

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-3-2024/grade/rr42>

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

Submissions:

Submission Selection

1 Submission [active] 4/30/2024 10:00:44 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



 Task #2 - Points: 1

[^ EXPAND ^](#)

Text: Updating Entities

Handle Data Association (1 pt.)

[COLLAPSE](#)

Task #1 - Points: 1

Text: Screenshots of the code

[EXPAND](#)

Task #2 - Points: 1

Text: Screenshot of the association table(s)

[EXPAND](#)

Task #3 - Points: 1

Text: Explain solution

[EXPAND](#)

Task #4 - Points: 1

Text: Add related links

[EXPAND](#)

Current User's Association Page (2 pts.)

[COLLAPSE](#)

Task #1 - Points: 1

Text: Screenshots of this page

[EXPAND](#)

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 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 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

The screenshot shows a dark-themed web application interface titled "My Favorites". At the top, there are three size selection buttons: "Small", "Medium", and "Large". Below the buttons is a browser header with the URL "it202-rr42-dev-fb744d733bee.herokuapp.com/Project/favorites.php". The main content area has a heading "My Favorites" and a sub-heading "Total favorites: 4". A "Display Limit" input field with the value "10" and a blue "Apply" button are present. Four items are listed in a grid:

- Views** by Drake (Artist: Drake) with a "Remove Favorite" button.
- Under Pressure** by Logic (Artist: Logic) with a "Remove Favorite" button.
- Bounce** by Logic (Artist: Logic) with a "Remove Favorite" button.
- Trance** by Metro Boomin, Travis Scott & Young Thug (Artist: Metro Boomin, Travis Scott & Young Thug) with a "Remove Favorite" button.

A red "Remove All Favorites" button is located at the bottom left of the item grid. The bottom of the page features a dark bar with the text "My favorites (favorites.php)".

Checklist Items (3)

#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

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

This screenshot shows the same application interface as the previous one, but with a lower "Display Limit" of 2. Only the first two items from the previous list are visible:

- Views** by Drake (Artist: Drake) with a "Remove Favorite" button.
- Under Pressure** by Logic (Artist: Logic) with a "Remove Favorite" button.

The "Total favorites: 4" and "2 items displayed" counts are still present at the top. The bottom of the page features a dark bar with the text "My favorites (favorites.php)".

[Artist: Drake](#)[Remove Favorite](#)[Artist: Logic](#)[Remove Favorite](#)[Remove All Favorites](#)

My favorites (favorites.php) (continued)

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 type="checkbox"/> #1	1	Show the code related to fetching the user's associations (including the query)
<input type="checkbox"/> #2	1	Show the code related to the display of the results
<input type="checkbox"/> #3	1	Each record should have a button/link for single view
<input 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 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 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 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 type="checkbox"/> #8	1	Show the logic related to filter/sort (limit should be constrained to 1-100 otherwise default to 10)
<input type="checkbox"/> #9	1	Include ucid/date comments for each code screenshot
<input type="checkbox"/> #10	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

 favorites.php M X

```
public_html > Project >  favorites.php > ...
You, 1 second ago | 1 author (You)
1 <?php //rr42 4/30/2024
2 require(__DIR__ . "/../../partials/nav.php");
3 require( DIR . "/../../partials/flash.php");
```



```

4     is_logged_in();
5
6     $userFavorites = getUserFavorites(get_user_id());
7
8     $limit = isset($_GET['limit']) ? $_GET['limit'] : 10;
9     if (!is_numeric($limit) || $limit < 1 || $limit > 100) {
10         flash("Invalid limit value. Please enter a number between 1 and 100.", "danger");
11         header("Location: {$_SERVER['PHP_SELF']}");
12         exit;
13     }
14
15     $favoriteSongs = [];
16     $displayedItemCount = 0;
17     foreach ($userFavorites as $songId) {
18         if ($displayedItemCount >= $limit) {
19             break;
20         }
21
22         $stmt = $db->prepare("SELECT * FROM SONGS WHERE label = ?");
23         $stmt->execute([$songId]);
24         $song = $stmt->fetch(PDO::FETCH_ASSOC);
25         if ($song) {
26             $favoriteSongs[] = $song;
27             $displayedItemCount++;
28         }
29     }
30
31     $totalSongs = count($userFavorites);

```

favorites.php

Checklist Items (3)

