

# Submission Worksheet

**CLICK TO GRADE**

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-3-2024/grade/jn426>

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

### Submissions:

Submission Selection

1 Submission [active] 4/29/2024 4:21:03 AM

### Instructions

[^ COLLAPSE ^](#)

Implement the Milestone 3 features from the project's proposal document: <https://docs.google.com/document/d/1XF96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88E>  
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
<input type="checkbox"/> #1	1	What's the concept/association?
<input type="checkbox"/> #2	1	What sort of relationship is it (one to many, many to one, many to many, etc)
<input type="checkbox"/> #3	1	Note any other considerations

Response:

The concept is that it is each user's watchlist. It is a many-to-many relationship. Each user can have many different shows on their watchlist and one different shows can be assigned to many users.

### Task #2 - Points: 1

Text: Updating Entities

#### Checklist

\*The checkboxes are for your own tracking

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

Response:

When the user updates the data, they will see the new version along with the old version because the watchlist continuously will grow.

### Handle Data Association (1 pt.)

[COLLAPSE](#)

### Task #1 - Points: 1

Text: Screenshots of the code

#### Checklist

\*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

JN426-IT202-008

File Edit Selection View Go Run ... ↺ ↻

EXPLORER

JN426-IT202-008

public\_html

project

admin

api

sql

delete\_show\_from\_user.php

edit\_show\_content.php

fetch\_show.php

helpers.js

home.php

index.php

login.php

logout.php

my\_shows.php

profile.php

register.php

shows.php

single\_view\_show.php

styles.css

testApi.php

challenge.html

index.php

README.md

test.php

.gitignore

.htaccess

composer.json

composer.lock

JN426\_Milestone2.pdf

Milestone1\_APIResearch.pdf

Module1.pdf

OPEN EDITORS

OUTLINE

TIMELINE

list\_shows.php

shows.php

delete\_show\_from\_user.php

my\_shows.php

show\_card.php

UserShows

public\_html > project > shows.php > div>container-fluid > form

You, 43 minutes ago | author (you)

1 <?php

2

3 require(\_\_DIR\_\_ . '../../../../../partials/nav.php');

4 is\_logged\_in(true);

5

6 //JN426 4/26/24

7

8 //Build search form

9 \$form = [

10 ["type" => "text", "name" => "title", "placeholder" => "Show Title", "label" => "Show Title", "include\_margin" => false],

11

12 ["type" => "text", "name" => "imdb\_rating", "placeholder" => "Show Rating", "label" => "Show Rating", "include\_margin" => false],

13 ["type" => "text", "name" => "rated", "placeholder" => "Show Audience", "label" => "Show Audience", "include\_margin" => false],

14

15 ["type" => "select", "name" => "sort", "label" => "Sort", "options" => [ "title" => "Title", "genres" => "Genres", "imdb\_rating" => "Rating", "rate" => "Rating"], "include\_margin" => false],

16 ["type" => "select", "name" => "order", "label" => "Order", "options" => [ "asc" => "<", "desc" => ">" ], "include\_margin" => false],

17

18 ["type" => "number", "name" => "limit", "label" => "Limit", "value" => "10", "include\_margin" => false],

19 ]; <- a9-10 \$form =

20

21 error\_log("Form data: " . var\_export(\$form, true));

22

23 \$total\_records = get\_total\_count("Shows" b LEFT JOIN "UserShows" ub on b.id = ub.show\_id);

24

25 \$query = "SELECT b.id, title, genres, imdb\_id, imdb\_rating, rated, ub.user\_id FROM 'Shows' b

26 LEFT JOIN 'UserShows' ub ON b.id = ub.show\_id

27 WHERE 1=1";

28 \$params = [];

29 \$session\_key = \$\_SERVER["SCRIPT\_NAME"];

30 \$is\_clear = isset(\$\_GET["clear"]);

31 if (\$is\_clear) {

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUIT

remote:

To github.com:jennematnguyen/JN426-IT202-008.git

\* [new branch] M53-Feat-Associated-Entities-Page -> M53-Feat-Associated-Entities-Page

Jenne@DESKTOP-CS8M9Q: MINGW64 /c/JN426-IT202-008 (M53-Feat-Associated-Entities-Page)

70°F Partly sunny

Type here to search

**part 1** This page allows the user to see all the available movies and add it to their watchlist

## Checklist Items (0)

The screenshot shows a Windows desktop environment with several open windows. The main focus is a Microsoft Visual Studio Code (VS Code) window titled 'JN426-IT202-008'. The left sidebar displays a file tree for the project, including files like 'index.php', 'show\_card.php', 'UserShows.php', and 'shows.php'. The 'shows.php' file is currently open in the editor, showing PHP code for handling form submissions. Below the editor are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', 'PORTS', and 'GITLENS'. The 'TERMINAL' tab is active, showing a command-line interface with a GitHub remote connection and a local terminal session. The bottom status bar shows the path 'M53-Feat-Associated-Entities-Page', the date '4/29/2024', and the time '7:19 PM'. The taskbar at the bottom includes icons for File Explorer, Task View, Start, Taskbar settings, and a search bar.

part 2

Checklist Items (0)

```
public_html > project > shows.php > ...
61     $genres = se($_GET, "genres", "", false);
62     if (!empty($genres) && $genres != "") {
63         $query .= " AND genres like :genres";
64         $params[":genres"] = "%$genres%";
65     }
66
67     $imdb_rating = se($_GET, "imdb_rating", "", false);
68     if (!empty($imdb_rating) && $imdb_rating != "") {
69         $query .= " AND imdb_rating like :imdb_rating";
70         $params[":imdb_rating"] = "%$imdb_rating%";
71     }
72
73     $rated = se($_GET, "rated", "", false);
74     if (!empty($rated) && $rated != "") {
75         $query .= " AND rated like :rated";
76         $params[":rated"] = "%$rated%";
77     }
78
79 //sort and order JN426 4/29/24
80 $sort = se($_GET, "sort", "created", false);
81 if (!in_array($sort, ["title", "genres", "imdb_rating", "rated"])) {
82     $sort = "created";
83 }
84 //tell mysql I care about the data from table "tv"
85 if ($sort === "created" || $sort === "modified") {
86     $sort = "tv." . $sort;
87 }
88
89 $order = se($_GET, "order", "desc", false);
90 if (!in_array($order, ["asc", "desc"])) {
91     $order = "desc";
92 }
93
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QMULNS

remote: To github.com/jenmatnguyen/JN426-IT202-008.git
 \* [new branch] M53-Feat-Associated-Entities-Page -> M53-Feat-Associated-Entities-Page

Denne@DESKTOP-CS8M4Q9 MINGW64 /c/JN426-IT202-008 (M53-Feat-Associated-Entities-Page)
\$ [ ]

Bash Paused 7:19 PM 4/29/2024

### part 3

## Checklist Items (0)

```
public_html > project > shows.php > ...
94     //IMPORTANT make sure you fully validate/trust $sort and $order (sql injection possibility)
95     $query .= " ORDER BY $sort $order";
96     //limit
97     try {
98         $limit = (int)se($_GET, "limit", "10", false);
99     } catch (Exception $e) {
100         $limit = 10;
101     }
102     if ($limit < 1 || $limit > 100) {
103         $limit = 10;
104     }
105     //IMPORTANT make sure you fully validate/trust $limit (sql injection possibility)
106     $query .= " LIMIT $limit";
107     //<- 846-107 if (count($_GET) > 0)
108
109
110 //*****
111 //JN426 4/26/24
112 $db = getDB();
113 $stmt = $db->prepare($query);
114 $results = [];
115 try {
116     $stmt->execute($params);
117     $r = $stmt->fetchAll();
118     if ($r) {
119         $results = $r;
120     }
121 }
122 catch (PDOException $e) {
123     error_log("Error fetching shows " . var_export($e, true));
124     flash("An unhandled error occurred", "danger");
125 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QMULNS

remote: To github.com/jenmatnguyen/JN426-IT202-008.git
 \* [new branch] M53-Feat-Associated-Entities-Page -> M53-Feat-Associated-Entities-Page

Denne@DESKTOP-CS8M4Q9 MINGW64 /c/JN426-IT202-008 (M53-Feat-Associated-Entities-Page)
\$ [ ]

You, 44 minutes ago 7:20 PM 4/29/2024

### Part 4

## Checklist Items (0)

The screenshot shows a code editor with a dark theme. The left sidebar displays a file tree for a project named 'JN426-IT202-008'. The current file is 'shows.php'. The code is a PHP script for displaying a list of TV shows. It includes logic for handling form submissions, rendering a search bar, displaying results, and providing links to view or add shows. The code is well-structured with comments and uses Bootstrap classes for styling.

```
public_html > project > shows.php > ...
121     }
122     } catch (PDOException $e) {
123         error_log("Error fetching shows " . var_export($e, true));
124         flash("Unhandled error occurred", "danger");
125     }
126 //JN426 4/28/24 <- 8127-132 Stable
127     ?>
128     <div class="container-fluid">
129         <h3>TV Shows</h3>
130         <form method="GET">
131             <div class="row mb-3" style="align-items: flex-end;">
132                 <?php foreach ($form as $k => $v) : ?>
133                     <div class="col">
134                         <?php render_input($v); ?>
135                     </div>
136                 <?php endforeach; ?>
137             </div>
138             <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
139             <a href="#">Clear</a>
140             <a href="?><?php echo get_url("fetch_show.php"); ?>" class="btn btn-danger">Add Unavailable Show</a>
141         </form>
142         <?php render_result_counts(count($results), $total_records); ?>
143         <div class="row w-100 row-cols-auto row-cols-sm-1 row-cols-md-2 row-cols-lg-3 row-cols-xl-4 row-cols-xxl-5 g-4">
144             <?php foreach ($results as $show) : ?>
145                 <div class="col">
146                     <?php render_list_show_card($show); ?>
147                 </div>
148             <?php endforeach; ?>
149         </div>
150     <?php require_once(__DIR__ . "/../../partials/flash.php"); ?>
```

## part 5

## Checklist Items (0)

The screenshot shows a code editor with a dark theme. The left sidebar displays a file tree for a project named 'JN426-IT202-008'. The current file is 'show\_card.php'. The code is a PHP template for displaying a show card. It checks if a show is provided and displays its title, rating, audience, and genres. If no user ID is provided, it adds the show to the user's watchlist. The code uses Bootstrap classes for styling.

```
public_html > project > show_card.php > ...
2     if (!isset($show)) {
3         error_log("Using Show partial without data");
4         flash("Dev Alert: Show called without data", "danger");
5     }
6     <?php if (isset($show)) : ?>
7         <div class="card mx-auto" style="width: 18rem;">
8             <div class="card-body">
9                 <h5 class="card-title"><?php se($show, "title", "Unknown"); ?></h5>
10                <div class="card-text">
11                    <ul class="list-group">
12                        <li class="list-group-item">Rating: <?php se($show, "imdb_rating", "N/A"); ?></li>
13                        <li class="list-group-item">Audience: <?php se($show, "rated", "N/A"); ?></li>
14                        <li class="list-group-item">Genre(s): <?php se($show, "genres", "N/A"); ?></li>
15                    </ul>
16                </div>
17            </div>
18        <?php if (!isset($show["user_id"]) || $show["user_id"] === "N/A") : ?>
19            <div class="card-body">
20                <a href="?><?php echo get_url('api/add_show.php?show_id=' . $show["id"]); ?>" class="btn btn-success">Add to Watchlist</a>
21            </div>
22        <?php else : ?>
23            <div class="card-body">
24                <div class="bg-warning text-dark text-center">Broker not available</div>
25            </div>
26        <?php endif; ?>
27    </div>
28    </div>
29    <?php endif; ?>
```

## Checklist Items (0)

## Task #2 - Points: 1

Text: Screenshot of the association table(s)

## Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the table(s) you made to handle the associations
<input type="checkbox"/> #2	1	Should have some example data
<input type="checkbox"/> #3	1	Make sure the heroku dev url is visible in the address bar
<input type="checkbox"/> #4	1	Clearly caption screenshots

## Task Screenshots:

## Gallery Style: Large View

Small

Medium

Large

The screenshot shows a web application interface for managing TV shows. At the top, there's a navigation bar with links for Home, Profile, My Shows, TV Shows, Admin, Show Management, Watchlist Management, and Logout. Below the navigation is a search/filter section with fields for Show Title, Show Rating, Show Audience, and Genre(s), along with dropdowns for Sort, Order, and Limit. There are also buttons for Filter, Clear, and Add Unavailable Show. The main content area displays a grid of TV show cards. Each card contains the show's title and a green 'Add to Watchlist' button. The cards are arranged in two rows of four. The bottom of the screen shows a taskbar with various icons and a system tray indicating it's 8:41 AM on 4/30/2024.

This is how the user will add shows to their watchlist

## Checklist Items (0)

This screenshot shows a single TV show card for 'Breaking Bad'. The card displays the show's title, a small thumbnail image, a brief description, and a large green 'Add to Watchlist' button. The background of the page is white, and the overall layout is clean and focused on the show details.

