

# Submission Worksheet

**CLICK TO GRADE**

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-3-2024/grade/tlj3>

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

### Submissions:

Submission Selection

1 Submission [active] 4/28/2024 11:26:35 PM

### Instructions

**▲ COLLAPSE ▲**

Implement the Milestone 3 features from the project's proposal document: <https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88E>  
Make sure you add your ucid/date as code comments where code changes are done  
All code changes should reach the Milestone3 branch  
Create a pull request from Milestone3 to dev and keep it open until you get the output PDF from this assignment.  
Gather the evidence of feature completion based on the below tasks.  
Once finished, get the output PDF and copy/move it to your repository folder on your local machine.  
Run the necessary git add, commit, and push steps to move it to GitHub  
Complete the pull request that was opened earlier  
Create and merge a pull request from dev to prod  
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   |
|----|--------|---|
| #1 | 1      | What's the concept/association?   |
| #2 | 1      | What sort of relationship is it (one to many, many to one, many to many, etc) |
| #3 | 1      | Note any other considerations   |

Response:

What I decided to do for the concept was make a Pokemon NFT website. My initial idea was vastly different from this. However, since I made the Pokemon Cards, it looked a lot like NFTs, so I decided that was a good way to associate it with the users since they would be purchasing those NFTs.

The relationship would be one to many. So, in the case of my website, the users can buy as many pokemon nfts as they want, but each pokemon nft would only be associated to one user.

### Task #2 - Points: 1

Text: Updating Entities

#### Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details   |
|----|--------|---|
| #1 | 1      | When an update occurs either manually or from the API how does it affect associated data? |
| #2 | 1      | Do users see the old data, new data, does data need to be reassociated, etc?              |

Response:

When an update occurs, whether from the API or manually, it would just change the data, whether it's associated with a user or not. I have a query set the updates all data that changes, whether manually or from the API. So, there's never a worry that something would go wrong with the associated data if it was subject to change.

Users wouldn't see the old data, they would see the new data and wouldn't have to worry about reassociation since their associated data would update too (which I did test by manually editing an associated 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   |
|----|--------|---|
| #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) |
| #2 | 1      | Include ucid/date comments for each code screenshot   |
| #3 | 1      | Clearly caption screenshots   |

## Task Screenshots:

## Gallery Style: Large View

code from Pokemons.php, where associations are done pt1

## Checklist Items (0)

```
(1) pokemons_info.json
(1) pokemons.php
(1) pokemons.php
(1) profile.php
(1) register.php
(1) styles.css
(1) testApi.php

(1) index.php
(1) placeholder.html
(1) README.md
(1) test_db.php
(1) test_fleahp
(1) t3j3_002-project-prep...
(1) gitignore
(1) .htaccess
(1) composer.json
(1) composer.lock
(1) partials - ShortcutLink
(1) Profile
(1) public.html - ShortcutLink
(1) README.md
(1) t3j3_002-api-project-md...
(1) t3j3_002-milestone-1...
(1) OUTLINE
(1) TIMELINE

109     $results = [];
110   try {
111     $stmt->execute($params);
112     $r = $stmt->fetchAll();
113     if ($r) {
114       $results = $r;
115     }
116   } catch (PDOException $e) {
117     error_log("Error fetching pokemons", var_export($e, true));
118     flash("Unhandled error occurred", "danger");
119   }
120
121   $state = ["data" => $results, "title" => "List of Pokemons", "ignored_columns" => ["id"], "view_url"=> get_url("pokemons.php")]
122 }
123
124 <div class="container-fluid">
125   <h1>Pokemons</h1>
126   <form method="GET">
127     <div class="row mb-3" style="align-items: flex-end;">
128       <?php foreach ($form as $k => $v);?>
129         <div class="col-2">
130           <?php render_input($v);?>
131         </div>
132       <?php endforeach;?>
133     </div>
134     <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
135     <a href="?clear" class="btn btn-secondary">Clear</a>
136   </form>
137   <?php render_result_counts(count($results), $state, records);?>
```

code from Pokemons.php, where associations are done pt2

## Checklist Items (0)



EXPLORER ... pokemons.php X purchase\_pokemon.php pokemon\_associations.php delete\_relationship.php my\_pokemons.php pokemon\_card

TLJ3-IT202-008 public\_html > Project > pokemons.php > ...

```
93     <div class="container-fluid">
108         <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 row-cols-lg-4 rows-cols-xl-5 rows-cols-xxl-6 g-4">
109             <?php foreach($results as $pokemon):?>
110                 <div class="col">
111                     <?php render_pokemon_card($pokemon);?>
112                 </div>
113             <?php endforeach;?>
114             <?php if(count($results) === 0):?>
115                 <div class="col">
116                     | No results to show
117                 </div>
118             <?php endif;?>
119         </div>
120     </div>
121
122     <?php
123         //note we need to go up 1 more directory
124         require_once(__DIR__ . "/../../partials/flash.php");
125     ?>
```

code from Pokemons.php, where associations are done pt3

### Checklist Items (0)

### Task #2 - Points: 1

#### **Text: Screenshot of the association table(s)**

| Checklist |        | *The checkboxes are for your own tracking         |
|-----------|--------|---|
| #         | Points | Details   |
| 1         | 1      | Please attach a copy of your last bill statement. |

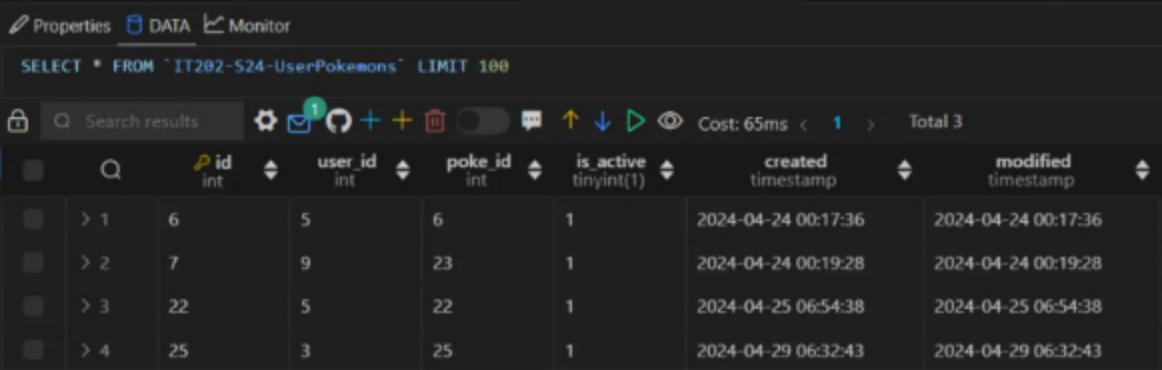
|    |   |   |
|----|---|---|
| #1 | 1 | Show the table(s) you made to handle the associations |
| #2 | 1 | Should have some example data                         |
| #3 | 1 | Clearly caption screenshots                           |

### Task Screenshots:

**Gallery Style: Large View**

---

Small      Medium      Large



The screenshot shows a database interface with a sidebar containing schema information for 'IT202 8.0.36-Ubuntu' and 'tj3 208k'. The main area displays a table named 'UserPokemons' with the following data:

|   | p_id | user_id | poke_id | is_active | created             | modified            |
|---|------|---------|---------|-----------|---------------------|---------------------|
| 1 | 6    | 5       | 6       | 1         | 2024-04-24 00:17:36 | 2024-04-24 00:17:36 |
| 2 | 7    | 9       | 23      | 1         | 2024-04-24 00:19:28 | 2024-04-24 00:19:28 |
| 3 | 22   | 5       | 22      | 1         | 2024-04-25 06:54:38 | 2024-04-25 06:54:38 |
| 4 | 25   | 3       | 25      | 1         | 2024-04-29 06:32:43 | 2024-04-29 06:32:43 |

table used for association

### Checklist Items (0)

#### Task #3 - Points: 1

Text: Explain solution

### Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details   |
|----|--------|---|
| #1 | 1      | Mention which option your application handles regarding association |
| #2 | 1      | Describe each column/association table                              |

Response:

The option that my application handles regarding association is Option 1, which is that related pages will have a button to do the association (in this case, it would be the "Purchase Pokemon" button).

With the columns in the UserPokemon association table, the id column is the unique identifier for each record that's in the table (it would auto-increment each new entry), the user\_id would represent the user who owns the Pokemon (linked as a foreign key to the id in the Users table), poke\_id represents the Pokemon that's associated to the user (linked as a foreign key to id in the "IT202\_S24\_Pokemon" table), is\_active would signify whether they're active or not (1 would mean they are, 0 would mean they aren't), created indicates the time the record was created, and modified indicates the time the record was updated last.

#### Task #4 - Points: 1

Text: Add related links

##### Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details   |
|----|--------|---|
| #1 | 1      | Include the heroku prod link for the page that creates the association  |
| #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://tlj3-prod-5077e065f26e.herokuapp.com/Project/pokemons.php>

##### URL #2

<https://github.com/TiaraJenks/tlj3-it202-008/pull/66>

#### Current User's Association Page (2 pts.)

#### Task #1 - Points: 1

Text: Screenshots of this page

##### Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details   |
|----|--------|---|
| #1 | 1      | Show the summary of the results with relevant information per entity  |
| #2 | 1      | Show the single view buttons/links, delete button/links, and delete all button/link                                   |
| #3 | 1      | Show variations of the number of shown items count and show the count of total number of associated items to the user |
| #4 | 1      | Show variations of the filter/sort including no results (proper message should be visible)                            |