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

#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)

```

32 <div class="container-fluid"> <br><br>4/28/2024-->
33     <h1>My Favorites</h1>
34     <div class="row">
35         <div class="col-md-12">
36             <div>
37                 <total_favorites><php echo $totalSongs; </></p>
38                 <displayedItemCount><php echo $displayedItemCount; ></p>
39             </div>
40         </div>
41     </div>
42
43     <div class="row">
44         <div class="col-md-12">
45             <form method="POST">
46                 <label for="limit">Display limit:</label>
47                 <input type="number" id="limit" name="limit" value="<php echo $limit; >" min="1" max="100">
48                 <button type="submit" class="btn btn-primary">Apply</button>
49             </form>
50         </div>
51     </div>
52
53     <div class="row">
54         <php foreach ($favoriteSongs as $song) : ?>
55             <div class="col-md-3">
56                 <div class="card shadow-sm">
57                     <a href="#"><php echo urlencode($song['title']); ></a>
58                     
59                     <div class="card-body">
60                         <h3 class="card-title"><php echo htmlspecialchars($song['title']); ></h3>
61                         <p class="card-text">Artist: <php echo htmlspecialchars($song['artist']); ></p>
62                     </div>
63                     <form method="POST">
64                         <input type="hidden" name="user_id" value="<php echo get_user_id(); >">
65                         <input type="hidden" name="song_label" value="<php echo $song['label']; >">
66                         <button type="submit" class="btn btn-danger" onclick="return confirm('Are you sure you want to remove this favorite?')">Remove Favorite</button>
67                     </form>
68                 </div>
69             </php endforeach; ?>
70         </div>
71
72         <div class="row">
73             <div class="col-md-12">
74                 <form method="POST">
75                     <input type="hidden" name="remove_all" value="true">
76                     <button type="submit" class="btn btn-danger" onclick="return confirm('Are you sure you want to remove all favorites?')">Remove All Favorites</button>
77                 </form>
78             </div>
79         </div>
80     </div>
81
82     <php
83     if (isset($_POST['remove_all']) && $_POST['remove_all'] === "true") {
84         removeAllFavorites(get_user_id());
85         flash("All favorites removed successfully", "success");
86         header("refresh:0");
87     }
88
89     if (($_SERVER['REQUEST_METHOD'] == "POST" && isset($_POST['user_id']) && isset($_POST['song_label'])) &&
90         $user_id == $_POST['user_id'] && $song_label == $_POST['song_label']) {
91         $songId = $_POST['song_label'];
92         removeFavorite($user_id, $songId);
93         flash("Favorite removed successfully", "success");
94         header("refresh:0");
95     }
96 >
```

Checklist Items (4)

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

#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

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

```

favorites_functions.php M X
lib > favorites_functions.php > ...
...
1 | <?php //rr42 4/30/2024
2 | function addToFavorites($userId, $songId){
3 |     $db = getDB();
4 |     $stmt = $db->prepare("INSERT INTO UserSongs (user_id, song_label) VALUES (?, ?)");
5 |     $stmt->execute([$userId, $songId]);
6 |
7 |
8 | function removeFromFavorites($userId, $songId){
9 |     $db = getDB();
10 |    $stmt = $db->prepare("DELETE FROM UserSongs WHERE user_id = ? AND song_label = ?");
11 |    $stmt->execute([$userId, $songId]);
12 |
13 |
14 | function getUserFavorites($userId){
15 |     $db = getDB();
16 |     $stmt = $db->prepare("SELECT song_label FROM UserSongs WHERE user_id = ?");
17 |     $stmt->execute([$userId]);
18 |     $favorites = $stmt->fetchAll(PDO::FETCH_COLUMN);
19 |     return $favorites;
20 |
21 |
22 | function removeAllFavorites($userId){
23 |     $db = getDB();
24 |     $stmt = $db->prepare("DELETE FROM UserSongs WHERE user_id = ?");
25 |     $stmt->execute([$userId]);
26 |
27 |?> You, 1 second ago * Uncommitted changes

```

favorites_functions.php

Checklist Items (2)

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

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

Task #3 - Points: 1

Text: Explain the solution

#	Points	Details
<input type="checkbox"/> #1	1	Mention how you determine the result list (include the association logic and filters)
<input type="checkbox"/> #2	1	Mention the logic behind the two counts (total items and visible result count)
<input type="checkbox"/> #3	1	Mention the logic for single delete and delete all associations for the specific user

Response:

The users favorites are fetched by using a SELECT statement to get all record in the UserSongs table where the user_id matches the user id of the current logged in user.

The total favorites associated with the user is found by counting the total number of entries fetched from the database. The items displayed count gets incremented each time a favorite song is added to the list of songs to display until it hits the display limit or runs out of available favorited songs.

The single delete function (removeFromFavorites) takes in the user's id and the song id to be deleted and removes the relevant record with a DELETE statement. The multi-delete for all record of the user function (removeAllFavorites) takes the user's id as a parameter and removes all records where the user id matches with a DELETE statement.

