

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-2-2024/grade/jn426>

1.

IT202-008-S2024 - [IT202] API Project Milestone 2 2024

3.

4.

Submissions:

Submission Selection

1 Submission [active] 4/14/2024 8:35:20 PM

8.

9.

10. Instructions

[^ COLLAPSE ^](#)

Implement the Milestone 2 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 Milestone2 branch

Create a pull request from Milestone2 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: Milestone2

Tasks: 29 **Points:** 10.00

 Define Appropriate Tables for Data (1 pt.)

[^ COLLAPSE ^](#)

 Task #1 - Points: 1

Text: Screenshots of Table SQL

Checklist

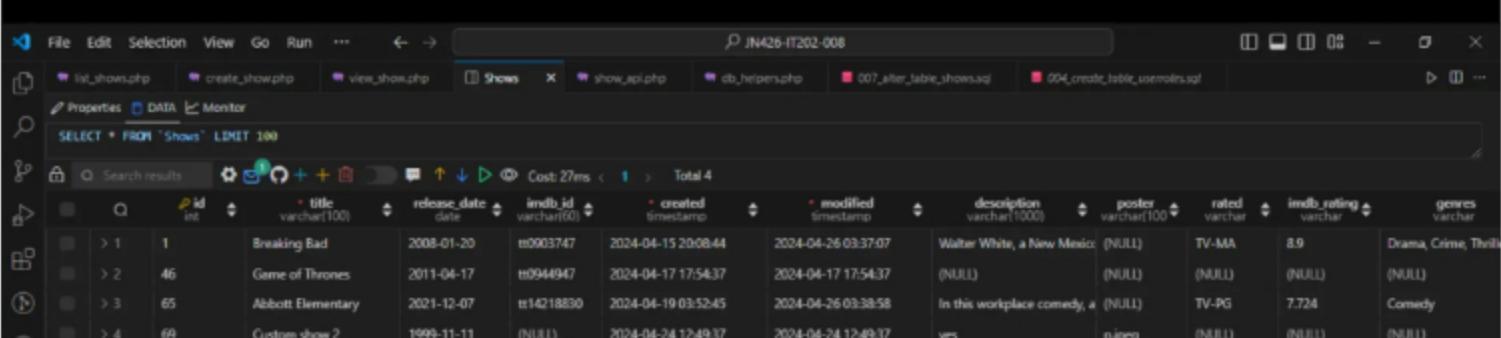
*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Table(s) should have the 3 core columns we'll always be using (id, created, modified) plus additional columns for the incoming API data
<input type="checkbox"/> #2	1	Columns should be logical and thought out (not valid to have a single field of JSON data or similar)
<input type="checkbox"/> #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



	id int	title varchar(100)	release_date date	imdb_id varchar(100)	created timestamp	modified timestamp	description varchar(1000)	poster varchar	rated varchar	imdb_rating varchar	genres varchar
> 1	1	Breaking Bad	2008-01-20	tt0903747	2024-04-15 20:08:44	2024-04-26 03:37:07	Walter White, a New Mexico	(NULL)	TV-MA	8.9	Drama, Crime, Thriller
> 2	46	Game of Thrones	2011-04-17	tt0944947	2024-04-17 17:54:37	2024-04-17 17:54:37	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
> 3	65	Abbott Elementary	2021-12-07	tt14218890	2024-04-19 03:52:45	2024-04-26 03:38:58	In this workplace comedy, a	(NULL)	TV-PG	7.724	Comedy
> 4	69	Custom show 2	1999-11-11	(NULL)	2024-04-24 12:49:37	2024-04-24 12:49:37	yes	jpeg	(NULL)	(NULL)	(NULL)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```

Counting objects: 1088 (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 1088 (8/8), done.
Writing objects: 1088 (8/8), 1003 bytes | 581.00 KiB/s, done.
Total 8 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 1000 (7/7), completed with 7 local objects.
To github.com:sonatrnguyen/JM26-IT202-008.git
 3de661c..61b3e5c JM2-API-Integration-Admin -> JM2-API-Integration-Admin
Jenna@DESKTOP-CSBPHQ9 MINGW64 /c/JM26-IT202-008 (JM2-API-Integration-Admin)
$ 
```

12:33 AM 4/26/2024

This is my table with titles of tv shows, their release date and imdb ids. It also has the description, rating, audience rating, and genres. It has a field for a poster image but that is not being used anymore and I forgot to drop it when taking this screenshot.

Checklist Items (0)

Task #2 - Points: 1

Text: Explain the design

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Note the different fields and their purpose
		Note if you needed to normalize the data into separate tables or if one table fit your

#2

1

needs

Response:

The title, release date, description, rating, genres, and audience rating are information that I want to project to the user. The imdb_id is used to fetch data from an additional endpoint. The imdb_id field also differentiates which shows are fetched from the API and which are not .

Task #3 - Points: 1

Text: Add the pull request link for the branch related to this feature

i Details:

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/68>

Data Creation Page (2 pts.)

Data Creation Page (2 pts.)

Task #1 - Points: 1

Text: Screenshots of the creation page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show potentially valid data filled in for the custom creation page
<input type="checkbox"/> #2	1	Show how the API data is fetched for API data (must be server-side)
<input type="checkbox"/> #3	1	Show examples of validation messages
<input type="checkbox"/> #4	1	Show an example of successful creation message
<input type="checkbox"/> #5	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
<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

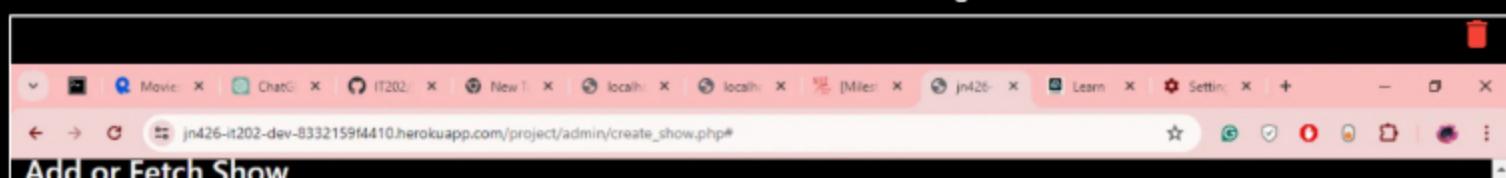
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Add or Fetch Show

Add or Fetch Show

Fetch Add

Show Title

Custom Show 2

Release Date

01/11/1999

Show Description

this is a custom data

Show Rating

Show Rating (/10)

Show Audience

PG

Genre(s)

Comedy

Add



Potentially valid data

Checklist Items (1)

#1 Show potentially valid data filled in for the custom creation page

The screenshot shows a browser window with a red header bar containing the URL "jn426-it202-dev-8332159f4410.herokuapp.com/project/admin/create_show.php#". Below the header is a navigation bar with links for "Show Tracker", "Home", "Profile", "Admin", "Show Management", and "Logout". A green success message box displays the text "Created record with id array { 'rowCount' => 1, 'lastInsertId' => '72' }". The main content area has a heading "Add or Fetch Show" and two buttons: "Fetch" and "Add". Below these buttons is a text input field labeled "Show Title" with the value "Custom Show 2" and a blue "Search" button. At the bottom of the browser window is a dark taskbar with various pinned icons.

success create

Checklist Items (0)

Show Tracker Home Profile Admin Show Management Logout

Add or Fetch Show

Fetch Add

spongebob

Search

Fetched from API

Checklist Items (1)

#2 Show how the API data is fetched for API data (must be server-side)

You must provide a title

Invalid data passed

Add or Fetch Show

Fetch Add

Show Title

Search

Console