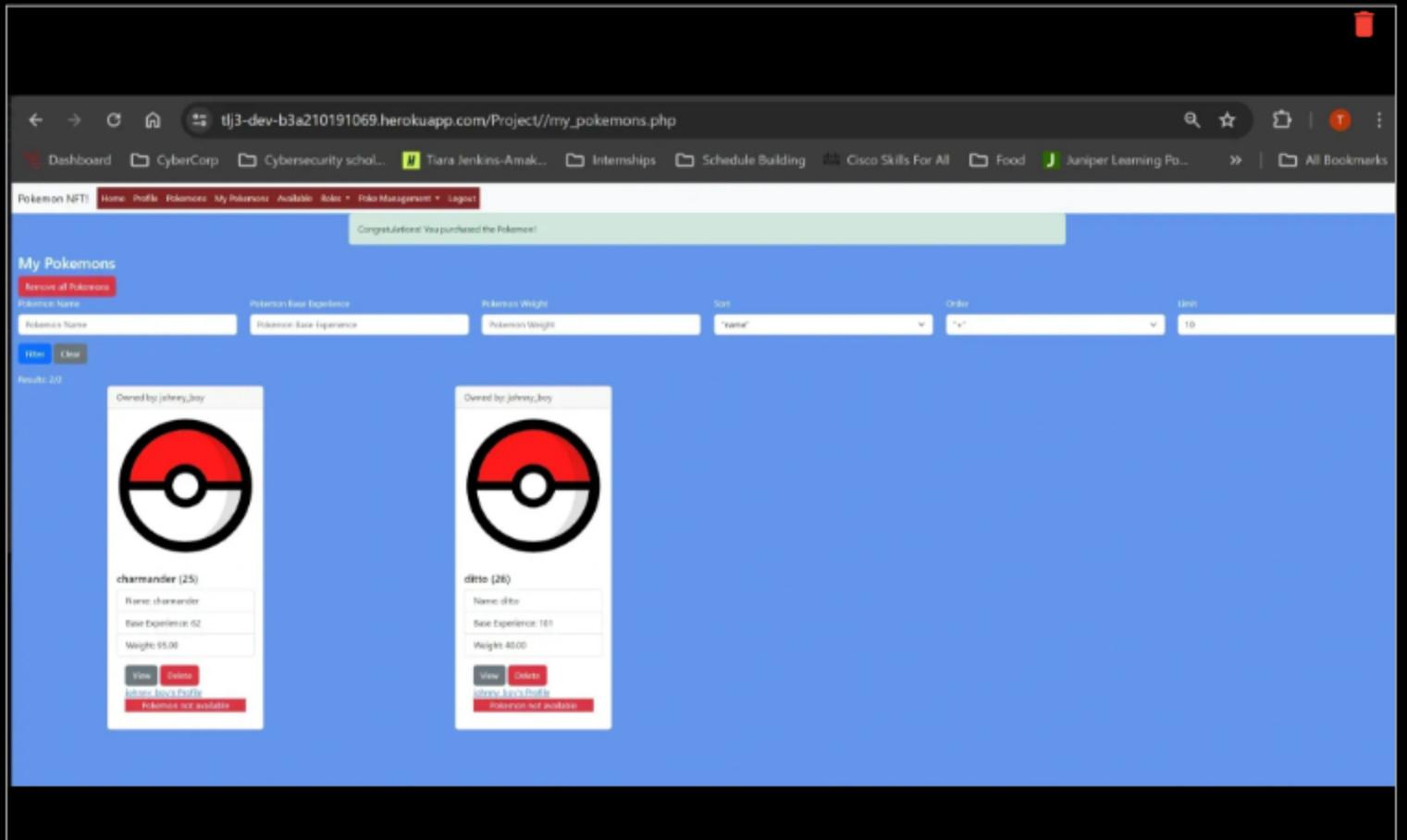
|    |   |  |
|----|---|--|
| #5 | 1 | Make sure the heroku dev url is visible in the address bar |
| #6 | 1 | Clearly caption screenshots                                |

## Task Screenshots:

**Gallery Style: Large View**

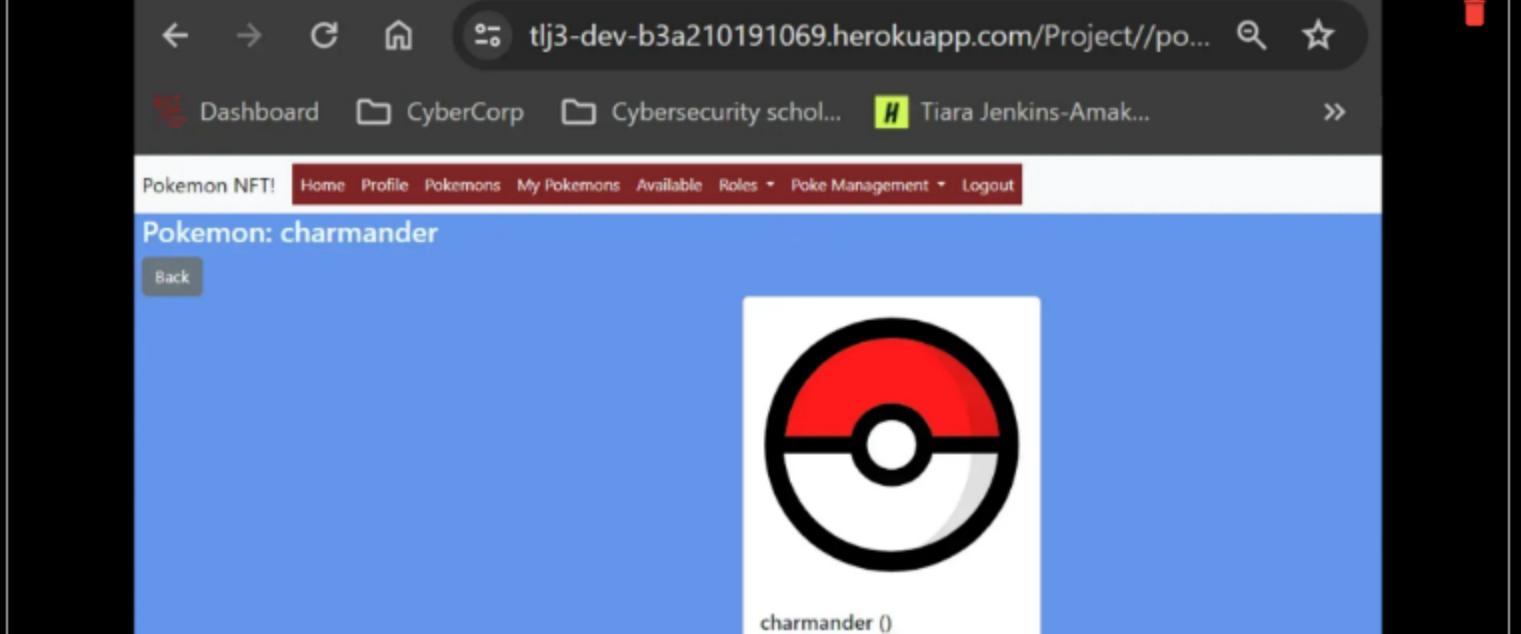
---

Small      Medium      Large



main page that shows the user's associations (includes count, filter/sort, view, delete, remove all, and a message stating that the entity is associated with a user)

## Checklist Items (0)



|                                  |
|----------------------------------|
| Name: charmander                 |
| Base Experience: 62              |
| Weight: 95.00                    |
| <a href="#">Purchase Pokemon</a> |

what's shown when view button is clicked

## Checklist Items (0)

The screenshot shows a web browser window with the URL [tlj3-dev-b3a210191069.herokuapp.com/Project//my\\_pokemons.php](http://tlj3-dev-b3a210191069.herokuapp.com/Project//my_pokemons.php). The page title is "Pokemon NFT!". The main content area is titled "My Pokemons". A message at the top says "Successfully removed your Pokemon!". Below it, there is a search/filter section with fields for "Pokemon Name", "Pokemon Base Experience", "Pokemon Weight", "Sort", "Order", and a "Link" dropdown set to "10". Underneath is a "Results: 1/1" section. It displays a card for "charmander (25)" owned by "johnny\_boy". The card contains the following details:

- Pokemon Name: charmander
- Base Experience: 62
- Weight: 95.00

At the bottom of the card are two buttons: "View" (grey) and "Delete" (red). Below the card is a link "johnny\_boy's Profile" and a note "Pokemon not available".

when delete button is clicked (success message shown)

## Checklist Items (0)

The screenshot shows a web browser window with the URL [tlj3-dev-b3a210191069.herokuapp.com/Project//my\\_pokemons.php](http://tlj3-dev-b3a210191069.herokuapp.com/Project//my_pokemons.php). The page title is "Pokemon NFT!". The main content area is titled "My Pokemons". A message at the top says "Successfully removed your Pokemon!". Below it, there is a search/filter section with fields for "Pokemon Name", "Pokemon Base Experience", "Pokemon Weight", "Sort", "Order", and a "Link" dropdown set to "10". Underneath is a "Results: 1/2" section. It displays a card for "ditto" owned by "johnny\_boy". The card contains the following details:

- Pokemon Name: ditto
- Base Experience: 62
- Weight: 95.00

At the bottom of the card are two buttons: "View" (grey) and "Delete" (red). Below the card is a link "johnny\_boy's Profile" and a note "Pokemon not available".

ditto (26)

|                      |
|----------------------|
| Name: ditto          |
| Base Experience: 101 |
| Weight: 40.00        |

[View](#) [Delete](#)

johnny\_boy's Profile

Pokemon not available

## filter/sort variation and count variation pt1

### Checklist Items (0)

Dashboard CyberCorp Cybersecurity schol... Tiara Jenkins-Amak...

Pokemon NFT! Home Profile Pokemons My Pokemons Available Roles Poke Management Logout

### My Pokemons

Remove all Pokemons

| Pokemon Name | Pokemon Base Experience | Pokemon Weight | Sort   | Order |
|--------------|-------------------------|----------------|--------|-------|
| Pokemon Name | 101                     | Pokemon Weight | "name" | "a"   |

Filter Clear

Results: 1/2

Owned by: johnny\_boy



ditto (26)

|                      |
|----------------------|
| Name: ditto          |
| Base Experience: 101 |
| Weight: 40.00        |

[View](#) [Delete](#)

johnny\_boy's Profile

Pokemon not available

## filter/sort variation and count variation pt2

### Checklist Items (0)

Dashboard CyberCorp Cybersecurity schol... Tiara Jenkins-Amak...

Pokemon NFT! Home Profile Pokemons My Pokemons Available Roles Poke Management Logout

### My Pokemons

Remove all Pokemons

| Pokemon Name | Pokemon Base Experience | Pokemon Weight | Sort   | Order |
|--------------|-------------------------|----------------|--------|-------|
| Pokemon Name | Pokemon Base Experience | 95             | "name" | "a"   |

Filter Clear

Results: 1/2

Owned by: johnny\_boy



ditto (26)

|                      |
|----------------------|
| Name: ditto          |
| Base Experience: 101 |
| Weight: 40.00        |

[View](#) [Delete](#)

johnny\_boy's Profile

Pokemon not available

charmander (25)

Name: charmander

Base Experience: 62

Weight: 95.00

[View](#)

[Delete](#)

[johnny\\_boy's Profile](#)

Pokemon not available

## filter/sort variation pt3

Checklist Items (0)

The screenshot shows a web application interface for managing Pokemons. At the top, there is a navigation bar with links for Dashboard, CyberCorp, Cybersecurity schol..., and a user profile for Tiara Jenkins-Amak... Below the navigation bar is a sub-navigation menu for 'Pokemon NFT!' with options like Home, Profile, Pokemons, My Pokemons, Available, Roles, Poke Management, and Logout.