### Add Unavailable Show

Show Title

Search



If they seek a show that is not available then they can fetch it themselves

## Checklist Items (0)

The screenshot shows the MySQL Workbench interface. On the left is the Database Navigator with a tree view of the database structure:

- IT202Server (8.0.35-Ubuntu20.04.1):
  - information\_schema
  - jn426 224k
  - Query
    - show columns Users
    - userColumn
  - Tables (6):
    - Roles
    - Samples
    - Show fixing columns for fetch by id: 8
    - UserRoles
      - columns
        - id int
        - user\_id int
        - role\_id int
        - is\_active tinyint(1)
        - created timestamp
        - modified timestamp
      - index
      - partitions
    - UserShows 1
    - Users 9
    - Views
    - Procedures
    - Functions

The main pane displays a query result grid for the 'UserShows' table:

	id	user_id	show_id	created	modified
1	16	15	158	2024-04-30 06:26:17	2024-04-30 06:26:17
2	19	15	1	2024-04-30 06:40:22	2024-04-30 06:40:22
3	24	8	70	2024-04-30 07:10:34	2024-04-30 07:10:34
4	26	8	46	2024-04-30 09:59:44	2024-04-30 09:59:44
5	27	8	74	2024-04-30 10:02:22	2024-04-30 10:02:22

At the bottom, the taskbar shows the current session as 'MS3-Feast-Admin-Management\*' and the system tray includes icons for battery, signal, volume, and date/time.

This is the table that handles associations

### Checklist Items (0)

 ^COLLAPSE ^

### Task #3 - Points: 1

**Text:** Explain solution

#### Checklist

\*The checkboxes are for your own tracking

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

**Response:**

Option 1: A related page showcasing a list of the available shows will have an option for the user to add it to their watchlist. If the show they desire is not on the page, it will give them the option to fetch it by sending them to the "Add unavailable show" page. On the association table it associates the specific user id with a show id .



 ^COLLAPSE ^

### 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://github.com/jennatnguyen/JN426-IT202-008/pull/84>

**URL #2**

<https://jn426-it202-prod-a94514875a20.herokuapp.com/project/shows.php>



Current User's Association Page (2 pts.)

 ^COLLAPSE ^



 ^COLLAPSE ^

### Task #1 - Points: 1

**Text:** Screenshots of this page

#### Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the summary of the results with relevant information per entity
<input type="checkbox"/> #2	1	Show the single view buttons/links, 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/cort including no results (proper message should be visible)

<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 web application titled "Show Tracker" with a red header bar containing links for Home, Profile, My Shows, TV Shows, Admin, Show Management, Watchlist Management, and Logout. Below the header is a search/filter section with fields for Show Title, Show Rating, Show Audience, and Genre(s), along with dropdowns for Sort (Title), Order (+), and Limit (10). Buttons for Filter, Clear, and Remove All Shows are also present. The main content area displays three TV show cards in a grid. Each card contains the show's title, rating, audience, genres, and three buttons: View (blue), Edit (grey), and Delete (red). The cards are labeled "Large" at the top. The browser's address bar shows the URL: jn426-it202-dev-8332159f4410.herokuapp.com/project/my\_shows.php. The operating system taskbar at the bottom shows various open applications like Movies, Learn, and View.

-Show the summary of the results with relevant information per entity -Show the single view buttons/links, delete button/links, and delete all button/link -Show variations of the number of shown items count and show the count of total number of associated items to the user

## Checklist Items (0)

The screenshot shows the same "Show Tracker" application with a different set of results. The main content area now displays two TV show cards: "Game of Thrones" and "The Sopranos". The browser's address bar shows the URL: jn426-it202-dev-8332159f4410.herokuapp.com/project/my\_shows.php?title=th&imdb\_rating=&rated=&genres=&sort=title&order=asc&limit... . The operating system taskbar at the bottom shows various open applications like Movies, Learn, and View.

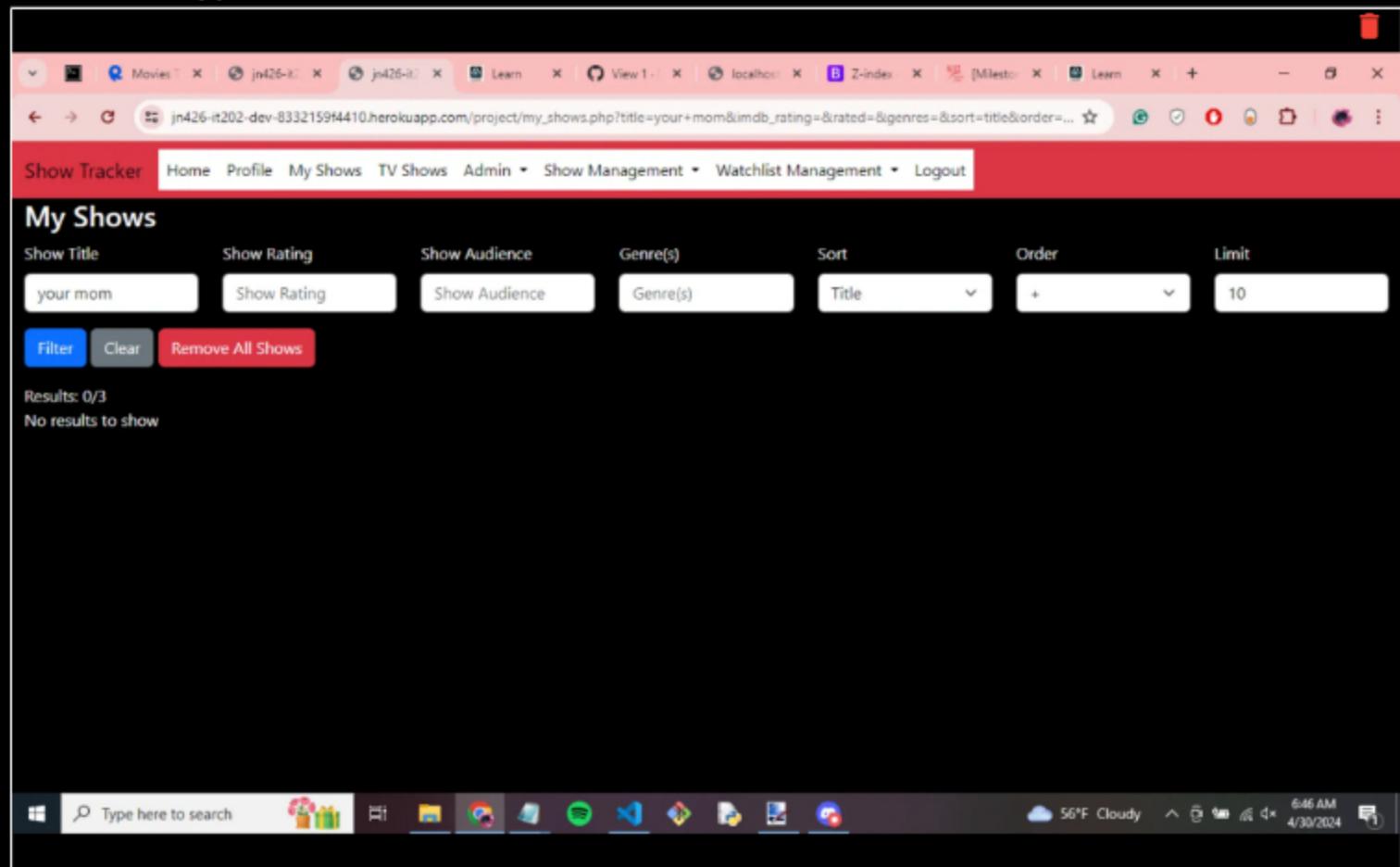
Fiction, Drama, Action,  
Adventure, Mystery, War,  
Science-fiction

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



## filter example 1

Checklist Items (0)



Show Tracker Home Profile My Shows TV Shows Admin Show Management Watchlist Management Logout

### My Shows

Show Title	Show Rating	Show Audience	Genre(s)	Sort	Order	Limit
your mom	Show Rating	Show Audience	Genre(s)	Title	+	10

Filter Clear Remove All Shows

Results: 0/3  
No results to show

no results

Checklist Items (0)

**Task #2 - Points: 1**  
**Text: Screenshot the code**

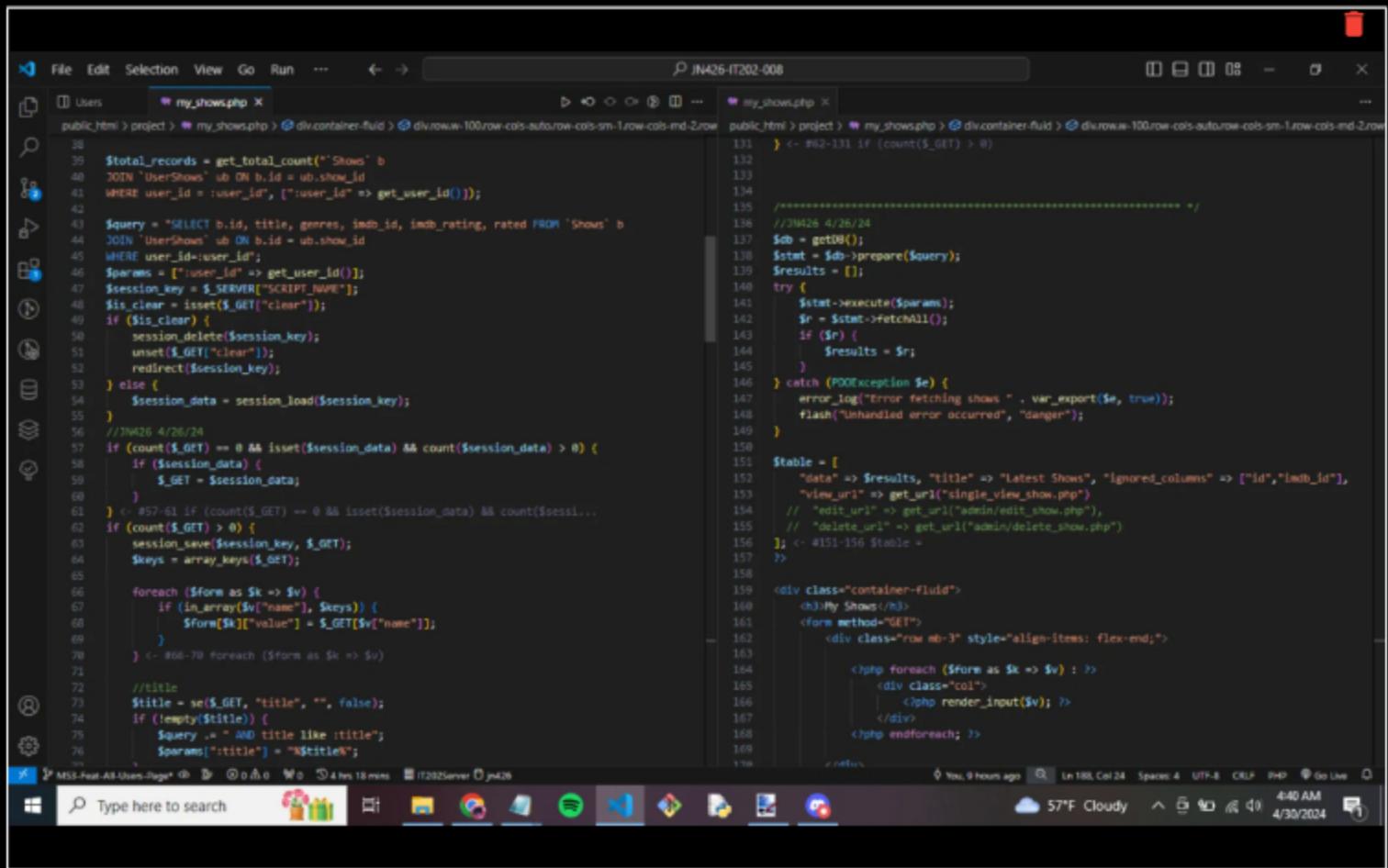
Checklist			*The checkboxes are for your own tracking
#	Points	Details	
<input type="checkbox"/> #1	1	Show the code related to fetching the user's associations (including the query)	
<input type="checkbox"/> #2	1	Show the code related to the display of the results	
<input type="checkbox"/> #3	1	Each 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



