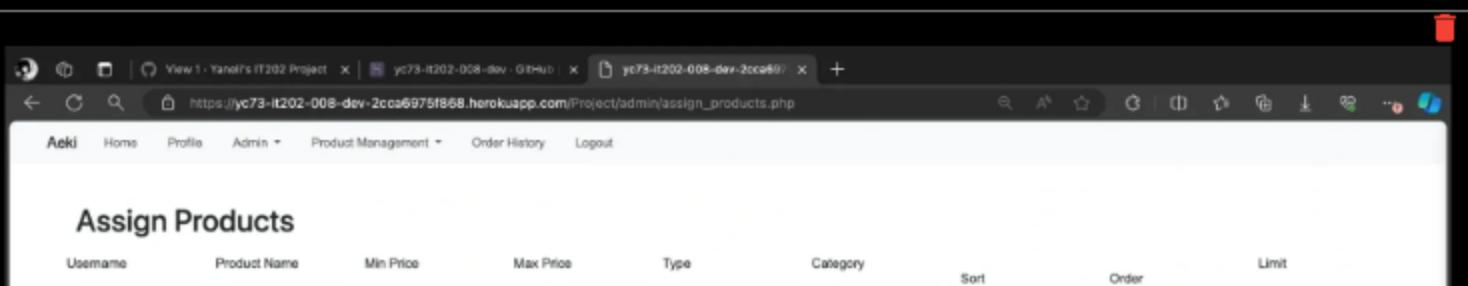
*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the search form with valid data
<input type="checkbox"/> #2	1	Show the results of the search
<input type="checkbox"/> #3	1	Show the result of entities and users being associated and unassociated
<input type="checkbox"/> #4	1	Make sure the heroku dev url is visible in the address bar
<input type="checkbox"/> #5	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Showing product association page (admin) - with search form and checkboxes

Username	Name	Min Price	Max Price	Type	Category	Created	Order	Limit																																																
								25																																																
<div style="display: flex; justify-content: space-between;"> <button>Search</button> <button>Clear</button> </div>																																																								
Users					Products to Assign																																																			
<table border="1"> <tbody> <tr><td>bob</td><td>No Products</td></tr> <tr><td>carl</td><td>No Products</td></tr> <tr><td>gab</td><td>No Products</td></tr> <tr><td>yan</td><td>LACK (19.99) (active), FRIHETEN (899.00) (active), FRIHETEN (749.00) (active), FINNALA (1699.00) (active)</td></tr> <tr><td></td><td>KALLAX (79.99) (active), MATCHSPEL (289.99) (active)</td></tr> <tr><td>yan2</td><td></td></tr> </tbody> </table>					bob	No Products	carl	No Products	gab	No Products	yan	LACK (19.99) (active), FRIHETEN (899.00) (active), FRIHETEN (749.00) (active), FINNALA (1699.00) (active)		KALLAX (79.99) (active), MATCHSPEL (289.99) (active)	yan2		<table border="1"> <tbody> <tr><td>KALLAX</td><td>79.99</td><td>Shelf unit</td><td>Living room & entryway tables</td></tr> <tr><td>LACK</td><td>19.99</td><td>TV unit</td><td>TV & media furniture</td></tr> <tr><td>FRIHETEN</td><td>899.00</td><td>Sleeper sectional,3 seat w/storage</td><td>Sofas & sectionals</td></tr> <tr><td>UPPLAND</td><td>449.00</td><td>Armchair</td><td>Chairs</td></tr> <tr><td>LINANÄS</td><td>399.00</td><td>Sofa</td><td>Sofas & sectionals</td></tr> <tr><td>JOOKMOKK</td><td>199.99</td><td>Table and 4 chairs</td><td>Chairs</td></tr> <tr><td>KIVIK</td><td>1449.00</td><td>Sectional, 5-seat corner</td><td>Sofas & sectionals</td></tr> <tr><td>FRIHETEN</td><td>749.00</td><td>Sleeper sofa</td><td>Sofas & sectionals</td></tr> <tr><td>FINNALA</td><td>1699.00</td><td>Sectional, 4-seat</td><td>Sofas & sectionals</td></tr> </tbody> </table>				KALLAX	79.99	Shelf unit	Living room & entryway tables	LACK	19.99	TV unit	TV & media furniture	FRIHETEN	899.00	Sleeper sectional,3 seat w/storage	Sofas & sectionals	UPPLAND	449.00	Armchair	Chairs	LINANÄS	399.00	Sofa	Sofas & sectionals	JOOKMOKK	199.99	Table and 4 chairs	Chairs	KIVIK	1449.00	Sectional, 5-seat corner	Sofas & sectionals	FRIHETEN	749.00	Sleeper sofa	Sofas & sectionals	FINNALA	1699.00	Sectional, 4-seat	Sofas & sectionals
bob	No Products																																																							
carl	No Products																																																							
gab	No Products																																																							
yan	LACK (19.99) (active), FRIHETEN (899.00) (active), FRIHETEN (749.00) (active), FINNALA (1699.00) (active)																																																							
	KALLAX (79.99) (active), MATCHSPEL (289.99) (active)																																																							
yan2																																																								
KALLAX	79.99	Shelf unit	Living room & entryway tables																																																					
LACK	19.99	TV unit	TV & media furniture																																																					
FRIHETEN	899.00	Sleeper sectional,3 seat w/storage	Sofas & sectionals																																																					
UPPLAND	449.00	Armchair	Chairs																																																					
LINANÄS	399.00	Sofa	Sofas & sectionals																																																					
JOOKMOKK	199.99	Table and 4 chairs	Chairs																																																					
KIVIK	1449.00	Sectional, 5-seat corner	Sofas & sectionals																																																					
FRIHETEN	749.00	Sleeper sofa	Sofas & sectionals																																																					
FINNALA	1699.00	Sectional, 4-seat	Sofas & sectionals																																																					
					<div style="text-align: right;"> <button>Toggle Products</button> </div>																																																			

Checklist Items (1)