The main content area is titled 'My Pokemons'. It features a search bar with fields for 'Pokemon Name', 'Pokemon Base Experience', 'Pokemon Weight', 'Sort' (set to "-name"), and 'Order' (set to "ASC"). There are also 'Filter' and 'Clear' buttons. The results section displays two entries:

- charmander (25)**  
Name: charmander  
Base Experience: 62  
Weight: 95.00  
[View](#) [Delete](#)  
[johnny\\_boy's Profile](#)  
Pokemon not available
- ditto (26)**  
Name: ditto  
Base Experience: 101  
Weight: 40.00  
[View](#) [Delete](#)  
[johnny\\_boy's Profile](#)  
Pokemon not available

## filter/sort variation pt4

Checklist Items (0)

The screenshot shows a confirmation dialog box overlaid on the previous 'My Pokemons' page. The dialog box contains the text "tlj3-dev-b3a210191069.herokuapp.com says" and "Are you sure?". At the bottom right of the dialog box are two buttons: "OK" and "Cancel".

The underlying page remains the same, displaying the two Pokemons from the previous screenshot.

ditto (26)  
Name: ditto  
Base Experience: 101  
Weight: 40.00  
View Delete  
johnny\_bay's Profile  
Pokemon not available

charmander (25)  
Name: charmander  
Base Experience: 62  
Weight: 95.00  
View Delete  
johnny\_bay's Profile  
Pokemon not available

friendly message asking the user if they're user they want to remove all pokemon associations (this happens when clicking "Remove All Pokemons")

### Checklist Items (0)

Successfully removed all Pokemons!

### My Pokemons

Remove all Pokemons

| Pokemon Name | Pokemon Base Experience | Pokemon Weight | Sort   | Order | Limit |
|--------------|-------------------------|----------------|--------|-------|-------|
| Pokemon Name | Pokemon Base Experience | Pokemon Weight | "name" | "+"   | 10    |

Filter Clear

Results: 0/0  
No results to show

success message when confirming that you want to remove all pokemons, and associations are removed

### Checklist Items (0)

Successfully removed all Pokemons!

### My Pokemons

Remove all Pokemons

| Pokemon Name | Pokemon Base Experience | Pokemon Weight | Sort   | Order | Limit |
|--------------|-------------------------|----------------|--------|-------|-------|
| ivysaur      | Pokemon Base Experience | Pokemon Weight | "name" | "+"   | 10    |

[Filter](#)[Clear](#)

Results: 0/2

No results to show

no results

## Checklist Items (0)

## 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 the user's associations (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      | 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  | 1      | Show the logic for deleting all associations for the user (this may be admin-only but should be present for the specific role)  |
| #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)   |
| #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 (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

```

DPLORER   ...
  ● pokemons.php   ● purchase_pokemon.php   ● pokemon_associations.php   ● delete_relationship.php   ● my_pokemons.php X   ● pokemon_card.php   ● assign_roles.php   ● assign_entries.php   +○
  ✓ TUS-IT202-008
  ✓ public_html
    ✓ Project
      ✓ admin
        ✓ api
          ● purchase_pokemon...
            ✓ sql
              ■ 001_create_table_us...
              ■ 002_alter_table_user...
              ■ 003_create_table_rel...
              ■ 004_create_table_us...
              ■ 005_insert_poke_id...
              ■ 006_create_table_po...
              ■ 007_alter_table_pk...
              ■ 008_create_table_us...
              ■ init_db.php
            ⓘ README.md
            ● available_pokemons...
            ⓘ dont use
            ⓘ helpers.js
            ● home.php
            ● index.php
            ● login.php
            ● logout.php
            ● my_pokemons.php
            ● pokemon_info.json
            ● pokemon.php
            ● pokemons.php
            ● profile.php
            ● register.php
            ● styles.css
            ● testApi.php
            ● index.php
            # placeholder_file
            ⓘ README.ind
            # test_db.php
            # test_file.php
            ↳ IT202-project-prep...
            ⓘ .gitignore
            ⓘ .htaccess
            ⓘ composer.json
            ⓘ composer.lock
            # partials - ShortcutLink
            ⓘ Profile
            # public_html - ShortcutLink
            ⓘ README.ind
            > OUTLINE
            > TIMELINE
            ↳ my_pokemons.php

  public_html > Project > my_pokemons.php ...
  You 5 days ago | Author (You)
  1  $query = "DELETE FROM `IT202_524-UserPokemons` WHERE user_id = :user_id";
  2  //note we need to go up 1 more directory
  3  require(__DIR__ . "/../../partials/nav.php");
  4  $db = getDB();
  5
  6  //remove all associations
  7  if(isset($_GET["remove"])){
  8    $query = "DELETE FROM `IT202_524-UserPokemons` WHERE user_id = :user_id";
  9
  10   try{
  11     $stmt = $db->prepare($query);
  12     $stmt->execute(["user_id"=>get_user_id()]);
  13     flash("Successfully removed all Pokemons!", "success");
  14   }catch(PDOException $e){
  15     error_log("Error removing Pokemon associations". var_export($e, true));
  16     flash("Error removing Pokemon associations", "danger");
  17   }
  18
  19   redirect("my_pokemons.php");
  20 } <- 07:19 If(isset($_GET["remove"]))
  21
  22 //build search form
  23 $form = [
  24   ["type" => "text", "name" => "name", "placeholder" => "Pokemon Name", "label" => "Pokemon Name", "include_margin"=>false],
  25   ["type" => "number", "name" => "base_experience", "placeholder" => "Pokemon Base Experience", "label" => "Pokemon Base Experience", "include_margin"=>false],
  26   ["type" => "number", "name" => "weight", "placeholder" => "Pokemon Weight", "label" => "Pokemon Weight", "include_margin"=>false],
  27   {"type" => "select", "name" => "sort", "label" => "Sort", "options" => ["name" => "name", "base_experience" => "base_experience", "weight" => "weight"], "include_margin"=>false},
  28   {"type" => "number", "name" => "order", "label" => "Order", "options" => ["asc" => "+", "desc" => "-"], "include_margin"=>false}
  29 ];
  30 <- 07:21 $form =
  31 />error_Log("Form data: ". var_export($form, true));
  32
  33 $total_records = get_total_count("IT202_524_Pokemon" p
  34 JOIN "IT202_524-UserPokemons" up ON p.id = up.poke_id
  35 WHERE user_id = :user_id", [":user_id"=>get_user_id()]);
  36
  37 $query = "SELECT username, p.id, name, base_experience, weight, user_id FROM `IT202_524_Pokemon` p
  38 JOIN `IT202_524-UserPokemons` up ON p.id = up.poke_id LEFT JOIN Users u ON u.id = up.user_id
  39 WHERE user_id = :user_id";
  40 $params = [":user_id"=>get_user_id()];
  41
  42 if (count($_GET) > 0) {
  43   $keys = array_keys($_GET);
  44
  45   foreach ($form as $k => $v) {
  46     if (in_array($v["name"], $keys)) {
  47       $form[$k]["value"] = $_GET[$v["name"]];
  48     }
  49   }
  50 } <- add-50 foreach ($form as $k => $v)
  51   />/name
  52   $name = se($_GET, "name", "", false);
  53   if(!empty($name))
  
```

## my\_pokemons.php (including queries, filter/sort, code for remove all, and the button for it) p1

### Checklist Items (0)

```

DPLORER   ...
  ● pokemons.php   ● purchase_pokemon.php   ● pokemon_associations.php   ● delete_relationship.php   ● my_pokemons.php X   ● pokemon_card.php   ● assign_roles.php
  ✓ TUS-IT202-008
  ✓ public_html
    ✓ Project
      ✓ admin
        ✓ api
          ● purchase_pokemon...
            ✓ sql
              ■ 001_create_table_us...
              ■ 002_alter_table_user...
              ■ 003_create_table_rel...
              ■ 004_create_table_us...
              ■ 005_insert_poke_id...
              ■ 006_create_table_po...
              ■ 007_alter_table_pk...
              ■ 008_create_table_us...
              ■ init_db.php
            ⓘ README.md
            ● available_pokemons...
            ⓘ dont use
            ⓘ helpers.js
            ● home.php
            ● index.php
            ● login.php
            ● logout.php
            ● my_pokemons.php
            ● pokemon_info.json
            ● pokemon.php
            ● pokemons.php
            ● profile.php
            ● register.php
            ● styles.css
            ● testApi.php
            ● index.php
            # placeholder_file
            ⓘ README.ind
            # test_db.php
            # test_file.php
            ↳ IT202-project-prep...
            ⓘ .gitignore
            ⓘ .htaccess
            ⓘ composer.json
            ⓘ composer.lock
            # partials - ShortcutLink
            ⓘ Profile
            # public_html - ShortcutLink
            ⓘ README.ind
            > OUTLINE
            > TIMELINE
            ↳ my_pokemons.php

  public_html > Project > my_pokemons.php ...
  You 5 days ago | Author (You)
  50   $query .= " AND name like :name";
  51   $params["name"] = "%$name%";
  52
  53   //base experience
  54   $base_experience = se($_GET, "base_experience", "", false);
  55   if(empty($base_experience) && $base_experience > 1){
  56     $query .= " AND base_experience = :base_experience";
  57     $params["base_experience"] = $base_experience;
  58
  59   //weight
  60   $weight = se($_GET, "weight", "", false);
  61   if(empty($weight) && $weight > 1){
  62     $query .= " AND weight = :weight";
  63     $params["weight"] = $weight;
  64
  65   //sort and order
  66   $sort = se($_GET, "sort", "", false);
  67   if(in_array($sort, ["name", "base_experience", "weight"])){
  68     $sort = "base_experience";
  69   }
  70   //tell my sql I care about the data from table "p"
  71   if($sort === "base_experience" || $sort === "weight"){
  72     $sort = "p.id . $sort";
  73   }
  74   $order = se($_GET, "order", "", false);
  75   if(in_array($order, ["asc", "desc"])){
  76     $order = "desc";
  77   }
  78   //IMPORTANT make sure you fully validate/test $sort and $order (sql injection possibility)
  79   $query .= " ORDER BY $sort $order";
  80
  81   try{
  82     $stmt = $db->prepare($query);
  83     $results = [];
  84     try {
  85       $stmt->execute($params);
  86       $r = $stmt->fetchAll();
  87       if ($r) {
  88         $results = $r;
  89       }
  90     } catch (PDOException $e){
  91       error_log("Error fetching pokemons". var_export($e, true));
  92       flash("Unhandled Error Occurred", "danger");
  93     }
  94   } <- add-93 If(count($_GET) > 0)
  95
  96   $stmt = $db->prepare($query);
  97   $results = [];
  98   try {
  99     $stmt->execute($params);
  100     $r = $stmt->fetchAll();
  101     if ($r) {
  102       $results = $r;
  103     }
  104   } catch (PDOException $e){
  105     error_log("Error fetching pokemons". var_export($e, true));
  106     flash("Unhandled Error Occurred", "danger");
  107   }
  