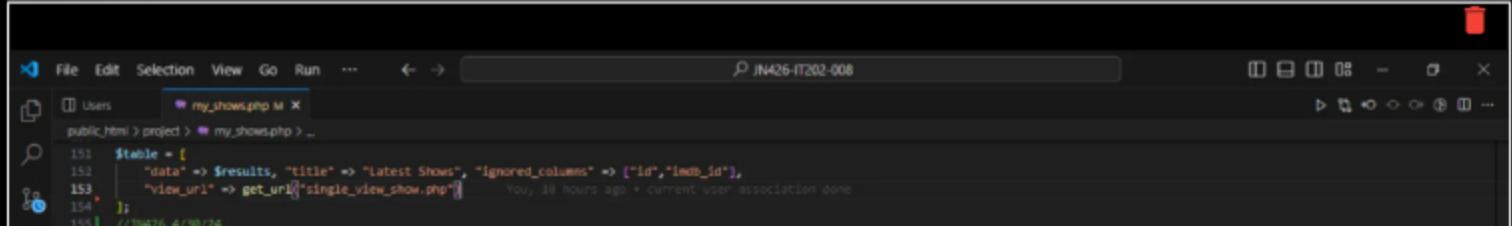
```

38
39 $total_records = get_total_count('Shows' b
40 JOIN 'UserShows' ub ON b.id = ub.show_id
41 WHERE user_id = :user_id OR ("user_id" => get_user_id()));
42
43 $query = "SELECT b.id, title, genres, imdb_id, imdb_rating, rated FROM 'Shows' b
44 JOIN 'UserShows' ub ON b.id = ub.show_id
45 WHERE user_id=:user_id";
46 $params = ["user_id" => get_user_id()];
47 $session_key = $_SERVER["SCRIPT_NAME"];
48 $is_clear = isset($_GET["clear"]);
49 if ($is_clear) {
50     session_delete($session_key);
51     unset($_GET["clear"]);
52     redirect($session_key);
53 } else {
54     $session_data = session_load($session_key);
55 }
//JN426 4/26/24
56 if (count($_GET) == 0 && !isset($session_data) && count($session_data) > 0) {
57     if ($session_data) {
58         $_GET = $session_data;
59     }
60 } <- #57-61 IF (count($_GET) == 0 && !isset($session_data) && count($sesi...
61 if (count($_GET) > 0) {
62     session_save($session_key, $_GET);
63     $keys = array_keys($_GET);
64
65     foreach ($form as $k => $v) {
66         if (in_array($v["name"], $keys)) {
67             $form[$k]["value"] = $_GET[$v["name"]];
68         }
69     }
70 } <- 466-70 foreach ($form as $k => $v)
71
72 //title
73 $title = $_GET["title", "", false];
74 if (!empty($title)) {
75     $query .= " AND title like :title";
76     $params["title"] = "%$title%";
77 }
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133 } <- #62-131 IF (count($_GET) > 0)
134
135 //*****
136 //JN426 4/26/24
137 $db = getDB();
138 $stmt = $db->prepare($query);
139 $results = [];
140 try {
141     $stmt->execute($params);
142     $r = $stmt->fetchAll();
143     if ($r) {
144         $results = $r;
145     }
146 } catch (PDOException $e) {
147     error_log("Error fetching shows " . var_export($e, true));
148     flash("Unhandled error occurred", "danger");
149 }
150
151 $table = [
152     "data" => $results, "title" => "Latest Shows", "ignored_columns" => ["id", "imdb_id"],
153     "view_url" => get_url("single_view_show.php"),
154     // "edit_url" => get_url("admin/edit_show.php"),
155     // "delete_url" => get_url("admin/delete_show.php")
156 ] <- #151-156 $table =
157 >
158
159 <div class="container-fluid">
160     <h3>My Shows</h3>
161     <form method="GET">
162         <div class="row mb-3" style="align-items: flex-end;">
163
164             <php foreach ($form as $k => $v) : >
165                 <div class="col">
166                     <php render_input($v); >
167                 </div>
168             <php endforeach; >
169         </div>
170     </form>
171 </div>

```

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

## Checklist Items (0)



```

151 $table = [
152     "data" => $results, "title" => "Latest Shows", "ignored_columns" => ["id", "imdb_id"],
153     "view_url" => get_url("single_view_show.php") // You, 18 hours ago + current user association done
154 ];
//JN426 4/26/24

```

```
157
158 <div class="container-fluid">
159   <h3>My Shows</h3>
160   <form method="GET">
161     <div class="row mb-3" style="align-items: flex-end;">
162       <?php foreach ($form as $k => $v) : ?>
163         <div class="col">
164           <?php render_input($v); ?>
165         </div>
166       <?php endforeach; ?>
167     </div>
168   </div>
169
170   <?php render_button([["text" => "Search", "type" => "submit", "text" => "Filter"]]); ?>
171   <a href="?clear" class="btn btn-secondary">Clear</a>
172   <a href="?remove" onclick="confirm('Are you sure?') || event.preventDefault()" class="btn btn-danger">Remove All Shows</a>
173
174 </form>
175 <?php render_result_counts(count($results), $total_records); ?>
176 <div class="row w-100 row-cols-auto row-cols-sm-1 row-cols-md-2 row-cols-lg-3 row-cols-xl-4 row-cols-xxl-5 g-4">
177   <?php foreach ($results as $show) : ?>
178     <div class="col">
179       <?php render_show_card($show); ?>
180     </div>
181   <?php endforeach; ?>
182   <?php if (count($results) === 0) : ?>
183     <div class="col">
184       No results to show
185     </div>
186   <?php endif; ?>
187
188 </div>
```

Show the code related to the display of the results.

## Checklist Items (0)

The screenshot shows a code editor interface with the following details:

- Title Bar:** JN426-IT202-008
- File Explorer:** Shows the project structure under "JN426-IT202-008". Key files include "lib", "functions.php", "get\_url.php", "load\_api\_keys.php", "README.md", "redirect.php", "render\_functions.php", "reset\_session.php", "safer\_echo.php", "sanitizers.php", "session\_store.php", "show\_card.php", "user\_helpers.php", and various "partial" files like "all\_users\_card.php", "button.php", "flash.php", "input\_field.php", "nav.php", "result\_counts.php", "show\_card\_list.php", and "single\_show\_card.php".
- Editor Area:** Displays the content of the "show\_card.php" file. The code is a partial template for displaying a show card. It includes logic to check if a show is set, logs an error if it's not, and displays a flash message. It then checks if the show has a date and formats it. The main content area contains a card with a title ("Unknown" if none), a list group for rating, audience, and genres, and a card body with links for view, edit, and delete actions.
- Bottom Navigation:** Includes buttons for "OPEN EDITORS", "OUTLINE", and "TIMELINE".

```
<?php
if (!isset($show)) {
    error_log("Using Show partial without data");
    flash("Dev Alert: Show called without data", "danger");
}
<?php if (isset($show) &gt;= JM26 4/30/24) >
<div class="card mx-auto" style="width: 18rem;">
<div class="card-body">
<h5 class="card-title"><?php se($show, "title", "Unknown"); ></h5>
<div class="card-text">
<ul class="list-group">
<li class="list-group-item">Rating: <?php se($show, "imdb_rating", "N/A"); ></li>
<li class="list-group-item">Audience: <?php se($show, "rated", "N/A"); ></li>
<li class="list-group-item">Genre(s): <?php se($show, "genres", "N/A"); ></li>
</ul>
</div>
<div class="card-body">
<a href=<?php echo get_url('single_view_show.php?id' . $show["id"]); >> class="btn btn-info">View</a>
<a href=<?php echo get_url('edit_show_content.php?id' . $show["id"]); >> class="btn btn-secondary">Edit</a>
<a href=<?php echo get_url('delete_show_from_user.php?id' . $show["id"]); >> class="btn btn-danger">Delete</a>
</div>
</div>
</div>
<?php endif; >
```

All buttons are on the show card that was rendered -Each record should have a button/link for single view -Each record should have a button/link for delete

## Checklist Items (0)

A screenshot of a web browser window. The address bar shows the URL "http://127.0.0.1:5501/JN426-IT202-008". The page title is "JN426-IT202-008". The content area displays a search result for "JN426-IT202-008" with three items listed: "my\_shows.php", "show\_card.php", and "delete\_show\_from\_user.php".

```
1 </?php
2
3 session_start();
4 require(__DIR__ . "/../../lib/functions.php");
5
6 $id = $_GET['id', -1, false];
7 if ($id < 1) {
8     flash("Invalid id passed to delete", "danger");
9     redirect("my_shows.php");
10 }
11
12 $db = getDB();
13 $query = "DELETE FROM 'UserShows' WHERE show_id = :id";
14 try {
15     $stmt = $db->prepare($query);
16     $stmt->execute([':id' => $id]);
17     flash("Deleted record with id $id", "success");
18 } catch (Exception $e) {
19     error_log("Error deleting stock $id" . var_export($e, true));
20     flash("Error deleting record", "danger");
21 }
22 redirect("my_shows.php"); //jn426 4/24/18
```

delete code for specific entity

## Checklist Items (0)

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar displays a file tree for a project named 'JN426-IT202-008'. The 'EXPLORER' view shows files like 'view\_show2.php', 'watchlist\_associations.php', 'api.php', 'sql.php', 'delete\_show\_from\_user.php', 'delete\_show\_from\_watchlist.php', 'edit\_show\_content.php', 'fetch\_show.php', 'helpers.php', 'home.php', 'index.php', 'login.php', 'logout.php', 'my\_shows.php', 'profile.php', 'register.php', 'shows.php', 'single\_view\_show.php', 'single\_view.php', 'styles.css', 'testApi.php', 'challenge.html', 'index.php', 'README.md', 'test\_db.php', and '.gitignore'. The 'OPEN EDITORS' section shows one editor tab for 'my\_shows.php'. The main 'Code Editor' window displays the 'my\_shows.php' file, which contains PHP code for managing user show associations. The status bar at the bottom provides information about the file, including its size (41 KB), creation date (4 hrs 18 min ago), and last save (4 hrs 18 min ago). It also shows the current file path (JN426-IT202-008), the number of lines (57), and the current line (57). The bottom right corner shows the system status (57°F Cloudy), date (4/30/2024), and time (4:48 AM).

```
public_html > project > my_shows.php ... my_shows.php X show_card.php delete_show_from_user.php delete_show_from_watchlist.php

1 <?php
2
3 require(__DIR__ . "/../../partials/nav.php");
4 is_logged_in(true);
5
6 $db = getDB();
7 //remove all associations
8 if (isset($_GET["remove"])) {
9     $query = "DELETE FROM `UserShows` WHERE user_id = :user_id";
10    try {
11        $stmt = $db->prepare($query);
12        $stmt->execute(["user_id" => get_user_id()]);
13        flash("Successfully removed all shows", "success");
14    } catch (PDOException $e) {
15        error_log("Error removing broker associations: " . var_export($e, true));
16        flash("Error removing broker associations", "danger");
17    }
18
19    redirect("my_shows.php");
20 } <- 40-20 if (isset($_GET["remove"]))
21
22 //JN426 4/26/24
23
24 //build search form
25 $form = [
26     ["type" => "text", "name" => "title", "placeholder" => "Show Title", "label" => "Show Title", "include_margin" => false],
27
28     ["type" => "text", "name" => "imdb_rating", "placeholder" => "Show Rating", "label" => "Show Rating", "include_margin" => false],
29     ["type" => "text", "name" => "rated", "placeholder" => "Show Audience", "label" => "Show Audience", "include_margin" => false],
30     ["type" => "text", "name" => "genres", "placeholder" => "Genre(s)", "label" => "Genre(s)", "include_margin" => false],
31     ["type" => "select", "name" => "sort", "label" => "Sort", "options" => ["title" => "Title", "genres" => "Genre(s)", "imdb_rating" => "Rating", "rated" => "RATED"],
32     ["type" => "select", "name" => "order", "label" => "Order", "options" => ["asc" => "+", "desc" => "-"], "include_margin" => false],
33
34     ["type" => "number", "name" => "limit", "label" => "Limit", "value" => "10", "include_margin" => false],
35 ]; <- 22-35 $form =
36
37 error_log("Form data: " . var_export($form, true));
38
39 $total_records = get_total_count("Shows");

```

the logic for deleting all associations for the user

## Checklist Items (0)

© 2013 McGraw-Hill Education. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has determined that any suppressed content does not materially affect the overall learning experience. McGraw-Hill's Higher Education is not responsible for the content or functionality of any included books that may be accompanied by the eBook.