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://it202-rr42-prod-dc703459f313.herokuapp.com/Project/favorites.php>

URL #2

<https://github.com/rRahmanM6/rr42-it202-008/pull/70>

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

[COLLAPSE](#)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

#	Points	Details

#	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

Title	Artist	Total Users	Favorited By	Action
Venmo	Drake	2	abc,m42	<button>Delete</button>
Under Pressure	Logic	1	m42	<button>Delete</button>
Trance	Metro Boomin, Travis Scott & Young Thug	1	m42	<button>Delete</button>

All favorites page (admin page)

Checklist Items (4)

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

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

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

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

The screenshot shows a web application interface for managing favorite items. At the top, there's a navigation bar with links like 'Search', 'Profile', 'Favorites', 'Create', 'Role List', 'Roles Assign', 'Roles View', 'Database', 'Create Song', 'All Favorites', 'New Association', 'Non-Associated', and 'Logout'. Below the navigation, the title 'All Favorites' is displayed. Underneath it, a message says 'Total number of items: 5'. There are two input fields: 'Records per page:' containing '0' and 'Filter by username:' containing 'zzzz'. A blue button labeled 'Apply Filter' is visible. The main area shows a table header with columns 'Title', 'Artist', 'Total Users', 'Favorited By', and 'Action'. Below the header, a message says 'No matching records found'.

All favorites page (admin page) (continued)

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 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

#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**

allFavorites.php

Checklist Items (2)

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

#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)

```
allviewsonphp N X

public_html > Project > admin > allviewsonphp > html > body

91 <html lang="en">
92   <head> <!-- class="d-flex justify-content-between" -->
93     <meta class="container-fluid">
94       <div class="row" style="height: 100%; align-items: center; justify-content: center; margin-top: 10px;">
95         <div class="col-12 col-md-6" style="text-align: center; width: 100%; height: 100%; position: relative;">
96           <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 1;">
97             <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 2;">
98               <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 3;">
99                 <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 4;">
100                   <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 5;">
101                     <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 6;">
102                       <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 7;">
103                         <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 8;">
104                           <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 9;">
105                             <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 10;">
106                               <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 11;">
107                                 <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 12;">
108                                   <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 13;">
109                                     <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 14;">
110                                       <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 15;">
111                                         <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 16;">
112                                           <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 17;">
113                                             <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 18;">
114                                               <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 19;">
115                                                 <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 20;">
116                                                   <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 21;">
117                                                     <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 22;">
118                                                       <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 23;">
119                                                         <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 24;">
120                                                           <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 25;">
121                                                             <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 26;">
122                                                               <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 27;">
123                                                                 <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 28;">
124                                                                   <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 29;">
125                                                                     <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 30;">
126                                                                       <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 31;">
127                                                                         <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 32;">
128                                                                           <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 33;">
129                                                                             <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 34;">
130                                                                               <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 35;">
131                                                                                 <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 36;">
132                                                                                   <div class="row" style="width: 100%; height: 100%; position: absolute; top: 0; left: 0; background-color: black; opacity: 0.5; z-index: 37;">
133                                                                                     <div class="col-12 col-md-6" style="margin: auto; position: relative; z-index: 38;">
134                       </div>
135                     </div>
136                   </div>
137                 </div>
138               </div>
139             </div>
140           </div>
141         </div>
142       </div>
143     </div>
144   </div>
145 </body>
146 </html>
```

```

56 <form method="GET" class="form-inline">
57   <label for="perPage">Records per page:</label>
58   <input type="number" id="perPage" name="perPage" value="50" min="0" max="300" class="form-control mr-2">
59   <label for="username">Filter by username:</label>
60   <input type="text" id="username" name="username" value="<php echo htmlspecialchars($_GET[username]); >" class="form-control mr-2">
61   <button type="submit" class="btn btn-primary">Apply Filter</button>
62 </form>
63
64 </div>
65
66 <div class="row mt-4">
67   <div class="col-md-12">
68     <table class="table table-striped">
69       <thead>
70         <tr>
71           <th><a href="#">Song title</a></th>
72           <th><a href="#">Artist</a></th>
73           <th><a href="#">Total users</a></th>
74           <th>Favorited By</th>
75           <th>Actions</th>
76         </tr>
77       </thead>
78       <tbody>
79         <tr>
80           <td><?php if(count($favoriteSongs) > 0) : ?>
81             <?php foreach ($favoriteSongs as $song) : ?>
82               <td>
83                 <a href="<?php echo url('search.php') . '?song=' . urlencode($song['title']) . '&' . $song['artist']; ?>"><?php echo htmlspecialchars($song['title']); ?></a>
84                 <?php echo $song['total_users']; ?></td>
85                 <?php echo htmlspecialchars($song['favorited_by']); ?></td>
86               <td>
87                 <form method="POST" style="display: inline-block;">
88                   <input type="hidden" name="delete_song" value="<?php echo $song['id']; ?>">
89                   <button type="submit" class="btn btn-danger" onclick="return confirm('Are you sure you want to delete this favorite?')">Delete</button>
90                 </form>
91               </td>
92             </?php endforeach; ?>
93           <?php else : ?>
94             <td colspan="5">No matching records found!</td>
95           </tr>
96           <tr>
97             <td colspan="5" style="text-align: center; font-weight: bold;"><?php endif; ?>