```

## my\_pokemons.php (including queries, filter/sort, code for remove all, and the button for it) p2

## Checklist Items (0)

my\_pokemons.php (including queries, filter/sort, code for remove all, and the button for it) p3

## Checklist Items (0)

## code for how count is done in db\_helpers.php

### Checklist Items (0)



```

<?php
    You 5 hours ago | 1 author (You)
    </div>
    <div class="card-body">
        <p>You 5 hours ago | 1 author (You)</p>
        <?php
            if (!isset($pokemons)) {
                error_log("Using Pokemom partial without data");
                flash("Dev Alert: Pokemom called without data", "danger");
            }
        </?php>
        <?php if (isset($pokemons)) : ?>
            <!-- https://upload.wikimedia.org/wikipedia/commons/5/53/Pok%C3%A9MON_ball_icon.svg -->
            <div class="card m-0 w-100" style="width: 100px;">
                <?php if (isset($pokemons["username"])) : ?>
                    <div class="card-header">
                        Owned by: <?php echo $pokemons["username"]; ?>
                    </div>
                <?php endif; ?>
                <img alt="https://upload.wikimedia.org/wikipedia/commons/5/53/Pok%C3%A9MON_ball_icon.svg" class="card-img-top" alt="..." data-bbox="130 150 200 180"/>
                <div class="card-title"><?php echo $pokemons["name"]; ?> ( <?php echo $pokemons["id"]; ?> )</div>
                <div class="card-text">
                    <ul class="list-group">
                        <li class="list-group-item">Name: <?php echo $pokemons["name"]; ?></li>
                        <li class="list-group-item">Base Experience: <?php echo $pokemons["base_experience"]; ?></li>
                        <li class="list-group-item">Height: <?php echo $pokemons["height"]; ?></li>
                    </ul>
                </div>
                <div class="card-body">
                    <?php if (isset($pokemons["id"])) : ?>
                        <a href="#" class="btn btn-secondary" href="<?php echo get_url('purchase_pokemon.php?id=' . $pokemons["id"]); ?>">View</a>
                    <?php endif; ?>
                    <br/>
                    <div class="d-flex justify-content-end">
                        <?php if ($pokemons["user_id"] == $user_id) || $pokemons["user_id"] === "N/A" : ?>
                            <a href="#" class="btn btn-primary" href="<?php echo get_url('admin/delete_relationship.php?poke_id=' . $pokemons["id"] . '&user_id=' . $pokemons["user_id"]); ?>">click="confirm('Are you sure?')";>Delete</a>
                        <?php endif; ?>
                    </div>
                </div>
            </div>
        </?php>
    </div>
    <div class="card-body">
        <?php if (isset($pokemons["user_id"])) || $pokemons["user_id"] === "N/A" : ?>
            <a href="#" class="btn btn-secondary" href="<?php echo get_url('admin/delete_relationship.php?poke_id=' . $pokemons["id"] . '&user_id=' . $pokemons["user_id"]); ?>">click="confirm('Are you sure?')";>Delete</a>
        <?php endif; ?>
        <br/>
        <?php if (isset($pokemons["user_id"])) || $pokemons["user_id"] === "N/A" : ?>
            <?php
                $id = isset($pokemons["id"]) ? $pokemons["id"] : (isset($_GET["id"]) ? $_GET["id"] : -1);
                <br/>
                <a href="#" class="btn btn-link" href="<?php echo get_url('purchase_pokemon.php?poke_id=' . $id); ?>">Purchase Pokemom</a>
            <?php else : ?>
                <a href="#" class="btn btn-link" href="<?php echo get_url('profile.php?id=' . $pokemons["user_id"]); ?>"><?php echo $pokemons["username"]; ?> Profile</a>
            <?php endif; ?>
        </div>
    </div>
</?php>

```

## code for what's shown in view and delete (includes buttons)

### Checklist Items (0)



```

EXPLORER ... pokemons.php purchase_pokemon.php pokemon_associations.php delete_relationship.php my_pokemons.php pokemon_card.php assign_r...
TJ3-IT202... D+ E+ U public_html > Project > admin > delete_relationship.php > ...
    You 5 hours ago | 1 author (You)
    <?php
        session_start();
        require(__DIR__ . "/../../../../lib/functions.php");
        if (!has_role("Admin")) {
            flash("You don't have permission to view this page", "warning");
            redirect("home.php");
        }
        if(isset($_GET["user_id"]) && isset($_GET["poke_id"])){
            $user_id = $_GET["user_id"];
            $poke_id = $_GET["poke_id"];
            $query = "DELETE FROM `IT202-524-UserPokemons` WHERE user_id = :user_id AND poke_id = :poke_id";
            $db = getDB();
            $stmt = $db->prepare($query);
            $stmt->execute([":user_id" => $user_id, ":poke_id" => $poke_id]);
            flash("Successfully removed your Pokemon!", "success");
        }catch(PDOException $e){
            error_log("Error removing Pokemon association: " . var_export($e, true));
            flash("Error removing Pokemon association", "danger");
        }
    } <- #18-23 if(isset($_GET["user_id"]) && isset($_GET["poke_id"]))
    redirect("my_pokemons.php");
    You 5 hours ago + Handling Data Association
    >
    > M4

```

code that I used for delete relationship

## Checklist Items (0)



```
partials > result_counts.php > ...
You, 7 days ago | 1 author (You)
1 <?php You, 7 days ago + Pokemon Associations
2 if(!isset($result_count)){
3     $message = "Dev Note: result_count not set";
4     error_log($message);
5     flash($message);
6     $result_count = -1;
7 } <- #2-7 if(!isset($result_count))
8 if(!isset($total_count)){
9     $message = "Dev Note: total_count not set";
10    error_log($message);
11    flash($message);
12    $total_count = -1;
13 } <- #8-13 if(!isset($total_count))
14 ?>
15 <header>
16     <div class="row">
17         <div class="col-1">Results: <?php se("$result_count/$total_count"); ?></div>
18     </div>
19 </header>
```

code for results

## 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) |
| #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:

So with the association logic, the association between users and the Pokemons they're associated with would be established and recorded in the UserPokemons table. The query would obtain the Pokemon associate user by joining the UserPokemons table with the Pokemon table based on the ID. This would make it so only pokemon associated with users are fetched, which also ensures that users only see their own pokemon in "my\_pokemon." With the filters,

the script uses \$form to search through the UserPokemon and filter it based on the criteria of name, base experience, and weight. When the form is submitted, the script check for filter values and then they're added to the query to narrow down the search results. After that query fetches the relevant data, it would be displayed in a grid-like format with my pokemon cards, and above it. If the filters comes out with no matching records, then a message of "no results to show" would be displayed on the screen.

\$result\_count is the number that represents the number of pokemon cards thats being displayed. By the logic, you can see that it updates based on the filters that are applied. In "my\_pokemon.php", you can see that the result\_count is used as render\_result\_counts, and inside it counts the results that were fetched from the query. With total\_records, it represents the total number of pokemon cards overall, regardless of if theyre visible or not. This is obtained by querying the db to count the number of records that meet the criteria, without applying the filters. The brains behind this comes from get\_total\_count in the db\_helpers screenshot. This shows the table refs using regex to ensure that valid table names are included in the query. There are parameters set and a query is executed to fetch the result. If a result is obtained the total count would be extracted and returned as an integer. You can see get\_total\_count used in my\_pokemon with a query inside of the parenthesis and then rendered at the bottom with result\_count so that it can retrieve the total count in that database and display it to the user.

At the bottom of the my\_pokemon code, you can see that "remove" is called, which indicates that the user wants to remove all of the association. This is because the scripts check for the remove parameter in \$\_GET. Once remove is present, it would construct a query to Delete all the associations from the UserPokemons table where the user\_id matches the current user's id. The user\_id would be retrieved by using the get\_user\_id() function. If everything goes through successfully, it will do its job and delete all of the associations to that user and redirect you to the my\_pokemons page. With a single delete, I placed that button in the pokemon\_card code since it had to be there. The code for single delete is in delete\_relationship, where it will first check if the user is an admin to even proceed with this. If so, it will continue and the script will check if the user\_id and poke\_id parameters are present. If so, the script would query for the association between the user and the pokemon to be deleted from the UsersPokemons table. Once that's done and everything runs smoothly, the user would be met with a success message and redirected to the my\_pokemons page. If not, they'll receive a flash message stating that something went wrong.

#### Task #4 - Points: 1

Text: Add related links

#### Checklist

\*The checkboxes are for your own tracking

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

#### URL #1

[https://tlj3-prod-5077e065f26e.herokuapp.com/Project/my\\_pokemons.php](https://tlj3-prod-5077e065f26e.herokuapp.com/Project/my_pokemons.php)

#### URL #2

<https://github.com/TiaraJenks/tlj3-it202-008/pull/59>

[COLLAPSE ▲](#)



[COLLAPSE ▲](#)

## Task #1 - Points: 1

Text: Screenshots of this page

### Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details   |
|----|--------|---|
| #1 | 1      | Show the summary of the results with relevant information per entity                                      |
| #2 | 1      | Show the single view buttons/links, delete button/links   |
| #3 | 1      | Show the username related to the specific entity and that it's clickable                                  |
| #4 | 1      | Show variations of the number of shown items count and show the count of total number of associated items |
| #5 | 1      | Show variations of the filter/sort including no results (proper message should be visible)                |
| #6 | 1      | Make sure the heroku dev url is visible in the address bar  |
| #7 | 1      | Clearly caption screenshots   |

### Task Screenshots:

#### Gallery Style: Large View

Small

Medium

Large

The screenshot shows a web application interface for managing Pokemons. At the top, there's a navigation bar with links like Dashboard, CyberCorp, Cybersecurity school, Tiara Jenkins-Amak..., Internships, Schedule Building, Cisco Skills For All, Food, Juniper Learning Po..., and All Bookmarks. Below the navigation is a sub-navigation menu for "Pokemon NFT!": User, Profile, Pokemons, My Pokemons, Available, Roles, and Poke Management.