Play 10 hours ago | 48 | In 1h Cut 1h | Season 4 | 17% | CME | PNP | Go Live | 0

Show the logic related to the count of all associated items to the user (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 the filter applied)

`$total_records -> get total count function -> render result counts`

## Checklist Items (0)

```
public_html > project > my_shows.php > ...
46 $params = ['user_id' => get_user_id()];
47 $session_key = $_SERVER['SCRIPT_NAME'];
48 $is_clear = isset($_GET['clear']);
49 if ($is_clear) {
50     session_delete($session_key);
51     unset($_GET['clear']);
52     redirect($session_key);
53 } else {
54     $session_data = session_load($session_key);
55 }
//JN426 4/26/24
57 if (count($_GET) == 0 && isset($session_data) && count($session_data) > 0) {
58     if ($session_data) {
59         $GET = $session_data;
60     }
61 } <- #57-61 If (count($_GET) == 0 && isset($session_data) && count($sess...
62 if (count($_GET) > 0) {
63     session_save($session_key, $_GET);
64     $keys = array_keys($_GET);
65
66     foreach ($form as $k => $v) {
67         if (!in_array($v['name'], $keys)) {
68             $form[$k]['value'] = $_GET[$v['name']];
69         }
70     } <- #66-70 foreach ($form as $k => $v)
71
72 //title
73 $title = se($_GET, "title", "", false);
74 if (!empty($title)) {
75     $query .= " AND title like :title";
76     $params[':title'] = "%$title%";
77 }
78
79 $genres = se($_GET, "genres", "", false);
80 if (!empty($genres) && $genres != "") {
81     $query .= " AND genres like :genres";
82     $params[':genres'] = "%$genres%";
83 }
84
85 $imdb_rating = se($_GET, "imdb_rating", "", false);
```

```
public_html > project > my_shows.php > ...
88 $params[':imdb_rating'] = "%$imdb_rating%";
89 }
90
91 $rated = se($_GET, "rated", "", false);
92 if (!empty($rated) && $rated != "") {
93     $query .= " AND rated like :rated";
94     $params[':rated'] = "%$rated%";
95 } You, 10 hours ago + current user association done
96
97 $sort = se($_GET, "sort", "created", false);
98 if (!in_array($sort, ["title", "genres", "imdb_rating", "rated"])) {
99     $sort = "created";
100 }
101 //tell mysql I care about the data from table "b"
102 if ($sort === "created" || $sort === "modified") {
103     $sort = "b." . $sort;
104 }
105
106 $order = se($_GET, "order", "desc", false);
107 if (!in_array($order, ["asc", "desc"])) {
108     $order = "desc";
109 }
110
111 //IMPORTANT make sure you fully validate/trust $sort and $order (sql injection possibility)
112 $query .= " ORDER BY $sort $order";
113 //limit
114 try {
115     $limit = (int)se($_GET, "limit", "10", false);
116 } catch (Exception $e) {
117     $limit = 10;
118 }
119 if ($limit < 1 || $limit > 100) {
120     $limit = 10;
121 }
122 //IMPORTANT make sure you fully validate/trust $limit (sql injection possibility)
123 $query .= " LIMIT $limit";
124 } <- #62-124 If (count($_GET) > 0)
```

You, 10 hours ago | In 95 Col 8 | Spaces 4 | UTF-8 | CRLF | PHP | Go Live

Type here to search

Line 95, Col 8 Spaces: 4 UTF-8 CRLF PHP Go Live

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

## Checklist Items (0)



[^COLLAPSE ^](#)

### 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 and delete all associations for the specific user

#### Response:

In this code, the result list is determined by querying the database to retrieve shows associated with the current user. This association logic is implemented by joining the Shows table with the UserShows table on the show\_id column. Filters for the search form, such as title, genres, IMDb rating, and audience, are applied as additional conditions in the SQL query's WHERE clause to refine the results based on user input. The total items count represents the total number of shows associated with the user, calculated by querying the database with a condition to count only the shows associated with the current user. This count is used to display the total number of items associated with the user. The visible result count is determined by counting the number of rows fetched from the database query result, representing the number of shows matching the search criteria and associated with the user. Additionally, the logic for single delete allows users to remove individual associations by clicking on the "Remove" button for each show. Furthermore, the "Remove All Shows" button triggers a process to delete all associations for the specific user, providing a way to clear all associations at once.



[^COLLAPSE ^](#)

### 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://github.com/jennatnguyen/JN426-IT202-008/pull/88>

#### URL #2

[https://jn426-it202-prod-a94514875a20.herokuapp.com/project/my\\_shows.php](https://jn426-it202-prod-a94514875a20.herokuapp.com/project/my_shows.php)



[^COLLAPSE ^](#)

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



### Task #1 - Points: 1

[COLLAPSE](#)

## TASK #1 - Points: 1

Text: Screenshots of this page

## Checklist

\*The checkboxes are for your own tracking

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

## Task Screenshots:

## Gallery Style: Large View

Small      Medium      Large

The screenshot shows a web application titled "Watchlist Associations". At the top, there are search filters for "Username", "Title", "Genres", "Rating", "Audience", and a "Limit" dropdown set to 10. Below the filters, a "Search" button and a "Clear Associations" button are visible. The main content area displays a table of shows on watchlists. The table has columns: "Username", "Watchlist Count", "Title", "Genre(s)", "Rating", "Audience", and "Actions". Three rows of data are shown:

Username	Watchlist Count	Title	Genre(s)	Rating	Audience	Actions
test21	1	Attack on Titan	Animation, Fantasy, Science-Fiction, Action, Adventure, Anime, Drama, Horror, War, Mystery, Science-fiction	8.66	TV-MA	<a href="#">View Show</a> <a href="#">Delete from Watchlist</a>
test20	1	SpongeBob SquarePants	Animation, Comedy, Family, Fantasy, Science-Fiction, Children	7.6	TV-Y7	<a href="#">View Show</a> <a href="#">Delete from Watchlist</a>
test20	1	The Sopranos	Drama, Crime, Family	8.64	TV-MA	<a href="#">View Show</a> <a href="#">Delete from Watchlist</a>

At the bottom of the screenshot, the Windows taskbar is visible with various pinned icons and the system tray showing the date and time.

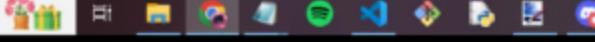
-Show the summary of the results with relevant information per entity  
-Show the single view buttons/links, delete button/links  
>Show the username related to the specific entity and that it's clickable  
>Show variations of the number of shown items count and show the count of total number of associated items

## Checklist Items (0)

Total Shows on Watchlists: 5  
 Total Items On Page: 4

Username	Title	Genres	Rating	Audience	Limit
<input type="text" value="Enter username"/>	<input type="text" value="Enter title"/>	<input type="text" value="Enter genres"/>	<input type="text" value="8"/>	<input type="text" value="Enter audience"/>	<input type="text" value="10"/>
<input type="button" value="Search"/> <input type="button" value="Clear Associations"/>					

Username	Watchlist Count	Title	Genre(s)	Rating	Audience	Actions
test21	1	Attack on Titan	Animation, Fantasy, Science-Fiction, Action, Adventure, Anime, Drama, Horror, War, Mystery, Science-fiction	8.66	TV-MA	<input type="button" value="View Show"/> <input type="button" value="Delete from Watchlist"/>
test20	1	The Sopranos	Drama, Crime, Family	8.64	TV-MA	<input type="button" value="View Show"/> <input type="button" value="Delete from Watchlist"/>
test20	1	Game of Thrones	Fantasy, Science-Fiction, Drama, Action, Adventure, Mystery, War, Science-fiction	8.449	TV-MA	<input type="button" value="View Show"/> <input type="button" value="Delete from Watchlist"/>
	1	Breaking Bad	Drama, Crime, Thriller, Western, Suspense	8.9	TV-MA	<input type="button" value="View Show"/> <input type="button" value="Delete from Watchlist"/>

Type here to search   Weather: 56°F Mostly cloudy  6:49 AM 4/30/2024

filter sort variation: rating

## Checklist Items (0)

Show Tracker Home Profile My Shows TV Shows Admin Show Management Watchlist Management Logout

### Watchlist Associations

Total Shows on Watchlists: 5  
 Total Items On Page: 0

Username	Title	Genres	Rating	Audience	Limit
<input type="text" value="your mom"/>	<input type="text" value="Enter title"/>	<input type="text" value="Enter genres"/>	<input type="text" value="Enter rating"/>	<input type="text" value="Enter audience"/>	<input type="text" value="10"/>
<input type="button" value="Search"/> <input type="button" value="Clear Associations"/>					

Username	Watchlist Count	Title	Genre(s)	Rating	Audience	Actions
No results available.						

Type here to search   Weather: 56°F Mostly cloudy  6:50 AM 4/30/2024

no results

## Checklist Items (0)



## 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 all 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 username field that is clickable to go to the user's profile page
<input type="checkbox"/> #5	1	Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity
<input type="checkbox"/> #6	1	Show the logic related to the count of all associated items (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 (should include a partial match for username ) (limit should be constrained to 1-100 otherwise default to 10)
<input type="checkbox"/> #9	1	Include uid/date comments for each code screenshot
<input type="checkbox"/> #10	1	Clearly caption screenshots

