

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-3-2024/grade/df39>

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

Submissions:

Submission Selection

1 Submission [active] 4/28/2024 9:05:43 PM

Instructions

[▲ COLLAPSE ▲](#)

Implement the Milestone 3 features from the project's proposal

document: <https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88E>

Make sure you add your ucid/date as code comments where code changes are done

All code changes should reach the Milestone3 branch

Create a pull request from Milestone3 to dev and keep it open until you get the output PDF from this assignment.

Gather the evidence of feature completion based on the below tasks.

Once finished, get the output PDF and copy/move it to your repository folder on your local machine.

Run the necessary git add, commit, and push steps to move it to GitHub

Complete the pull request that was opened earlier

Create and merge a pull request from dev to prod

Upload the same output PDF to Canvas

Branch name: Milestone3

Tasks: 26 Points: 10.00

 API (1 pt.)

[▲ COLLAPSE ▲](#)

 Task #1 - Points: 1

Text: Data Related to Users

Checklist

*The checkboxes are for your own tracking

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

Response:

So the data related to the users is movies and movies they have favorited. The relationship between the users and the movies is many to many. This is because many users can favorite the same movies.

Task #2 - Points: 1

Text: Updating Entities

Checklist

*The checkboxes are for your own tracking

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

Response:

If data is updated manually or from the API the users will automatically see the new data. The data does not need to be reassociated.

Handle Data Association (1 pt.)

[ACOLLAPSE](#)

Task #1 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

```

1 | <!-- //DF39 4/29/2024 -->
2 | <?php if(!isset($store)) {
3 | 
4 |     error_log("Using Movie Partial without data!");
5 |     flash("Dev Alert: Movie called without data", "danger");
6 | }
7 | ?>
8 | <?php if(isset($store)):?>
9 | <!-- https://i.kym-cdn.com/entries/icons/original/000/829/959/Screen_Shot_2019-06-05_at_1.26.32_PM.jpg -->
10 | <div class="card mx-auto" style="width: 10rem;">
11 | <?php if (isset($store["username"])) : ?>
12 |     <div class="card-header">
13 |         favorited By: <?php set$broker, "username", "N/A"; ?>
14 |     </div>
15 |     <?php endif; ?>
16 |     <?-- <img src="" class="card-img-top" alt="..."/> -->
17 |     <div class="card-body">
18 |         <h5 class="card-title"><?php set$broker, "title", "Unknown"; ?> </h5>
19 |         <div class="card-text">
20 |             <ul class="list-group">
21 |                 <li class="list-group-item">Movie Title: <?php set$broker, "title", "Unknown"; ?></li>
22 |                 <li class="list-group-item">Year: <?php set$broker, "year", "Unknown"; ?></li>
23 |                 <li class="list-group-item">ID: <?php set$broker, "id", "Unknown"; ?></li>
24 |             </ul>
25 |
26 |             @var never $broker
27 |             <?php if (isset($broker["id"])) : ?>
28 |                 <a class="btn btn-secondary" href=<?php echo get_url('user_movies.php?id=' . $broker["id"]); ?>>View</a>
29 |             </?php endif; ?>
30 |             <?php if (isset($broker["user_id"])) || $broker["user_id"] == "N/A" : ?>
31 |                 <div class="card-body">
32 |                     <a href=<?php echo get_url('api/favorite_movie.php?movie_id=' . $broker["id"]); ?>> class="card-link">Favorite Movie</a>
33 |                 </div>
34 |             </?php else : ?>
35 |                 <div class="card-body">
36 |                     <a href="#">John echo get_url('profile.php?id=' . $broker["user_id"]); ?>> John set$broker, "username", "N/A"; ?>< Profile</a>
37 |

```

This is my movies.php page where the user has a list of the movies in the database with a sort/filter function at the top of the page and the ability to favorite and unfavorate movies.

Checklist Items (3)

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

#2 Include ucid/date comments for each code screenshot

#3 Clearly caption screenshots

```

1 | <!-- //DF39 4/29/2024 -->
2 | <?php if(!isset($broker)) {
3 | 
4 |     error_log("Using Movie Partial without data!");
5 |     flash("Dev Alert: Movie called without data", "danger");
6 | }
7 | ?>
8 | <?php if(isset($broker)):?>
9 | <!-- https://i.kym-cdn.com/entries/icons/original/000/829/959/Screen_Shot_2019-06-05_at_1.26.32_PM.jpg -->
10 | <div class="card mx-auto" style="width: 10rem;">
11 | <?php if (isset($store["username"])) : ?>
12 |     <div class="card-header">
13 |         favorited By: <?php set$broker, "username", "N/A"; ?>
14 |     </div>
15 |     <?php endif; ?>
16 |     <?-- <img src="" class="card-img-top" alt="..."/> -->
17 |     <div class="card-body">
18 |         <h5 class="card-title"><?php set$broker, "title", "Unknown"; ?> </h5>
19 |         <div class="card-text">
20 |             <ul class="list-group">
21 |                 <li class="list-group-item">Movie Title: <?php set$broker, "title", "Unknown"; ?></li>
22 |                 <li class="list-group-item">Year: <?php set$broker, "year", "Unknown"; ?></li>
23 |                 <li class="list-group-item">ID: <?php set$broker, "id", "Unknown"; ?></li>
24 |             </ul>
25 |
26 |             @var never $broker
27 |             <?php if (isset($broker["id"])) : ?>
28 |                 <a class="btn btn-secondary" href=<?php echo get_url('user_movies.php?id=' . $broker["id"]); ?>>View</a>
29 |             </?php endif; ?>
30 |             <?php if (isset($broker["user_id"])) || $broker["user_id"] == "N/A" : ?>
31 |                 <div class="card-body">
32 |                     <a href=<?php echo get_url('api/favorite_movie.php?movie_id=' . $broker["id"]); ?>> class="card-link">Favorite Movie</a>
33 |                 </div>
34 |             </?php else : ?>
35 |                 <div class="card-body">
36 |                     <a href="#">John echo get_url('profile.php?id=' . $broker["user_id"]); ?>> John set$broker, "username", "N/A"; ?>< Profile</a>
37 |

```

```

38
39
40
41         </h3>
42         <a href="php echo get_url('api/unfavorite_movie.php?movie_id=' . $broker['id']); ?" class="card-link">Unfavorite Movie</a>
43     </div>
44   </div>
45 <?php endif; ?>

```

This is my movie card.php file that has the favorite button that is used in movies.php where the user can favorite their movie of choice.