The main content area is titled "Associated Pokemons". It features a search bar with fields for Username and Pokemon Name, and a dropdown for Sort and Order. A "Filter" button is also present.

Below the search bar, there are four cards, each representing a different Pokemon:

- Wartortle (6)**: Name: Wartortle, Base Experience: 100, Weight: 112.00. Buttons: View, Delete, Update Profile.
- Glimmera (23)**: Name: glimmera, Base Experience: 184, Weight: 400.00. Buttons: View, Delete, Update Profile.
- Chatot (22)**: Name: chatot, Base Experience: 162, Weight: 19.00. Buttons: View, Delete, Update Profile.
- Charmander (25)**: Name: charmander, Base Experience: 62, Weight: 55.00. Buttons: View, Delete, Update Profile.

Each card also includes a note at the bottom stating "Pokemons not available".

summary of each user and their entities, the delete and view buttons, the name each entity is associated with, the count, and the filter/sort for username, poke name, poke base experience, , poke sort, order, and limit.

## Checklist Items (0)

The screenshot shows a web application interface for managing Pokémon associations. At the top, there is a navigation bar with links like Dashboard, CyberCorp, Cybersecurity schol..., Tiara Jenkins-Amak..., Internships, Schedule Building, Cisco Skills For All, Food, Juniper Learning Po..., and All Bookmarks. Below the navigation bar, there is a header with the text "Associated Pokemons".

There are several input fields for filtering: "Username" (set to "bobby"), "Pokemon Name" (empty), "Pokemon Base Experience" (empty), "Pokemon Weight" (empty), "Sort" (set to "Name"), and "Order" (set to "asc").

The results section displays two entries, both titled "Owned by: bobby":

- chatot (22)**
  - Name: chatot
  - Base Experience: 144
  - Weight: 19.00

[View](#) [Delete](#) [Details](#)  
[Profile](#) [Pokemon not available](#)
- Writer (8)**
  - Name: Writer
  - Base Experience: 100
  - Weight: 112.00

[View](#) [Delete](#) [Details](#)  
[Profile](#) [Pokemon not available](#)

At the bottom left, it says "Results: 2/5".

filter/sort and count variation 1

## Checklist Items (0)

The screenshot shows a web application interface for managing Pokémon associations, similar to the previous one. At the top, there is a navigation bar with links like Dashboard, CyberCorp, Cybersecurity schol..., Tiara Jenkins-Amak..., Internships, Schedule Building, Cisco Skills For All, Food, Juniper Learning Po..., and All Bookmarks. Below the navigation bar, there is a header with the text "Associated Pokemons".

There are several input fields for filtering: "Username" (set to "stuart1"), "Pokemon Name" (empty), "Pokemon Base Experience" (empty), "Pokemon Weight" (empty), "Sort" (set to "Name"), and "Order" (set to "asc").

The results section displays one entry, titled "Owned by: stuart1":

- glimmora (28)**
  - Name: glimmora
  - Base Experience: 184
  - Weight: 48.00

[View](#) [Delete](#) [Details](#)  
[Profile](#) [Pokemon not available](#)

At the bottom left, it says "Results: 1/5".

## filter/sort and count variation 2

### Checklist Items (0)

The screenshot shows a web browser window with a URL starting with `tij3-dev-b3a210191069.herokuapp.com/Project//admin/pokemon_associations.php?username=&name=orthworm&base_experience=&weight=`. The page title is "Associated Pokemons". There are search and filter fields for Username, Pokemon Name, Pokemon Base Experience, Pokemon Weight, Sort, and Order. The results table has columns for Username, Pokemon Name, Pokemon Base Experience, Pokemon Weight, Sort, and Order. One result is shown: "orthworm (38)" with name "orthworm", base experience 240, weight 3100.00, sort "Name", and order "ASC". Buttons for "View" and "Delete" are present, along with a link to "Johnny\_boy's profile".

## filter/sort and count variation 3

### Checklist Items (0)

The screenshot shows a web browser window with a URL starting with `tij3-dev-b3a210191069.herokuapp.com/Project//admin/pokemon_associations.php?username=&name=charmander&base_experience=&weight=95&s...`. The page title is "Associated Pokemons". There are search and filter fields for Username, Pokemon Name, Pokemon Base Experience, Pokemon Weight, Sort, and Order. The results table has columns for Username, Pokemon Name, Pokemon Base Experience, Pokemon Weight, Sort, and Order. One result is shown: "charmander (25)" with name "charmander", base experience 62, weight 95.00, sort "Name", and order "ASC". Buttons for "View" and "Delete" are present, along with a link to "Johnny\_boy's profile".

Pokémon not available

## filter/sort and count variation 4

### Checklist Items (0)

Associated Pokemons

| Owner | Pokemon Name    | Pokemon Base Experience                                   | Pokemon Weight | Sort   | Order |
|-------|-----------------|---|----------------|--------|-------|
| tj3   | orthworm (38)   | Name: orthworm<br>Base Experience: 240<br>Weight: 1160.00 |                | "name" | "asc" |
| tj3   | glimmora (23)   | Name: glimmora<br>Base Experience: 104<br>Weight: 450.00  |                |        |       |
| tj3   | Whirlie (6)     | Name: Whirlie<br>Base Experience: 100<br>Weight: 112.00   |                |        |       |
| tj3   | charmander (25) | Name: charmander<br>Base Experience: 62<br>Weight: 91.00  |                |        |       |

Results: 3/5

No results to show

## filter/sort and count variation 5

### Checklist Items (0)

Associated Pokemons

| Owner | Pokemon Name | Pokemon Base Experience | Pokemon Weight | Sort   | Order |
|-------|--------------|-------------------------|----------------|--------|-------|
| jim   |              |                         |                | "name" | "asc" |

Results: 0/5

No results to show

no results

## Checklist Items (0)

## 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 associations (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      | Each record should have a username field that is clickable to go to the user's profile page   |
| #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 |
| #6  | 1      | Show the logic related to the count of all associated items (even the ones not shown 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**

```
 ⑤  public/Net/Project/Admin | ~gozumon/associations.php > ...
    ⑥      1  You're too aged | author /root
    ⑦          1  c1ghp
    ⑧      2  //note we need to go up 1 more directory
    ⑨      3  require(__DIR__ . "/../../../../partials/nav.php");
    ⑩      4
    ⑪      5  if($user['role'] == 'admin') {
    ⑫          6  flash("You don't have permission to view this page", "warning")
    ⑬      7  
```

**poke\_association** 1 (the following 3 poke association screenshots consists of everything in checkbox 1, 2, 7, and 8)

## Checklist Items (0)

```
✓ TUS-MAN_ [5] ✘ ⌂ public_html>Project>admin>Pokemon_associations.php ...  
+ lib  
  - .env  
  - api_helper.php  
  - config.php  
  - db_helpers.php  
  - db.php  
  - duplicate_user_details.php  
  - flash_messages.php  
  - functions.php  
  - get_user.php  
  - load_api_keys.php  
  - pokemon_api.php  
  - README.html  
  - redirect.php  
  - render_functions.php  
  - reset_session.php  
  - save_edu.php  
  - sanitizers.php  
  - user_helpers.php  
+ partials  
  - buttons.php  
  - flash.php  
  - input_validation.php  
  - nav.php  
  - pokemon_card.php  
  - README.html  
  - result_counts.php  
  - table.php  
+ public_html  
  - .htaccess  
  - .htpasswd  
  - .htaccess  
+ Project  
  - admin  
    - assign_entities.php  
    - assign_roles.php  
    - create_pokemon.php  
    - create_role.php  
    - delete_pokemon.php  
    - delete_relationship.php  
    - edit_pokemon.php  
    - list_pokemons.php  
    - list_roles.php  
    + pokemon_associations.php  
    - view_pokemon.php  
+ api  
+ OUTPUT  
+ TESTING  
107
```

```
54     $params["base_experience"] = $base_experience;  
55   }  
56   //weight  
57   $weight = $_GET["weight"] ?? "", false);  
58   if(!empty($weight) && $weight > 1){  
59     $query .= " AND weight = :weight";  
60     $params["weight"] = $weight;  
61   }  
62   //sort and order  
63   $sort = $_GET["sort"] ?? "", false);  
64   if(in_array($sort, ["name", "base_experience", "weight"])){  
65     $sort = "base_experience";  
66   }  
67   //tell my sql i care about the data from table "p"  
68   if($sort === "base_experience" || $sort === "weight"){  
69     $sort = "p." . $sort;  
70   }  
71   $order = $_GET["order"] ?? "", false);  
72   if(in_array($order, ["asc", "desc"])){  
73     $order = "desc";  
74   }  
75   //IMPORTANT make sure you fully validate/trust $sort and $order (sql injection possibility)  
76   $query .= " ORDER BY $sort $order";  
77  
78   try{  
79     $limit = (int) $_GET["limit"] ?? 10, false);  
80   catch(\Exception $e){  
81     $limit = 10;  
82   }  
83   //IMPORTANT make sure you fully validate/trust $limit (sql injection possibility)  
84   $query .= " LIMIT $limit";  
85  
86   } <- 450-06 IF (count($_GET) > 0)  
87  
88   $db = getDB();  
89   $stmt = $db->prepare($query);  
90   $results = [];  
91   try {  
92     $stmt->execute($params);  
93     $r = $stmt->fetchAll();  
94     if ($r) {  
95       $results = $r;  
96     }  
97   } catch (\PDOException $e) {  
98     error_log("Error fetching pokemons" . var_export($e, true));  
99     flash("Unhanlded Error Occurred", "danger");  
100   }  
101  
102   $table = ["data" => $results, "title" => "List of Pokemons", "ignored_columns" => ["id"], "view_url" => get_url("pokemon.p  
103   }  
104   div class="container-fluid">  
105     <h3>Associated Pokemons</h3>  
106     <form method="GET">  
107       <div class="row mb-3" style="align-items: flex-end;">
```

### **poke\_association**

### Checklist Items (0)

```
api_helper.php
config.php
db_helpers.php
db.php
duplicate_user_details.php
flash_messages.php
functions.php
get_url.php
load_api_keys.php
pokemon_api.php
 README.md
 redirect.php
 render_functions.php
 reset_session.php
 safer_echo.php
 sanitizers.php
 user_helpers.php
partials
button.php
flash.php
input_field.php
nav.php
pokemon_card.php
 README.md
result_counts.php
table.php
public_html
> hm2
> M4
Project

105 <?php
106     <h3>Associated Pokemons</h3>
107     <form method="GET">
108         <div class="row mb-3" style="align-items: flex-end;">
109             <?php foreach ($form as $k => $v):?>
110                 <div class="col-2">
111                     <?php render_input($v);?>
112                 </div>
113             </?php endforeach;?>
114         </div>
115         <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
116         <a href="?clear" class="btn btn-secondary">Clear</a>
117     </form>
118     <?php render_result_counts(count($results), $total_records);?>
119     <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 row-cols-lg-4 row-cols-xl-5 row-cols-xxl-6 g-4">
120         <?php foreach($results as $pokemon):?>
121             <div class="col">
122                 <?php render_pokemon_card($pokemon);?>
123             </div>
124         <?php endforeach;?>
125         <?php if(count($results) === 0):?>
126             <div class="col">
127                 No results to show
128             </div>
129         <?php endif;?>
130     </div>
131 </div>
132
133
134 <?php
135 //note we need to go up 1 more directory
136 require_once(__DIR__ . '../../../../../partials/flash.php');
137 ?>
138     <?php #24-138 $total_records = get_total_count
```

## **poke\_association**

## Checklist Items (0)

this shows content in checkbox 3-

## Checklist Items (0)

this shows the content in check box

## Checklist Items (0)

TLJ3-IT202...

partials > result\_counts.php > ...