### Task Screenshots:

#### Gallery Style: Large View

Small

Medium

Large

```

public_html > project > admin > watchlist_associations.php
public_html > project > admin > watchlist_associations.php > table > tbody > tr > td

26
27
28 $username = isset($_GET['username']) ? $_GET['username'] : '';
29 $genres = isset($_GET['genres']) ? $_GET['genres'] : '';
30 $title = isset($_GET['title']) ? $_GET['title'] : '';
31 $released = isset($_GET['imdb_rating']) ? $_GET['imdb_rating'] : '';
32 $rated = isset($_GET['rated']) ? $_GET['rated'] : '';
33 $limit = isset($_GET['limit']) ? intval($_GET['limit']) : 10; // Default to 10
34 $limit = min($limit, 100);
35
36 $query = "SELECT s.id, title, genres, imdb_id, imdb_rating, rated,
37 COUNT(DISTINCT us.user_id) AS watchlist_count, u.username AS Username
38 FROM `UserShows` us
39 JOIN `Shows` s ON us.show_id = s.id
40 JOIN `Users` u ON us.user_id = u.id
41 WHERE 1 = 1";
42 $params = [];
43 //username: JN426 4/30/24 You, 16 minutes ago + Uncommitted changes
44 $username = se($_GET, "username", "", false);
45 if (!empty($username)) {
46   $query .= " AND u.username like :username";
47   $params[":username"] = "%$username%";
48 }
49
50 //title
51 $title = se($_GET, "title", "", false);
52 if (!empty($title)) {
53   $query .= " AND title like :title";
54   $params[":title"] = "%$title%";
55 }
56
57 $genres = se($_GET, "genres", "", false);
58 if (!empty($genres) && $genres != "") {
59   $query .= " AND genres like :genres";
60   $params[":genres"] = "%$genres%";
61 }

public_html > project > admin > watchlist_associations.php > table > tbody > tr > td
public_html > project > admin > watchlist_associations.php > table > tbody > tr > td

24
25 $query .= " GROUP BY s.id, title, genres, imdb_rating, rated, u.id, u.username
26 ORDER BY s.created DESC LIMIT $limit";
27
28 // Execute the query
29 $db = getDB();
30 $stmt = $db->prepare($query);
31 try {
32   $stmt->execute($params);
33   $results = $stmt->fetchAll();
34 } catch (PDOException $e) {
35   error_log("Error fetching shows: " . $e->getMessage());
36   flash("An error occurred while fetching shows", "danger");
37 }
38
39 $totalItemsQuery = "SELECT COUNT(*) AS total_items FROM UserShows";
40 $stmt = $db->prepare($totalItemsQuery);
41 $stmt->execute();
42 $row = $stmt->fetch(PDO::FETCH_ASSOC);
43 $totalItemCount = $row['total_items'];
44
45 ?>
46
47 <div style="text-align: center;">
48   <div class="container-fluid">
49     <h2>Watchlist Associations</h2>
50     <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
51     <h3>Total Items On Page: <php echo count($results); ></h3>
52
53     <div style="text-align: left;">
54       <form method="GET" class="row g-3 justify-content-center">
55         <div class="col-auto">
56           <label for="username" class="form-label">Username</label>
57           <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
58         </div>
59         <div class="col-auto">
60           <label for="title" class="form-label">Title</label>
61           <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
62         </div>
63         <div class="col-auto">
64           <label for="genres" class="form-label">Genres</label>
65           <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
66         </div>
67       </form>
68     </div>
69   </div>
70 </div>
71
72 <table border="1">
73   <thead>
74     <tr>
75       <th>Show ID</th>
76       <th>Title</th>
77       <th>Genre(s)</th>
78       <th>IMDb Rating</th>
79       <th>Released Year</th>
80       <th>Watchlist Count</th>
81       <th>Last Updated</th>
82     </tr>
83   </thead>
84   <tbody>
85     <tr>
86       <td><php echo se($row['id'], "id", "", false); ></td>
87       <td><php echo se($row['title'], "title", "", false); ></td>
88       <td><php echo se($row['genres'], "genres", "", false); ></td>
89       <td><php echo se($row['imdb_rating'], "imdb_rating", "", false); ></td>
90       <td><php echo se($row['released'], "released", "", false); ></td>
91       <td><php echo se($row['watchlist_count'], "watchlist_count", "", false); ></td>
92       <td><php echo se($row['created'], "created", "", false); ></td>
93     </tr>
94   </tbody>
95 </table>
96
97 <div style="text-align: center;">
98   <div class="container-fluid">
99     <h2>Watchlist Associations</h2>
100    <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
101    <h3>Total Items On Page: <php echo count($results); ></h3>
102
103    <div style="text-align: left;">
104      <form method="GET" class="row g-3 justify-content-center">
105        <div class="col-auto">
106          <label for="username" class="form-label">Username</label>
107          <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
108        </div>
109        <div class="col-auto">
110          <label for="title" class="form-label">Title</label>
111          <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
112        </div>
113        <div class="col-auto">
114          <label for="genres" class="form-label">Genres</label>
115          <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
116        </div>
117      </form>
118    </div>
119  </div>
120 </div>
121
122 <table border="1">
123   <thead>
124     <tr>
125       <th>Show ID</th>
126       <th>Title</th>
127       <th>Genre(s)</th>
128       <th>IMDb Rating</th>
129       <th>Released Year</th>
130       <th>Watchlist Count</th>
131       <th>Last Updated</th>
132     </tr>
133   </thead>
134   <tbody>
135     <tr>
136       <td><php echo se($row['id'], "id", "", false); ></td>
137       <td><php echo se($row['title'], "title", "", false); ></td>
138       <td><php echo se($row['genres'], "genres", "", false); ></td>
139       <td><php echo se($row['imdb_rating'], "imdb_rating", "", false); ></td>
140       <td><php echo se($row['released'], "released", "", false); ></td>
141       <td><php echo se($row['watchlist_count'], "watchlist_count", "", false); ></td>
142       <td><php echo se($row['created'], "created", "", false); ></td>
143     </tr>
144   </tbody>
145 </table>
146
147 <div style="text-align: center;">
148   <div class="container-fluid">
149     <h2>Watchlist Associations</h2>
150     <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
151     <h3>Total Items On Page: <php echo count($results); ></h3>
152
153     <div style="text-align: left;">
154       <form method="GET" class="row g-3 justify-content-center">
155         <div class="col-auto">
156           <label for="username" class="form-label">Username</label>
157           <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
158         </div>
159         <div class="col-auto">
160           <label for="title" class="form-label">Title</label>
161           <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
162         </div>
163         <div class="col-auto">
164           <label for="genres" class="form-label">Genres</label>
165           <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
166         </div>
167       </form>
168     </div>
169   </div>
170 </div>
171
172 <table border="1">
173   <thead>
174     <tr>
175       <th>Show ID</th>
176       <th>Title</th>
177       <th>Genre(s)</th>
178       <th>IMDb Rating</th>
179       <th>Released Year</th>
180       <th>Watchlist Count</th>
181       <th>Last Updated</th>
182     </tr>
183   </thead>
184   <tbody>
185     <tr>
186       <td><php echo se($row['id'], "id", "", false); ></td>
187       <td><php echo se($row['title'], "title", "", false); ></td>
188       <td><php echo se($row['genres'], "genres", "", false); ></td>
189       <td><php echo se($row['imdb_rating'], "imdb_rating", "", false); ></td>
190       <td><php echo se($row['released'], "released", "", false); ></td>
191       <td><php echo se($row['watchlist_count'], "watchlist_count", "", false); ></td>
192       <td><php echo se($row['created'], "created", "", false); ></td>
193     </tr>
194   </tbody>
195 </table>
196
197 <div style="text-align: center;">
198   <div class="container-fluid">
199     <h2>Watchlist Associations</h2>
200     <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
201     <h3>Total Items On Page: <php echo count($results); ></h3>
202
203     <div style="text-align: left;">
204       <form method="GET" class="row g-3 justify-content-center">
205         <div class="col-auto">
206           <label for="username" class="form-label">Username</label>
207           <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
208         </div>
209         <div class="col-auto">
210           <label for="title" class="form-label">Title</label>
211           <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
212         </div>
213         <div class="col-auto">
214           <label for="genres" class="form-label">Genres</label>
215           <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
216         </div>
217       </form>
218     </div>
219   </div>
220 </div>
221
222 <table border="1">
223   <thead>
224     <tr>
225       <th>Show ID</th>
226       <th>Title</th>
227       <th>Genre(s)</th>
228       <th>IMDb Rating</th>
229       <th>Released Year</th>
230       <th>Watchlist Count</th>
231       <th>Last Updated</th>
232     </tr>
233   </thead>
234   <tbody>
235     <tr>
236       <td><php echo se($row['id'], "id", "", false); ></td>
237       <td><php echo se($row['title'], "title", "", false); ></td>
238       <td><php echo se($row['genres'], "genres", "", false); ></td>
239       <td><php echo se($row['imdb_rating'], "imdb_rating", "", false); ></td>
240       <td><php echo se($row['released'], "released", "", false); ></td>
241       <td><php echo se($row['watchlist_count'], "watchlist_count", "", false); ></td>
242       <td><php echo se($row['created'], "created", "", false); ></td>
243     </tr>
244   </tbody>
245 </table>
246
247 <div style="text-align: center;">
248   <div class="container-fluid">
249     <h2>Watchlist Associations</h2>
250     <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
251     <h3>Total Items On Page: <php echo count($results); ></h3>
252
253     <div style="text-align: left;">
254       <form method="GET" class="row g-3 justify-content-center">
255         <div class="col-auto">
256           <label for="username" class="form-label">Username</label>
257           <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
258         </div>
259         <div class="col-auto">
260           <label for="title" class="form-label">Title</label>
261           <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
262         </div>
263         <div class="col-auto">
264           <label for="genres" class="form-label">Genres</label>
265           <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
266         </div>
267       </form>
268     </div>
269   </div>
270 </div>
271
272 <table border="1">
273   <thead>
274     <tr>
275       <th>Show ID</th>
276       <th>Title</th>
277       <th>Genre(s)</th>
278       <th>IMDb Rating</th>
279       <th>Released Year</th>
280       <th>Watchlist Count</th>
281       <th>Last Updated</th>
282     </tr>
283   </thead>
284   <tbody>
285     <tr>
286       <td><php echo se($row['id'], "id", "", false); ></td>
287       <td><php echo se($row['title'], "title", "", false); ></td>
288       <td><php echo se($row['genres'], "genres", "", false); ></td>
289       <td><php echo se($row['imdb_rating'], "imdb_rating", "", false); ></td>
290       <td><php echo se($row['released'], "released", "", false); ></td>
291       <td><php echo se($row['watchlist_count'], "watchlist_count", "", false); ></td>
292       <td><php echo se($row['created'], "created", "", false); ></td>
293     </tr>
294   </tbody>
295 </table>
296
297 <div style="text-align: center;">
298   <div class="container-fluid">
299     <h2>Watchlist Associations</h2>
300     <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
301     <h3>Total Items On Page: <php echo count($results); ></h3>
302
303     <div style="text-align: left;">
304       <form method="GET" class="row g-3 justify-content-center">
305         <div class="col-auto">
306           <label for="username" class="form-label">Username</label>
307           <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
308         </div>
309         <div class="col-auto">
310           <label for="title" class="form-label">Title</label>
311           <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
312         </div>
313         <div class="col-auto">
314           <label for="genres" class="form-label">Genres</label>
315           <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
316         </div>
317       </form>
318     </div>
319   </div>
320 </div>
321
322 <table border="1">
323   <thead>
324     <tr>
325       <th>Show ID</th>
326       <th>Title</th>
327       <th>Genre(s)</th>
328       <th>IMDb Rating</th>
329       <th>Released Year</th>
330       <th>Watchlist Count</th>
331       <th>Last Updated</th>
332     </tr>
333   </thead>
334   <tbody>
335     <tr>
336       <td><php echo se($row['id'], "id", "", false); ></td>
337       <td><php echo se($row['title'], "title", "", false); ></td>
338       <td><php echo se($row['genres'], "genres", "", false); ></td>
339       <td><php echo se($row['imdb_rating'], "imdb_rating", "", false); ></td>
340       <td><php echo se($row['released'], "released", "", false); ></td>
341       <td><php echo se($row['watchlist_count'], "watchlist_count", "", false); ></td>
342       <td><php echo se($row['created'], "created", "", false); ></td>
343     </tr>
344   </tbody>
345 </table>
346
347 <div style="text-align: center;">
348   <div class="container-fluid">
349     <h2>Watchlist Associations</h2>
350     <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
351     <h3>Total Items On Page: <php echo count($results); ></h3>
352
353     <div style="text-align: left;">
354       <form method="GET" class="row g-3 justify-content-center">
355         <div class="col-auto">
356           <label for="username" class="form-label">Username</label>
357           <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
358         </div>
359         <div class="col-auto">
360           <label for="title" class="form-label">Title</label>
361           <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
362         </div>
363         <div class="col-auto">
364           <label for="genres" class="form-label">Genres</label>
365           <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
366         </div>
367       </form>
368     </div>
369   </div>
370 </div>
371
372 <table border="1">
373   <thead>
374     <tr>
375       <th>Show ID</th>
376       <th>Title</th>
377       <th>Genre(s)</th>
378       <th>IMDb Rating</th>
379       <th>Released Year</th>
380       <th>Watchlist Count</th>
381       <th>Last Updated</th>
382     </tr>
383   </thead>
384   <tbody>
385     <tr>
386       <td><php echo se($row['id'], "id", "", false); ></td>
387       <td><php echo se($row['title'], "title", "", false); ></td>
388       <td><php echo se($row['genres'], "genres", "", false); ></td>
389       <td><php echo se($row['imdb_rating'], "imdb_rating", "", false); ></td>
390       <td><php echo se($row['released'], "released", "", false); ></td>
391       <td><php echo se($row['watchlist_count'], "watchlist_count", "", false); ></td>
392       <td><php echo se($row['created'], "created", "", false); ></td>
393     </tr>
394   </tbody>
395 </table>
396
397 <div style="text-align: center;">
398   <div class="container-fluid">
399     <h2>Watchlist Associations</h2>
400     <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
401     <h3>Total Items On Page: <php echo count($results); ></h3>
402
403     <div style="text-align: left;">
404       <form method="GET" class="row g-3 justify-content-center">
405         <div class="col-auto">
406           <label for="username" class="form-label">Username</label>
407           <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
408         </div>
409         <div class="col-auto">
410           <label for="title" class="form-label">Title</label>
411           <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
412         </div>
413         <div class="col-auto">
414           <label for="genres" class="form-label">Genres</label>
415           <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
416         </div>
417       </form>
418     </div>
419   </div>
420 </div>
421
422 <table border="1">
423   <thead>
424     <tr>
425       <th>Show ID</th>
426       <th>Title</th>
427       <th>Genre(s)</th>
428       <th>IMDb Rating</th>
429       <th>Released Year</th>
430       <th>Watchlist Count</th>
431       <th>Last Updated</th>
432     </tr>
433   </thead>
434   <tbody>
435     <tr>
436       <td><php echo se($row['id'], "id", "", false); ></td>
437       <td><php echo se($row['title'], "title", "", false); ></td>
438       <td><php echo se($row['genres'], "genres", "", false); ></td>
439       <td><php echo se($row['imdb_rating'], "imdb_rating", "", false); ></td>
440       <td><php echo se($row['released'], "released", "", false); ></td>
441       <td><php echo se($row['watchlist_count'], "watchlist_count", "", false); ></td>
442       <td><php echo se($row['created'], "created", "", false); ></td>
443     </tr>
444   </tbody>
445 </table>
446
447 <div style="text-align: center;">
448   <div class="container-fluid">
449     <h2>Watchlist Associations</h2>
450     <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
451     <h3>Total Items On Page: <php echo count($results); ></h3>
452
453     <div style="text-align: left;">
454       <form method="GET" class="row g-3 justify-content-center">
455         <div class="col-auto">
456           <label for="username" class="form-label">Username</label>
457           <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
458         </div>
459         <div class="col-auto">
460           <label for="title" class="form-label">Title</label>
461           <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
462         </div>
463         <div class="col-auto">
464           <label for="genres" class="form-label">Genres</label>
465           <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
466         </div>
467       </form>
468     </div>
469   </div>
470 </div>
471
472 <table border="1">
473   <thead>
474     <tr>
475       <th>Show ID</th>
476       <th>Title</th>
477       <th>Genre(s)</th>
478       <th>IMDb Rating</th>
479       <th>Released Year</th>
480       <th>Watchlist Count</th>
481       <th>Last Updated</th>
482     </tr>
483   </thead>
484   <tbody>
485     <tr>
486       <td><php echo se($row['id'], "id", "", false); ></td>
487       <td><php echo se($row['title'], "title", "", false); ></td>
488       <td><php echo se($row['genres'], "genres", "", false); ></td>
489       <td><php echo se($row['imdb_rating'], "imdb_rating", "", false); ></td>
490       <td><php echo se($row['released'], "released", "", false); ></td>
491       <td><php echo se($row['watchlist_count'], "watchlist_count", "", false); ></td>
492       <td><php echo se($row['created'], "created", "", false); ></td>
493     </tr>
494   </tbody>
495 </table>
496
497 <div style="text-align: center;">
498   <div class="container-fluid">
499     <h2>Watchlist Associations</h2>
500     <h3>Total Shows on Watchlists: <php echo $totalItemCount; ></h3>
501     <h3>Total Items On Page: <php echo count($results); ></h3>
502
503     <div style="text-align: left;">
504       <form method="GET" class="row g-3 justify-content-center">
505         <div class="col-auto">
506           <label for="username" class="form-label">Username</label>
507           <input type="text" class="form-control" id="username" name="username" value="<php echo se($username, "username", "", false); >">
508         </div>
509         <div class="col-auto">
510           <label for="title" class="form-label">Title</label>
511           <input type="text" class="form-control" id="title" name="title" value="<php echo se($title, "title", "", false); >">
512         </div>
513         <div class="col-auto">
514           <label for="genres" class="form-label">Genres</label>
515           <input type="text" class="form-control" id="genres" name="genres" value="<php echo se($genres, "genres", "", false); >">
516         </div>
517       </form>
518     </div>
519   </div>
520 </div>
521
522 <table border="1">
523   <thead>
524     <tr>
525       <th>Show ID</th>
526       <th>Title</th>
527       <th>Genre(s)</th>
528       <th>IMDb Rating</th>
529       <th>Released Year</th>
530       <th>Watchlist Count</th>
531       <th>Last
```

```
        <input type="text" class="form-control" id="title" name="title" value=<?php echo $imdb_rating ?> />
```