#1 Show the search form with valid data

Showing the search form with valid data

Username	Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit																																																						
yan2	kallax	Min Price	Max Price	Type	Category	Created	-	25																																																						
<div style="display: flex; justify-content: space-between;"> <button>Search</button> <button>Clear</button> </div>																																																														
Users					Products to Assign																																																									
<table border="1"> <tbody> <tr><td><input type="checkbox"/></td><td>bob</td><td>No Products</td></tr> <tr><td><input type="checkbox"/></td><td>carl</td><td>No Products</td></tr> <tr><td><input type="checkbox"/></td><td>gab</td><td>No Products</td></tr> <tr><td><input type="checkbox"/></td><td>yan</td><td>LACK (19.99) (active), FRIHETEN (899.00) (active), FRIHETEN (749.00) (active), FINNALA (1699.00) (active)</td></tr> <tr><td><input type="checkbox"/></td><td></td><td>KALLAX (79.99) (active), MATCHSPEL (289.99) (active)</td></tr> <tr><td><input type="checkbox"/></td><td>yan2</td><td></td></tr> </tbody> </table>					<input type="checkbox"/>	bob	No Products	<input type="checkbox"/>	carl	No Products	<input type="checkbox"/>	gab	No Products	<input type="checkbox"/>	yan	LACK (19.99) (active), FRIHETEN (899.00) (active), FRIHETEN (749.00) (active), FINNALA (1699.00) (active)	<input type="checkbox"/>		KALLAX (79.99) (active), MATCHSPEL (289.99) (active)	<input type="checkbox"/>	yan2		<table border="1"> <tbody> <tr><td>KALLAX</td><td>79.99</td><td>Shelf unit</td><td>Living room & entryway tables</td></tr> <tr><td>LACK</td><td>19.99</td><td>TV unit</td><td>TV & media furniture</td></tr> <tr><td>FRIHETEN</td><td>899.00</td><td>Sleeper sectional,3 seat w/storage</td><td>Sofas & sectionals</td></tr> <tr><td>UPPLAND</td><td>449.00</td><td>Armchair</td><td>Chairs</td></tr> <tr><td>LINANÄS</td><td>399.00</td><td>Sofa</td><td>Sofas & sectionals</td></tr> <tr><td>JOOKMOKK</td><td>199.99</td><td>Table and 4 chairs</td><td>Chairs</td></tr> <tr><td>KIVIK</td><td>1449.00</td><td>Sectional, 5-seat corner</td><td>Sofas & sectionals</td></tr> <tr><td>FRIHETEN</td><td>749.00</td><td>Sleeper sofa</td><td>Sofas & sectionals</td></tr> <tr><td>FINNALA</td><td>1699.00</td><td>Sectional, 4-seat</td><td>Sofas & sectionals</td></tr> </tbody> </table>				KALLAX	79.99	Shelf unit	Living room & entryway tables	LACK	19.99	TV unit	TV & media furniture	FRIHETEN	899.00	Sleeper sectional,3 seat w/storage	Sofas & sectionals	UPPLAND	449.00	Armchair	Chairs	LINANÄS	399.00	Sofa	Sofas & sectionals	JOOKMOKK	199.99	Table and 4 chairs	Chairs	KIVIK	1449.00	Sectional, 5-seat corner	Sofas & sectionals	FRIHETEN	749.00	Sleeper sofa	Sofas & sectionals	FINNALA	1699.00	Sectional, 4-seat	Sofas & sectionals
<input type="checkbox"/>	bob	No Products																																																												
<input type="checkbox"/>	carl	No Products																																																												
<input type="checkbox"/>	gab	No Products																																																												
<input type="checkbox"/>	yan	LACK (19.99) (active), FRIHETEN (899.00) (active), FRIHETEN (749.00) (active), FINNALA (1699.00) (active)																																																												
<input type="checkbox"/>		KALLAX (79.99) (active), MATCHSPEL (289.99) (active)																																																												
<input type="checkbox"/>	yan2																																																													
KALLAX	79.99	Shelf unit	Living room & entryway tables																																																											
LACK	19.99	TV unit	TV & media furniture																																																											
FRIHETEN	899.00	Sleeper sectional,3 seat w/storage	Sofas & sectionals																																																											
UPPLAND	449.00	Armchair	Chairs																																																											
LINANÄS	399.00	Sofa	Sofas & sectionals																																																											
JOOKMOKK	199.99	Table and 4 chairs	Chairs																																																											
KIVIK	1449.00	Sectional, 5-seat corner	Sofas & sectionals																																																											
FRIHETEN	749.00	Sleeper sofa	Sofas & sectionals																																																											
FINNALA	1699.00	Sectional, 4-seat	Sofas & sectionals																																																											
					<div style="text-align: right;"> <button>Toggle Products</button> </div>																																																									

Checklist Items (1)

#1 Show the search form with valid data

Assign Products

Username	Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit																		
yan2	kallax	Min Price	Max Price	Type	Category	Created	-	25																		
<input type="button" value="Search"/> <input type="button" value="Clear"/>																										
Users					Products to Assign																					
<table border="1"> <tr><td>yan2</td><td>KALLAX (79.99) (active), MATCHSPEL (289.99) (active)</td></tr> </table>					yan2	KALLAX (79.99) (active), MATCHSPEL (289.99) (active)	<table border="1"> <tr><td>KALLAX</td><td>79.99</td><td>Shelf unit</td><td>Living room & entryway tables</td></tr> <tr><td>KALLAX</td><td>44.99</td><td>Shelf unit</td><td>Shelving furniture</td></tr> <tr><td>KALLAX</td><td>64.99</td><td>Shelf unit with doors</td><td>Shelving furniture</td></tr> <tr><td>KALLAX</td><td>79.99</td><td>Shelf unit with 2 inserts</td><td>Shelving furniture</td></tr> </table>				KALLAX	79.99	Shelf unit	Living room & entryway tables	KALLAX	44.99	Shelf unit	Shelving furniture	KALLAX	64.99	Shelf unit with doors	Shelving furniture	KALLAX	79.99	Shelf unit with 2 inserts	Shelving furniture
yan2	KALLAX (79.99) (active), MATCHSPEL (289.99) (active)																									
KALLAX	79.99	Shelf unit	Living room & entryway tables																							
KALLAX	44.99	Shelf unit	Shelving furniture																							
KALLAX	64.99	Shelf unit with doors	Shelving furniture																							
KALLAX	79.99	Shelf unit with 2 inserts	Shelving furniture																							
<input type="button" value="Toggle Products"/>																										