You, 7 days ago | 1 author (You)

```
1 <?php You, 7 days ago * Pokemon Associations
2 if(!isset($result_count)){
3     $message = "Dev Note: result_count not set";
4     error_log($message);
5     flash($message);
6     $result_count = -1;
7 } <- #2-7 if(!isset($result_count))
8 if(!isset($total_count)){
9     $message = "Dev Note: total_count not set";
10    error_log($message);
11    flash($message);
12    $total_count = -1;
13 } <- #8-13 if(!isset($total_count))
14 ?>
15 <header>
16     <div class="row">
17         <div class="col-1">Results: <?php se("$result_count/$total_count"); ?></div>
18     </div>
19 </header>
```

shows content in checkbox 7

## Checklist Items (0)

 COLLAPSE

### 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   |
| #4 | 1      | Mention the logic for handling the username requirements                              |

#### Response:

The result list is retrieved the same way as mentioned in the previous explanation, but just all done from the Pokemon\_associations page. The association between the users and pokemons they're associated with would be established in the UserPokemons table. The query would obtain the pokemon associated user by joining the UserPokemons table with the Pokemon table based on the ID. This makes it so only the associated pokemons with users are fetched. With the filters, it uses \$form to do the search and filter through UserPokemon based on the criteria given in name, base experience, and weight. Once that form is submitted, it would check for filter values in pokemon\_association and then query it to narrow down the results. After that's done, it would display all of the data that matched the fetch criteria in a grid-like form with my pokemon cards. If the filter goes through the query and no records match, then it would print out "No results to show."

The logic behind result\_count and total count are the same as mentioned in the previous explanation, but all done in the pokemon\_association page. The result count and total count would be in the parenthesis of the "render\_result\_count," where it would count the \$results after it has gone through the filters/sorting. The total count is displayed at the top using the get\_total\_count function from the db\_helpers code(uses sanitization by using regex, includes a query to aid in obtaining the total count, and handles results so it will print out the total count that's extracted from the query provided inside the total\_count parenthesis in poke\_association) with a query inside that will be used to count the total amount of pokemon cards in the dataset.

The logic for single delete is the same as mentioned previously since it is placed on the pokemon card. The code for single delete is in delete\_relationship, where it will check if you are an admin before doing the deletion. If you are, it proceeds with the script checking if the user\_id and poke\_id parameters are present. If so, it would be queried for the association between the user and the pokemon to be deleted from the UserPokemons table. Once that's done and everything runs smoothly, the user would be prompted with an "are you sure" message to ensure that is what they want to do, and if they say yes, it will complete the delete and redirect them to their pokemons page. The delete button is linked to the pokemon card, where it will show up everywhere besides in "Available\_Pokemons.php."

The username requirements are basically handle just like the filter/sort. When the username of a user is inputted, the script would modify the query to include a condition that would filter the results based on that specific username. In the query, there is a JOIN for the User table (JOIN Users u on u.id = up.user\_id), which would ensure that the associated user's username is made available to filter. Once a user name is inputted and submitted, a query is used to bind that username value to the query. After executing the query, the script would fetch the results for the pokemons associated with that username and display it. If there's no results listed under that username, then it will print "No

results to show."

#### Task #4 - Points: 1

Text: Add related links

##### Checklist

\*The checkboxes are for your own tracking

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

##### URL #1

[https://tlj3-prod-5077e065f26e.herokuapp.com/Project/admin/pokemon\\_associations.php](https://tlj3-prod-5077e065f26e.herokuapp.com/Project/admin/pokemon_associations.php)

##### URL #2

<https://github.com/TiaraJenks/tlj3-it202-008/pull/63>

#### Unassociated Page (2 pts.)

COLLAPSE

#### Task #1 - Points: 1

Text: Screenshots of this page

##### Checklist

\*The checkboxes are for your own tracking

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

## Task Screenshots:

### Gallery Style: Large View

Small

Medium

Large

This screenshot shows a 'Large' view of the application's content area. It displays a grid of 8 items, each represented by a Pokéball icon. Below each icon is a small card containing the name of the Pokémon and some numerical values. The names visible are mew, ivysaur, defairy, ditto, pikachu, and charizard. Each card also includes a 'View' and a 'Purchase' button.

Pokémon Name: mew (18)  
Name: mew  
Base Experience: 200  
Weight: 40.00  
[View](#) [Purchase](#)

Pokémon Name: ivysaur (20)  
Name: ivysaur  
Base Experience: 142  
Weight: 130.00  
[View](#) [Purchase](#)

Pokémon Name: defairy (21)  
Name: defairy  
Base Experience: 113  
Weight: 75.00  
[View](#) [Purchase](#)

Pokémon Name: ditto (26)  
Name: ditto  
Base Experience: 101  
Weight: 40.00  
[View](#) [Purchase](#)

Pokémon Name: pikachu (30)  
Name: pikachu  
Base Experience: 112  
Weight: 80.00  
[View](#) [Purchase](#)

Pokémon Name: charizard (36)  
Name: charizard  
Base Experience: 267  
Weight: 165.00  
[View](#) [Purchase](#)

this shows the content in checkbox 1, 2, and 3

### Checklist Items (0)

This screenshot shows a 'Large' view of the application's content area. It displays a grid of 2 items, each represented by a Pokéball icon. Below each icon is a small card containing the name of the Pokémon and some numerical values. The names visible are pikachu and charizard. Each card also includes a 'View' and a 'Purchase' button.

Pokémon Name: pikachu (30)  
Name: pikachu  
Base Experience: 112  
Weight: 80.00  
[View](#) [Purchase](#)

Pokémon Name: charizard (36)  
Name: charizard  
Base Experience: 267  
Weight: 165.00  
[View](#) [Purchase](#)

## filter/sort and count variations 1 (checkbox 3 too)

### Checklist Items (0)

A screenshot of a web browser displaying a search results page for 'rayquaza'. The URL in the address bar is [tlj3-dev-b3a210191069.herokuapp.com/Project//available\\_pokemons.php?name=&base\\_experience=340&weight=&sort=name&order=asc&limit=10](http://tlj3-dev-b3a210191069.herokuapp.com/Project//available_pokemons.php?name=&base_experience=340&weight=&sort=name&order=asc&limit=10). The page title is 'Pokemon NFT1'. The search results show one entry for 'rayquaza (40)'. The entry includes the name, base experience (340), weight (2063.00), and two buttons: 'View' and 'Purchase Pokemon'.

## filter/sort and count variations 2

### Checklist Items (0)

A screenshot of a web browser displaying a search results page for 'clefairy'. The URL in the address bar is [tlj3-dev-b3a210191069.herokuapp.com/Project//available\\_pokemons.php?name=&base\\_experience=&weight=75&sort=name&order=asc&limit=10](http://tlj3-dev-b3a210191069.herokuapp.com/Project//available_pokemons.php?name=&base_experience=&weight=75&sort=name&order=asc&limit=10). The page title is 'Pokemon NFT1'. The search results show one entry for 'clefairy (21)'. The entry includes the name, base experience (112), weight (75.00), and two buttons: 'View' and 'Purchase Pokemon'.

## filter/sort and count variations 3

### Checklist Items (0)

The screenshot shows a web browser window with the URL [https://tj3-dev-b3a210191069.herokuapp.com/Project//available\\_pokemons.php?name=bob&base\\_experience=&weight=&sort=name&order=asc...](https://tj3-dev-b3a210191069.herokuapp.com/Project//available_pokemons.php?name=bob&base_experience=&weight=&sort=name&order=asc...). The page title is "Pokemon NFT". The main content area is titled "Available Pokemons" and includes a search form with fields for "Pokemon Name" (containing "bob"), "Pokemon Base Experience", "Pokemon Weight", "Sort" (set to "name"), "Order" (set to "asc"), and "Limit" (set to 10). Below the search form, there are "Title" and "Clear" buttons. A message at the bottom left says "Results(0) No results to show".

no results (checkbox 4)

### Checklist Items (0)

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

|    |   |   |
|----|---|---|
| #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 pokemons 1** (the first 3 screenshots will display the content in checkbox 1, 2, 4, 6)

## Checklist Items (0)

## Available\_pokemons

## Checklist Items (0)

TLJ3-IT202... [ ] ⌂ ⌂ ⌂ ⌂

- public\_html
- Project
- admin
  - create\_pokemon.php
  - create\_role.php
  - delete\_pokemon.php
  - delete\_relationship.php
  - edit\_pokemon.php
  - list\_pokemons.php
  - list\_roles.php
  - pokemon\_association.php
  - view\_pokemon.php
- api
  - purchase\_pokemon.php
- sql
  - 001\_create\_table\_user.php
  - 002\_alter\_table\_user.php
  - 003\_create\_table\_role.php
  - 004\_create\_table\_us...
  - 005\_insert\_roles\_admin.php
  - 006\_create\_table\_pokemon.php
  - 007\_alter\_table\_pokemon.php
  - 008\_create\_table\_us...
- init\_db.php
- README.md

available\_pokemons.php

```
public_html > Project > available_pokemons.php > ...
97     <div class="container-fluid">
98         <form method="GET">
99             <div class="row mb-3" style="align-items: flex-end;">
100                 </div>
101                 <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
102                 <a href="?clear" class="btn btn-secondary">Clear</a>
103             </div>
104             <?php render_result_counts(count($results), $total_records); ?>
105             <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 row-cols-lg-4 rows-cols-xl-5 rows-cols-xxl-6 g-4" style="margin-top: 20px;">
106                 <?php foreach($results as $pokemon):?>
107                     <div class="col">
108                         <?php render_pokemon_card($pokemon); ?>
109                     </div>
110                 <?php endforeach; ?>
111                 <?php if(count($results) === 0):?>
112                     <div class="col">
113                         No results to show
114                     </div>
115                 <?php endif; ?>
116             </div>
117         </div>
118     <?php
119     //note we need to go up 1 more directory
120     require_once(__DIR__ . "/../../../../partials/flash.php");
121 ?>
```