```
Uncaught ReferenceError: showTitle is not defined
```

Fetched from API

Type here to search

Checklist Items (0)

Task #2 - Points: 1

Text: Screenshots of creation page code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Form should have correct data types for each property being requested
<input type="checkbox"/> #2	1	Form should have correct validation for each field (HTML, JS, and PHP)
<input type="checkbox"/> #3	1	Successful creation should have a user-friendly message
<input type="checkbox"/> #4	1	Any errors should have user-friendly messages
<input type="checkbox"/> #5	1	Include the form/process for fetching API data
<input type="checkbox"/> #6	1	Include some indicator between custom data and API data
<input type="checkbox"/> #7	1	Include any other rules like role guards and login checks
<input type="checkbox"/> #8	1	Include ucid/date comments for each code screenshot
<input type="checkbox"/> #9	1	Clearly caption screenshots

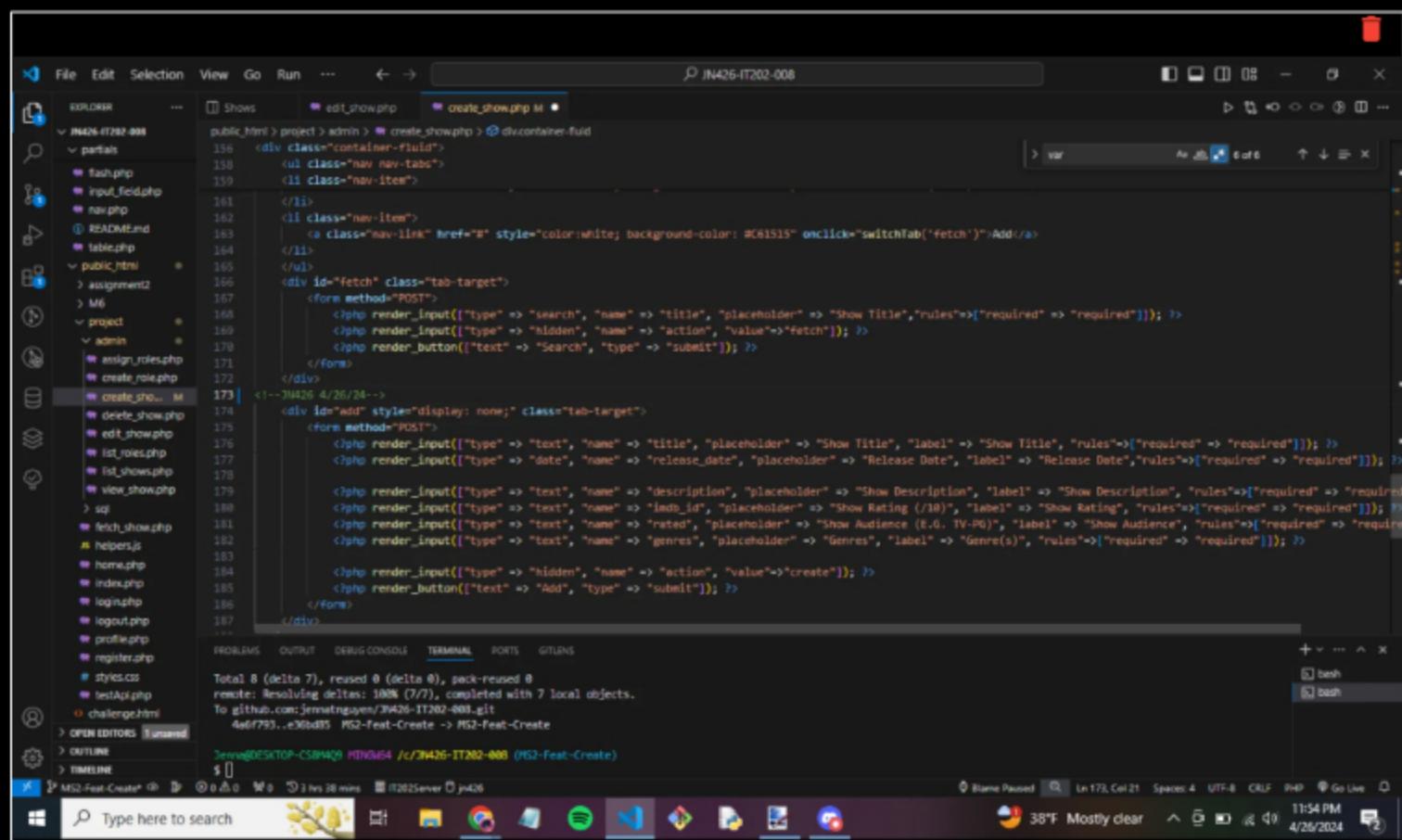
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



```

File Edit Selection View Go Run ... ← → JN426-IT202-008
EXPLORER JN426-IT202-008 edit_show.php create_show.php
public_html > project > admin > create_show.php > @div.container-fluid
156 <div class="container-fluid">
158   <ul class="nav nav-tabs">
159     <li class="nav-item">
161       </li>
162     <li class="nav-item">
163       <a class="nav-link" href="#" style="color:white; background-color: #6C5151" onclick="switchTab('fetch')">>Add</a>
164     </li>
165   </ul>
166   <div id="fetch" class="tab-target">
167     <form method="POST">
168       <php render_input(["type" => "search", "name" => "title", "placeholder" => "Show Title", "rules"=>["required" => "required"]); >>
169       <php render_input(["type" => "hidden", "name" => "action", "value"=>"fetch"]); >>
170       <php render_button(["text" => "Search", "type" => "submit"]); >>
171     </form>
172   </div>
173   <!--JN426 4/26/24-->
174   <div id="add" style="display: none;" class="tab-target">
175     <form method="POST">
176       <php render_input(["type" => "text", "name" => "title", "placeholder" => "Show Title", "label" => "Show Title", "rules"=>["required" => "required"]); >>
177       <php render_input(["type" => "date", "name" => "release_date", "placeholder" => "Release Date", "label" => "Release Date", "rules"=>["required" => "required"]); >>
178       <php render_input(["type" => "text", "name" => "description", "placeholder" => "Show Description", "label" => "Show Description", "rules"=>["required" => "required"]); >>
179       <php render_input(["type" => "text", "name" => "imdb_id", "placeholder" => "Show Rating (/10)", "label" => "Show Rating", "rules"=>["required" => "required"]); >>
180       <php render_input(["type" => "text", "name" => "rated", "placeholder" => "Show Audience (E.O. TV-PG)", "label" => "Show Audience", "rules"=>["required" => "required"]); >>
181       <php render_input(["type" => "text", "name" => "genres", "placeholder" => "Genres", "label" => "Genre(s)", "rules"=>["required" => "required"]); >>
182       <php render_input(["type" => "hidden", "name" => "action", "value"=>"create"]); >>
183       <php render_button(["text" => "Add", "type" => "submit"]); >>
184     </form>
185   </div>
186
187 </div>

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

Total 8 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), completed with 7 local objects.
To github.com:jennetinguyen/JN426-IT202-008.git
4a6f793...e36bd45 M52-Fest-Create -> M52-Fest-Create

Devra@DESKTOP-CSBHQ4Q MINIM64 /c/JN426-IT202-008 (M52-Fest-Create)

File Blame Paused Line 173, Col 21 Spaces: 4 UTF-8 CRLF PHP Go Live

38°F Mostly clear 11:54 PM 4/26/2024

Form has correct data types being requested

Checklist Items (1)