Checklist Items (3)

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

#2 Include ucid/date comments for each code screenshot

#3 Clearly caption screenshots

```

62 //DF39 4/29/2024
63 //sort and order
64 $sort = isset($_GET['sort'], 'name', false);
65 if (!in_array($sort, ['title', 'year'])) {
66     $sort = 'name';
67 }
68 $order = isset($_GET['order'], 'desc', false);
69 if (!in_array($order, ['asc', 'desc'])) {
70     $order = 'desc';
71 }
72 //IMPORTANT make sure you fully validate/trust sort and order (sql injection possibility)
73 $query .= " ORDER BY $sort $order";
74 //limit
75 try {
76     $limit = (int) $_GET['limit', '10', false];
77 } catch (Exception $e) {
78     $limit = 10;
79 }
80 if ($limit < 1 || $limit > 100) {
81     $limit = 10;
82 }
83 //IMPORTANT make sure you fully validate/trust limit (sql injection possibility)
84 $query .= " LIMIT $limit";
85 }
86 //DF39 4/29/2024
87 $db = getDB();
88 $stmt = $db->prepare($query);
89 $results = [];
90 try{
91     $stmt->execute($params);
92     $r = $stmt->fetchAll();
93     if($r)
94     {
95         $results = $r;
96     }
97     else{
98         flash("No Results Found From Filter", "warning");
99     }
100 } catch (\PDOException $e) {
101     error_log("Error Fetching Movies" . var_export($e, true));
102     flash("Unable to Find Movie ID", "danger");
103 }
104 if (empty($results)) {
105     flash("No Movies Found With The Used Filters", "danger");
106 }
107 $table = [
108     'data' => $results, 'title' => 'Latest Movies', 'ignored_columns' => ['id'],
109     'view_url' => get_url('user_movies.php'),
110     'favorite_url' => get_url('api/favorite_movie.php')
111 ];
112 
```

This is the second part of movies.php

Checklist Items (3)

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

#2 Include ucid/date comments for each code screenshot

#3 Clearly caption screenshots

```

<!-- DF39 4/19/2024 -->
<div class="container-fluid">
    <h2>Movies </h2>

```

```

<!--> Movies-->
<form method="GET">
    <div class="row mb-3" style="align-items: flex-end;">
        <?php foreach ($form as $k => $v) : ?>
            <div class="col">
                <?php render_input($v); ?>
            </div>
        <?php endforeach; ?>
    </div>
    <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
    <a href="#clear" class="btn btn-secondary">Clear</a>
</form>
<?php render_result_counts(count($results), $total_records); ?>
<div class="row w-100 row-cols-auto row-cols-sm-1 row-cols-md-2 row-cols-lg-3 row-cols-xl-4 row-cols-xxl-5 g-4">
    <?php foreach ($results as $broker) : ?>
        <div class="col">
            <?php render_movie_card($broker); ?>
        </div>
    <?php endforeach; ?>
    <?php if(count($results) === 0) : ?>
        <div class="col">
            No Results to Show
        </div>
    <?php endif; ?>
    </div>
</div>

<?php
require_once(__DIR__ . "/../../../../partials/flash.php");
?>

```

This is the form of movies.php that calls the movie card containing the favorite button.

Checklist Items (3)

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

#2 Include ucid/date comments for each code screenshot

#3 Clearly caption screenshots

Task #2 - Points: 1

Text: Screenshot of the association table(s)

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows the MySQL Workbench interface. On the left, the database tree for 'IT202-008 8.0.36-0ubuntu1' is visible, including tables like Movies, Roles, and UserMovies. The central area shows the 'Properties' tab selected, displaying the schema for the 'UserMovies' table:


```
CREATE TABLE `UserMovies` (
      `id` int NOT NULL,
      `user_id` int NOT NULL,
      `movie_id` int NOT NULL,
      `is_active` tinyint(1) NOT NULL,
      `created` timestamp NOT NULL,
      `modified` timestamp NOT NULL
    )
```

 Below this is the 'DATA' tab, showing the results of the query:


```
SELECT * FROM `UserMovies` LIMIT 100
```

 The results grid contains 6 rows of data:

	id	user_id	movie_id	is_active	created	modified
> 1	63	22	529	1	2024-04-28 19:45:25	2024-04-28 19:45:25
> 2	64	22	532	1	2024-04-28 19:45:29	2024-04-28 19:45:29
> 3	66	26	660	1	2024-04-28 23:30:08	2024-04-28 23:30:08
> 4	71	26	632	1	2024-04-29 00:18:40	2024-04-29 00:18:40
> 5	72	26	656	1	2024-04-29 00:20:09	2024-04-29 00:20:09
> 6	78	11	526	1	2024-04-29 16:49:52	2024-04-29 16:49:52

This is my table UserMovies that associates the user_id from the Users table to the movie_id from the Movies table.

Checklist Items (3)

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

#2 Should have some example data

#3 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain solution

Checklist

*The checkboxes are for your own tracking

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

Response:

I chose option 1 which was to create a page (movies.php) that any logged in user can see. In this page there is a list of movies with a filter/sort function at the top of the page. Under each movie in the list there is a favorite button that is presented. This favorite button is being rendered by movie_card.php which will have a view button and a favorite button no matter where it is called. Clicking this button will call favorite.php which is going to take the users_id and the movie_id of the movie the user clicked the favorite button under and insert it into UserMovies which is the table of the associated data between the user and the movies. This table has 6 columns movie_id, user_id, id, created, modified and is_active. Currently i use every column except for is_active which is what i copied from the UserRoles table. The

and is_active. Currently I use every column except for is_active which is what I copied from the UserRoles table. The movie_id column is the id of the movie passed from the Movies table, user_id is the users id passed in from the Users table and created/modifies is when the row was created and or modified. In my case the column gets created but not modified unless it is deleted from the user unfavoriting the movie.