Show the results of the search

Checklist Items (1)

#2 Show the results of the search

Assign Products

Username	Product Name	Min Price	Max Price	Type	Category	Sort	Order	Limit																																																								
Username	Name	Min Price	Max Price	Type	Category	Created	-	25																																																								
<input type="button" value="Search"/> <input type="button" value="Clear"/>																																																																
Users					Products to Assign																																																											
<table border="1"> <tr><td><input type="checkbox"/></td><td>bob</td><td>No Products</td></tr> <tr><td><input type="checkbox"/></td><td>carl</td><td>No Products</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>gab</td><td>No Products</td></tr> <tr><td><input type="checkbox"/></td><td>yan</td><td>LACK (19.99) (active), FRIHETEN (899.00) (active), FRIHETEN (749.00) (active), FINNALA (1699.00) (active)</td></tr> <tr><td><input type="checkbox"/></td><td>yan2</td><td>KALLAX (79.99) (active), MATCHSPEL (289.99) (active)</td></tr> </table>					<input type="checkbox"/>	bob	No Products	<input type="checkbox"/>	carl	No Products	<input checked="" type="checkbox"/>	gab	No Products	<input type="checkbox"/>	yan	LACK (19.99) (active), FRIHETEN (899.00) (active), FRIHETEN (749.00) (active), FINNALA (1699.00) (active)	<input type="checkbox"/>	yan2	KALLAX (79.99) (active), MATCHSPEL (289.99) (active)	<table border="1"> <tr><td>KALLAX</td><td>79.99</td><td>Shelf unit</td><td>Living room & entryway tables</td></tr> <tr><td>LACK</td><td>19.99</td><td>TV unit</td><td>TV & media furniture</td></tr> <tr><td>FRIHETEN</td><td>899.00</td><td>Sleeper sectional, 3 seat w/storage</td><td>Sofas & sectionals</td></tr> <tr><td>UPPLAND</td><td>449.00</td><td>Armchair</td><td>Chairs</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>LINANÄS</td><td>399.00</td><td>Sofa</td><td>Sofas & sectionals</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>JOHANNES</td><td>199.99</td><td>Table and 4 chairs</td><td>Chairs</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>KIVIK</td><td>1449.00</td><td>Sectional, 5-seat corner</td><td>Sofas & sectionals</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>FRIHETEN</td><td>749.00</td><td>Sleeper sofa</td><td>Sofas & sectionals</td></tr> <tr><td><input checked="" type="checkbox"/></td><td>FINNALA</td><td>1699.00</td><td>Sectional, 4-seat</td><td>Sofas & sectionals</td></tr> </table>				KALLAX	79.99	Shelf unit	Living room & entryway tables	LACK	19.99	TV unit	TV & media furniture	FRIHETEN	899.00	Sleeper sectional, 3 seat w/storage	Sofas & sectionals	UPPLAND	449.00	Armchair	Chairs	<input checked="" type="checkbox"/>	LINANÄS	399.00	Sofa	Sofas & sectionals	<input checked="" type="checkbox"/>	JOHANNES	199.99	Table and 4 chairs	Chairs	<input checked="" type="checkbox"/>	KIVIK	1449.00	Sectional, 5-seat corner	Sofas & sectionals	<input checked="" type="checkbox"/>	FRIHETEN	749.00	Sleeper sofa	Sofas & sectionals	<input checked="" type="checkbox"/>	FINNALA	1699.00	Sectional, 4-seat	Sofas & sectionals
<input type="checkbox"/>	bob	No Products																																																														
<input type="checkbox"/>	carl	No Products																																																														
<input checked="" type="checkbox"/>	gab	No Products																																																														
<input type="checkbox"/>	yan	LACK (19.99) (active), FRIHETEN (899.00) (active), FRIHETEN (749.00) (active), FINNALA (1699.00) (active)																																																														
<input type="checkbox"/>	yan2	KALLAX (79.99) (active), MATCHSPEL (289.99) (active)																																																														
KALLAX	79.99	Shelf unit	Living room & entryway tables																																																													
LACK	19.99	TV unit	TV & media furniture																																																													
FRIHETEN	899.00	Sleeper sectional, 3 seat w/storage	Sofas & sectionals																																																													
UPPLAND	449.00	Armchair	Chairs																																																													
<input checked="" type="checkbox"/>	LINANÄS	399.00	Sofa	Sofas & sectionals																																																												
<input checked="" type="checkbox"/>	JOHANNES	199.99	Table and 4 chairs	Chairs																																																												
<input checked="" type="checkbox"/>	KIVIK	1449.00	Sectional, 5-seat corner	Sofas & sectionals																																																												
<input checked="" type="checkbox"/>	FRIHETEN	749.00	Sleeper sofa	Sofas & sectionals																																																												
<input checked="" type="checkbox"/>	FINNALA	1699.00	Sectional, 4-seat	Sofas & sectionals																																																												
<input type="button" value="Toggle Products"/>																																																																

Showing BEFORE user and product association (user: gab & product: LINANÄS)

Checklist Items (1)

#3 Show the result of entities and users being associated and unassociated

The screenshot shows the 'Assign Products' interface. At the top, there's a search bar with fields for Username, Product Name, Min Price, Max Price, Type, Category, Sort, Order, and Limit. Below the search bar is a button to 'Search' and another to 'Clear'. On the left, under 'Users', there's a list of users: bob, carl, gab (selected), yan, and yan2. On the right, under 'Products to Assign', there's a list of products: MATCHSPEL, SEKTION, TILLREDA, LAGAN, FRYSSBAR, and ÖVERSÅDLIG. Each product has its name, price, description, and category.