#1 Form should have correct data types for each property being requested

The screenshot shows a code editor with two tabs open: `edit_show.php` and `create_show.php`. The `create_show.php` tab is active, displaying PHP code for validating form inputs. The code includes checks for title, genres, rating, and description, ensuring they are not empty and correctly typed (e.g., numbers for rating). It also uses the `isNan()` function to check if the rating is a number. The code is annotated with comments explaining the validation logic.

```
public_html > project > admin > create_show.php > script
70 <script> //JN426 4/26/24
71 function validation(form) {
72     let title = form.title.value;
73     let genres = form.genres.value;
74     let rated = form.rated.value;
75     let imdb_rating = form.imdb_rating.value;
76     let description = form.description.value;
77     let valid = true;
78
79     if(title.length == 0) {
80         flash("Must include title", "warning");
81         valid = false;
82     }
83
84     if(genres.length == 0) {
85         flash("Must include genre", "warning");
86         valid = false;
87     }
88
89     if(imdb_rating.length == 0) {
90         flash("Must include rating", "warning");
91         valid = false;
92     }
93
94     if(isNaN(imdb_rating)) {
95         flash("Rating must be a number", "warning");
96         valid = false;
97     }
98
99     if(description.length == 0) {
100        flash("Must include description", "warning");
101        valid = false;
102    }
103
104    return valid;
105 } <-- #71-105 function validation(form)
106 </script>
107
108 <?php
109 if (isset($_POST["create"])) {
110     $title = se($_POST, "title", "", false);
111     $genres = se($_POST, "genres", "", false);
112     $rated = se($_POST, "rated", "", false);
113     $imdb_rating = se($_POST, "imdb_rating", "", false);
114     $description = se($_POST, "description", "", false);
115     //JN426 4/26/24
116     $hasError = false;
117
118     if (empty($title)) {
119         flash("Title must not be empty", "danger");
120         $hasError = true;
121     }
122
123     if (empty($genres)) {
124         flash("Genre must not be empty", "danger");
125         $hasError = true;
126     }
127     if (empty($rated)) {
128         flash("Audience rating must not be empty", "danger");
129         $hasError = true;
130     }
131
132     if (empty($imdb_rating)) {
133         flash("Rating must not be empty", "danger");
134         $hasError = true;
135     }
136
137     if (empty($description)) {
138         flash("Description password must not be empty", "danger");
139         $hasError = true;
140     }
141
142     if ($hasError) {
143         echo 4;
144         $stmt = $db->prepare("INSERT INTO Shows (title, genres, rated, imdb_rating, description) VALUES(:title, :genres, :rated, :imdb_rating, :description)");
145         $stmt->execute(":title" => $title, ":genres" => $genres, ":rated" => $rated, ":imdb_rating" => $imdb_rating, ":description" => $description);
146         flash("Successfully created!", "success");
147         catch (PDOException $e) {
148             users_check_duplicate($e->errorInfo);
149         }
150     }
151
152     if ($hasError) {
153         flash("Form has errors", "danger");
154     }
155 }
```

JS Validation

Checklist Items (1)

#2 Form should have correct validation for each field (HTML, JS, and PHP)

The screenshot shows a code editor with two tabs open: `edit_show.php` and `create_show.php`. The `create_show.php` tab is active, displaying PHP code for validating form inputs. The code includes checks for title, genres, rating, and description, ensuring they are not empty and correctly typed (e.g., numbers for rating). It also uses the `isNan()` function to check if the rating is a number. The code is annotated with comments explaining the validation logic.

```
public_html > project > admin > create_show.php > script
126 by($rated) {
127     sh("Audience rating must not be empty", "danger");
128     $hasError = true;
129
130     by($imdb_rating) {
131         sh("Rating must not be empty", "danger");
132         $hasError = true;
133
134         by($description) {
135             sh("Description password must not be empty", "danger");
136             $hasError = true;
137
138             if ($hasError) {
139                 echo 4;
140
141                 $stmt = $db->prepare("INSERT INTO Shows (title, genres, rated, imdb_rating, description) VALUES(:title, :genres, :rated, :imdb_rating, :description)");
142                 $stmt->execute(":title" => $title, ":genres" => $genres, ":rated" => $rated, ":imdb_rating" => $imdb_rating, ":description" => $description);
143                 flash("Successfully created!", "success");
144                 catch (PDOException $e) {
145                     users_check_duplicate($e->errorInfo);
146
147                     if ($hasError) {
148                         flash("Form has errors", "danger");
149                     }
150                 }
151             }
152         }
153     }
154 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
Total 8 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), completed with 7 local objects.
To github.com:JennettNguyen/JN426-IT202-008.git
 4a6f793..e30bd85 MS2-Feat-Create -> MS2-Feat-Create
```

Jennet@DESKTOP-CSBHQ09 MINIM64 /c/JN426-IT202-008 (MS2-Feat-Create)

MS2-Feat-Create

Type here to search

Blame Paused | Line 115, Col 20 | Spaces: 4 | UTF-8 | CR/LF | PHP | Go Live | 11:57 PM | 4/26/2024

PHP Validation

Checklist Items (0)

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

```
Shows edit_show.php create_show.php M ...
```

```
public_html > project > admin > create_show.php M ...
```

```
37     flash("You must provide a title", "warning");
38   }
39   //insert data
40   try {
41     //optional options for debugging and duplicate handling
42     //JN426 4/26/24
43     $opts = [
44       "debug" => true, "update_duplicate" => false, "columns_to_update" => []
45     ];
46     $result = insert("Shows", $quote, $opts);
47     if (!$result) {
48       flash("Unhandled error", "warning");
49     } else {
50       flash("Created record with id " . var_export($result, true), "success");
51     }
52   } catch (InvalidArgumentException $e1) {
53     error_log("Invalid arg" . var_export($e1, false));
54     flash("Invalid data passed", "danger");
55   } catch (PDOException $e2) {
56     if ($e2->errorInfo[1] == 1062) {
57       flash("An entry for this title already exists for today", "warning");
58     } else {
59       error_log("Database error" . var_export($e2, true));
60       flash("Database error", "danger");
61     }
62   } catch (Exception $e3) {
63     error_log("Invalid data records" . var_export($e3, false));
64     flash("Invalid data records", "danger");
65   }
66 } // #15-64 if (isset($_POST["action"]));
67 // handle manual create stock
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
Total 8 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), completed with 7 local objects.
To github.com:JennettNguyen/JN426-IT202-008.git
 4a6f793..e30bd85 MS2-Feat-Create -> MS2-Feat-Create
```

Jennet@DESKTOP-CSBHQ09 MINIM64 /c/JN426-IT202-008 (MS2-Feat-Create)

MS2-Feat-Create

Type here to search

Blame Paused | Line 40, Col 24 | Spaces: 4 | UTF-8 | CR/LF | PHP | Go Live | 11:58 PM | 4/26/2024

Creation success message and error messages

Checklist Items (0)

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

```
Shows edit_show.php create_show.php M ...
```

```
public_html > project > admin > create_show.php M ...
```

```
1 <?php
2 //note we need to go up 1 more directory
3 //JN425 4/26/24
4 require __DIR__ . "/../../../../partials/nav.php";
5
6 if (!has_role("Admin")) {
7   flash("You don't have permission to view this page", "warning");
8   die(header("Location: $BASE_PATH" . "/home.php"));
9   redirect("home.php");
10 } // #5-9 if (!has_role("Admin"))
11 ?>
12
13 <?php
14
15 //160 handle show fetch
16 if (isset($_POST["action"])) {
17   $action = $_POST["action"];
18   $title = strtoupper(se($_POST, "title", "", false));
19   $quote = [];
20   if ($title) {
21     if ($action === "fetch") {
22       $result = fetch_show_by_title($title);
23       error_log("Data from API" . var_export($result, true));
24       if ($result) {
25         $quote = $result;
26         // $quote["is_apl"] = 1;
27       }
28     }
29   }
30 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
Shows edit_show.php create_show.php M ...
```

```
128   ty($rated)) {
129     sh("audience rating");
130     sError = true;
131   }
132
133   ty($imdb_rating)) {
134     sh("rating must not be");
135     sError = true;
136   }
137   ty($description)) {
138     sh("description pass");
139     sError = true;
140   }
141
142   asError) {
143     echo 4;
144   }
145   = getDB();
146   $stmt = $db->prepare("
147   ");
148   $stmt->execute(["$title"]);
149   flash("Successfully");
150   catch (PDOException $e) {
151     users_check_duplicate();
152   }
153   41-152 if (!($hasError));
154   53 if (isset($_POST));
155
156 }
```