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://df39-it202-008-prod-8b39da6847aa.herokuapp.com/Project/movies.php>

URL #2

<https://github.com/dunnfall/df39-it202-008/pull/66>

Current User's Association Page (2 pts.)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

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

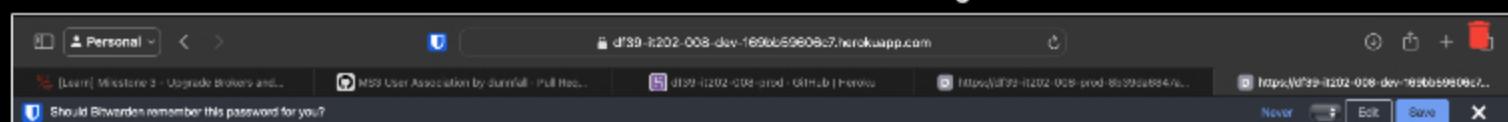
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



My Favorited Movies

Remove All Favorited Movies

Title	Year	Sort	Order	Limit
<input type="text" value="Title"/>	<input type="text" value="Year"/>	<input type="text" value="Sort"/> Title	<input type="button" value="+"/>	<input type="text" value="10"/>

Filter Clear

Results: 3/570

The Scarface Mob

Movie Title: The Scarface Mob

Year: 1969

ID: 535

[View](#)

[Unfavorite Movie](#)

The Making of the Lion King

Movie Title: The Making of the Lion King

Year: 1994

ID: 544

[View](#)

[Unfavorite Movie](#)

An Awesome Action Movie

Movie Title: An Awesome Action Movie

Year: 2023

ID: 545

[View](#)

[Unfavorite Movie](#)

This screenshot shows the page with all my users favorite movies with the movie cards and the link to unfavorite and the link to view a single card. This also shows my heroku dev link in the url bar.

Checklist Items (1)

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

Personal < >  d39-k202-008-dev-1696659609cf.herokuapp.com    

File [Open] Milestones 3 - Upgrade Brokers and ... GitHub User Association by Jurnalit - Pull Requ... d39-k202-008-prod - GitHub | Heroku https://d39-k202-008-prod-8b591da8544a... https://d39-k202-008-dev-1696659609cf... Heroku Edit Beta X

Movie Watch Home Profile Create/Fetch Movies Favorite Movies Non-Favorite Movies Movies Admin Logout

No Results Found From Filter

My Favorited Movies

Remove All Favorited Movies

Title	Year	Sort	Order	Limit
<input type="text" value="y"/>	<input type="text" value="Year"/>	<input type="text" value="Sort"/> Title	<input type="button" value="+"/>	<input type="text" value="10"/>

Filter Clear

Results: 0/570

No Results to Show

This shows no results from the filter/sort and a friendly message, with count showing no results as well

Checklist Items (1)

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

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "My Favorited Movies". At the top, there is a search bar with "Scar" entered, and a dropdown menu set to "Title". Below the search bar are buttons for "Filter" and "Clear". The main content area displays a single search result for "The Scarface Mob". The result includes the movie title, year (1989), ID (535), and a "View" button. There is also a link to "Unfavorite Movie". The background of the page is yellow.

This shows filter/results showing one result and count showing 1 item on the page

Checklist Items (3)

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

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

#5 Make sure the heroku dev url is visible in the address bar

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the code related to fetching the user's associations (including the query)
<input checked="" type="checkbox"/> #2	1	Show the code related to the display of the results
<input checked="" type="checkbox"/> #3	1	Each record should have a button/link for single view
		Each record should have a button/link for delete (this may be an admin-only thing but

#4	1	should be present for the specific role) Note: this is to delete the relationship and not the specific entity
#5	1	Show the logic for deleting all associations for the user (this may be admin-only but should be present for the specific role)
#6	1	Show the logic related to the count of all associated items to the user (even the ones not shown in the filtered results)
#7	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
#8	1	Show the logic related to filter/sort (limit should be constrained to 1-100 otherwise default to 10)
#9	1	Include ucid/date comments for each code screenshot
#10	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

This screen shot shows the query of the users associated data coming from the UserMovies table. This screenshot also shows the logic for removing all favorited movies from the UserMovies database.

Checklist Items (1)

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

```

76     //year
77     $year = $this->GET('year', 10, 'int');
78     if (!empty($year)) {
79         $query .= " AND year LIKE year'{$year}'";
80     }
81
82     //sort and order
83     $sort = $this->GET('sort', 'rand', 'int');
84     if ($sort >= 0) {
85         $sort = "score{$sort}";
86     }
87
88     $order = $this->GET('order', 'desc', 'int');
89     if ($order >= 0) {
90         $order = "desc{$order}";
91     }
92
93     //DEPRECATE me! use your fully validated sort and order (sql injection possibility)
94     $query .= " ORDER BY {$sort} {$order}";
95
96     try {
97         $stmt = $this->prepare($query);
98         $stmt->execute();
99         $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
100
101         $stmt->close();
102         $stmt = null;
103
104         $count = count($results);
105         if ($count < 1 || $count > 100) {
106             $count = 0;
107         }
108
109         //DEPRECATE me! use your fully validated limit (sql injection possibility)
110         $query .= " LIMIT {$count}";
111
112     } catch (\PDOException $e) {
113         $error_log("Error Fetching Novels", 'var_export($e, true)');
114         $error_log("Please make sure you have MySQL 2.1+, 'danger'!");
115     }
116
117     $results = [
118         'total' => $count,
119         'title' => 'Recent Novels',
120         'ignored_columns' => (isOff),
121         'view_url' => get_lrU('user_novels.php'),
122         'reverse_url' => get_lrU('user_novels_reverse.php'),
123     ];
124
125     return $results;
126 }

```

This screenshot shows the code for the logic of filtering and sorting based on title and year.

Checklist Items (3)

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

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

```

136 <!-- DF39 4/19/2024 -->
137 <div class="container-fluid">
138     <h3>My Favorited Novels</h3>
139     <div>
140         <a href="#">?remove</a> onclick="confirm('Are You Sure You Want To Remove All Favorites?')?event.preventDefault(): Remove All Favorited Movies</a>
141     </div>
142     <form method="GET">
143         <div class="row mb-3" style="align-items: flex-end;">
144
145             <?php foreach ($form as $k => $v) : ?>
146                 <div class="col">
147                     <?php render_input($v); ?>
148                 </div>
149             <?php endforeach; ?>
150
151         </div>
152         <?php render_button(['text' => 'Search', 'type' => 'submit', 'text' => 'Filter']); ?>
153         <a href="#">?clear</a> class="btn btn-secondary">Clear</a>
154     </form>
155     <?php render_result_counts(count($results), $total_records); ?>
156     <div class="row w-100 row-cols-auto row-cols-sm-1 row-cols-md-2 row-cols-lg-3 row-cols-xl-4 row-cols-xxl-5 g-4">
157         <?php foreach ($results as $broker) : ?>
158             <div class="col">
159                 <?php render_favorite_movie_card($broker); ?>
160             </div>
161         <?php endforeach; ?>
162         <?php if(count($results) === 0): ?>
163             <div class="col">
164                 No Results to Show
165             </div>
166         <?php endif; ?>
167     </div>
168 </div>
169
170 <?php
171 require_once(__DIR__ . '../../../../../partials/_flash.php');
172 ?>