Product	Price	Description	Category
MATCHSPEL	289.99	Gaming chair, Bomstad black	Chairs
SEKTION	185.00	Top cabinet for fridge/freezer	Kitchen cabinets
SEKTION	185.00	Top cab f fridge/freezer w push-op	Kitchen cabinets
SEKTION	133.00	Top cab f fridge/freezer w 2 doors	Kitchen cabinets
TILLREDA	119.00	Refrigerator	Appliances
LAGAN	749.00	Top-freezer refrigerator	Appliances
FRYSSBAR	829.00	Top-freezer refrigerator	Appliances
LAGAN	299.00	Fridge with freezer compartment	Kitchen systems
ÖVERSÅDLIG	2999.00	French door refrigerator	Appliances

Showing result AFTER user and product association (user: gab & product: LINANÄS)

Checklist Items (1)

#3 Show the result of entities and users being associated and unassociated

The screenshot shows the 'Assign Products' interface. The 'Users' section now lists bob, carl, gab (no longer selected), yan, and yan2. The 'Products to Assign' section remains the same as in the previous screenshot, showing the same list of products and their details.

Product	Price	Description	Category
MATCHSPEL	289.99	Gaming chair, Bomstad black	Chairs
SEKTION	185.00	Top cabinet for fridge/freezer	Kitchen cabinets
SEKTION	185.00	Top cab f fridge/freezer w push-op	Kitchen cabinets
SEKTION	133.00	Top cab f fridge/freezer w 2 doors	Kitchen cabinets
TILLREDA	119.00	Refrigerator	Appliances
LAGAN	749.00	Top-freezer refrigerator	Appliances
FRYSSBAR	829.00	Top-freezer refrigerator	Appliances
LAGAN	299.00	Fridge with freezer compartment	Kitchen systems
ÖVERSÅDLIG	2999.00	French door refrigerator	Appliances

	LAGAN	749.00	Top-freezer refrigerator	Appliances
	FRYSBAR	829.00	Top-freezer refrigerator	Appliances
	LAGAN	299.00	Fridge with freezer compartment	Kitchen systems
	ÖVERSÄKDLIG	2999.00	French door refrigerator	Appliances

Showing result AFTER user and product UNassociation (user: gab)

Checklist Items (1)

#3 Show the result of entities and users being associated and unassociated

The screenshot shows the MySQL Workbench interface with the database 'it202' selected. The left sidebar shows tables like Products, Roles, Samples, UserProducts, UserRoles, and Users. The 'UserProducts' table is currently selected. A query window displays the following SQL command:

```
SELECT * FROM 'UserProducts' LIMIT 100
```

The results show 7 rows of data:

	id	user_id	product_id	product_name	price	quantity	is_active	created
1	48	2	1	KALLAX	79.99	(NULL)	1	2024-04-26 20:28:53
2	200	1	2	LACK	19.99	(NULL)	1	2024-04-28 16:18:32
3	204	2	393	MATCHSPEL	289.99	(NULL)	1	2024-04-28 16:28:53
4	207	1	3	FRIHETEN	899.00	(NULL)	1	2024-04-28 20:08:31
5	208	1	8	FRIHETEN	749.00	(NULL)	1	2024-04-29 01:29:12
6	209	1	9	FINNALA	1699.00	(NULL)	1	2024-04-29 01:29:12
7	213	5	6	LINANÄS	399.00	(NULL)	1	2024-04-29 17:29:11

Showing table AFTER user and product association (user: gab & product: LINANÄS)

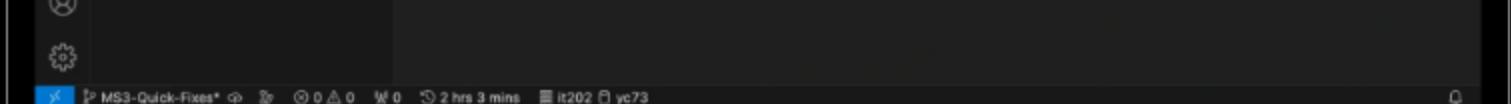
Checklist Items (0)

The screenshot shows the MySQL Workbench interface with the database 'it202' selected. The left sidebar shows tables like Products, Roles, Samples, UserProducts, UserRoles, and Users. The 'UserProducts' table is currently selected. A query window displays the following SQL command:

```
SELECT * FROM 'UserProducts' LIMIT 100
```

The results show 6 rows of data, identical to the previous screenshot but with one additional row for user_id 5 and product_id 6:

	id	user_id	product_id	product_name	price	quantity	is_active	created
1	48	2	1	KALLAX	79.99	(NULL)	1	2024-04-26 20:28:53
2	200	1	2	LACK	19.99	(NULL)	1	2024-04-28 16:18:32
3	204	2	393	MATCHSPEL	289.99	(NULL)	1	2024-04-28 16:28:53
4	207	1	3	FRIHETEN	899.00	(NULL)	1	2024-04-28 20:08:31
5	208	1	8	FRIHETEN	749.00	(NULL)	1	2024-04-29 01:29:12
6	209	1	9	FINNALA	1699.00	(NULL)	1	2024-04-29 01:29:12
7	213	5	6	LINANÄS	399.00	(NULL)	1	2024-04-29 17:29:11



Checklist Items (0)

Task #2 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Search form field for finding a partial match of usernames
<input checked="" type="checkbox"/> #2	1	Search form field for finding a partial match of entities
<input checked="" type="checkbox"/> #3	1	Code related to getting a max of 25 results for each field (i.e., 25 users and 25 entities limit)
<input checked="" type="checkbox"/> #4	1	Code that generates the checkboxes next to each list (users and entities)
<input checked="" type="checkbox"/> #5	1	Code related to submitting the checkbox lists
<input checked="" type="checkbox"/> #6	1	Code related to applying the associations upon submission (i.e., add the relationship if it doesn't exist and remove the relationship if it does exist)
<input checked="" type="checkbox"/> #7	1	Include ucid/date comments for each code screenshot
<input checked="" type="checkbox"/> #8	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