```

I did not make the username clickable to go to the user's profile because I was not sure if any user should be able to go to the profile.php we have already created where the fields are editable. Or if I should create a brand new separate profile page where the profile info is not editable.

Checklist Items (5)

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

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

#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

#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)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Mention how you determine the result list (include the association logic and filters)
<input type="checkbox"/> #2	1	Mention the logic behind the two counts (total items and visible result count)
<input type="checkbox"/> #3	1	Mention the logic for single delete
<input type="checkbox"/> #4	1	Mention the logic for handling the username requirements

Response:

The list is fetched from the UserSongs table. After fetching the list, a concatenated list of usernames who have favorited each song is also generated. If the username filter is applied, the record only with at least a partial match of the username column are displayed, and a button to delete all displayed associations is shown. The list can also be ordered by title, artist name, or total number of users who have favorited.

The total number of items count just displays the total number of songs in the UserSongs table. The displayed items count increments as songs are displayed to the page.

The single delete function (deleteFavorite) takes the song label as a parameter which is a unique value generated for each song in the SONGS table. It uses a DELETE statement to delete the requested song.

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://it202-rr42-prod-dc703459f313.herokuapp.com/Project/admin/allFavorites.php>

URL #2

<https://github.com/rRahmanM6/rr42-it202-008/pull/72>

Unassociated Page (2 pts.)

[COLLAPSE](#)

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
<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
<input checked="" type="checkbox"/> #4	1	Show variations of the filter/sort including no results (proper error handling)

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

The screenshot shows a web browser window with the URL <http://it202-rr42-dev-fb744d733bee.herokuapp.com/>. The page title is "No Association". It displays a table with three rows of song data:

Title	Artist
Do Not Disturb	Drake
Power Trip	Yeat
How I Been	YoungBoy Never Broke Again

Below the table, a large yellow banner displays the text "No Association".

Checklist Items (3)

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

#2 Show the single view buttons/links

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

The screenshot shows a web browser window with the URL <http://it202-rr42-dev-fb744d733bee.herokuapp.com/>. The page title is "No Association". At the top, there is a search bar with the placeholder "Search Profile Favorites Create Role List Roles Assign Roles View Database Create Song All Favorites New Association Non-Associated Logout". Below the search bar, a yellow banner displays the text "No matching results found.".

Search:
sdflsdflgdfgfdg

Display Limit:

10

Apply

Title

Artist

[Do Not Disturb](#)

Drake

[Power Trip](#)

Yeat

[How I Been](#)

YoungBoy Never Broke Again

[SAM](#)

Logic

No Association (continued)

Checklist Items (1)

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

The screenshot shows a web browser window with the URL <http://it202-rr42-dev-fb744d733bee.herokuapp.com/>. The page title is "No Association". The search bar contains "logic". The display limit is set to 10. The results table has two columns: "Title" and "Artist". There is one result listed: "SAM" by Logic.

Title	Artist
SAM	Logic

No Association (continued)

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 all unassociated entities (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	Show the logic related to the count of all unassociated items (even the ones not shown in the filtered results)
<input checked="" type="checkbox"/> #5	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"/> #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)
<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**

```

noAssociation.php M X
public_html > Project > admin > noAssociation.php > dw.container-fluid
You, 1 second ago | 1 author (You)
<?php //rr42 4/30/2024
1 require(__DIR__ . "/../../partials/nav.php");
2 require(__DIR__ . "/../../partials/flash.php");
3
4
5 if (!has_role("Admin")) {
6     flash("You don't have permission to view this page", "warning");
7     header("location: $BASE_PATH" . "/search.php");
8     exit;
9 }
10 $db = getDB();
11 $totalStat = $db->query("SELECT COUNT(*) FROM SONGS WHERE label NOT IN (SELECT song_label FROM UserSongs)");
12 $totalNoAssociation = $totalStat->fetchColumn();
13 $limit = isset($_GET['limit']) ? $_GET['limit'] : 10;
14 if (!is_numeric($limit) || $limit < 1 || $limit > 100) {
15     flash("Invalid limit value. Please enter a number between 1 and 100.", "danger");
16     header("location: {$PHP_SELF}");
17     exit;
18 }
19 $search = isset($_GET['search']) ? $_GET['search'] : '';
20 if (empty($search)) {
21     $stmt = $db->prepare("SELECT * FROM SONGS WHERE (title LIKE CONCAT('%', :search, '%') OR artist LIKE CONCAT('%', :search, '%')) AND label NOT IN (SELECT song_label FROM UserSongs) LIMIT :limit");
22     $stmt->bindValue(":search", $search, PDO::PARAM_STR);
23     $stmt->bindValue(":limit", $limit, PDO::PARAM_INT);
24 } else {
25     $stmt = $db->prepare("SELECT * FROM SONGS WHERE label NOT IN (SELECT song_label FROM UserSongs) LIMIT :limit");
26     $stmt->bindValue(":limit", $limit, PDO::PARAM_INT);
27 }
28 $stmt->execute();
29 $noAssociationSongs = $stmt->fetchAll(PDO::FETCH_ASSOC);
30 $displayedRecords = count($noAssociationSongs);
31 if (empty($noAssociationSongs)) {
32     flash("No matching results found.", "warning");
33     header("location: {$PHP_SELF}");
34     exit;
35 }
36 >>