```

This screenshot shows the code for how the data is displayed to the user. It does so using the favorite_movie_card.php file

Checklist Items (3)

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

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

```
partials > 📄 result_counts.php > ...
1
2  <?php
3  //DF39 4/29/2024
4  if (!isset($result_count)) {
5      $message = "Dev Note: result_count not set";
6      error_log($message);
7      flash($message);
8      $result_count = -1;
9  }
10 if (!isset($total_count)) {
11     $message = "Dev Note: total_count not set";
12     error_log($message);
13     flash($message);
14     $total_records = -1;
15 }
16 ?>
17 <header>
18     <div class="row">
19         <div class="col-1">Results: <?php se("$result_count/$total_count"); ?></div>
20     </div>
21 </header>
```

This code shows the logic of the count results of items associated to the user.

Checklist Items (2)

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

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

```
partials > 📄 movie_card.php > ...
1  <!-- /DF39 4/29/2024 -->
2  <?php if (!isset($broker)) {
3  {
4      error_log("Using Movie Partial without data");
5      flash("Dev Alert: Movie called without data", "danger");
6  }
7 ?>
8 <?php if (!isset($broker)) :?>
9 <! -- https://i.kym-cdn.com/entries/icons/original/000/029/959/Screen_Shot_2019-06-05_at_3.26.32_PM.jpg -->
10 <div class="card mx-auto" style="width: 10rem;">
11 <?php if (!isset($broker['username'])) :?>
12     <div class="card-header">
13         Favorited by: <?php se($broker, "username", "N/A"); ?>
14     </div>
15     <?php endif; ?>
16     <div style="background-color: #f0f0f0; padding: 10px; border-radius: 5px; margin-top: 10px;">
```

```

1<div class="card card-inverse" style="background-color: #f2f2f2; border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;>
2    <div class="card-body">
3        <h5 class="card-title"><?php set$broker, "title", "Unknown"; ?></h5>
4        <div class="card-text">
5            <ul class="list-group">
6                <li class="list-group-item">Movie Titles <?php se($broker, "title", "Unknown"); ?></li>
7                <li class="list-group-item">Years <?php se($broker, "year", "Unknown"); ?></li>
8                <li class="list-group-item">IDs <?php se($broker, "id", "Unknown"); ?></li>
9            </ul>
10        </div>
11    </div>
12    <?php if (isset($broker["id"])) : ?>
13        <a class="btn btn-secondary" href="<?php echo get_url("user_movies.php?id" . $broker["id"]); ?>">View</a>
14        <?php endif; ?>
15    <?php if (!isset($broker["user_id"]) || $broker["user_id"] == "N/A") : ?>
16        <div class="card-body">
17            <a href="<?php echo get_url("api/favorite_movie.php?movie_id" . $broker["id"]); ?>" class="card-link">Favorite Movie</a>
18        </div>
19    <?php else : ?>
20        <div class="card-body">
21            <a href="<?php echo get_url("profile.php?id" . $broker["user_id"]); ?>"><?php se($broker, "username", "N/A"); ?>'s Profile</a>
22            <br>
23            <h3>
24                <a href="<?php echo get_url("api/unfavorite_movie.php?movie_id" . $broker["id"]); ?>" class="card-link">Unfavorite Movie</a>
25            </h3>
26        </div>
27    <?php endif; ?>
28</div>
29<?php endif; ?>

```

This code shows the file favorite_movie_card.php which displays the single view button and unfavorite (delete) link.

Checklist Items (2)

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

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

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

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

Response:

When a user favorites a movie, the user_id and movie_id is stored in the new table UserMovies. With this said to be able to view the movies favorited by the user there is a query of the UserMovies table where it searches for all records matching the current users user_id. With this query you then join the Movies table and the matching records of the user_id tied to those movie_ids would get passed into the Movies table to pull the information tied to the movie_ids. When a user deletes (unfavorites) a movie they get checked for their user_id and the movie_id of the movie they are trying to delete. Then a delete sql query is prepared and executed to delete the movie associated with the user_id from the UserMovies table. The logic for the count works by calling the get_total_count function which takes a string as the sql query of the tables. The function then calls the UserMovies table and Movies table. In this case it fetches the count of all the movies in the database and all the movies that are favorited from the user in the UserMovies database and displays the results to the user using the render function.

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/dunnfall/df39-it202-008/pull/66>

URL #2

https://df39-it202-008-prod-8b39da6847aa.herokuapp.com/Project/my_movies.php

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

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<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 interface for managing movie associations. At the top, there is a navigation bar with links for Personal, Learn, and Logout. Below the navigation is a header titled "Movie Association". A search/filter form is present, featuring fields for Username, Title, Year, Sort, Order, Limit, and buttons for Filter and Clear. The main content area displays a table with columns for Title, Year, and Sort. The table contains several rows of data, with the first row being highlighted in yellow.

Title	Year	Sort
Movie 1	2022	Original
Movie 2	2023	Revised
Movie 3	2021	Original
Movie 4	2022	Revised
Movie 5	2023	Original

Results:
11/50

Favorited By: admin Toy Story 400 Movie Title: Toy Story 400 Year: 2019 ID: 526 View admin's Profile Unfavorite Movie	Favorited By: admin The Scarface Mob Movie Title: The Scarface Mob Year: 1959 ID: 535 View admin's Profile Unfavorite Movie	Favorited By: admin The Making of the Lion King Movie Title: The Making of the Lion King Year: 1994 ID: 544 View admin's Profile Unfavorite Movie	Favorited By: admin An Awesome Action Movie Movie Title: An Awesome Action Movie Year: 2023 ID: 545 View admin's Profile Unfavorite Movie	Favorited By: prodbyfall Toy Story Movie Title: Toy Story Year: 1995 ID: 529 View prodbyfall's Profile Unfavorite Movie
Favorited By: prodbyfall Scarface Movie Title: Scarface Year: 1983	Favorited By: dunfall Charlie: A Toy Story Movie Title: Charlie: A Toy Story Year: 2010	Favorited By: dunfall 1 Billion Poops Movie Title: 1 Billion Poops Year: 2018	Favorited By: dunfall 100 Million BC Movie Title: 100 Million BC Year: 2008	Favorited By: dunfall 'A Muzzarell' Movie Title: 'A Muzzarell' Year: 2024

This screen shot shows the movies favorited by different users with the username at the top of the cards as well as the view button, the profile link and the unfavorite link. The heroku dev link is in the url bar as well.