A screenshot of a browser window displaying a code editor. The URL bar shows "yc73-it202-008". The code editor contains a large block of PHP code. The code includes file inclusion, permission checks, and a form building loop. It uses variables like \$form, \$type, \$name, and \$placeholder, and includes logic for handling different input types like text, number, select, and checkboxes.

```

26     ["type" => "number", "name" => "limit", "label" => "Limit", "value" => "25", "include_margin" => false],
27   ] <- #13-28 $form =
28 //error_log("Form data: " . var_export($form, true));
29
30
31 /* yc73 4/26/23 */
32 //attempt to apply
33 if (isset($_POST["users"])) { //isset($_POST["products"])
34   $user_ids = $_POST["users"]; //sel doesn't like arrays so we'll just do this
35   $product_ids = $_POST["products"]; //sel doesn't like arrays so we'll just do this
36   if (empty($user_ids) || empty($product_ids)) {
37     flash("Both users and products need to be selected", "warning");
38   } else {
39     //for sake of simplicity, this will be a tad inefficient
40     $db = getDB();
41     foreach ($user_ids as $user_id) {
42       foreach ($product_ids as $product_id) {
43         try {
44           $check_query = "SELECT is_active FROM UserProducts WHERE user_id = :user_id AND product_id = :product_id";
45           ...
46         }
47       }
48     }
49   }
50 }
51
52 
```

⌚ You, 14 hours ago Ln 13, Col 1 (20 selected) Spaces: 4 UTF-8 LF PHP

Showing search form fields for finding a partial match of usernames & products

Checklist Items (2)

#1 Search form field for finding a partial match of usernames

#2 Search form field for finding a partial match of entities

The screenshot shows a code editor window with the title bar "yc73-#202-008". The code is written in PHP and is located in the file "assign_products.php". The code implements a search function for users based on their username. It starts by initializing an empty array for users and getting the database connection. It then constructs a SQL query to select users whose username contains the search term from the POST variable. The query uses GROUP_CONCAT to handle multiple matches. It also handles session management, including clearing the session key and loading session data. The code includes error handling and logging.

```

199
200
201 /* yc73 4/26/23 */
202 //search for user by username
203 $users = [];
204 $db = getDB();
205
206 $query = "SELECT u.id, username, (SELECT GROUP_CONCAT(name, ', ', price) FROM Products pr ON user_product_id = pr.id
207 WHERE upr.user_id = u.id) AS product FROM users u";
208
209 $params = [];
210 $session_key = $_SERVER['SCRIPT_NAME'];
211 $is_clear = isset($_GET['clear']);
212 if ($is_clear) {
213   session_delete($session_key);
214   unset($_GET['clear']);
215   redirect($session_key);
216 } else {
217   $session_data = session_load($session_key);
218 }
219
220 if (count($_POST) == 0 && isset($session_data) && count($session_data) > 0) {
221   if ($session_data) {
222     $_POST = $session_data;
223   }
224 } <- #209-223 if (count($_POST) == 0 && isset($session_data) && count($session_data) > 0)
225 if (count($_POST) > 0) {
226   session_set($session_key, $_POST);
227   $keys = array_keys($_POST);
228
229   foreach ($keys as $k => $v) {
230     if (is_array($v)) {
231       $keys[$k] = $_POST[$k]['value'];
232     }
233   }
234 }
235
236 $username = $_POST['username'];
237 if (empty($username)) {
238   $query .= " WHERE username like :username";
239   $params['username'] = "%$username%";
240 }
241
242 $stmt = $db->prepare($query);
243 try {
244   $stmt->execute($params);
245   $results = $stmt->fetchALL(PDO::FETCH_ASSOC);
246   if ($results) {
247     $users = $results;
248   }
249 } catch (PDOException $e) {
250   flash('Error: ' . $e->getMessage(), true, "danger");
251 }
252
253
254 
```

Showing search form field logic for finding a partial match of usernames

Checklist Items (1)

#1 Search form field for finding a partial match of usernames

The screenshot shows a code editor window with the title bar "yc73-#202-008". The code is written in PHP and is located in the file "assign_products.php". The code implements a search function for products based on their type name. It starts by initializing an empty array for products and getting the database connection. It then constructs a SQL query to select products whose type name contains the search term from the POST variable. The query uses GROUP_CONCAT to handle multiple matches. It also handles session management, including clearing the session key and loading session data. The code includes error handling and logging.

```

79
80
81 //get products
82 $get_products = [];
83 $db = getDB();
84
85 $query = "SELECT id, api_id, name, price, measurement, typeName, image, contextualImageUrl, imageAlt,
86   url, categoryPath, stock, is_api
87   FROM `Products` WHERE 1";
88
89 
```

Showing search form fields logic for finding a partial match of products

Checklist Items (1)

#2 Search form field for finding a partial match of entities

Showing code related to getting a max of 25 results for each field (i.e., 25 users and 25 products limit)

Checklist Items (1)

#3 Code related to getting a max of 25 results for each field (i.e., 25 users and 25 entities limit)

Showing code that generates the checkboxes next to each list (users and products)

```

public_html > Project > admin > assign_products.php > ...
296     <table class="table">
297         <thead class="tableHeader">
298             <th>Users</th>
299             <th>Products to Assign</th>
300         </thead>
301         <tbody>
302             <tr>
303                 <td>
304                     <div class="assignProd-userContent">
305                         <table class="table">
306                             <?php foreach ($users as $user) : ?>
307                                 <tr>
308                                     <td>
309                                         <!-- yc73 4/26/23 (checkbox) -->
310                                         <?php render_input("type" => "checkbox", "id" => "user_.$user['id']", "", false, "name" => "users[]", "label" => $user['username'], "", false), "value" => $user['id'], "", false)) ?>
311                                     </td>
312                                     <td><?php sel($user, "product", "No Products"); ?></td>
313                                 </tr>
314                             <?php endforeach; ?>
315                         </table>
316                     </div>
317                 </td>
318                 <td class="assignTable">
319                     <div class="assignProd-content">
320                         <table class="table">
321                             <?php foreach ($get_products as $prod) : ?>
322                                 <tr>
323                                     <!-- yc73 4/26/23 (checkbox) -->
324                                     <td><?php render_input("type" => "checkbox", "id" => "product_.$prod['id']", "", false, "name" => "products[]", "label" => $prod['name'], "", false), "value" => $prod['id'], "", false)) ?></td>
325                                     <td><?php sel($prod, "price", "No Info"); ?></td>
326                                     <td><?php sel($prod, "typeName", "No Info"); ?></td>
327                                     <td><?php sel($prod, "categoryPath", "No Info"); ?></td>
328                                 </tr>
329                             <?php endforeach; ?>
330                         </table>
331                     </div>
332                 </td>
333             </tr>
334         </tbody>
335     </table>
336     <div class="arCont-button">
337         <?php render_button(["text" => "Toggle Products", "type" => "submit", "color" => "secondary"]); ?>
338     </div>
339 </form>