```

noAssociation.php

Checklist Items (4)

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

#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)

#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)

```
38 <div class="container-fluid"> <!--rr42 4/30/2024-->
39 <h1>No Association</h1>
40 <div class="row">
41     <div class="col-md-12">
42         <p>Total records without association: <?php echo $totalNoAssociation; ?></p>
43         <p><?php echo $displayedRecords; ?> records displayed</p>
44     </div>
45 </div>
46 <div class="row">
47     <div class="col-md-12">
48         <form method="GET">
49             <div class="form-group">
50                 <label for="search">Search:</label>
51                 <input type="text" id="search" name="search" value="<?php echo $search; ?>" class="form-control" placeholder="Enter title or artist">
52             </div>
53             <div class="form-group">
54                 <label for="limit">Display limit:</label>
55                 <input type="number" id="limit" name="limit" value="<?php echo $limit; ?>" min="1" max="100" class="form-control">
56             </div>
57             <button type="submit" class="btn btn-primary">Apply</button>
58         </form>
59     </div>
60 </div>
61 <?php if (!empty($noAssociationSongs)) : ?>
62     <div class="row mt-4">
63         <div class="col-md-12">
64             <table class="table table-striped">
65                 <thead>
66                     <tr>
67                         <th>Title</th>
68                         <th>Artist</th>
69                     </tr>
70                 </thead>
71                 <tbody>
72                     <?php foreach ($noAssociationSongs as $song) : ?>
73                         <tr>
74                             <td><a href="<?php echo $BASE_PATH ?>/search.php?song=<?php echo urlencode($song['title']) . ' ' . $song['artist']; ?>"><?php echo htmlspecialchars($song['title']); ?></a></td>
75                             <td><?php echo htmlspecialchars($song['artist']); ?></td>
76                         </tr>
77                     <?php endforeach; ?>
78                 </tbody>
79             </table>
80         </div>
81     </div>
82 <?php endif; ?>
83 </div>
```

noAssociation.php (continued)

Checklist Items (2)

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

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

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 code)

unassociated logic and filters)

#2

1

Mention the logic behind the two counts (total items and visible result count)

Response:

The base logic for the list is a SELECT statement that gets all records in SONGS that are not present in the UserSongs table, meaning they are not favorited by any users. The search filter searches for partial matches in the title and artist columns and uses the base logic as well.

The total records without association count checks the count of total records fetched from the base SELECT statement. The records displayed increments as the records are displayed on the page.

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://it202-rr42-prod-dc703459f313.herokuapp.com/Project/admin/noAssociation.php>

URL #2

<https://github.com/rRahmanM6/rr42-it202-008/pull/74>

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 checked="" type="checkbox"/> #1	1	Show the search form with valid data
<input checked="" 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 checked="" type="checkbox"/> #4	1	Make sure the heroku dev url is visible in the address bar
<input checked="" type="checkbox"/> #5	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a web application interface titled "New Association". At the top, there are search fields for "Search Users:" containing "abc" and "Search Songs:" containing "logic". Below these are sections for "Users" (listing "abc") and "Songs" (listing "Bounce - Logic", "SAM - Logic", and "Under Pressure - Logic"). A green "Apply Association" button is visible. The background is dark, and the overall layout is clean.

My new association page is not fully functioning. While the search function and check boxes fully work, the Apply Association button does not function as of now.