Checklist Items (5)

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

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

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

#6 Make sure the heroku dev url is visible in the address bar

#7 Clearly caption screenshots

The screenshot shows a web application titled "Movie Association". At the top, there is a navigation bar with links for "Movie Watch", "Home", "Profile", "Create/Patch Movies", "Favorite Movies", "Non-Favorite Movies", "Movies", "Admin", and "Logout". Below the navigation bar, there is a search form with fields for "Username" (set to "admin"), "Title", "Year", "Sort" (set to "Title"), "Order" (set to "+"), and "Limit" (set to "10"). There are also "Filter" and "Clear" buttons. The results section displays four movie cards, each with its title, year, ID, a "View" button, and links to "admin's Profile" and "Unfavorite Movie". The cards are: "An Awesome Action Movie" (Year: 2023, ID: 545), "The Making of the Lion King" (Year: 1994, ID: 544), "The Scarface Mob" (Year: 1959, ID: 535), and "Toy Story 400" (Year: 2019, ID: 526). The URL in the browser's address bar is "https://M39-t202-008-dev-169bb69606c7.herokuapp.com".

This screen shot shows a filter of favorited movies by the admin user only. It also shows the count being 4 results of 570 movies in the movies table.

Checklist Items (1)

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

No Results Found From Filter

Movie Association

Username	Title	Year	Sort	Order	Limit
kjsdinc	Title	Year	Title	+	10

Filter Clear

Results:
0/570
No Results to Show

This screenshot shows the filter with no results due to a username that is not in the Users table. It also shows the count being 0 for no results with a message

Checklist Items (1)

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

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
■ #1	1	Show the code related to fetching all associations (including the query)
■ #2	1	Show the code related to the display of the results
■ #3	1	Explain what the Movie class does with its associations

#3	1	Each record should have a button/link for single view
#4	1	Each record should have a username field that is clickable to go to the user's profile page
#5	1	Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity
#6	1	Show the logic related to the count of all associated items (even the ones not shown in the filtered results)
#7	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
#8	1	Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)
#9	1	Include ucid/date comments for each code screenshot
#10	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

This screenshot shows the code to fetching the associations with the query. This screenshot also shows the logic for get total count and result count.

Checklist Items (5)

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

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

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

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

This screenshot shows the filter code of the username, year and title.

Checklist Items (1)

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

```

151     <!-- reference class="bit-bit-secondary" bit-->
152     <div>
153       <span render-result-count>{{count(result)},&nbsp;&nbsp;}>
154       <div class="list-item" style="border-bottom: 1px solid #ccc; padding: 10px 0; margin: 0; font-size: 14px; font-weight: bold; font-family: inherit; color: inherit; border-left: 2px solid transparent; position: relative; width: 100%;">
155         <span> {{result}} (Results as Blocker) : <br>
156           <div class="list-item">
157             <span render-modify-command> >
158             <div>
159               <span>{{result}} >
160               <span> {{count(result)}}(result)>
161                 <div class="list-item">
162                   <span> {{result}} (Results to Block) >
163                   <div>
164                     <span render-modify-command> >
165                     <div>
166                       <span>{{result}} >
167                     </div>
168                   </div>
169                   <span> {{result}}_DIR : {{result}}/ </span>
170                 </div>
171               </div>
172             </div>
173           </div>
174         </div>
175       </div>
176     </div>

```

This screenshot shows the code that displays the information on the page as well as the `render_movie_card` function which displays the card that has the profile and unfavorite link and the view button.

Checklist Items (4)

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

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

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

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

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

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

Response:

The result list is determined from the UserMovies table, Users table and the Movies table. A SQL query is created to pull the username, title, year, user_id, movie_id from all three of these tables. The query does this to look for movie_ids in the Movies table that are tied to user_ids in the UserMovies table. If they are found in the UserMovies table it means they are a favorited movie by a user. The users table is queried here as well to find the username of said users that have a movie_id tied to their user_id in the UserMovies table. This is put into a result list and displayed using the render movie card function. This query will also change depending on the filters applied by the user. If the user inputs a certain username/year/title the query is appended with this information and looking for specific results tied to the username, year and or title. I was unable to get the logic for single delete working for admins. The logic for the count works by calling the get_total_count function which takes a string as the sql query of the tables. The function then calls the UserMovies table and Movies table. In this case it fetches the count of all the movies in the database and all the movies that are favorited from the user in the UserMovies database and displays the results to the user using the render function.

COLLAPSE

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://df39-it202-008-prod-8b39da6847aa.herokuapp.com/Project/admin/movie_association.php

URL #2

<https://github.com/dunnfall/df39-it202-008/pull/68>

Unassociated Page (2 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Movie Watch Home Profile Create/Fetch Movies Favorite Movies Non-Favorite Movies Movies Admin Logout

Non-Favorite Movies

Title	Year	Sort	Order	Limit
<input type="text" value="Title"/>	<input type="text" value="Year"/>	<input type="text" value="Titles"/>	<input type="button" value="+"/>	<input type="text" value="10"/>
<input type="button" value="Filter"/>	<input type="button" value="Clear"/>			

Results: 560/560

[Toy Story 400](#)

[Toy Story of Terror!](#)

[Toy Story That Time Forgot](#)

[IT202 The Movie](#)

[Captain Scarface](#)

Movie Title: Toy Story 400 Year: 2019 ID: 526 View Favorite Movie	Movie Title: Toy Story of Terror! Year: 2018 ID: 527 View Favorite Movie	Movie Title: Toy Story That Time Forgot Year: 2014 ID: 528 View Favorite Movie	Movie Title: IT202 The Movie Year: 2024 ID: 530 View Favorite Movie	Movie Title: Captain Scaface Year: 1953 ID: 531 View Favorite Movie
---	--	--	---	---

This screenshot shows movie not favorited by any users in the UserMovies table. This also shows the single view link with the ability to favorite the movie link. It also shows the heroku dev url in the url bar.