```

Checklist Items (1)

#4 Code that generates the checkboxes next to each list (users and entities)

Showing code that generates the checkboxes next to each list (users and products)

```

public_html > Project > admin > assign_products.php > <div>.container-fluid arBody </div>.all-products-container > <div>.form > <div>.row mb-3
291     <form method="POST"><!-- yc73 4/26/23 -->
292     <?php if (!isset($username) || !empty($username)) : ?>
293         <input type="hidden" name="username" value="<?php sel($username, false); ?>" />
294     <?php endif; ?>
295     <table class="table">
296         <thead class="tableHeader">
297             <th>Users</th>
298             <th>Products to Assign</th>
299         </thead>
300         <tbody>
301             <tr>
302                 <td>
303                     <div class="assignProd-userContent">
304                         <table class="table">
305                             <?php foreach ($users as $user) : ?>
306                                 <tr>
307                                     <td><!-- yc73 4/26/23 (checkbox) -->
308                                         <?php render_input("type" => "checkbox", "id" => "user_.$user['id']", "", false, "name" => "users[]", "label" => $user['username'], "", false), "value" => $user['id'], "", false)) ?>
309                                     </td>
310                                     <td><?php sel($user, "product", "No Products"); ?></td>
311                                 </tr>
312                             <?php endforeach; ?>
313                         </table>
314                     </div>
315                 </td>
316                 <td class="assignTable">
317                     <div class="assignProd-content">
318                         <table class="table">
319                             <?php foreach ($get_products as $prod) : ?>
320                                 <tr>
321                                     <td><!-- yc73 4/26/23 (checkbox) -->
322                                         <?php render_input("type" => "checkbox", "id" => "product_.$prod['id']", "", false, "name" => "products[]", "label" => $prod['name'], "", false), "value" => $prod['id'], "", false)) ?></td>
323                                     <td><?php sel($prod, "price", "No Info"); ?></td>
324                                     <td><?php sel($prod, "typeName", "No Info"); ?></td>
325                                     <td><?php sel($prod, "categoryPath", "No Info"); ?></td>
326                                 </tr>
327                             <?php endforeach; ?>
328                         </table>
329                     </div>
330                 </td>
331             </tr>
332         </tbody>
333     </table>
334     <div class="arCont-button"><?php render_button(["text" => "Toggle Products", "type" => "submit", "color" => "secondary"]); ?></div>
335 </form>

```

Showing code related to submitting the checkbox lists

Checklist Items (1)

#5 Code related to submitting the checkbox lists

The screenshot shows a code editor window with a file named `assign_products.php`. The code is written in PHP and handles the submission of a form where users can associate products with users via checkboxes. It includes logic to check if arrays were submitted, handle empty arrays, and execute database queries to either insert or delete records from the `UserProducts` table based on the checked boxes. The code uses MySQL syntax for queries and prepared statements.

```
public_html > Project > admin > assign_products.php > ...  
26  
27     "type" => "number", "name" => "limit", "label" => "Limit", "value" => "25", "include_merges" => false,  
28     ]); -- $form->error_log("Form data: " . var_export($form, true));  
29  
30     /* yc23 4/26/23 */  
31     //Attempt to apply  
32     if (isset($_POST['users']) && isset($_POST['products'])) {  
33         $user_ids = $_POST['users']; //sel doesn't like arrays so we'll just do this  
34         $product_ids = $_POST['products']; //sel doesn't like arrays so we'll just do this  
35         if (empty($user_ids) || empty($product_ids)) {  
36             flash("Both users and products need to be selected", "warning");  
37         } else {  
38             //for sake of simplicity, this will be a tad inefficient  
39             $db = getDB();  
40             foreach ($user_ids as $user_id) {  
41                 foreach ($product_ids as $product_id) {  
42                     try {  
43                         $check_query = "SELECT is_active FROM UserProducts WHERE user_id = :user_id AND product_id = :product_id";  
44                         $check_stmt = $db->prepare($check_query);  
45                         $check_stmt->execute([':user_id' => $user_id, ':product_id' => $product_id]);  
46                         $is_active = $check_stmt->fetchColumn();  
47  
48                         if ($is_active == 1) {  
49                             $delete_query = "DELETE FROM UserProducts WHERE user_id = :user_id AND product_id = :product_id";  
50                             $delete_stmt = $db->prepare($delete_query);  
51                             $delete_stmt->execute([':user_id' => $user_id, ':product_id' => $product_id]);  
52                             $db->commit();  
53                         } else {  
54                             $query = "INSERT INTO UserProducts (user_id, product_id, product_name, price, is_active) VALUES (:user_id, :product_id,  
55                             (SELECT name FROM Products WHERE id = :product_id), (SELECT price FROM Products WHERE id = :product_id), 1)  
56                             ON DUPLICATE KEY UPDATE is_active = :is_active";  
57                             $stmt = $db->prepare($query);  
58                             $stmt->execute([':user_id' => $user_id, ':product_id' => $product_id]);  
59                             $db->commit();  
60                         }  
61                         flash("Updated product", "success");  
62                     } catch (PDOException $e) {  
63                         flash("Error updating product", "danger");  
64                         error_log("Error applying: " . var_export($e->errorInfo, true));  
65                     }  
66                 } -- #34-35 foreach ($user_ids as $user_id)  
67             } -- #34-35 if (isset($_POST['users']) && isset($_POST['products']))  
68         }  
69         else if (isset($_POST['users']) || isset($_POST['products'])) {  
70             flash("Both users and products need to be selected", "warning");  
71         }  
72     } -- #34-35 /* yc23 4/26/23 */  
73     /* MS3-Quick-Fixer */  
74     $form->error_log("Form data: " . var_export($form, true));  
75 
```