Checklist Items (2)

#1 Show the search form with valid data

#2 Show the results of the search

Task #2 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

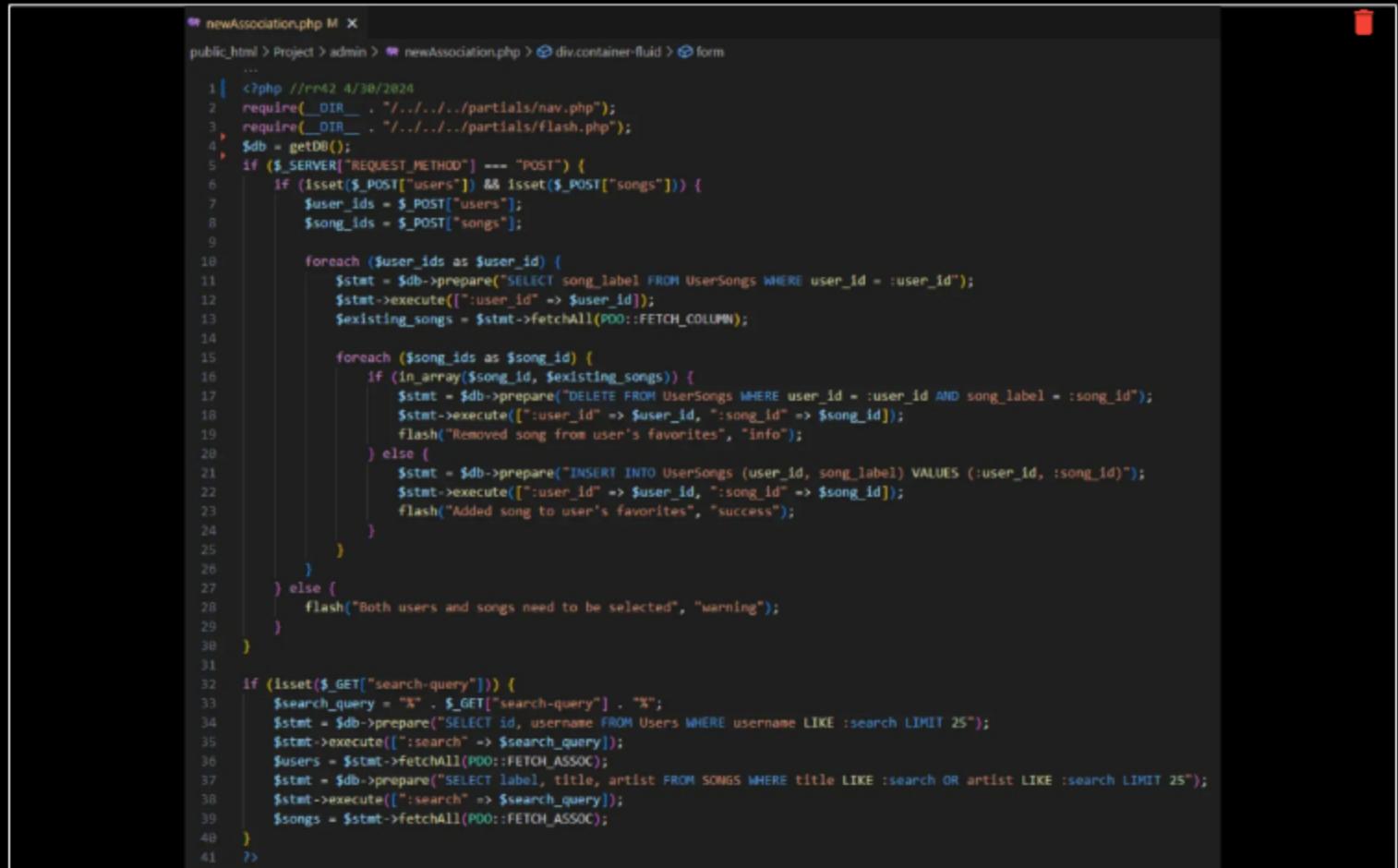
#	Points	Details
<input type="checkbox"/> #1	1	Search form field for finding a partial match of usernames
<input type="checkbox"/> #2	1	Search form field for finding a partial match of entities
<input type="checkbox"/> #3	1	Code related to getting a max of 25 results for each field (i.e., 25 users and 25 entities limit)