Blame Paused | Line 40, Col 24 | Spaces: 4 | UTF-8 | CR/LF | PHP | Go Live | 11:58 PM | 4/26/2024

```

20   > exit or function --- create()
21   > foreach ($_POST as $k => $v) {
22     if (!in_array($k, ["title", "release_date", "description", "rated", "imdb_rating", "genres"])) {
23       continue;
24     }
25     $data[$k] = $v;
26   }
27   $data['imdb_id'] = $imdb_id;
28   $data['created'] = date('Y-m-d H:i:s');
29   $data['modified'] = date('Y-m-d H:i:s');
30

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS + v ... x

Total 8 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), completed with 7 local objects.
To github.com:Jennatnguyen/JN426-IT202-008.git
 4a6f793..e36bd85 MS2-Feat-Create -> MS2-Feat-Create
jennat@DESKTOP-CS8PAQ9 MINIM64 /c/JN426-IT202-008 (MS2-Feat-Create)

156 157

11:59 PM 4/26/2024

role check, indicator between API data is automatically built into api

Checklist Items (1)

#7 Include any other rules like role guards and login checks

Task #3 - Points: 1

Text: Screenshot of records from DB

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show at least one record fetched from the API
<input type="checkbox"/> #2	1	Show at least one record created via the creation form
<input type="checkbox"/> #3	1	Note what differs
<input type="checkbox"/> #4	1	Clearly caption screenshots

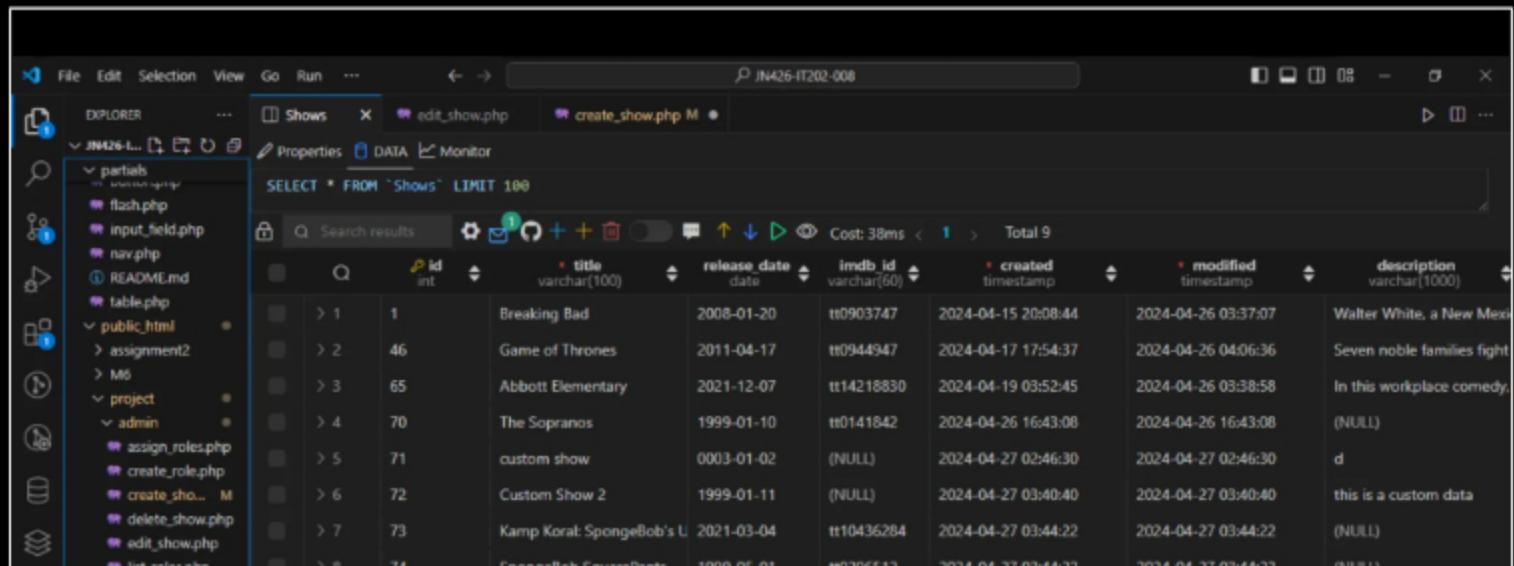
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