Showing code related to applying the associations upon submission (i.e., add the relationship if it doesn't exist and remove the relationship if it does exist)

Checklist Items (1)

#6 Code related to applying the associations upon submission (i.e., add the relationship if it doesn't exist and remove the relationship if it does exist)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Describe the steps for the search and how it works for users and entities
<input checked="" type="checkbox"/> #2	1	Mention how you built the UI for this
<input checked="" type="checkbox"/> #3	1	Describe the steps for the associate/unassociate logic for the combination of users and entities

Response:

When users use the search to find products:

- It starts by building a query. The query starts by SELECTing the products id, api_id, name, price, measurement, typeName, image, contextualImageUrl, imageAlt, url, categoryPath, stock, is_api from the "Products" table.
- To search for products (name, min/max price, type, category) it:
 - grabs what they entered and assigns it to a variable
 - checks if the variable is not empty & if it's a number (min/max price) it checks if it's greater than -1
 - if it passes: the entered filter gets added to the query with AND as well as LIKE - to get partial matches (for product name, type, and category), the value is also assigned to the corresponding parameter with "%" - which is a wildcard that helps look for partial matches
 - for sort and order, it checks if the user chose one of the options in the selection (array), and if not it defaults sort to created & order to descending and gets added to the query with ORDER BY
 - for limit, it checks to see if the entered number falls within a range: if it's less than 1 or greater than 100 it fails and defaults to 25, if valid the entered limit number gets added to the query with LIMIT
- The query is prepared and executed. If it successfully selects it will fetch and store the \$results in a variable. Each product in the result variable will be rendered as a checkbox.

When users use the search to find users:

- It starts by building a query. The query starts by SELECTing the user's id and username. Then, using a subquery, it selects their associated product's name, price, & active status and concatenates them into a single string. The earlier selected user info is grabbed from the "UserProducts" table. It JOINs with the "Products" table based on the product's id and user's id from the "Users" table.
- To search for usernames it:
 - grabs what they entered and assigns it to a variable
 - checks if the variable is not empty
 - if it passes: the entered search gets added to the query with AND (where the username matches) as well as LIKE - to get partial matches, the value is also assigned to the corresponding parameter with "%" - which is a wildcard that helps look for partial matches
 - for limit, it checks to see if the entered number falls within a range: if it's less than 1 or greater than 100 it fails and defaults to 25, if valid the entered limit number gets added to the query with LIMIT
- The query is prepared and executed. If it successfully selects it will fetch and store the \$results in a variable. Each product in the result variable will be rendered as a checkbox.

For the UI of this page, I put the search fields at the top of the page, and under it the table that holds the user/product association form. Within this table, I made two data cells: the left cell displays a list of users with checkboxes (to select users) & the right cell displays a list of products with checkboxes (to select products). For each user checkbox item, I also wanted to show the product name, product price, and active status- to make it easier to know what products are associated). For each user checkbox item, I also wanted to show the product price, type, and category to its easier to search for a specific item rather than just looking at the product name. I also set heights for both cells so they can scroll within that box. This way if there's a long result, user's won't have to scroll all the way to the bottom of the actual page just to click the toggle button, they just scroll within that set box.

It first checks if the checkbox for the user and product was both selected. If not it flashes a warning to the user that both must be selected. If they are selected then it proceeds to grab the selected user(s) and product(s) and store them into (seperate) variables. It checks if those variables are empty. If they're not empty, then the outer loop goes through each user and the inner loop goes through each product for that particular user. For each product, it uses a query to SELECT the is_active column from the "UserProducts" table based on that particular user & product. By grabbing the is_active column it can check whether it's already active (associated) or not.

- To associate a combination of users and products:
 - If it's NOT active (not associated), it runs the else condition and makes a query to INSERT a record (into the UserProducts table) of the chosen user (id), the chosen product (its id, name, price), and makes it

active. This insertion creates an association between the user and the product.

- To UNassociate a combination of users and products:
 - It checks if is_active contains a 1, the 1 means that the association does currently exist. For my site, I didn't want to just mark it as inactive, I wanted to treat it like the return button that removes the association completely. So, after confirming the is_active is 1, it makes a query to DELETE the association record (from the UserProducts table) based on the chosen user's id and chosen product's id. Thus, deleting the association.
- After a successful association or unassociation, the user flashes a success message.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input checked="" type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

https://yc73-it202-008-prod-35e9bd30f553.herokuapp.com/Project/admin/assign_products.php

URL #2

<https://github.com/yaneliii/yc73-it202-008/pull/72>

Misc (1 pt.)

[^ COLLAPSE ^](#)