Checklist Items (4)

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

#2 Show the single view buttons/links

#5 Make sure the heroku dev url is visible in the address bar

#6 Clearly caption screenshots

Movie Watch Home Profile Create/Fetch Movies Favorite Movies Non-Favorite Movies Movies Admin Logout

Non-Favored Movies

Title Year Sort Order Limit

Toy

Filter Clear

Results: 5/500

Making Toy Story Movie Title: Making Toy Story Year: 1995 ID: 633 View Favorite Movie	Toy Story 2 Movie Title: Toy Story 2 Year: 1999 ID: 634 View Favorite Movie	Toy Story 400 Movie Title: Toy Story 400 Year: 2019 ID: 526 View Favorite Movie	Toy Story of Terror! Movie Title: Toy Story of Terror! Year: 2013 ID: 527 View Favorite Movie	Toy Story That Time Forgot Movie Title: Toy Story That Time Forgot Year: 2014 ID: 528 View Favorite Movie
--	--	--	--	--

This screenshot shows the count number showing 5 results of 560 total unfavorited movies. It also shows the filtering of the movie with title Toy.

Checklist Items (2)

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

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

No Results Found From Filter

Non-Favored Movies

Title	Year	Sort	Order	Limit
Toy		Title	+	10

Filter Clear

Results: 0/560
No Results to Show

This screenshot shows a filter that brings no results with the count being 0

Checklist Items (2)

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

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

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the code related to fetching all unassociated entities (including the query)
<input checked="" type="checkbox"/> #2	1	Show the code related to the display of the results

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

Task Screenshots:

Gallery Style: Large View

This screenshot shows the code that shows fetching all the associated data and the query. This screenshot also shows the code of sorting and filtering and the count logic.

Checklist Items (5)

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

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

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

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots

```
1 <?php
2 //include database connection
3
4 $query = "SELECT * FROM movies ORDER BY rating DESC";
5
6 $result = mysqli_query($conn, $query);
7
8 if ($result) {
9     $movies = [];
10
11     while ($row = mysqli_fetch_assoc($result)) {
12         $movies[] = [
13             'id' => $row['id'],
14             'title' => $row['title'],
15             'rating' => $row['rating'],
16             'genre' => $row['genre']
17         ];
18     }
19
20     echo json_encode($movies);
21 }
22 else {
23     die("Error: " . mysqli_error($conn));
24 }
25
26 //close connection
27 mysqli_close($conn);
28
29 //function to get movie details
30 function getMovie($id) {
31     global $conn;
32
33     $query = "SELECT * FROM movies WHERE id = $id";
34
35     $result = mysqli_query($conn, $query);
36
37     if ($result) {
38         $row = mysqli_fetch_assoc($result);
39
40         return [
41             'id' => $row['id'],
42             'title' => $row['title'],
43             'rating' => $row['rating'],
44             'genre' => $row['genre']
45         ];
46     }
47     else {
48         die("Error: " . mysqli_error($conn));
49     }
50 }
51
52 //function to insert movie
53 function insertMovie($title, $rating, $genre) {
54     global $conn;
55
56     $query = "INSERT INTO movies (title, rating, genre) VALUES (?, ?, ?)";
57
58     $stmt = mysqli_prepare($conn, $query);
59
60     mysqli_stmt_bind_param($stmt, "sss", $title, $rating, $genre);
61
62     mysqli_stmt_execute($stmt);
63
64     mysqli_stmt_close($stmt);
65 }
66
67 //function to update movie
68 function updateMovie($id, $title, $rating, $genre) {
69     global $conn;
70
71     $query = "UPDATE movies SET title = ?, rating = ?, genre = ? WHERE id = ?";
72
73     $stmt = mysqli_prepare($conn, $query);
74
75     mysqli_stmt_bind_param($stmt, "ssss", $title, $rating, $genre, $id);
76
77     mysqli_stmt_execute($stmt);
78
79     mysqli_stmt_close($stmt);
80 }
81
82 //function to delete movie
83 function deleteMovie($id) {
84     global $conn;
85
86     $query = "DELETE FROM movies WHERE id = ?";
87
88     $stmt = mysqli_prepare($conn, $query);
89
90     mysqli_stmt_bind_param($stmt, "i", $id);
91
92     mysqli_stmt_execute($stmt);
93
94     mysqli_stmt_close($stmt);
95 }
```

This code shows how the results are displayed and this is with the movie_card.php file. This is also where the button for viewing and favoriting the movie lie as well.

Checklist Items (5)

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

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

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

#7 Include uid/date comments for each code screenshot

#8 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
---	--------	---------

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

Response:

The result list logic is determined by the query that looks for movie ids that are in the Movies table but are also NOT in the UserMovies table. This result is done using an SQL query and assigned to a results variable. The results of this variable also change depending on the sorting/filtering options used by the user. If a filter based on title or year this input gets appended to the query to look for those results for the user. The count logic in this file is used by passing a query that looks for the number of movies that are in Movies that are also not in the UserMovies table. This number is then displayed next to the amount of results displayed on the page.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://df39-it202-008-prod-8b39da6847aa.herokuapp.com/Project/non_favorited_movies.php

URL #2

<https://github.com/dunnfall/df39-it202-008/pull/68>

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large

This screenshot shows the 'Assign Movies' page of the Movie Watch application. At the top, there are three view modes: 'Small', 'Medium', and 'Large'. Below the header, a navigation bar includes links for 'Movie Watch', 'Home', 'Profile', 'Create/Fetch Movies', 'Favorite Movies', 'Non-Favorite Movies', 'Movies', 'Admin', and 'Logout'. A search bar at the top has the letter 'd' typed into it. A 'Title Search' input field is also present. A 'Search' button is highlighted in blue.

The main content area is titled 'Assign Movies' and contains two sections: 'Users' and 'Movies to Assign'. The 'Users' section lists several users with their names and movie counts:

User	Movies
Danny	No Movies
admin	The Scarface Mob;The Making of the Lion King;An Awesome Action Movie
ksdmc	No Movies
welkjdnwed	No Movies
dur	No Movies
prodbyjell	Toy Story;Scarface
adminadmin	No Movies
dunfall	Charlie; A Toy Story;I Billion Poops;100 Million BC;A Muzzarell';(Naj)zajom!

The 'Movies to Assign' section lists several movies with checkboxes:

Movie
I!Women Art Revolution
'A Muzzarell'
'Nest the Arizona Skies
'Twas the Night Before Christmas
(Naj)zajom!
@urFRENZ
#Dowwozn!

A 'Assign/Unassign Movies' button is located at the bottom left of the main content area.

This screenshot shows the form with valid data and a result of a search of users with the letter d in it. With the heroku dev in the url.