<input type="checkbox"/> #4	1	Code that generates the checkboxes next to each list (users and entities)
<input type="checkbox"/> #5	1	Code related to submitting the checkbox lists
<input 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 type="checkbox"/> #7	1	Include ucid/date comments for each code screenshot
<input type="checkbox"/> #8	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



```

newAssociation.php M ✘
public_html > Project > admin > newAssociation.php > div.container-fluid > form
...
1 | <?php //cm62 4/30/2024
2 | require(__DIR__ . "/../../../../partials/nav.php");
3 | require(__DIR__ . "/../../../../partials/flash.php");
4 | $db = getDB();
5 | if ($_SERVER["REQUEST_METHOD"] === "POST") {
6 |     if (isset($_POST["users"]) && isset($_POST["songs"])) {
7 |         $user_ids = $_POST["users"];
8 |         $song_ids = $_POST["songs"];
9 |
10 |         foreach ($user_ids as $user_id) {
11 |             $stmt = $db->prepare("SELECT song_label FROM UserSongs WHERE user_id = :user_id");
12 |             $stmt->execute([":user_id" => $user_id]);
13 |             $existing_songs = $stmt->fetchAll(PDO::FETCH_COLUMN);
14 |
15 |             foreach ($song_ids as $song_id) {
16 |                 if (in_array($song_id, $existing_songs)) {
17 |                     $stmt = $db->prepare("DELETE FROM UserSongs WHERE user_id = :user_id AND song_label = :song_id");
18 |                     $stmt->execute([":user_id" => $user_id, ":song_id" => $song_id]);
19 |                     flash("Removed song from user's favorites", "info");
20 |                 } else {
21 |                     $stmt = $db->prepare("INSERT INTO UserSongs (user_id, song_label) VALUES (:user_id, :song_id)");
22 |                     $stmt->execute([":user_id" => $user_id, ":song_id" => $song_id]);
23 |                     flash("Added song to user's favorites", "success");
24 |                 }
25 |             }
26 |         }
27 |     } else {
28 |         flash("Both users and songs need to be selected", "warning");
29 |     }
30 | }
31 |
32 | if (isset($_GET["search-query"])) {
33 |     $search_query = "%" . $_GET["search-query"] . "%";
34 |     $stmt = $db->prepare("SELECT id, username FROM Users WHERE username LIKE :search LIMIT 25");
35 |     $stmt->execute([":search" => $search_query]);
36 |     $users = $stmt->fetchAll(PDO::FETCH_ASSOC);
37 |     $stmt = $db->prepare("SELECT label, title, artist FROM SONGS WHERE title LIKE :search OR artist LIKE :search LIMIT 25");
38 |     $stmt->execute([":search" => $search_query]);
39 |     $songs = $stmt->fetchAll(PDO::FETCH_ASSOC);
40 | }
41 | ?>

```

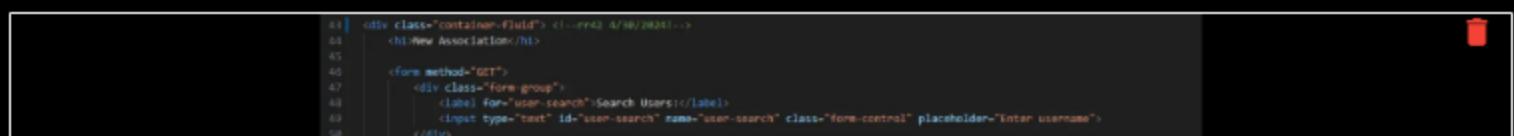
newAssociation.php Like i said, the majority of the page is functioning except for the application of the associations to the users.

Checklist Items (3)

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

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

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



```

<div class="container-fluid" > (cm62 4/30/2024) ...
41 <h1>New Association

```

```

51 <div class="form-group">
52   <label for="song-search">Search Songs:</label>
53   <input type="text" id="song-search" name="song-search" class="form-control" placeholder="Enter title or artist">
54 </div>
55
56 <button type="submit" class="btn btn-primary">Search</button>
57 </form>
58
59 <?php
60 if (isset($_GET["user-search"]) || isset($_GET["song-search"])) {
61   if (isset($_GET["user-search"])) {
62     $user_search = "% " . $_GET["user-search"] . "%";
63     $stmt = $db->prepare("SELECT id, username FROM users WHERE username LIKE :search LIMIT 25");
64     $stmt->execute([":search" => $user_search]);
65     $users = $stmt->fetchall(PDO::FETCH_ASSOC);
66   }
67   if (isset($_GET["song-search"])) {
68     $song_search = "% " . $_GET["song-search"] . "%";
69     $stmt = $db->prepare("SELECT id, title, artist FROM songs WHERE title LIKE :search OR artist LIKE :search LIMIT 25");
70     $stmt->execute([":search" => $song_search]);
71     $songs = $stmt->fetchall(PDO::FETCH_ASSOC);
72   }
73 }
74 <?php if (!empty($users)) : ?>
75 <h3>Users</h3>
76 <?php foreach ($users as $user) : ?>
77   <div class="form-check">
78     <input class="form-check-input" type="checkbox" name="users[]" value="<?php echo $user['id']; ?>" checked="checked" />
79     <label class="form-check-label" for="user-<?php echo $user['id']; ?>">
80       <?php echo htmlspecialchars($user['username']); ?>
81     </label>
82   </div>
83 <?php endforeach; ?>
84 <?php endif; ?>
85 <?php if (!empty($songs)) : ?>
86 <h3>Songs</h3>
87 <?php foreach ($songs as $song) : ?>
88   <div class="form-check">
89     <input class="form-check-input" type="checkbox" name="songs[]" value="<?php echo $song['label']; ?>" checked="checked" />
90     <label class="form-check-label" for="song-<?php echo $song['label']; ?>">
91       <?php echo htmlspecialchars($song['title'] . " - " . $song['artist']); ?>
92     </label>
93   </div>
94 <?php endforeach; ?>
95 <?php endif; ?>
96 <button type="submit" class="btn btn-success">Apply Association</button>
97 </form>
98 </div>
99 <?php
100 ?>