```

SELECT * FROM `Shows` LIMIT 100
+----+----+-----+-----+-----+-----+-----+-----+-----+
| id | title | release_date | imdb_id | created | modified | description |
+----+----+-----+-----+-----+-----+-----+-----+
| 1  | Breaking Bad | 2008-01-20 | tt0903747 | 2024-04-15 20:08:44 | 2024-04-26 03:37:07 | Walter White, a New Mexi |
| 2  | Game of Thrones | 2011-04-17 | tt0944947 | 2024-04-17 17:54:37 | 2024-04-26 04:06:36 | Seven noble families fight |
| 3  | Abbott Elementary | 2021-12-07 | tt14218830 | 2024-04-19 03:52:45 | 2024-04-26 03:38:58 | In this workplace comedy |
| 4  | The Sopranos | 1999-01-10 | tt0141842 | 2024-04-26 16:43:08 | 2024-04-26 16:43:08 | (NULL) |
| 5  | custom show | 2000-01-02 | (NULL) | 2024-04-27 02:46:30 | 2024-04-27 02:46:30 | d |
| 6  | Custom Show 2 | 1999-01-11 | (NULL) | 2024-04-27 03:40:40 | 2024-04-27 03:40:40 | this is a custom data |
| 7  | Kamp Koral: SpongeBob's L | 2021-03-04 | tt10436284 | 2024-04-27 03:44:22 | 2024-04-27 03:44:22 | (NULL) |
| 8  | SpongeBob SquarePants | 1999-05-01 | tt0206512 | 2024-04-27 03:44:22 | 2024-04-27 03:44:22 | (NULL) |
| 9  | The SpongeBob SquarePant | 2020-07-23 | tt12895414 | 2024-04-27 03:44:22 | 2024-04-27 03:44:22 | (NULL) |
+----+----+-----+-----+-----+-----+-----+-----+

```

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

EXPLORER JN426-IT202-008 Properties DATA Monitor

partials

edit_show.php create_show.php M

SELECT * FROM `Shows` LIMIT 100

Cost: 38ms Total 9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS + v ... x

remote: Resolving deltas: 100% (7/7), completed with 7 local objects.
To github.com:jennatnguyen/JN426-IT202-008.git
 4a6f793..e36bd85 MS2-Feat-Create -> MS2-Feat-Create

bash bash

Spongebob shows fetched from API, Custom show fetched from creation form. Difference is that the API fetches have an imdb_id

Checklist Items (0)

Task #4 - Points: 1

Text: Explain the process

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Provide a high-level step-by-step of how fetching API data works and gets added to your DB (include how duplicates are handled)
<input type="checkbox"/> #2	1	Provide a high-level step-by-step of how creating custom data works and gets added to your DB (include how duplicates are handled)
<input type="checkbox"/> #3	1	Briefly describe the validations for the applicable fields
<input type="checkbox"/> #4	1	Describe how duplicate data is handled

Response:

The process of fetching API data and adding it to the database involves users inputting a show title, which triggers a server request to an external API for relevant show details. Upon successful retrieval, the data is stored and optionally modified before insertion into the "Shows" table. Duplicate handling options like debugging and update handling can be applied during insertion. For this code duplicates are not allowed during creation and therefore will throw an error. Creating custom data involves users filling a form with show details, which undergo server-side validation to ensure completeness. Upon validation, the data is inserted into the database, with duplicate detection managed by the database. If a duplicate entry is detected, a warning message alerts the user. Field validations include ensuring the title, genres, rated, IMDb rating, and description are all provided, with IMDb rating specifically checked for numeric format. Throughout both processes, error handling mechanisms catch exceptions like invalid arguments or database errors, displaying appropriate flash messages to users to aid in troubleshooting.

Task #5 - 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 this page
<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/77>

URL #2

https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/create_show.php

● Data List Page (many entities) (2 pts.)

[COLLAPSE ^](#)

● Task #1 - Points: 1

Text: Screenshots of the list page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the page of your entities listed (have a reasonable number shown)
<input type="checkbox"/> #2	1	Show the filter/sort form based on your data and the required limit field
<input type="checkbox"/> #3	1	Demonstrate a few varied filters/sorts
<input type="checkbox"/> #4	1	Demonstrate a filter that doesn't have any records (should show an appropriate message)
<input type="checkbox"/> #5	1	Each list item should have a link of single view (i.e., details), edit, and delete (some of which may only be visible to admin users)
<input type="checkbox"/> #6	1	Each list item should have a summary of the entity (likely won't be the entire entity data)
<input type="checkbox"/> #7	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
<input type="checkbox"/> #8	1	Make sure the heroku dev url is visible in the address bar
<input type="checkbox"/> #9	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a web application interface for managing shows. At the top, there's a navigation bar with links like Home, Profile, Admin, Show Management, and Logout. Below that 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. The main area displays a table titled "Latest Shows" with two rows of data. Each row contains the show's title, its genres (separated by commas), the show's ID (imdb_id), its rating (imdb_rating), its rating status (rated), and a "Actions" column with three buttons: "View", "Edit", and "Delete".

title	genres	imdb_id	imdb_rating	rated	Actions
Breaking Bad	Drama, Crime, Thriller, Western, Suspense	tt0903747	8.9	TV-MA	View Edit Delete
Game of Thrones	Fantasy, Science-Fiction, Drama, Action, Adventure, Mystery, War, Science-fiction	tt0944947	8.4	TV-MA	View Edit

Abbott Elementary	Comedy	tt14218830	7.724	TV-PG	Delete
The Sopranos		tt0141842		View	Edit

Show the page of your entities listed (have a reasonable number shown)

Checklist Items (0)

Show Tracker							
Show Title		Show Rating		Show Audience		Genre(s)	
Show Title	Show Rating	Show Audience	Genre(s)	Sort	Order	Limit	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Title"/> <input type="button" value="Genre(s)"/> <input type="button" value="Rating"/>	<input type="button" value="+"/>	<input type="text" value="10"/>	
Filter	Clear						

List Shows

Latest Shows

title	genres	imdb_id	imdb_rating	rated	Actions
Breaking Bad	Drama, Crime, Thriller, Western, Suspense	tt0903747	8.9	TV-MA	View Edit
Game of Thrones	Fantasy, Science-Fiction, Drama, Action, Adventure, Mystery, War, Science-fiction	tt0944947	8.4	TV-MA	View Edit
Abbott Elementary	Comedy	tt14218830	7.724	TV-PG	View Edit
The Sopranos		tt0141842			View Edit

Show the filter/sort form based on your data and the required limit field

Checklist Items (0)

Show Tracker							
jn426-it202-dev-8332159f4410.herokuapp.com/project/admin/list_shows.php?title=b&imdb_rating=&rated=&genres=&sort=title&order=as... View Edit							
Show Title		Show Rating		Show Audience		Genre(s)	
<input type="text" value="b"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Title"/>	<input type="button" value="+"/>	<input type="text" value="10"/>
Filter	Clear						

List Shows

Latest Shows

title	genres	imdb_id	imdb_rating	rated	Actions
Abbott Elementary	Comedy	tt14218830	7.724	TV-PG	View Edit Delete
Breaking Bad	Drama, Crime, Thriller, Western, Suspense	tt0903747	8.9	TV-MA	View Edit Delete

Kamp Koral: SpongeBob's Under Years

tt10436284

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

SpongeBob SquarePants

tt0206512

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

The SpongeBob SquarePants Anime

tt12895414

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

Filter example 1, title b filter

Checklist Items (0)

Windows taskbar at the top.

Show Tracker - Home - Profile - Admin - Show Management - Logout

List Shows

Show Title	Show Rating	Show Audience	Genre(s)	Sort	Order	Limit
Show Title	7	Show Audience	Genre(s)	Title	+	10
Filter Clear						

Latest Shows

title	genres	imdb_id	imdb_rating	rated	Actions
Abbott Elementary	Comedy	tt14218830	7.724	TV-PG	View Edit Delete

Windows taskbar at the bottom.

filter by show rating 7

Checklist Items (0)

Windows taskbar at the top.

Show Tracker - Home - Profile - Admin - Show Management - Logout

List Shows

Show Title	Show Rating	Show Audience	Genre(s)	Sort	Order	Limit
Show Title	Show Rating	Show Audience	Comedy	Title	-	10
Filter Clear						

Latest Shows

title	genres	imdb_id	imdb_rating	rated	Actions
Custom Show 2	Comedy			PG	View Edit Delete



Filter by genre and in descending order

Checklist Items (0)

Screenshot of a web browser displaying a show tracking application. The URL is jn426-it202-dev-8332159f4410.herokuapp.com/project/admin/list_shows.php?title=your+mom&imdb_rating=&rated=&genres=&sort=title&.... The page shows a search/filter form with fields for Show Title, Show Rating, Show Audience, Genre(s), Sort (Title, Order (- or +), Limit (10)), and buttons for Filter and Clear. Below the form, a section titled "Latest Shows" displays the message "No records to show".



no records

Checklist Items (0)

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

Checklist			*The checkboxes are for your own tracking
#	Points	Details	
<input type="checkbox"/> #1	1	Show the filter/sort form generation	
<input type="checkbox"/> #2	1	Show the DB query and how the filter/sort is handled (including the restriction on the limit field)	

<input type="checkbox"/> #3	1	Show how the output is generated and displayed
<input type="checkbox"/> #4	1	Show any restrictions like role guard or login checks
<input type="checkbox"/> #5	1	Include ucid/date comments for each code screenshot
<input type="checkbox"/> #6	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```

public_html > project > admin > list_shows.php > ...
...
1 <?php
2 //note we need to go up 1 more directory
3 require_once __DIR__ . "/../../../../partials/nav.php";
4 //JW426 4/26/24
5 if (!has_role("Admin")) {
6     flash("You don't have permission to view this page", "warning");
7     redirect("home.php");
8 }
9
10 //build search form
11 $form = [
12     ["type" => "text", "name" => "title", "placeholder" => "Show Title", "label" => "Show Title", "include_margin" => false],
13     ["type" => "text", "name" => "imdb_rating", "placeholder" => "Show Rating", "label" => "Show Rating", "include_margin" => false],
14     ["type" => "text", "name" => "rated", "placeholder" => "Show Audience", "label" => "Show Audience", "include_margin" => false],
15     ["type" => "text", "name" => "genres", "placeholder" => "Genre(s)", "label" => "Genre(s)", "include_margin" => false],
16     ["type" => "text", "name" => "sort", "label" => "Sort", "options" => ["title" => "Title", "genres" => "Genre(s)", "imdb_rating" => "Rating", "rated" => "Rated"], "include_margin" => false],
17     ["type" => "select", "name" => "order", "label" => "Order", "options" => ["asc" => "+", "desc" => "-"], "include_margin" => false],
18     ["type" => "number", "name" => "limit", "label" => "Limit", "value" => "10", "include_margin" => false],
19 ];
20 <?11-21 $form ->
21
22 error_log("Form data: " . var_export($form, true));
23
24 $query = "SELECT id, title, genres, imdb_id, imdb_rating, rated FROM Shows WHERE 1=1";
25 $params = [];
26 $session_key = $_SERVER["SCRIPT_NAME"];
27 $is_clear = isset($_GET["clear"]);
28 if ($is_clear) {
29     session_delete($session_key);
30 }
31
32 $query .= " AND title like :title";
33 $params[":title"] = "%$title%";
34
35 $query .= " AND genres like :genres";
36 $params[":genres"] = "%$genres%";
37
38 $query .= " AND imdb_rating like :imdb_rating";
39 $params[":imdb_rating"] = "%$imdb_rating%";
40
41 $sort = se($_GET, "sort", "date", false);
42 if ($sort == "date") {
43     $query .= " ORDER BY $sort $order";
44 }
45
46 foreach ($form as $k => $v) {
47     if (in_array($v["name"], $keys)) {
48         $form[$v["name"]] = $_GET[$v["name"]];
49     }
50 }
51
52 //sort and order
53 $title = se($_GET, "title", "", false);
54 if (!empty($title)) {
55     $query .= " AND title like :title";
56     $params[":title"] = "%$title%";
57 }
58
59 $genres = se($_GET, "genres", "", false);
60 if (!empty($genres) && $genres != "") {
61     $query .= " AND genres like :genres";
62     $params[":genres"] = "%$genres%";
63 }
64
65 $imdb_rating = se($_GET, "imdb_rating", "", false);
66 if (!empty($imdb_rating) && $imdb_rating != "") {
67     $query .= " AND imdb_rating like :imdb_rating";
68     $params[":imdb_rating"] = "%$imdb_rating%";
69 }
70
71 $rated = se($_GET, "rated", "", false);
72 if (!empty($rated) && $rated != "") {
73     $query .= " AND rated like :rated";
74     $params[":rated"] = "%$rated%";
75 }
76
77 //sort and order
78 $sort = se($_GET, "sort", "date", false);
79 if (in_array($sort, ["title", "genres", "imdb_rating", "rated"])) {
80     $sort = "date";
81 }
82 $order = se($_GET, "order", "desc", false);
83 if (in_array($order, ["asc", "desc"])) {
84     $order = "desc";
85 }
86
87 //IMPORTANT make sure you fully validate/trust $sort and $order
88 $query .= " ORDER BY $sort $order";
89 //limit
90 try {
91     $limit = (int)se($_GET, "limit", "10", false);
92 } catch (Exception $e) {
93     $limit = 10;
94 }
95 if ($limit < 1 || $limit > 100) {
96     $limit = 10;
97 }
98
99 //IMPORTANT make sure you fully validate/trust $limit (sql injection)
100 $query .= " LIMIT $limit";