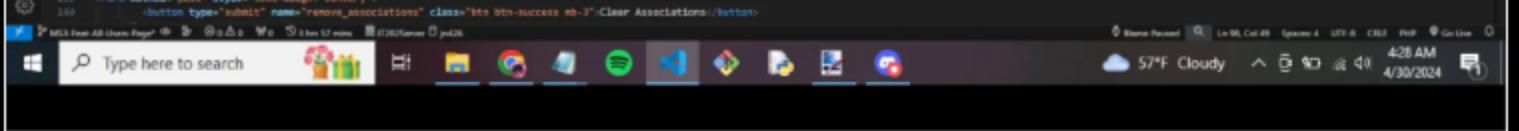
Show the code related to fetching all associations (including the query)

### Checklist Items (0)

Show the code related to the display of the results + Each record should have a button/link for single view + Each record should have a username field that is clickable to go to the user's profile page

## Checklist Items (0)

```
File Edit Selection View Go Run ... JN426-IT202-008  
User watchlist_associations.php  
public_html > project > admin > watchlist_associations.php > div > div.container-fluid  
33 $total_results = $row['total_items'];  
34  
35 //  
36 <div style="text-align: center;">  
37     <div class="container-fluid">//38426 A/38/24  
38     <h2>Watchlist Associations</h2>  
39     <h3>Total Shows on Watchlists: <php echo $total_results;>  
40     <h3>Total Items On Page: <php echo count($results); >  
41  
42     <div style="text-align: left;">  
43         <form method="GET" class="row g-3 justify-content-center">  
44             <div class="col-auto">  
45                 <label for="username" class="form-label">Username:</label>  
46                 <input type="text" class="form-control" id="username" name="username" value=<php echo htmlspecialchars($username); > placeholder="Enter username">  
47             </div>  
48             <div class="col-auto">  
49                 <label for="title" class="form-label">Title:</label>  
50                 <input type="text" class="form-control" id="title" name="title" value=<php echo htmlspecialchars($title); > placeholder="Enter title">  
51             </div>  
52             <div class="col-auto">  
53                 <label for="genres" class="form-label">Genres:</label>  
54                 <input type="text" class="form-control" id="genres" name="genres" value=<php echo htmlspecialchars($genres); > placeholder="Enter genres">  
55             </div>  
56             <div class="col-auto">  
57                 <label for="imdb_rating" class="form-label">Rating:</label>  
58                 <input type="text" class="form-control" id="imdb_rating" name="imdb_rating" value=<php echo htmlspecialchars($imdb_rating); > placeholder="Enter rating">  
59             </div>  
60             <div class="col-auto">  
61                 <label for="rated" class="form-label">Audience:</label>  
62                 <input type="text" class="form-control" id="rated" name="rated" value=<php echo htmlspecialchars($rated); > placeholder="Enter audience">  
63             </div>  
64             <div class="col-auto">  
65                 <label for="limit" class="form-label">Limit:</label>  
66                 <input type="number" class="form-control" id="limit" name="limit" value=<php echo $limit; > min="1" max="300" placeholder="Enter limit">  
67             </div>  
68         </div>  
69         <div class="col-12">  
70             <div style="text-align: center;">  
71                 <button type="submit" class="btn btn-primary">Search  
72             </div>  
73         </div>  
74     </div>
```



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

### Checklist Items (0)

```
public_html > project > admin > watchlist_associations.php > ...
78 // Execute the query
79 $db = getDB();
80 $stmt = $db->prepare($query);
81 try {
82     $stmt->execute($params);
83     $results = $stmt->fetchAll();
84 } catch (PDOException $e) {
85     error_log("Error fetching shows: " . $e->getMessage());
86     flash("An error occurred while fetching shows", "danger");
87 }
88 //JN426 4/30/24
89 $totalItems = "SELECT COUNT(*) AS total_items FROM UserShows";
90 $stmt = $db->prepare($totalItems);
91 $stmt->execute();
92 $row = $stmt->fetch(PDO::FETCH_ASSOC);
93 $total_results = $row['total_items'];
94
95 ?>
96
97 <div style="text-align: center;">
98     <div class="container-fluid">
99         <h2>Watchlist Associations</h2>
100        <h5>Total Shows on Watchlists: <?php echo $total_results; ?></h5>
101        <h5>Total Items On Page: <?php echo count($results); ?></h5>
102
103        <div style="text-align: left;">
104            <form method="GET" class="row g-3 justify-content-center">
105                <div class="col-auto">
106                    <label for="username" class="form-label">Username</label>
107                    <input type="text" class="form-control" id="username" name="username" value="<?php echo htmlspecialchars($username); ?>" placeholder="Enter username">
108                </div>
109                <div class="col-auto">
110                    <button type="submit" class="btn btn-primary mb-3">Search</button>
111                </div>
112            </form>
113        </div>
114    </div>
115 </div>
116
117 <form method="post" style="text-align: center;">
118     <button type="submit" name="remove_associations" class="btn btn-success mb-3">Clear Associations</button>
```

```
public_html > project > admin > watchlist_associations.php > ...
79 $query = "SELECT s.id, title, genres, imdb_id, imdb_rating, rated,
80 COUNT(DISTINCT us.user_id) AS watchlist_count, us.username AS username
81 FROM `UserShows` us
82 JOIN Shows s ON us.show_id = s.id
83 JOIN Users u ON us.user_id = u.id
84 WHERE 1 = 1";
85 $params = [];
86 //username JN426 4/30/24
87 $username = $_GET['username'];
88 if (!empty($username)) {
89     $query .= " AND u.username like :username";
90     $params[':username'] = "%$username%";
91 }
92
93 //title
94 $title = $_GET['title'];
95 if (!empty($title)) {
96     $query .= " AND title like :title";
97     $params[':title'] = "%$title%";
98 }
99
100 $genres = $_GET['genres'];
101 if (!empty($genres) && $genres != "") {
102     $query .= " AND genres like :genres";
103     $params[':genres'] = "%$genres%";
104 }
```

## Show the logic related to the count of all associated 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 the filter applied)

### Checklist Items (0)

```
public_html > project > admin > watchlist_associations.php > ...
79 $query = "SELECT s.id, title, genres, imdb_id, imdb_rating, rated,
80 COUNT(DISTINCT us.user_id) AS watchlist_count, us.username AS username
81 FROM `UserShows` us
82 JOIN Shows s ON us.show_id = s.id
83 JOIN Users u ON us.user_id = u.id
84 WHERE 1 = 1";
85 $params = [];
86 //username JN426 4/30/24
87 $username = $_GET['username'];
88 if (!empty($username)) {
89     $query .= " AND u.username like :username";
90     $params[':username'] = "%$username%";
91 }
92
93 //title
94 $title = $_GET['title'];
95 if (!empty($title)) {
96     $query .= " AND title like :title";
97     $params[':title'] = "%$title%";
98 }
99
100 $genres = $_GET['genres'];
101 if (!empty($genres) && $genres != "") {
102     $query .= " AND genres like :genres";
103     $params[':genres'] = "%$genres%";
104 }
```

```

$imdb_rating = $_GET['imdb_rating'];
if (empty($imdb_rating) || $imdb_rating == "") {
    $query .= " AND imdb_rating like '$imdb_rating'";
    $params["imdb_rating"] = "%$imdb_rating%";
}

$rated = $_GET['rated'];
if (empty($rated) || $rated == "") {
    $query .= " AND rated like '$rated'";
    $params["rated"] = "%$rated%";
}

$Query .= " GROUP BY s.id, title, genres, imdb_rating, rated, s.id, username
            ORDER BY s.created DESC LIMIT $limit";

```

Now, 22 minutes ago | In 8h, Col 14 | Spaces 4 | UTF-8 | CLIP | PHP | Go Live | 431 AM | 4/30/2024

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)

Checklist Items (0)

JN426-IT202-008

File Edit Selection View Go Run ... ← →

EXPLORER JN426-IT202-008 public\_html > project > admin > delete\_show\_from\_watchlist.php

```

1 session_start();
2 require(__DIR__ . "/../../lib/functions.php");
3
4 $id = $_GET['id', -1, false];
5 if ($id < 1) {
6     flash("Invalid id passed to delete", "danger");
7     redirect("admin/watchlist_associations.php");
8 }
9
10 $db = getDB();
11 $query = "DELETE FROM `UserShows` WHERE show_id = :id AND user_id = :user_id";
12 try {
13     // Fetch user_id from session JN426 4/30/24
14     $user_id = get_user_id();
15
16     $stmt = $db->prepare($query);
17     $stmt->execute([":id" => $id, ":user_id" => $user_id]);
18     Flash("Deleted record with id $id from the watchlist", "success");
19 } catch (Exception $e) {
20     error_log("Error deleting show $id from watchlist: " . $e->getMessage());
21     Flash("Error deleting record", "danger");
22 }
23 redirect("admin/watchlist_associations.php");
24
25

```

File Edit Selection View Go Run ... ← →

File Explorer Taskbar

code behind deleting from watchlist

Checklist Items (0)

**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"/> #3	1	Mention the logic for single delete
<input type="checkbox"/> #4	1	Mention the logic for handling the username requirements

Response:

The result list in the code is determined by querying the database for shows that are associated with users. This is achieved by joining the UserShows table with the Shows table and the Users table, filtering out associations where a user is linked to a particular TV show. The list is further refined based on various filters such as username, title, genres, rating, and audience.

"Total Shows on Watchlists" counts all the associations stored in the UserShows table, providing an overview of the total number of TV shows currently on users' watchlist through displaying the total show count from a query counting it. On the other hand, "Total Items On Page" calculates the number of visible results based on applied filters, offering insight into the quantity of TV shows displayed after filtering. This is based on what the \$results number is because the \$results is what is on the page.

The logic for single deletion involves handling a POST request triggered by clicking the "Clear Associations" button. Upon submission, the code removes associations for the specified user, utilizing a DELETE query targeting the UserShows table.

The username parameter is retrieved from the query parameters using \$\_GET . The username filter is then applied within the SQL query's WHERE clause using a LIKE comparison, allowing for partial or full matches. If the username parameter is present in the GET request, the query is appended with a condition to filter results accordingly. The username was put as a button so that when the user clicks on it, it will send them to its profile.

#### 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://github.com/jennatnguyen/JN426-IT202-008/pull/90>

#### URL #2

[https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/watchlist\\_associations.php](https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/watchlist_associations.php)

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

### Task Screenshots:

#### Gallery Style: Large View

Small

Medium

Large

The screenshot shows a web application titled "Unassociated Shows". The URL in the address bar is [jn426-it202-dev-8332159f4410.herokuapp.com/project/admin/not\\_associated.php](http://jn426-it202-dev-8332159f4410.herokuapp.com/project/admin/not_associated.php). The page header includes "Show Tracker" and various navigation links like Home, Profile, My Shows, TV Shows, Admin, Show Management, Watchlist Management, and Logout.

**Unassociated Shows**

Total Shows not on Watchlists: 6

Total Items On Page: 6

Show Title	Show Rating	Show Audience	Genre(s)	Sort	Order	Limit
Show Title	Show Rating	Show Audience	Genre(s)	Title	+	10

**Shows**

title	imdb_rating	rated	genres	Actions
Abbott Elementary	7.724	TV-PG	Comedy	<button>View</button>
Custom Show 2		PG	Comedy	<button>View</button>
Kamp Koral: SpongeBob's Under Years	7.2	TV-Y7	Animation, Comedy, Family, Children, Fantasy	<button>View</button>
The SpongeBob SquarePants Anime	7	TV-14	Action, Adventure, Animation, Drama	<button>View</button>
Attack on Titan: Junior High	8.2	TV-14	Animation, Action, Adventure, Comedy, Anime	<button>View</button>

At the bottom, there is a taskbar with icons for search, file operations, and system status.

- Show the summary of the results with relevant information per entity
- Show the single view buttons/links
- Show variations of the number of shown items count and show the count of total number of associated items

### Checklist Items (0)

This screenshot shows the same "Unassociated Shows" page as the previous one, but it appears to be a different instance or a later state of the application. The URL is identical: [jn426-it202-dev-8332159f4410.herokuapp.com/project/admin/not\\_associated.php](http://jn426-it202-dev-8332159f4410.herokuapp.com/project/admin/not_associated.php). The page structure and data are the same, showing the table of shows with their respective details and "View" buttons.