```

newAssociation.php (continued)

Checklist Items (5)

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

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

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

#5 Code related to submitting the checkbox lists

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

Response:

The search functions take in strings and look for partial matches in the Users table for usernames and the SONGS table for songs/artists using SELECT statements.

I have one column for the usernames and one column for the songs. I used similar code from the assign roles page for the checkboxes (form-check-input checkbox).

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://it202-rr42-prod-dc703459f313.herokuapp.com/Project/admin/newAssociation.php>

URL #2

<https://github.com/rRahmanM6/rr42-it202-008/pull/75>

Misc (1 pt.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

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

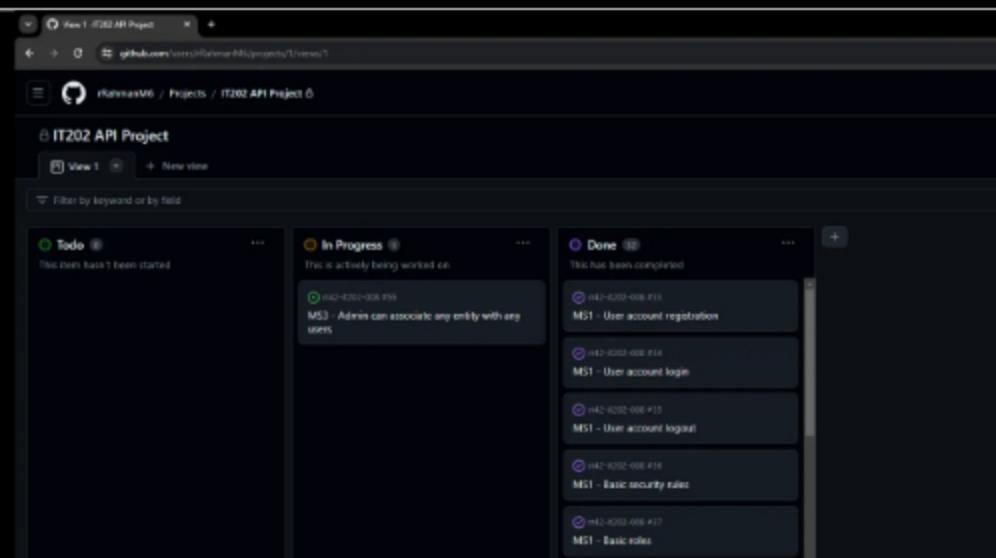
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



+ 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/rRahmanM6/projects/1/views/1>

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

The main issue I had and still have is the New associations page is not fully functioning.

As of the last assignment, I was still a bit shaky on the pull request/issues/branches workflow, but as of this milestone, I think i've fully got the hang of it.

Task #4 - Points: 1

Text: WakaTime Screenshot

i Details:

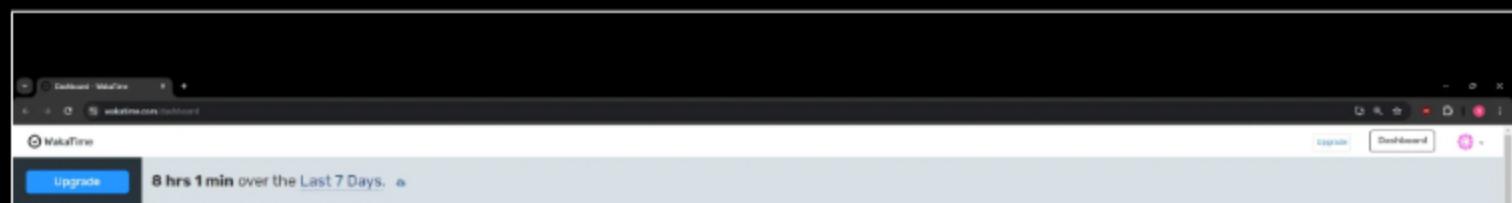
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large





wakatime

End of Assignment