```

the filter/sort form generation AND role guard (admin)

Checklist Items (0)

```

public_html > project > admin > list_shows.php > ...
...
1 <?php
2 //note we need to go up 1 more directory
3 require_once __DIR__ . "/../../../../partials/nav.php";
4 //JW426 4/26/24
5 if (!has_role("Admin")) {
6     flash("You don't have permission to view this page", "warning");
7     redirect("home.php");
8 }
9
10 //build search form
11 $form = [
12     ["type" => "text", "name" => "title", "placeholder" => "Show Title", "label" => "Show Title", "include_margin" => false],
13     ["type" => "text", "name" => "imdb_rating", "placeholder" => "Show Rating", "label" => "Show Rating", "include_margin" => false],
14     ["type" => "text", "name" => "rated", "placeholder" => "Show Audience", "label" => "Show Audience", "include_margin" => false],
15     ["type" => "text", "name" => "genres", "placeholder" => "Genre(s)", "label" => "Genre(s)", "include_margin" => false],
16     ["type" => "text", "name" => "sort", "label" => "Sort", "options" => ["title" => "Title", "genres" => "Genre(s)", "imdb_rating" => "Rating", "rated" => "Rated"], "include_margin" => false],
17     ["type" => "select", "name" => "order", "label" => "Order", "options" => ["asc" => "+", "desc" => "-"], "include_margin" => false],
18     ["type" => "number", "name" => "limit", "label" => "Limit", "value" => "10", "include_margin" => false],
19 ];
20 <?11-21 $form ->
21
22 error_log("Form data: " . var_export($form, true));
23
24 $query = "SELECT id, title, genres, imdb_id, imdb_rating, rated FROM Shows WHERE 1=1";
25 $params = [];
26 $session_key = $_SERVER["SCRIPT_NAME"];
27 $is_clear = isset($_GET["clear"]);
28 if ($is_clear) {
29     session_delete($session_key);
30     unset($_GET["clear"]);
31     redirect($session_key);
32 }
33
34 $query .= " AND title like :title";
35 $params[":title"] = "%$title%";
36
37 $query .= " AND genres like :genres";
38 $params[":genres"] = "%$genres%";
39
40 $query .= " AND imdb_rating like :imdb_rating";
41 $params[":imdb_rating"] = "%$imdb_rating%";
42
43 $sort = se($_GET, "sort", "date", false);
44 if ($sort == "date") {
45     $query .= " ORDER BY $sort $order";
46 }
47
48 foreach ($form as $k => $v) {
49     if (in_array($v["name"], $keys)) {
50         $form[$v["name"]] = $_GET[$v["name"]];
51     }
52 }
53
54 //sort and order
55 $title = se($_GET, "title", "", false);
56 if (!empty($title)) {
57     $query .= " AND title like :title";
58     $params[":title"] = "%$title%";
59 }
60
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
80 $sort = se($_GET, "sort", "date", false);
81 if (in_array($sort, ["title", "genres", "imdb_rating", "rated"])) {
82     $sort = "date";
83 }
84 $order = se($_GET, "order", "desc", false);
85 if (in_array($order, ["asc", "desc"])) {
86     $order = "desc";
87 }
88
89 //IMPORTANT make sure you fully validate/trust $sort and $order
90 $query .= " ORDER BY $sort $order";
91 //limit
92 try {
93     $limit = (int)se($_GET, "limit", "10", false);
94 } catch (Exception $e) {
95     $limit = 10;
96 }
97 if ($limit < 1 || $limit > 100) {
98     $limit = 10;
99 }
100
101 //IMPORTANT make sure you fully validate/trust $limit (sql injection)
102 $query .= " LIMIT $limit";

```

```

51 //title
52 $title = se($_GET, "title", "", false);
53 if (!empty($title)) {
54
78 $sort = se($_GET, "sort", "date", false);
79 if (in_array($sort, ["title", "genres", "imdb_rating",
80     $sort = "date";
81 })
100 + 842-100 if (count($_GET) > 0)
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
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149

```

the DB query and how the filter/sort is handled (including the restriction on the limit field)

Checklist Items (0)

```

File Edit Selection View Go Run ... ← → JN426-IT202-008
File Edit Selection View Go Run ... ← → JN426-IT202-008
public_html > project > admin > list_shows.php > ...
180
181 ****
182
183
184 //JN426 4/26/24
185 $db = getDB();
186 $stmt = $db->prepare($query);
187 $results = [];
188 try {
189     $stmt->execute($params);
190     $r = $stmt->fetchAll();
191     if ($r) {
192         $results = $r;
193     }
194 } catch (PDOException $e) {
195     error_log("Error fetching shows " . var_export($e, true));
196     flash("Unhandled error occurred", "danger");
197 }
198
199 $table = [
200     "data" => $results,
201     "title" => "Latest Shows",
202     "ignored_columns" => ["id"],
203     "view_url" => get_url("admin/view_show.php"),
204     "edit_url" => get_url("admin/edit_show.php"),
205     "delete_url" => get_url("admin/delete_show.php")
206 ];
207 <- #120-125 $table =
208
209 </div>
210 <h3>List Shows</h3>
211 <form method="GET">
212     <div class="row mb-3" style="align-items: flex-end;">
213         <?php foreach ($form as $k => $v) : ?>
214             <div class="col">
215                 <?php render_input($v); ?>
216             </div>
217         </?php endforeach; ?>
218     </div>
219     <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
220     <a href="#clear" class="btn btn-secondary">Clear</a>
221 </form>
222 <?php render_table($table); ?>
223 </div>
224
225 <?php
226 //note we need to go up 1 more directory
227 require_once(__DIR__ . "/../../../../partials/flash.php");
228 ?>

```

more of db query and Show how the output is generated and displayed

Checklist Items (0)

Task #3 - Points: 1

Text: Explain how the page works

Checklist		*The checkboxes are for your own tracking
#	Points	Details
<input type="checkbox"/> #1	1	Provide the high-level steps of how the filter/sorting works and how data is displayed
<input type="checkbox"/> #2	1	Summarize what you're showing on the screen
<input type="checkbox"/> #3	1	Mention your design/style choice

Response:

The filtering and sorting functionality lets users refine their search criteria. Through a form containing input fields like show title, IMDb rating, audience rating, genres, and sorting options for title, genres, IMDb rating, and audience rating, users specify their preferences. Upon form submission, the server processes the input, constructs a SQL query based on the provided parameters, and executes it against the database. The query retrieves relevant show data from the "Shows" table, filtered and sorted according to the user's selections. Error handling ensures reliability, with flash messages alerting users to any encountered issues during data retrieval. The results are then presented in a table format, showcasing the latest shows with options for viewing, editing, and deleting individual entries. Users can further refine their search or reset the filters using the provided controls. On the screen is a summary view list of all the shows with the option to view, delete, and edit each one. It gives the option to go through the list through different filters. The style is just bootstrap. The only users who can view, edit, and delete are the users with the admin role.