Total Items On Page: 2

Show Title	Show Rating	Show Audience	Genre(s)	Sort	Order	Limit
Show Title	Show Rating	PG	Genre(s)	Title	+	10

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

## Shows

title	imdb_rating	rated	genres	Actions
Abbott Elementary	7.724	TV-PG	Comedy	<a href="#">View</a>
Custom Show 2		PG	Comedy	<a href="#">View</a>



## Variation sort 1

### Checklist Items (0)

Unassociated Shows

Total Shows not on Watchlists: 6

Total Items On Page: 0

Show Title	Show Rating	Show Audience	Genre(s)	Sort	Order	Limit
your mom	Show Rating	Show Audience	Genre(s)	Title	+	10

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

**Shows**

No records to show

no results

### Checklist Items (0)

Task #2 - Points: 1

Text: Screenshot the code

COLLAPSE

## Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the code related to fetching all unassociated entities (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	Show the logic related to the count of all unassociated items (even the ones not shown in the filtered results)
<input 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 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 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

The screenshot shows a terminal window with a dark theme. The title bar says "JN426-IT202-008". The code displayed is a PHP script for a search form. It includes logic for filtering by genre, rating, and order. The code is as follows:

```
<?php
require(__DIR__ . "/../../partials/nav.php");
if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    redirect("home.php");
}

//JN426 4/26/24
//build search form
$form = [
    "type" => "text", "name" => "title", "placeholder" => "Show Title", "label" => "Show Title", "include_margin" => false,
    "type" => "text", "name" => "imdb_rating", "placeholder" => "Show Rating", "label" => "Show Rating", "include_margin" => false,
    "type" => "text", "name" => "rated", "placeholder" => "Show Audience", "label" => "Show Audience", "include_margin" => false,
    "type" => "text", "name" => "genres", "placeholder" => "Genre(s)", "label" => "Genre(s)", "include_margin" => false,
    "type" => "select", "name" => "sort", "label" => "Sort", "options" => ["title" => "Title", "genres" => "Genre(s)", "imdb_rating" => "Rating", "rated" => "Rated"],
    "type" => "select", "name" => "order", "label" => "Order", "options" => ["desc" => "+", "desc" => "-"], "include_margin" => false,
    "type" => "number", "name" => "limit", "label" => "Limit", "value" => "10", "include_margin" => false,
];
-- $11-21 $form =
error_log("Form data: " . var_export($form, true));
$total_records = get_total_count("Shows" b WHERE b.id NOT IN (SELECT show_id FROM 'UserShows'"));
$Query = "SELECT s.id, title, imdb_id, imdb_rating, rated, genres FROM 'Shows' s
WHERE s.id NOT IN (SELECT show_id FROM 'UserShows') ";
$params = [];
$_SESSION['key'] = $_SERVER['SCRIPT_NAME'];
$_SESSION['clear'] = isset($_GET['clear']);
if ($_SESSION['clear']) {
    session_unset();
}
$_SESSION['selectedExecutionPlan'] = $plan;
```

The terminal window also shows a list of files in the current directory, including "not\_associated.php" which is currently selected. The bottom of the screen shows a taskbar with various icons and system status.

This code includes:  
-some of the code related to fetching all unassociated entities (including the query)  
-the logic related to the count of all unassociated items (even the ones not shown in the filtered results)

## Checklist Items (0)

```
public_html/project/admin/not_associated.php // not_associated.php > div.container-fluid > form > div.row.mb-3 > div.col
39
40 //JH26 4/26/24
41 if (count($_GET) == 0 && !isset($session_data) && count($session_data) > 0) {
42     if ($session_data) {
43         $_GET = $session_data;
44     }
45 } <- #41-45 if (count($_GET) == 0 && !isset($session_data) && count($session_data) > 0)
46 if (count($_GET) > 0) {
47     session_save($session_key, $_GET);
48     $keys = array_keys($_GET);
49
50     foreach ($form as $k => $v) {
51         if (in_array($v["name"], $keys)) {
52             $form[$k]["value"] = $_GET[$v["name"]];
53         }
54     } <- #50-54 foreach ($form as $k => $v)
55
56     //title
57     $title = se($_GET, "title", "", false);
58     if (!empty($title)) {
59         $query .= " AND title like :title";
60         $params[":title"] = "%$title%";
61     }
62
63     $genres = se($_GET, "genres", "", false);
64     if (!empty($genres) && $genres != "") {
65         $query .= " AND genres like :genres";
66         $params[":genres"] = "%$genres%";
67     }
68
69     $imdb_rating = se($_GET, "imdb_rating", "", false);
70     if (!empty($imdb_rating) && $imdb_rating != "") {
71         $query .= " AND imdb_rating like :imdb_rating";
72         $params[":imdb_rating"] = "%$imdb_rating%";
73     }
74
75     $rated = se($_GET, "rated", "", false);
76     if (!empty($rated) && $rated != "") {
77         $query .= " AND rated like :rated";
78         $params[":rated"] = "%$rated%";
79
80     }
81     //sort and order JH26 4/29/24
82     $sort = se($_GET, "sort", "created", false);
83     if (!in_array($sort, ["title", "genres", "imdb_rating", "rated"])) {
84         $sort = "created";
85     }
86     //tell MySQL I care about the data from table "b"
87     if ($sort === "created" || $sort === "modified") {
88         $sort = "b." . $sort;
89     }
90
91     $order = se($_GET, "order", "desc", false);
92     if (!in_array($order, ["asc", "desc"])) {
93         $order = "desc";
94     }
95
96     //IMPORTANT make sure you fully validate/trust $sort and $order (sql injection possibility)
97     $query .= " ORDER BY $sort $order";
98     //limit
99     try {
100         $limit = (int) se($_GET, "limit", "10", false);
101     } catch (Exception $e) {
102         $limit = 10;
103     }
104     if ($limit < 1 || $limit > 100) {
105         $limit = 10;
106     }
107     //IMPORTANT make sure you fully validate/trust $limit (sql injection possibility)
108     $query .= " LIMIT $limit";
109 } <- #46-109 if (count($_GET) > 0)
110
111
112 //*****
113 //JH26 4/26/24
114 $db = getDB();
115 $stmt = $db->prepare("SELECT * FROM movies WHERE $query");
116
117 //Get results
118 $stmt->execute();
119 $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
120
121 //Print results
122 $output = '';
123
124 //Loop through results
125 foreach ($results as $result) {
126     $output .= '

' . PHP_EOL;
127     $output .= '     0)
110
111
112
113 //***** //JN426 4/26/24
114 $db = getDB();
115 $stmt = $db->prepare($query);
116 $results = [];
117 try {
118     $stmt->execute($params);
119     $r = $stmt->fetchAll();
120     if ($r) {
121         $results = $r;
122     }
123 } catch (PDOException $e) {
124     error_log("Error fetching shows " . var_export($e, true));
125     flash("Unhandled error occurred", "danger");
126 }
127 //JN426 4/29/24
128 $table = [
129     "data" => $results, "title" => "Shows", "ignored_columns" => ["id", "imdb_id"],
130     "view_url" => get_url("admin/view_show2.php"),
131 ];
132
133
134 <div class="container-fluid">
135
136
137 <div style="text-align: center;">
138     <h2>Unassociated Shows</h2>
139     <h5>Total Shows not on Watchlists: <?php echo $total_records; ?></h5>
140     <h5>Total Items On Page: <?php echo count($results); ?></h5>
141 </div>
142     <form method="GET">
143         <div class="row mb-3" style="align-items: flex-end;">
You, 4 hours ago 57°F Cloudy 4:20 AM 4/30/2024
```

This code shows part 2 of the code related to fetching all unassociated entities (including the query)

## Checklist Items (0)

File Edit Selection View Go Run ... ← → JN426-IT202-008

Users watchlist\_associations.php not\_associated.php

public\_html>project>admin>not\_associated.php>div.container-fluid

```
126     flash("Unhandled error occurred", "danger");
127 }
128 //IM036 4/29/34
129 $table = [
130     "data" => $results, "title" => "Shows", "ignored_columns" => ["id", "imdb_id"],
131     "view_url" => get_url("admin/view_show2.php"),
132 ];
133 ?>
134
135 <div class="container-fluid">
136 |     You, 9 minutes ago * Uncommitted changes
137
138 <div style="text-align: center;">
139 |     <h2>Unassociated Shows</h2>
140 |     <h5>Total Shows not on Watchlists: <?php echo $total_records; ?></h5>
141 |     <h5>Total Items On Page: <?php echo count($results); ?></h5>
142 </div>
143 <form method="GET">
144     <div class="row mb-3" style="align-items: flex-end;">
145
146         <?php foreach ($rows as $k => $v) : ?>
147             <div class="col">
148                 <?php render_input($v); ?>
149             </div>
150         <?php endforeach; ?>
151
152     </div>
153     <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
154     <a href="?clear" class="btn btn-secondary">Clear</a>
155
156
157 </form>
158 <?php render_table($table); ?>
159 </div>
160
161 </?php
162
163 require_once(__DIR__ . "/../../../../partials/_flash.php");
164 ?>
```

MS-DOS All Users Page 3 hrs 57 mins IT202Server JN426

You, 9 minutes ago Ln 126, Col 1 Species: A UTF-8 CR/LF PHP Go Live

57°F Cloudy 4:22 AM 4/30/2024

This code shows the display of the results and the logic between the count of the items on the page as well as each record having a view page

## Checklist Items (0)

### 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 unassociated logic and filters)	
<input type="checkbox"/> #2	1	Mention the logic behind the two counts (total items and visible result count)	

### **Response:**

The code presents a page displaying TV shows that are not associated with any users, accomplished by querying the database for shows where their IDs do not exist in the UserShows table. It includes a form for filtering the results based on title, rating, audience, genres, sorting criteria, and limit. The page offers two counts: "Total Shows not on Watchlists" tallies the overall unassociated TV shows by counting records in the Shows table that lack associations in UserShows. Meanwhile, "Total Items On Page" counts the number of visible results based on applied filters. This design ensures users have insights into both the total pool of unassociated TV shows and the quantity displayed after filtering.

[^COLLAPSE ^](#)

## 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://github.com/jennatnguyen/JN426-IT202-008/pull/90>

### URL #2

[https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/not\\_associated.php](https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/not_associated.php)



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

[^COLLAPSE ^](#)

## Task #1 - Points: 1

**Text:** Screenshots of the page

### Checklist

\*The checkboxes are for your own tracking

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

### Task Screenshots:

#### Gallery Style: Large View

[Small](#)[Medium](#)[Large](#)

Users	Shows to Assign
<input type="checkbox"/> anothaonebiatch	<input type="checkbox"/> Abbott Elementary
	<input type="checkbox"/> Alvin and the Chipmunks

curaroiky

Attack on Titan

mother

Attack on Titan: Junior High

Custom Show 2

mt85

Game of Thrones

test1

Kamp Koral: SpongeBob's Under Years

test10

Shingeki no kyojin Attack on Titan: Hangeki no noroshi

SpongeBob SquarePants

### Show the search form with valid data (username)

#### Checklist Items (0)

Movie

Movies

Jn426-it...

Jn426-it...

Learn

View 1...

localhost

Z-index

[Milestone]

Learn

Show Tracker

Home Profile My Shows TV Shows Admin Show Management Watchlist Management Logout

## Assign Shows

Username Search

Search

Users	Shows to Assign
test1	Abbott Elementary
	Attack on Titan
test10	Attack on Titan: Junior High
test2	Breaking Bad
test20	Custom Show 2
test21	Game of Thrones
test3	Kamp Koral: SpongeBob's Under Years
	Shingeki no kyojin Attack on Titan: Hangeki no noroshi
	SpongeBob SquarePants

### Show the search form with valid data (show title)

#### Checklist Items (0)

Movie

Movies

Jn426-it...

Jn426-it...

Learn

View 1...

localhost

Z-index

[Milestone]

Learn

Show Tracker

Home Profile My Shows TV Shows Admin Show Management Watchlist Management Logout

## Assign Shows

test

Show Title Search

Search

Users	Shows to Assign
test	Abbott Elementary
	Attack on Titan
	Attack on Titan: Junior High
	Breaking Bad
	Custom Show 2
	Game of Thrones
	Kamp Koral: SpongeBob's Under Years
	Shingeki no kyojin Attack on Titan: Hangeki no noroshi
	SpongeBob SquarePants

- |                                 |   |
|---------------------------------|---|
| <input type="checkbox"/> test1  | <input type="checkbox"/> Abbott Elementary                                      |
| <input type="checkbox"/> test10 | <input type="checkbox"/> Attack on Titan  |
|                                 | <input type="checkbox"/> Attack on Titan: Junior High                           |
| <input type="checkbox"/> test2  | <input type="checkbox"/> Breaking Bad   |
|                                 | <input type="checkbox"/> Custom Show 2  |
| <input type="checkbox"/> test20 | <input type="checkbox"/> Game of Thrones  |
|                                 | <input type="checkbox"/> Kamp Koral: SpongeBob&#039;s Under Years               |
| <input type="checkbox"/> test21 | <input type="checkbox"/> Shingeki no kyojin Attack on Titan: Hangeki no noroshi |
| <input type="checkbox"/> test3  | <input type="checkbox"/> SpongeBob SquarePants                                  |



## results of username search

### Checklist Items (0)

The screenshot shows a web application titled "Assign Shows". At the top, there is a search bar with the placeholder "Username Search" and a button labeled "Search". Below the search bar, there are two columns: "Users" and "Shows to Assign".

**Users:**

- anothaonebiatch
- curaroiky
- mother
- mt85
- test1
- test10

**Shows to Assign:**

- Abbott Elementary
- Breaking Bad
- Kamp Koral: SpongeBob&#039;s Under Years
- SpongeBob SquarePants
- The SpongeBob SquarePants Anime

## result of show title search

### Checklist Items (0)

This screenshot shows the same "Assign Shows" page as the previous one, but with a different selection. In the "Users" column, the checkbox next to "test20" is checked. The rest of the users and shows listed are the same as in the previous screenshot.