Checklist Items (4)

#1 Show the search form with valid data

#2 Show the results of the search

#4 Make sure the heroku dev url is visible in the address bar

#5 Clearly caption screenshots

This screenshot shows the 'Assign Movies' page of the Movie Watch application. At the top, there are three view modes: 'Small', 'Medium', and 'Large'. Below the header, a navigation bar includes links for 'Movie Watch', 'Home', 'Profile', 'Create/Fetch Movies', 'Favorite Movies', 'Non-Favorite Movies', 'Movies', 'Admin', and 'Logout'. A search bar at the top has the letter 'd' typed into it. A 'Title Search' input field is also present. A 'Search' button is highlighted in blue.

The main content area is titled 'Assign Movies' and contains two sections: 'Users' and 'Movies to Assign'. The 'Users' section lists several users with their names and movie counts:

User	Movies
Danny	I!Women Art Revolution
admin	The Scarface Mob;The Making of the Lion King;An Awesome Action Movie

The 'Movies to Assign' section lists several movies with checkboxes:

Movie
I!Women Art Revolution
'A Muzzarell'
'Nest the Arizona Skies
'Twas the Night Before Christmas
(Naj)zajom!
@urFRENZ
#Dowwozn!

<input type="checkbox"/> ksomic	No Movies
<input type="checkbox"/> wekjdjwedd	No Movies
<input type="checkbox"/> dun	No Movies
<input type="checkbox"/> prodbyfall	Toy Story, Scarface
<input type="checkbox"/> adminadmin	No Movies
<input type="checkbox"/> cunfall	Charlie: A Toy Story, 1 Billion Poops, 100 Million BC, 'A Muzzarelli', (Nie)zajomi

Assign/Unassign Movies

This screenshot shows the movie woman art revolution being associated with the user danny.

Checklist Items (1)

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

The screenshot shows the 'Assign Movies' page of the Movie Watch application. The user 'danny' has the 'Assign' checkbox checked. The 'Assign/Unassign Movies' button is visible at the bottom of the page.

Users		Movies to Assign
<input type="checkbox"/> Danny	No Movies	<input type="checkbox"/> !Women-Art Revolution
<input type="checkbox"/> admin	The Scarface Mob, The Making of the Lion King, An Awesome Action Movie	<input type="checkbox"/> 'A Muzzarelli'
<input type="checkbox"/> ksomic	No Movies	<input type="checkbox"/> 'Neath the Arizona Skies
<input type="checkbox"/> wekjdjwedd	No Movies	<input type="checkbox"/> 'Twas the Fight Before Christmas
<input type="checkbox"/> dun	No Movies	<input type="checkbox"/> (Nie)zajomi
<input type="checkbox"/> prodbyfall	Toy Story, Scarface	<input type="checkbox"/> @urFRENZ
<input type="checkbox"/> adminadmin	No Movies	<input type="checkbox"/> #Gwwwemz
<input type="checkbox"/> cunfall	Charlie: A Toy Story, 1 Billion Poops, 100 Million BC, 'A Muzzarelli', (Nie)zajomi	

This screenshot shows the results of the movie woman art revolution being unfavorited from the user.

Checklist Items (1)

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

Task #2 - Points: 1
Text: Screenshots of the code

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```

mysql> SELECT * FROM username LIMIT 25;
+-----+
| id | name |
+-----+
| 1  | test1 |
| 2  | test2 |
| 3  | test3 |
| 4  | test4 |
| 5  | test5 |
| 6  | test6 |
| 7  | test7 |
| 8  | test8 |
| 9  | test9 |
| 10 | test10 |
| 11 | test11 |
| 12 | test12 |
| 13 | test13 |
| 14 | test14 |
| 15 | test15 |
| 16 | test16 |
| 17 | test17 |
| 18 | test18 |
| 19 | test19 |
| 20 | test20 |
| 21 | test21 |
| 22 | test22 |
| 23 | test23 |
| 24 | test24 |
| 25 | test25 |
+-----+
25 rows in set (0.00 sec)

mysql> SELECT * FROM movie LIMIT 25;
+-----+
| id | title |
+-----+
| 1  | test1 |
| 2  | test2 |
| 3  | test3 |
| 4  | test4 |
| 5  | test5 |
| 6  | test6 |
| 7  | test7 |
| 8  | test8 |
| 9  | test9 |
| 10 | test10 |
| 11 | test11 |
| 12 | test12 |
| 13 | test13 |
| 14 | test14 |
| 15 | test15 |
| 16 | test16 |
| 17 | test17 |
| 18 | test18 |
| 19 | test19 |
| 20 | test20 |
| 21 | test21 |
| 22 | test22 |
| 23 | test23 |
| 24 | test24 |
| 25 | test25 |
+-----+
25 rows in set (0.00 sec)

```

This screenshot shows the query of the usernames and movies being a limit of 25. This screenshot also shows applying the associations from the form to the database.

Checklist Items (4)

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

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

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots