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 this page
<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/72>

URL #2

https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/list_shows.php

View Details Page (single entity) (1 pt.)

 COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the details page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Entity should be fetch by id (via the url)
<input type="checkbox"/> #2	1	A missing id should redirect back to the list page with an applicable message
<input type="checkbox"/> #3	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
<input type="checkbox"/> #4	1	Data shown should be more detailed/inclusive than the summary view
<input type="checkbox"/> #5	1	There should be a link to edit the entity (this may be an admin-only thing, but it should be present)

		present for the respective role)
<input type="checkbox"/> #6	1	There should be a link to delete the entity (this may be an admin-only thing, but it should be present for the respective role)
<input type="checkbox"/> #7	1	Make sure the heroku dev url is visible in the address bar
<input type="checkbox"/> #8	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The Sopranos

Released: 1999-01-10

Audience: TV-MA

Rating: 8.6

Genre(s): Drama, Crime, Family

Description: The story of New Jersey-based Italian-American mobster Tony Soprano and the difficulties he faces as he tries to balance the conflicting requirements of his home life and the criminal organization he heads. Those difficulties are often highlighted through his ongoing professional relationship with psychiatrist Jennifer Melfi. The show features Tony's family members and Mafia associates in prominent roles and story arcs, most notably his wife Carmela and his cousin and protégé Christopher Moltisanti.

Back Edit Delete

-Entity should be fetched by id (via the url) -Data shown should be more detailed/inclusive than the summary view (has description unlike the summary view) -there is a link to delete and edit

Checklist Items (0)

Invalid id passed

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

Show Title Show Rating Show Audience Genre(s) Title + 10

Filter Clear

Latest Shows

title	genres	imdb_id	imdb_rating	rated	Actions	
Breaking Bad	Drama, Crime, Thriller, Western, Suspense	tt0903747	8.9	TV-MA	<button>View</button>	<button>Edit</button>
					<button>Delete</button>	
Game of Thrones	Fantasy, Science-Fiction, Drama, Action, Adventure, Mystery, War, Science-fiction	tt0944947	8.4	TV-MA	<button>View</button>	<button>Edit</button>
					<button>Delete</button>	
Abbott Elementary	Comedy	tt14218830	7.724	TV-PG	<button>View</button>	<button>Edit</button>
					<button>Delete</button>	

Missing ID redirection

Checklist Items (0)

Task #2 - Points: 1

Text: Screenshots of the details page code

Checklist

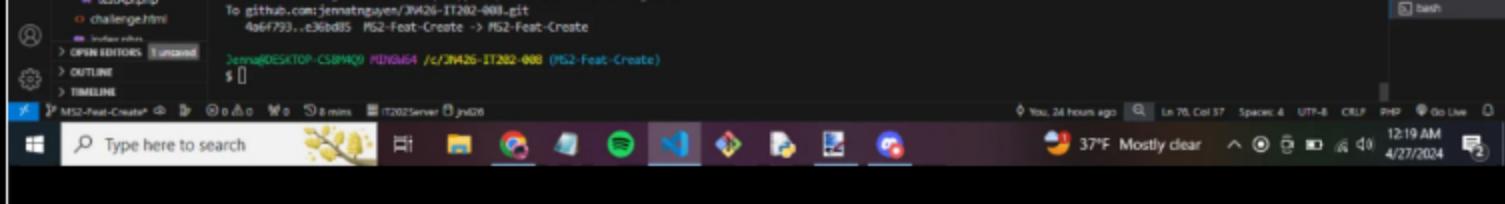
*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show how id is fetched
<input type="checkbox"/> #2	1	Show the DB query to get the record
<input type="checkbox"/> #3	1	Show the code related to presenting the data and showing the links
<input type="checkbox"/> #4	1	Include ucid/date comments for each code screenshot
<input type="checkbox"/> #5	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



- shows how id is fetched and db query - bonus fetch: fetches other API information from a different endpoint

Checklist Items (0)

```
public_html > project > admin > view_show.php > div.container-fluid
  61 } <- 857-61 Foreach ($show as $key => $value)
  62
  63
  64 ?>
<div class="container-fluid">
  <h3><?php echo $show['title']; ?></h3>
  <?php
    //Fetch jn426 4/26/24
    echo "<div style='text-align: center;'>";
    echo "<h2>{$show['title']}</h2>";
    echo "<p><strong>Released:</strong> <strong>{$show['release_date']}

remote: Resolving deltas: 100% (7/7), completed with 7 local objects.  
To github.com:jennnguyen/JN426-IT202-008.git  
 4a6f793..e36bd85 M52-Feat-Create -> M52-Feat-Create


```

how data is presented

Checklist Items (0)

Task #3 - Points: 1

Text: Explain how the page works

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Provide the high-level steps for handling the DB lookup and presenting the data

Response:

The script first checks if the user has admin privileges; if not, they're redirected with a warning. It then retrieves the show's data based on the provided ID from the "Shows" table in the database. Error handling ensures any issues, like database lookup errors, are appropriately managed. The retrieved show details are then presented on the page, including title, release date, ratings, genres, and description through HTML. If available, additional data may be fetched from an external API using the IMDB ID. Buttons for navigating back, editing, or deleting the show are provided, and

from an external API using the IMDB ID. Buttons for navigating back, editing, or deleting the show are provided, and flash messages give feedback on any encountered errors or successful operations.

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 this page
<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/72>

URL #2

https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/view_show.php?id=1

Edit Data Page (2 pts.)

[COLLAPSE](#)

Task #1 - Points: 1

Text: Screenshots of the edit page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show before and after screenshots of data you'll edit
<input type="checkbox"/> #2	1	Show examples of validation messages
<input type="checkbox"/> #3	1	Show an example of successful edit messages
<input type="checkbox"/> #4	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
<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

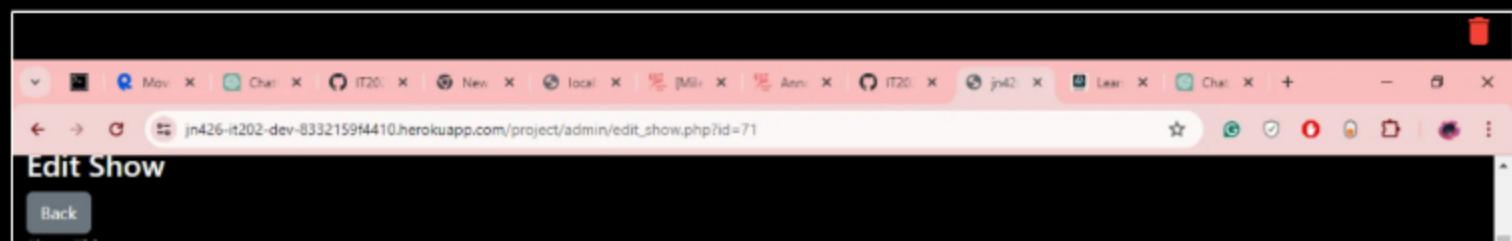
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Show Title
custom show

Release Date
01/02/0003

Show Description
d

Show Rating
Show Rating

Show Audience
d

Genre(s)
d

Update

37°F Mostly clear 12:24 AM 4/27/2024

Before

Checklist Items (0)

Edit Show

Back

Show Title
custom show

Release Date
01/02/0003

Show Description
edited

Show Rating
10

Show Audience
d

Genre(s)
d

Update

37°F Mostly clear 12:24 AM 4/27/2024

After

Checklist Items (0)

37°F Mostly clear 12:24 AM 4/27/2024

Show Tracker Home Profile Admin ▾ Show Management ▾ Logout

Updated record

Edit Show

Back

Show Title

custom show

Release Date

01/02/0003



Show Description

edited

Show Rating

10

Show Audience

d



successful edit

Checklist Items (0)

The screenshot shows a browser window with a red error message box containing two validation messages: "Title must not be empty" and "description must not be empty". The URL in the address bar is jn426-it202-dev-8332159f4410.herokuapp.com/project/admin/edit_show.php?id=71. The browser's developer tools are open, showing the "Console" tab with no issues. The page content is identical to the one above, showing the "Edit Show" form with empty fields.