## Available\_pokemons

## Checklist Items (0)

```
    $query = "SELECT count(1) as totalCount FROM $table_refs";
    $stmt = $db->prepare($query);
    foreach ($params as $key => $value) {
        $stmt->bindValue("$key", $value);
        error_log("binding value for :$key: $value");
    }
    $stmt->execute();
    $r = $stmt->fetch();
    if ($r) {
        return (int)$r["totalCount"];
    }
    return 0;
}
catch(PDOException $e){
    error_log("Error getting count for ". var_export($query, true) . ":" . var_export($e, true));
    flash("Error getting count", "danger");
}
return -1;
}
```

code related to count and total count (checkbox 4 and 5)

## Checklist Items (0)

[view link \(checkbox 3\)](#)

## Checklist Items (0)

### Task #3 - Points: 1

### **Text: Explain the solution**

## Checklist

\*The checkboxes are for your own tracking

|    |   |  |
|----|---|--|
| #1 | 1 | Mention how you determine the result list (include the unassociated logic and filters) |
| #2 | 1 | Mention the logic behind the two counts (total items and visible result count)         |

#### Response:

The result list is retrieved the same way as mentioned in the previous explanation, but just all done from the Pokemon\_associations page. The association between the users and pokemons they're associated with would be established in the UserPokemons table. However, the query being used here has a WHERE and NOT IN operator so that it can fetch the pokemons that have no associations with a user at all. The query would use the UserPokemons table to sift out the pokemons that are associated to a user from the Pokemon table, and you'll be left with all unassociated pokemons from the Pokemon table and now those are the only entities that will be fetched when using filters. With the filters, it uses \$form to do the search and filter through the Pokemon table based on the criteria given in name, base experience, and weight. Once that form is submitted, it would check for filter values in available\_pokemon and then query it to narrow down the results. After that's done, it would display all of the unassociated data that matched the fetch criteria in a grid-like form with my pokemon cards. If the filter goes through the query and no records match, then it would print out "No results to show."

The logic behind result\_count and total count are the same as mentioned in the previous explanation, but all done in the available\_pokemon page. The result count and total count would be in the parenthesis of the "render\_result\_count," where it would count the \$results after it has gone through the filters/sorting. The total count is displayed at the top using the get\_total\_count function from the db\_helpers code(uses sanitization by using regex, includes a query to aid in obtaining the total count, and handles results so it will print out the total count that's extracted from the query provided inside the total\_count parenthesis in poke\_association) with a query inside that will be used to count the total amount of available/unassociated pokemon cards in the dataset. This query is different than the other ones in the separate sections since this will be querying from the Pokemons using WHERE and NOT IN constraints to sift out the associated pokemons with the use of the UserPokemons table. This is so it will only be focusing on the result count and total count of the unassociated pokemons, not the associated ones.

#### Task #4 - Points: 1

Text: Add related links

#### Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details   |
|----|--------|---|
| #1 | 1      | Include the heroku prod link for the page that creates the association  |
| #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://tlj3-prod-5077e065f26e.herokuapp.com/Project/available\\_pokemons.php](https://tlj3-prod-5077e065f26e.herokuapp.com/Project/available_pokemons.php)

#### URL #2

<https://github.com/TiaraJenks/tlj3-it202-008/pull/65>

**COLLAPSE**



**COLLAPSE**

## Task #1 - Points: 1

**Text: Screenshots of the page**

### Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details   |
|----|--------|---|
| #1 | 1      | Show the search form with valid data                                    |
| #2 | 1      | Show the results of the search  |
| #3 | 1      | Show the result of entities and users being associated and unassociated |
| #4 | 1      | Make sure the heroku dev url is visible in the address bar              |
| #5 | 1      | Clearly caption screenshots   |

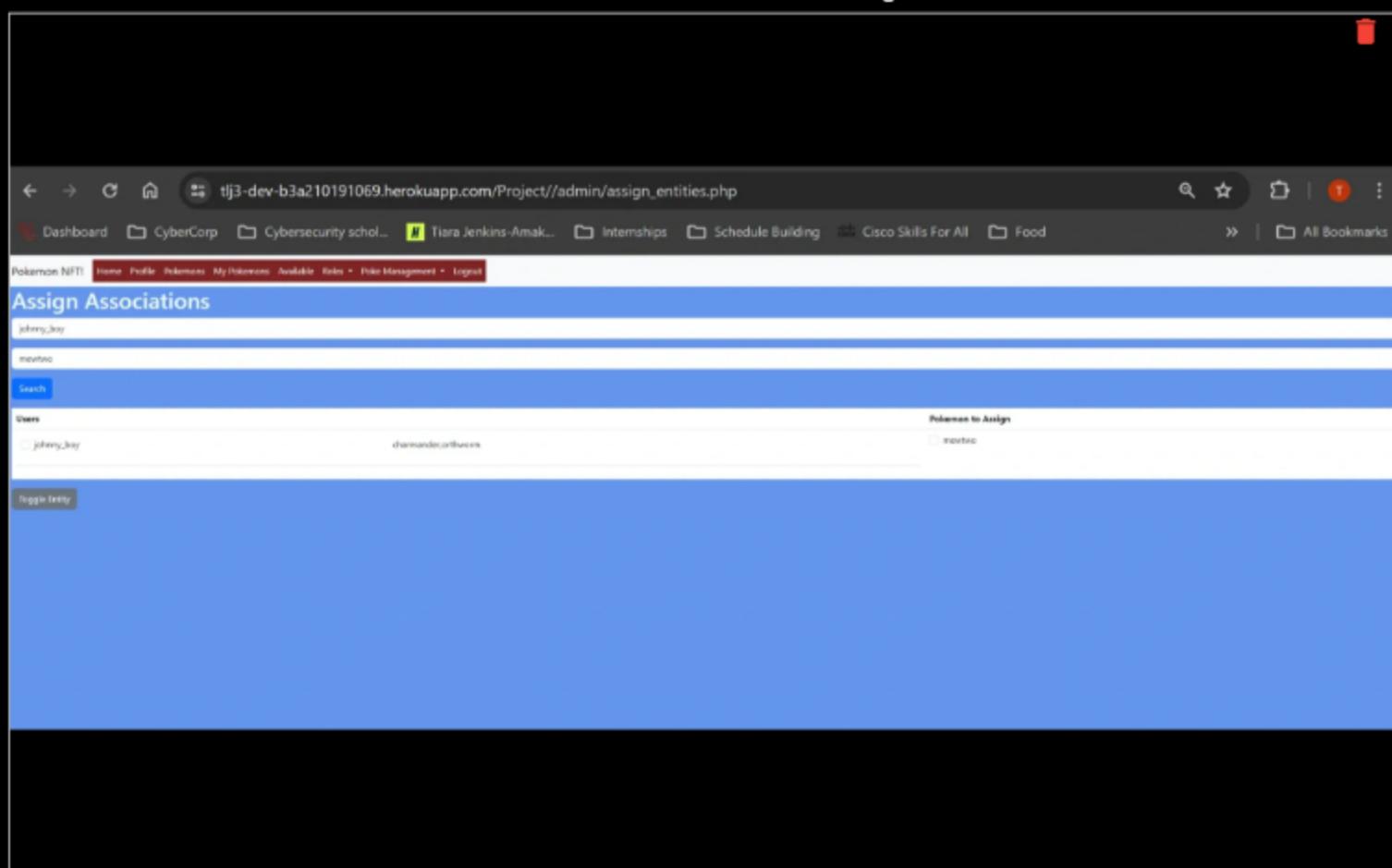
### Task Screenshots:

**Gallery Style: Large View**

**Small**

**Medium**

**Large**



this screenshot shows the content for checkbox 1 and 2

## Checklist Items (0)

Upgraded Association

### Assign Associations

Users

- johnny\_boy
- charmander.orthworm.mewtwo

Pokemon to Assign

- charmander.orthworm.mewtwo

Toggle Entity

associating an entity (mewtwo)

## Checklist Items (0)

Deleted association

### Assign Associations

Users

- johnny\_boy
- charmander.orthworm.mewtwo

Pokemon to Assign

- charmander.orthworm.mewtwo

Toggle Entity

unassociating an entity (orthworm)

## Checklist Items (0)



## Task #2 - Points: 1

## **Text: Screenshots of the code**

## Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details   |
|----|--------|---|
| #1 | 1      | Search form field for finding a partial match of usernames  |
| #2 | 1      | Search form field for finding a partial match of entities   |
| #3 | 1      | Code related to getting a max of 25 results for each field (i.e., 25 users and 25 entities limit)   |
| #4 | 1      | Code that generates the checkboxes next to each list (users and entities)   |
| #5 | 1      | Code related to submitting the checkbox lists   |
| #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) |
| #7 | 1      | Include ucid/date comments for each code screenshot   |
| #8 | 1      | Clearly caption screenshots   |

## Task Screenshots:

## Gallery Style: Large View

**Small      Medium      Large**

```
public_html>Project>admin>assign_entities.php >...
```

You, 3 days ago | 1 author (raw)

```
1 //php
2 // Note: We need to go up 3 more directory
3 require_once(__DIR__ . "/../../../../partials/nav.php");
4
5 if (!has_role("Admin")) {
6     flash("You don't have permission to view this page", "warning");
7     redirect("home.php");
8 }
9
10 // attempt to apply associations
11 if (isset($_POST["users"]) && isset($_POST["poke"])) {
12     $user_ids = $_POST["users"];
13     $pokemon_ids = $_POST["poke"];
14
15     if (empty($user_ids) || empty($pokemon_ids)) {
16         flash("Both users and pokemons need to be selected", "warning");
17     } else {
18         // For simplicity, this will be a tad inefficient
19         $db = getDB();
20         $stmt = $db->prepare("INSERT INTO `IT202-524-UserPokemon` (user_id, poke_id) VALUES (:uid, :pid)");
21         foreach ($user_ids as $uid) {
22             foreach ($pokemon_ids as $pid) {
23                 try {
24                     $stmt->execute([":uid" => $uid, ":pid" => $pid]);
25                     flash("Updated Association", "success");
26                 } catch (PDOException $e) {
27                     //flash("An error occurred: " . $e->getMessage(), true, "danger");
28                     $stmt2 = $db->prepare("DELETE FROM `IT202-524-UserPokemon` WHERE user_id = :uid AND poke_id = :pid");
29                     $stmt2->execute([":uid" => $uid, ":pid" => $pid]);
30                     flash("Deleted association", "success");
31                 }
32             }
33         }
34     }
35 }
36
37 // Search for users by username
38 $users = [];
39 $username = "";
40 if (isset($_POST["username"])) {
41     $username = trim($_POST["username"], "", false);
42 }
```

```
    if (empty($username)) {
        $db = getDB();
        $stmt = $db->prepare("SELECT uid, username,
        (SELECT GROUP_CONCAT(p.name)
         FROM ITI202_524_UserPokes AS p
         WHERE up.user_id = uid AND p.uid = up.poke_id) AS poke_name
        FROM users u WHERE username LIKE :username LIMIT 25");
        try {
            $stmt->execute([":username" => "%$username%"]);
            $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
            if ($results) {
```