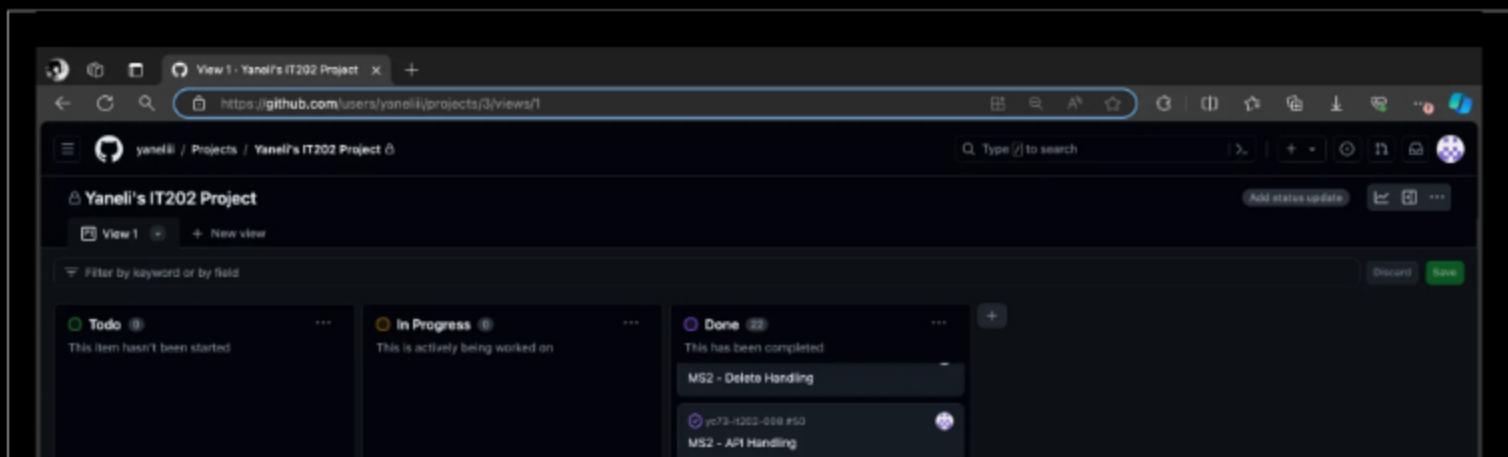
Task #1 - Points: 1

Text: Screenshot of your project board from GitHub (tasks should be in the proper column)

Task Screenshots:

Gallery Style: Large View

Small Medium Large



A screenshot of a GitHub project board. At the top, there are three columns with headers: 'Add item' (with a plus icon), 'Add item' (with a plus icon), and 'Add item' (with a plus icon). Below these are six issues, each represented by a card:

- yc73-i202-008 #67 MS3 - API Data Association
- yc73-i202-009 #68 MS3 - Handle the association of data to a user
- yc73-i202-008 #64 MS3 - Logged in user's associated entities page
- yc73-i202-009 #65 MS3 - All Users association page
- yc73-i202-008 #66 MS3 - Create a page that shows data not associated with any user
- yc73-i202-009 #67 MS3 - Admin can associate any entity with any users

Showing GitHub project board (with MS3 issues in the proper column)

Task #2 - Points: 1

Text: Provide a direct link to the project board on GitHub

URL #1

<https://github.com/users/yaneliii/projects/3>

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

Luckily for milestone 3 I didn't run into any major issues. The only part that sort of took me a while to understand was the admin association form - trying to associate products to existing users. More specifically trying to remove the association if the relationship existed because unlike the assign roles page, I didn't want to mark it as just "inactive". Marking it "inactive" still caused the product to appear in the users "order history". So, I had trouble thinking about how to I could get it to delete the association instead. Then, I realized that when its active, each record has a 1 for "is_active" and I used that to my advantage. When the user submits the form, it checks if "is_active" is (aka the relation already exists), so this means they are trying to remove it, thus I run a DELETE statement to remove the association from the table which also removes it from "order history".

Task #4 - Points: 1

Text: WakaTime Screenshot

Details:

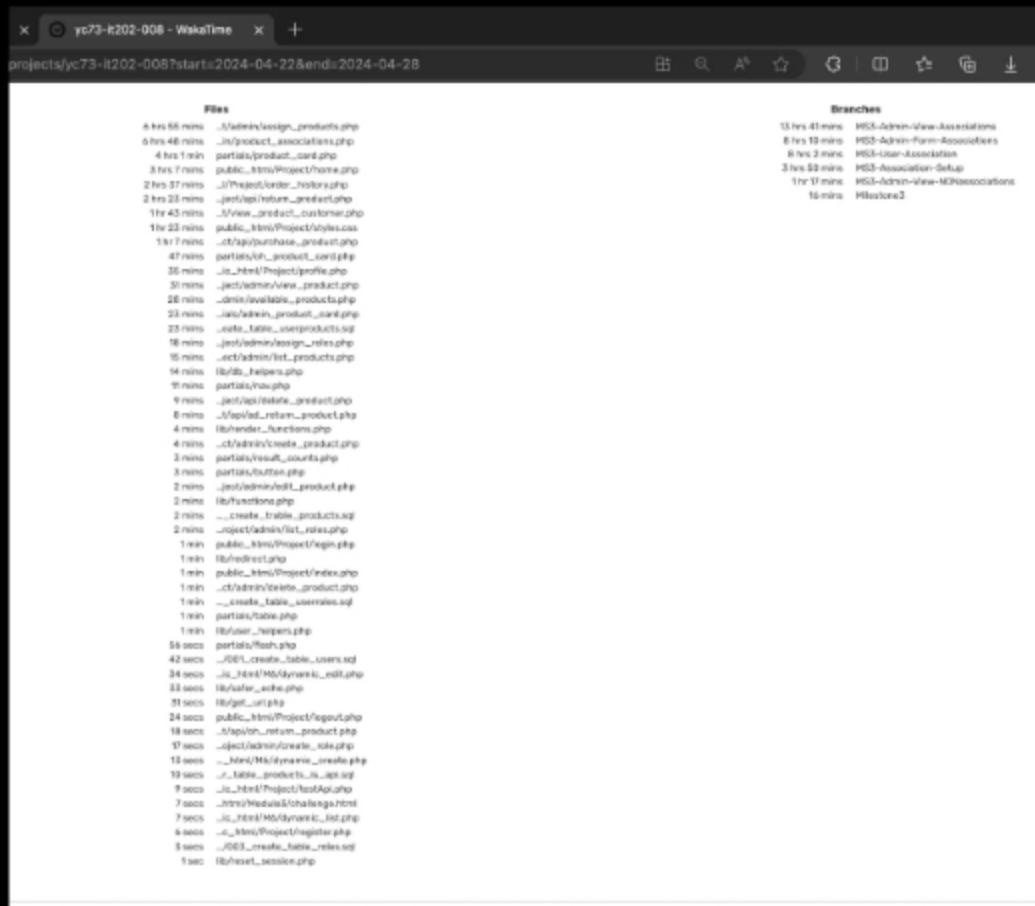
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Showing wakatime for milestone3

End of Assignment