```
<!-- 0030 4/29/2024 -->
<div class="row justify-content-between">
    <div><h3>Search</h3>
        <form method="POST">
            <div>
                <input type="text" name="username" placeholder="Username Search" value="">
                <input type="text" name="movies_title" placeholder="Title Search" value="">
            </div>
            <br>
            <input type="submit" value="Search" />
            <input type="submit" value="Submit" />
        </form>
    </div>
    <div>
        <table border="1">
            <thead>
                <tr>
                    <th>User Name</th>
                    <th>Movies</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>JohnDoe123</td>
                    <td>
                        <input checked="" type="checkbox" value="MovieA" />
                        <input checked="" type="checkbox" value="MovieB" />
                        <input checked="" type="checkbox" value="MovieC" />
                    </td>
                </tr>
                <tr>
                    <td>JaneDoe456</td>
                    <td>
                        <input type="checkbox" value="MovieA" />
                        <input type="checkbox" value="MovieB" />
                        <input type="checkbox" value="MovieC" />
                    </td>
                </tr>
            </tbody>
        </table>
    </div>
</div>
<script>
    require('./scripts/partials/search.js');
</script>
```

This screenshot shows the search form field for the username and movies. It also shows the checkboxes for movies and usernames and the last render button is the submit button for the checkbox lists.

Checklist Items (4)

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

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

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

#5 Code related to submitting the checkbox lists

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Describe the steps for the search and how it works for users and entities.

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

Response:

Once the form is submitted with a username and a movie title entered a function is ran assigning the username entry to the variable username and the title entry to the variable named movie_title. A query is then made to check the users table for usernames that match the users entry in the form and the movies they have favorited. The second query then selects the movies that match the movie title searched from the user. It is then displayed to the user using a for each loop with the movie_card.php file as the UI to display it to the user. The movie search is also wrapped in a scroll bar element. As far as associating and unassociating data once the form is submitted with a user and a movie selected to favorite or unfavorite the movie_id chosen and the user_id chosen will be selected from the UserMovies table if a record of the two are found it will delete them from the table essentially unfavoriting the movie from the user. If a record of them is not found it will favorite the movie for the user and a record will get inserted into the UserMovies table.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

<https://github.com/dunnfall/df39-it202-008/pull/70>

URL #2

https://df39-it202-008-prod-8b39da6847aa.herokuapp.com/Project/admin/admin_user_association.php

Misc (1 pt.)

[COLLAPSE](#)

Task #1 - Points: 1

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

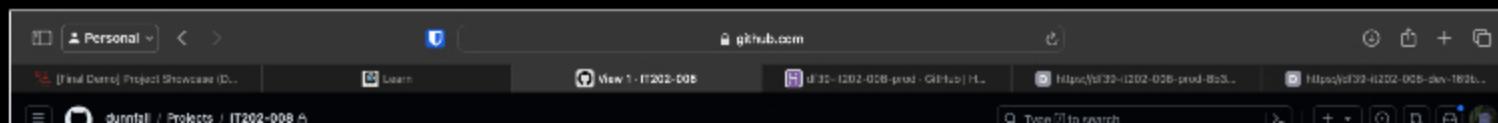
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



IT202-008

Add status update

+ New view Filter by keyword or by field

Todo	In Progress	Done
This item hasn't been started.	This is actively being worked on.	This has been completed. <ul style="list-style-type: none">MS2 - Delete HandlingMS2 - API HandlingMS3 - API Data AssociationMS3 - Handle The Association of Data To A UserMS3 - Logged In Users Associated Entities PageMS3 - All Users Association PageMS3 - Create a page that shows data not associated with any userMS3 - Admin Can Associate Any Entity with Any User
+ Add item	+ Add item	+ Add item

This shows my project board on GitHub

Task #2 - Points: 1

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

URL #1

<https://github.com/users/dunnfall/projects/1>

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

An issue I had in the assignment was getting an admin to unfavorite movies from the All user associations page. I couldn't seem to figure out how to be able to let admins do this, so I gave up on it.

Task #4 - Points: 1

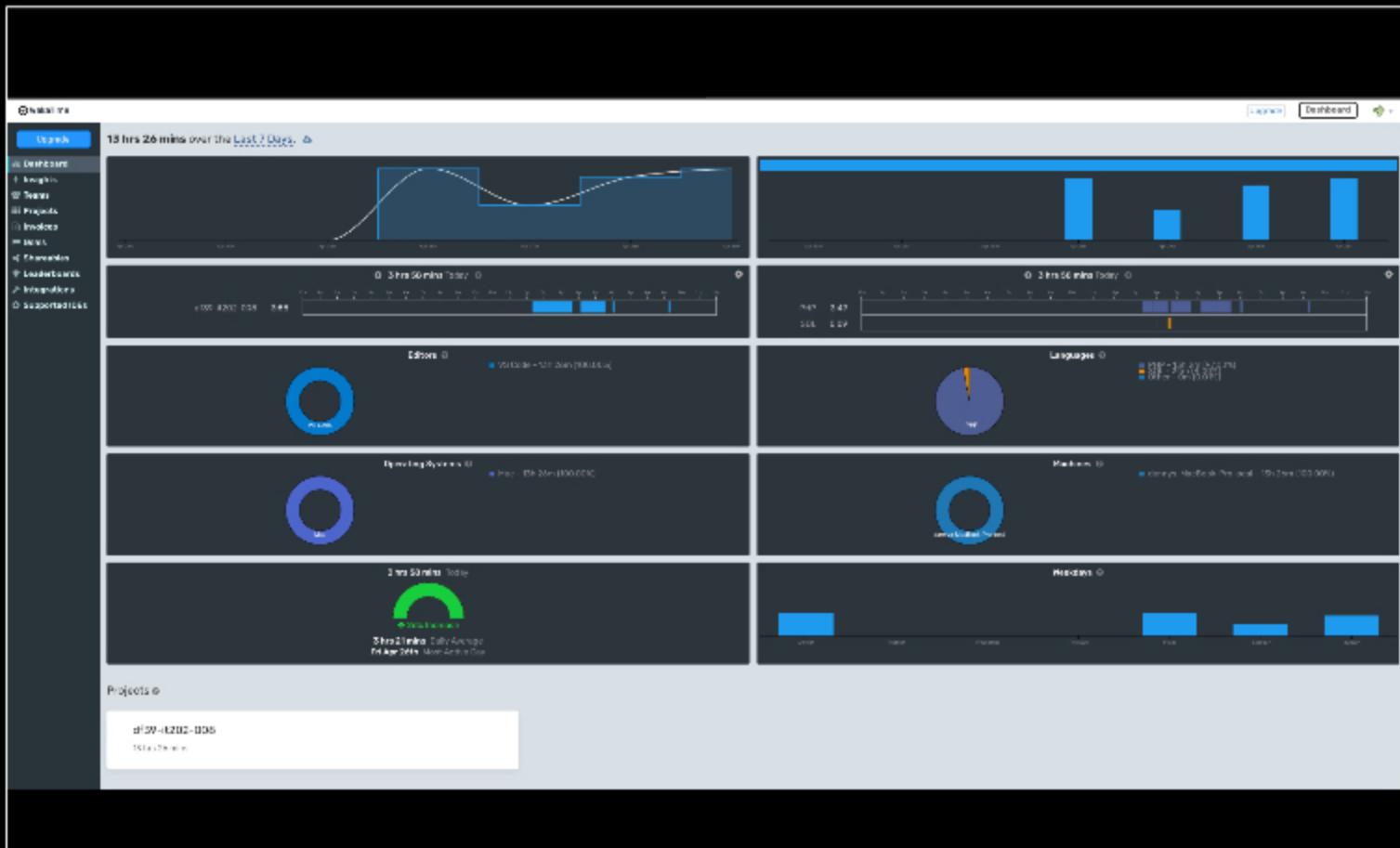
Text: WakaTime Screenshot

Details:

Small

Medium

Large



This screenshot shows my wakatime

End of Assignment