**assign\_entities** 1 (each screenshot fulfills all checkbox requirements)

## Checklist Items (0)

## assign\_entities

## Checklist Items (0)

```
tree pokemonsession
└── view_pokemon.php
    └── purchase_pokemon...
        └── sql
            ├── 001_create_table_us...
            ├── 002_alter_table_us...
            ├── 003_create_table_ud...
            ├── 004_create_table_us...
            └── 005_insert_roles_id...
```

## assign\_entities 3

### Checklist Items (0)

#### Task #3 - Points: 1

Text: Explain the solution

#### Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details  |
|----|--------|--|
| #1 | 1      | Describe the steps for the search and how it works for users and entities                        |
| #2 | 1      | Mention how you built the UI for this  |
| #3 | 1      | Describe the steps for the associate/unassociate logic for the combination of users and entities |

#### Response:

When searching for both the user and the entity(in this case, it would be pokemons), I put form fields for both, with the username field being named Username Search, and pokemon field being named Pokemon Search. In order to submit the form, I put an if statement under both to ensure that both fields are filled in prior to submitting since that is an important factor to fetch both user and entity. Once that form is submitted, the server would receive it and they both would go through a query. With the username query, it specifies to search for users whose usernames match the one that was submitted in the form with the use of the LIKE operator. With the pokemon query, it specifies to search for pokemon whose names match the one that was submitted in the form with the use of the LIKE operator. Once it confirms that data in the db matches the ones that were submitted in the form, it would execute those queries. If they don't match the ones in the db, the user would be flashed with a message indicating that the username and/or pokemon doesn't exist within the database. If there are results from the query, they are fetched and stored in their designated arrays (username would be stored in \$users, and pokemon would be stored in \$pokemon). The script would then display both of those results in the designated areas (user under Users, and pokemon under Pokemon to Assign), with a checkbox beside both of them, which allows for selection for association.

This is the way the entire thing works: The form submission is handled using the POST method. With the input sanitization (prevents SQL injections), it would use the se() function. When interacting with the database, the queries would be using PDO to do that. For flashing error messages, I used flash and error\_log so it can show the problem on the screen for the user, and inside the terminal. When handling the association, the selected and submitted ones are processed by inserting the records into the UserPokemons table. If it already exists (unique constraint), it's deleted before the reinsertion. Once everything is done, the user would be flashed with a success message, or if something goes wrong, they'll be flashed with a friendly message stating something went wrong.

The UI for this was pretty simple to create. All I added were the 2 form fields for username and pokemon search. Then

I added a blue search button that would submit that form for the server/script to query and fetch the requested data. Then I had a table with a side for the user search results (left side) and a side for the pokemon search results (right). The user search result side places the results from the search, which is username of the user and the pokemon they're associated with under the title "User." The pokemon result search side places the results from the search, which is the name of the pokemon under the title "Pokemon to Assign." That result was queried from the Pokemon table, and the username information was queried from the UserPokemon table. Lastly, I added a button called "Toggle Entity," which is used to associate and unassociate the users and the pokemons that was searched in the form. Also, when a association or unassociation is done after clicking the Toggle Entity button, a flash message will show on the ui that tells the user that it was successful.

First I would go over the association logic. When the user submits the form of the username and pokemon of choice, it would go through the script, where it validates if both fields have the users and pokemon selected. If not, nothing will occur and the association won't be done. If it passes the validation, then the script will try to insert the selected user and pokemon combination into the UserPokemons table, with the use of an SQL statement. Once its done and completed, the user would be flashed with a success message. For the unassociation logic, it would go through the same process as the association logic, however, instead of adding it into the UserPokemons table, it will be extracting/deleting the association from that table, where it uses "DELETE FROM" in the SQL statement instead of "INSERT INTO" like the SQL statement used in the association logic

#### Task #4 - Points: 1

Text: Add related links

##### Checklist

\*The checkboxes are for your own tracking

| #  | Points | Details   |
|----|--------|---|
| #1 | 1      | Include the heroku prod link for the page that creates the association  |
| #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://tlj3-prod-5077e065f26e.herokuapp.com/Project/admin/assign\\_entities.php](https://tlj3-prod-5077e065f26e.herokuapp.com/Project/admin/assign_entities.php)

##### URL #2

<https://github.com/TiaraJenks/tlj3-it202-008/pull/62>

#### 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

The screenshot shows a GitHub project board titled "IT202 S2024 Project". The board has three columns: "Todo", "In Progress", and "Done".

- Todo:** This item hasn't been started.
- In Progress:** This is actively being worked on.
- Done:** This has been completed.
  - 163-4202-000 404 MG2-Edit Data Page
  - 163-4202-000 410 MG2-Delete Handling
  - 163-4202-000 416 MG2-API Handling
  - 163-4202-000 418 MG3 - API Data Association
  - 163-4202-000 419 MG3 - Handle the association of data to a user
  - 163-4202-000 419 MG3 - Logged in user's associated entities page
  - 163-4202-000 471 MG3 - All Users association page
  - 163-4202-000 472 MG3 - Create a page that shows data not associated with any user
  - 163-4202-000 473 MG4 - Admin can associate any entity with any users

+ Add item      + Add item      + Add item

project board

Task #2 - Points: 1

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

URL #1

<https://github.com/users/TiaraJenks/projects/2/views/1>

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

To be honest, I didn't face that many issues in this milestone. The biggest one was with the assigning entities in the admin section. It had a lot to do with the queries and how I set them up. I also had a small issue with deleting individual pokemons. The link was the main problem, but with a little tweaking and help from the prof, it works out pretty well in the end. I'm happy with how I did in this milestone and I learned a lot.

Task #4 - Points: 1

[COLLAPSE](#)

## TASK #4 - POINTS: 1

Text: WakaTime Screenshot

### Details:

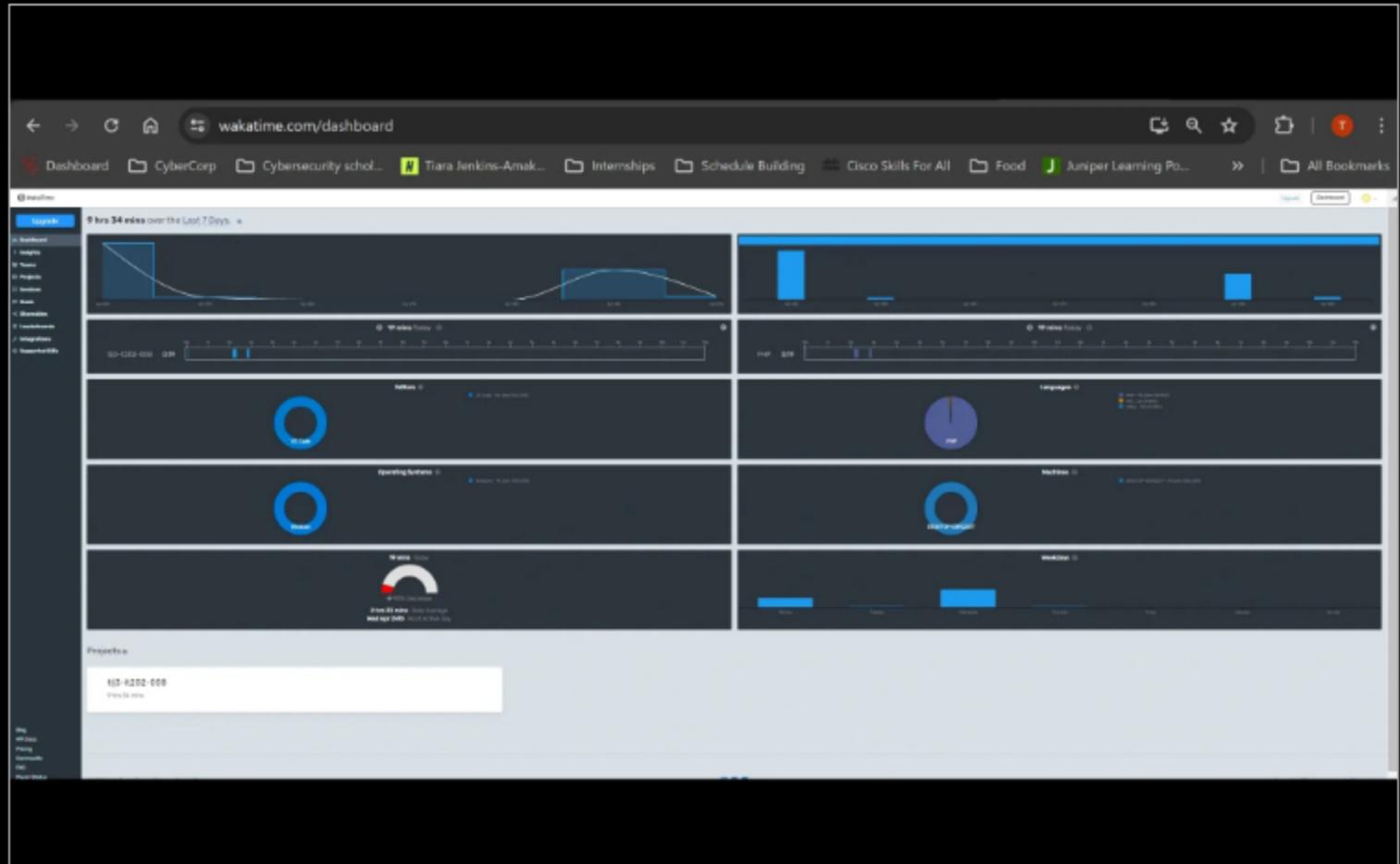
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



waka screenshot

End of Assignment