- Breaking Bad
- Custom Show 2
- Game of Thrones
- Kamp Koral: SpongeBob's Under Years
- Shingeki no kyojin Attack on Titan: Hangeki no noroshi
- SpongeBob SquarePants
- The Sopranos
- The SpongeBob SquarePants Anime

Assign Shows



## Before

### Checklist Items (0)

result of entities and users being associated and unassociated

### Checklist Items (0)

The screenshot shows the MySQL Workbench interface. On the left is a tree view of database objects: UserRoles, columns, index, partitions, UserShows (selected), Users, Views, Procedures, and Functions. The main pane displays the UserShows table with the following data:

	id	user_id	show_id	created	modified
1	23	24	8	2024-04-30 07:10:34	2024-04-30 07:10:34
2	24	26	8	2024-04-30 09:59:44	2024-04-30 09:59:44
3	25	27	8	2024-04-30 10:02:22	2024-04-30 10:02:22
4					
5					

At the bottom, the task bar shows "MS3-Feat-Admin-Management" and the system status: 56°F Mostly cloudy, 6:57 AM, 4/30/2024.

before

### Checklist Items (0)

This screenshot is identical to the one above, showing the MySQL Workbench interface with the UserShows table containing 5 rows of data. The task bar at the bottom remains the same.

After

### Checklist Items (0)

**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**

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

### Checklist Items (0)

```

11  if (isset($_POST['users'])) {
12     $user_ids = $_POST['users'];
13     $show_ids = $_POST['shows'];
14     if (empty($user_ids) || empty($show_ids)) {
15         flash("Both users and shows need to be selected", "warning");
16     } else {
17         $db = getDB();
18         $stmt = $db->prepare("INSERT INTO UserShows (user_id, show_id) VALUES (:uid, :sid)
19             ON DUPLICATE KEY UPDATE user_id = VALUES(user_id)");
20         foreach ($user_ids as $uid) {
21             foreach ($show_ids as $sid) {
22                 try {
23                     // Check if the association already exists
24                     $stmt_check = $db->prepare("SELECT * FROM UserShows WHERE user_id = :uid AND show_id = :sid");
25                     $stmt_check->execute(":uid" => $uid, ":sid" => $sid);
26                     $existing_association = $stmt_check->fetch(PDO::FETCH_ASSOC);
27                     if ($existing_association) {
28                         // If the association exists, remove it
29                         $stmt_delete = $db->prepare("DELETE FROM UserShows WHERE user_id = :uid AND show_id = :sid");
30                         $stmt_delete->execute(":uid" => $uid, ":sid" => $sid);
31                         flash("Removed show association", "info");
32                     } else {
33                         // If the association doesn't exist, insert it
34                         $stmt->execute(":uid" => $uid, ":sid" => $sid);
35                         flash("Updated show association", "success");
36                     }
37                 } catch (PDOException $e) {
38                     flash(var_export($e->errorInfo, true), "danger");
39                 }
40             }
41         }
42     }
43 } <- #16-43 else

```

MSI-Test-Admin-Management 5 hrs 53 mins IT202Server 4/4/2024 You, 1 second ago Le 10, Col 52 Spaces: 4 UTF-8 CRLF PHP Go Live

Type here to search

Cloud 56°F Cloudy 628 AM 4/30/2024

## Code related to applying the associations upon submission

### Checklist Items (0)

```

File Edit Selection View Go Run ... ← → JN426-IT202-008
Users my_shows.php assign_shows.php M show_card.php delete_show_from_user.php delete_show_from_watchlist.php

```

public\_html > project > admin > assign\_shows.php ...

```

111 <?php
112 <?php if (!isset($username) || empty($username)) : ?>
113     <input type="hidden" name="username" value=<?php se($username, false); ?> />
114     <input type="hidden" name="show_title" value=<?php se($show_title, false); ?> />
115     <input type="hidden" name="show_id" value=<?php se($show_id, false); ?> />
116     <input type="checkbox" name="user_id" value=<?php se($user_id, false); ?> />
117     <table class="table">
118         <thead>
119             <tr>
120                 <th>Shows to Assign</th>
121             </tr>
122         </thead>
123         <tbody>
124             <tr>
125                 <td>
126                     <table class="table">
127                         <thead>
128                             <tr>
129                                 <th>User</th>
130                             <th>Shows to Assign</th>
131                         </tr>
132                         </thead>
133                         <tbody>
134                             <tr>
135                                 <td><?php foreach ($users as $user) : ?>
136                                     <?php render_input(['type' => 'checkbox', 'id' => 'user_' . se($user, 'id', '', false), 'name' => 'users[]', 'label' => se($user, 'username', '', false), 'value' => se($user, 'id', '', false)]; ?>
137                                 </td>
138                                 <td><?php foreach ($shows as $show) : ?>
139                                     <div>
140                                         <?php render_input(['type' => 'checkbox', 'id' => 'show_' . se($show, 'id', '', false), 'name' => 'shows[]', 'label' => se($show, 'title', '', false), 'value' => se($show, 'id', '', false)]; ?>
141                                     </div>
142                                     <?php endforeach; ?>
143                                 </td>
144                             </tr>
145                         </tbody>
146                     </table>
147                     <?php render_button(['text' => 'Assign Show', 'type' => 'submit', 'color' => 'secondary']); ?>
148                 </td>
149             </tr>
150         </tbody>
151     </table>
152     <?php endif; ?>
153     <?php if (isset($_POST['users']) && isset($_POST['shows'])) : ?>
154         <?php once(_DIR_ . "/partials/flash.php"); ?>
155     </?php endif; ?>

```

Note: we need to go up 1 more directory 20426 4/30/24 You, 2 minutes ago + uncommitted changes

git-rebase -i --edit /.../.../partials/flash.php

MSI-Test-Admin-Management 2 minutes ago Le 168, Col 57 Spaces: 4 UTF-8 CRLF PHP Go Live

Type here to search

Cloud 56°F Cloudy 630 AM 4/30/2024

## Code that generates the checkboxes next to each list (users and entities)+ Code related to submitting the checkbox lists

### Checklist Items (0)

```

File Edit Selection View Go Run ... ← → JN426-IT202-008
Users my_shows.php assign_shows.php M show_card.php delete_show_from_user.php delete_show_from_watchlist.php

```

public\_html > project > admin > assign\_shows.php ...

```

111 <?php
112 <?php if (isset($_POST['users']) && isset($_POST['shows'])) : ?>
113     <?php once(_DIR_ . "/partials/flash.php"); ?>
114     <?php endif; ?>

```

MSI-Test-Admin-Management 5 hrs 53 mins IT202Server 4/4/2024 You, 2 minutes ago + uncommitted changes

Type here to search

Cloud 56°F Cloudy 628 AM 4/30/2024

```

74 // Search for show by title if provided
75 $show_title = "";
76 if (isset($_POST["show_title"])) {
77     $show_title = $_POST["show_title"];
78     if (!empty($show_title)) {
79         $db = getDB();
80         $stmt = $db->prepare("SELECT id, title FROM Shows WHERE title LIKE :title LIMIT 25");
81         try {
82             $stmt->execute([":title" => "%$show_title%"]);
83             $shows = $stmt->fetchAll(PDO::FETCH_ASSOC);
84         } catch (PDOException $e) {
85             flash(var_export($e->errorInfo, true), "danger");
86         }
87     } <- #98-107 if (!empty($show_title))
88 } <- #96-100 if (isset($_POST["show_title"]))
89 ?>
90
91 <div class="container-fluid">
92     <h1>Assign Shows</h1>
93     <form method="POST">
94         <div class="row mb-3">
95             <div class="col">
96                 <?php render_input(["type" => "search", "name" => "username", "placeholder" => "Username Search", "value" => $username]); ?>
97             </div>
98             <div class="col">
99                 <?php render_input(["type" => "search", "name" => "show_title", "placeholder" => "Show Title Search", "value" => $show_title]); ?>
100            </div>
101        </div>
102        <div class="row mb-3">
103            <div class="col">
104                <?php render_button(["text" => "Search", "type" => "submit"]); ?>
105            </div>
106        </div>
107    </form>
108 </div>
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125

```

M53-Feat-Admin-Management\* 0 0 0 0 5 hrs 56 mins IT202Server js426

You, 4 minutes ago Ln 94, Col 54 Spaces: 4 UTF-8 CRLF PHP Go Live

56°F Cloudy 6:32 AM 4/30/2024

## Search form field for finding a partial match of usernames + Search form field for finding a partial match of entities

### Checklist Items (0)

#### 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 code implements a feature enabling the assignment of shows to users. It incorporates search functionality for both usernames and show titles. When users input a partial username or show title and submit the form, PHP retrieves the input from the `$_POST` array, executes SQL queries to fetch matching users and shows respectively, and displays them in separate tables on the UI. The UI comprises search input fields for both criteria, accompanied by "Search" buttons, and two side-by-side tables showcasing users and shows. Each table row represents a user or show, with checkboxes for selection. Upon clicking the "Assign Shows" button, selected user-show associations are stored in the database. The logic checks for existing associations and either creates new ones or removes them based on user actions.

#### 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://github.com/jennatnguyen/JN426-IT202-008/pull/92>

### URL #2

[https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/assign\\_shows.php](https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/assign_shows.php)

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 "Milestones - JN426". The board has three columns: "Todo" (0 items), "In Progress" (0 items), and "Done" (20 items). The "Done" column contains four completed tasks:

- JN426-IT202-008 #93 MS3-Admin can associate any entity with any users
- JN426-IT202-008 #91 MS3- Create a page that shows data not associated with any user
- JN426-IT202-008 #89 MS3-All Users association page
- JN426-IT202-008 #86

The browser's address bar shows the URL: <https://github.com/jennatnguyen/JN426-IT202-008/issues/93>.

## Task #2 - Points: 1

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

URL #1

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

## Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

My main issues were more with the SQL side. I had trouble determining how to word the queries so that it would output the right data. A lot of errors were thrown in the process of putting this together. I also found that I had a huge time restraint during this because of how long the other Milestones took that I could not have too much time to really mess around with the code and make it more unique, which is something I wanted to do.

## Task #4 - Points: 1

Text: WakaTime Screenshot

### Details:

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a WakaTime dashboard with a sidebar containing links like Upgrade, Dashboard, Insights, Teams, Projects, Invoices, Goals, Shareables, Leaderboards, and Integrations. The main area displays a table of tracked files and branches. The table has three columns: 'Files' (listing file paths), 'Branches' (listing branch names), and a third column that is partially visible. The 'Files' column includes entries such as '4 hrs 5 mins ...project/admin/view\_show.php', '2 hrs 51 mins ...roject/admin/list\_shows.php', '2 hrs 21 mins ...ject/admin/create\_show.php', '2 hrs 21 mins .../watchlist\_associations.php', '1 hr 18 mins ...o\_html/project/my\_shows.php', '1 hr 9 mins ...user\_show\_associations.php', '1 hr 6 mins partials/show\_card.php', '1 hr 2 mins ...roject/admin/edit\_show.php', '59 mins public\_html/project/shows.php', '56 mins lib/show\_api.php', '49 mins ...roject/single\_view\_show.php', '48 mins ...ject/admin/assign\_shows.php', '42 mins partials/show\_card\_list.php', '42 mins ...ct/admin/not\_associated.php', '32 mins ...ject/admin/delete\_show.php', and '27 mins ...html/project/fetch\_show.php'. The 'Branches' column includes entries such as '8 hrs 9 mins MS2-API-Integration-Admin', '5 hrs 11 mins MS3-Feat-Associated-Entities-Page', '4 hrs 43 mins MS3-Feat-All-Users-Page', '2 hrs 35 mins MS2-Feat-Create', '2 hrs 3 mins MS3-Feat-Handle-Data-Association', '1 hr 10 mins MS3-Feat-Admin-Management', '55 mins Milestone2', '40 mins MS2-Feat-Edit-Page', and '2 mins Unknown'.

Files	Branches
4 hrs 5 mins ...project/admin/view_show.php	8 hrs 9 mins MS2-API-Integration-Admin
2 hrs 51 mins ...roject/admin/list_shows.php	5 hrs 11 mins MS3-Feat-Associated-Entities-Page
2 hrs 21 mins ...ject/admin/create_show.php	4 hrs 43 mins MS3-Feat-All-Users-Page
2 hrs 21 mins .../watchlist_associations.php	2 hrs 35 mins MS2-Feat-Create
1 hr 18 mins ...o_html/project/my_shows.php	2 hrs 3 mins MS3-Feat-Handle-Data-Association
1 hr 9 mins ...user_show_associations.php	1 hr 10 mins MS3-Feat-Admin-Management
1 hr 6 mins partials/show_card.php	55 mins Milestone2
1 hr 2 mins ...roject/admin/edit_show.php	40 mins MS2-Feat-Edit-Page
59 mins public_html/project/shows.php	2 mins Unknown
56 mins lib/show_api.php	
49 mins ...roject/single_view_show.php	
48 mins ...ject/admin/assign_shows.php	
42 mins partials/show_card_list.php	
42 mins ...ct/admin/not_associated.php	
32 mins ...ject/admin/delete_show.php	
27 mins ...html/project/fetch_show.php	

Supported IDEs

Blog  
API Docs  
Pricing  
Community  
FAQ

19 mins partials/nav.php  
17 mins \_t/delete\_show\_from\_user.php  
17 mins \_ic\_html/project/profile.php  
11 mins partials/single\_show\_card.php  
10 mins public\_html/project/styles.css  
9 mins lib/render\_functions.php  
9 mins \_ete\_show\_from\_watchlist.php  
9 mins lib/db\_helpers.php  
8 mins \_roject/admin/list\_roles.php



Lots of time spent

End of Assignment