validation message

Checklist Items (0)

Task #2 - Points: 1



Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Form should have correct data types for each property being requested
<input type="checkbox"/> #2	1	Form should have correct validation for each field (HTML, JS, and PHP)
<input type="checkbox"/> #3	1	Successful edit should have a user-friendly message
<input type="checkbox"/> #4	1	Any errors should have user-friendly messages
<input type="checkbox"/> #5	1	Include any other rules like role guards and login checks
<input type="checkbox"/> #6	1	Include ucid/date comments for each code screenshot
<input type="checkbox"/> #7	1	Clearly caption screenshots

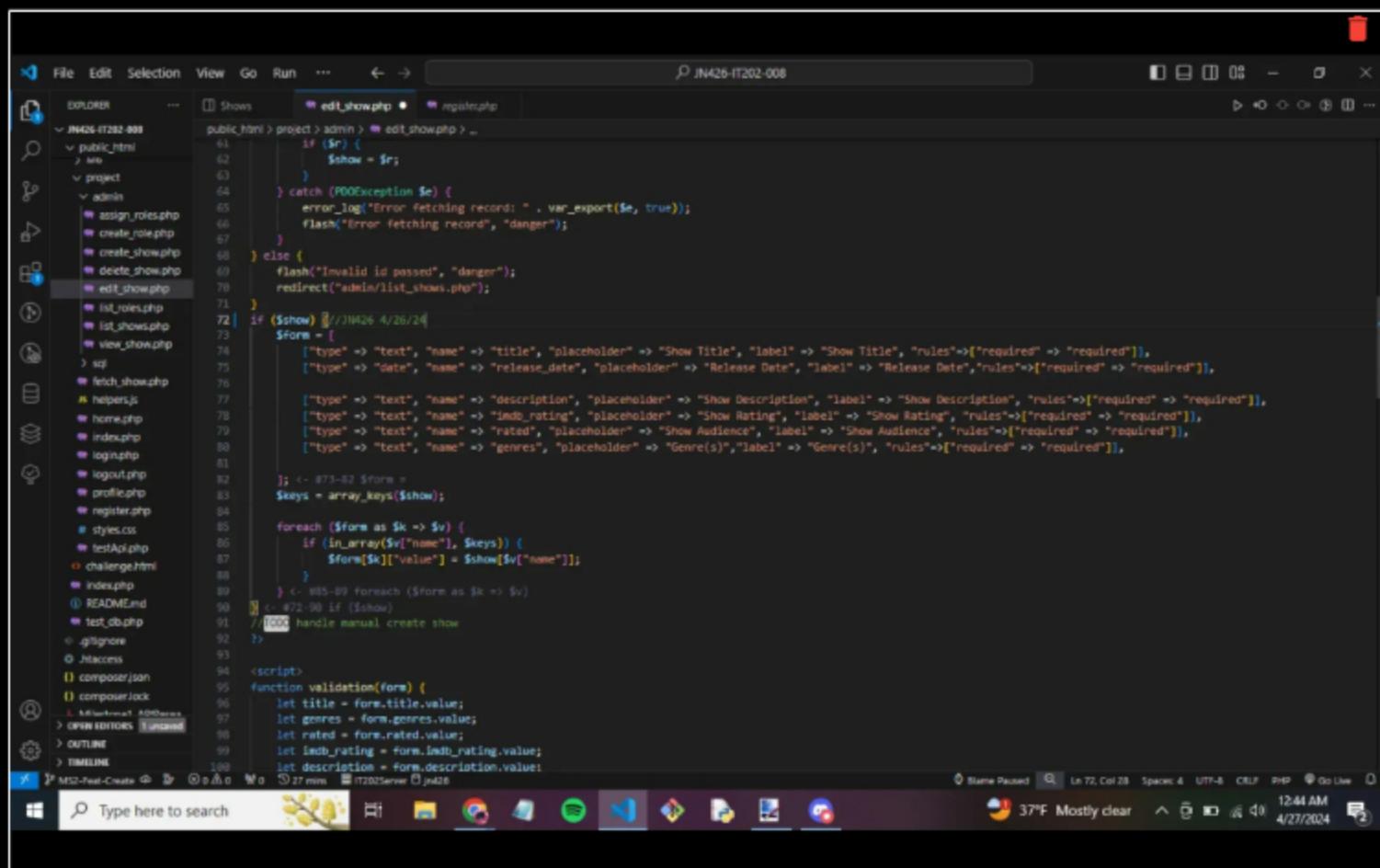
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



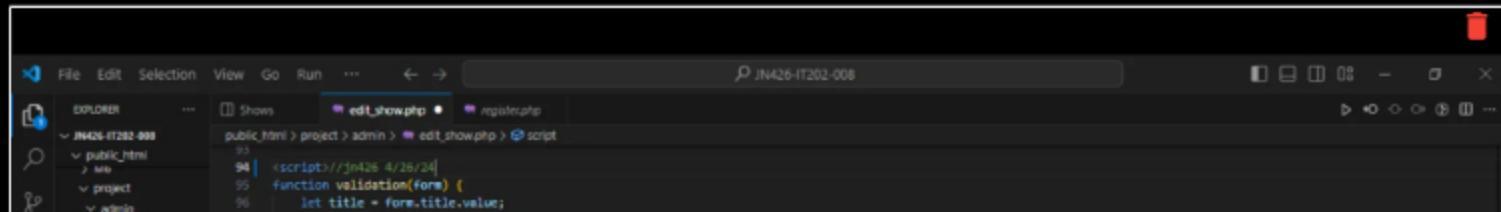
```

public_html > project > admin > edit_show.php > ...
  61 | if ($r) {
  62 |     $show = $r;
  63 |
  64 } catch (PDOException $e) {
  65     error_log("Error fetching record: " . var_export($e, true));
  66     flash("Error fetching record", "danger");
  67 }
  68 } else {
  69     flash("Invalid id passed", "danger");
  70     redirect("admin/list_shows.php");
  71 }
  72 if ($show) //JN426 4/26/24
  $form = [
    ["type" => "text", "name" => "title", "placeholder" => "Show Title", "label" => "Show Title", "rules"=>["required" => "required"]],
    ["type" => "date", "name" => "release_date", "placeholder" => "Release Date", "label" => "Release Date", "rules"=>["required" => "required"]], 
    ["type" => "text", "name" => "description", "placeholder" => "Show Description", "label" => "Show Description", "rules"=>["required" => "required"]],
    ["type" => "text", "name" => "imdb_rating", "placeholder" => "Show Rating", "label" => "Show Rating", "rules"=>["required" => "required"]],
    ["type" => "text", "name" => "rated", "placeholder" => "Show Audience", "label" => "Show Audience", "rules"=>["required" => "required"]],
    ["type" => "text", "name" => "genres", "placeholder" => "Genre(s)", "label" => "Genre(s)", "rules"=>["required" => "required"]], 
];
// JN426-82 $keys =
$keys = array_keys($show);
foreach ($form as $k => $v) {
    if (in_array($v["name"], $keys)) {
        $form[$k]["value"] = $show[$v["name"]];
    }
}
// JN426-90 foreach ($form as $k => $v)
// JN426-90 if ($show)
// JN426-90 handle manual create show
// JN426-90
<script>
function validation(form) {
    let title = form.title.value;
    let genres = form.genres.value;
    let rated = form.rated.value;
    let imdb_rating = form.imdb_rating.value;
    let description = form.description.value;
}

```

FORM HAS CORRECT DATA TYPES PASSED

Checklist Items (0)



```

public_html > project > admin > edit_show.php > script
  93 <script>//JN426 4/26/24
  94 function validation(form) {
  95     let title = form.title.value;
  96
  97     if (title === "") {
  98         alert("Title is required");
  99         return false;
  100    }
  101
  102    let genres = form.genres.value;
  103
  104    if (genres === "") {
  105        alert("Genre(s) is required");
  106        return false;
  107    }
  108
  109    let rated = form.rated.value;
  110
  111    if (rated === "") {
  112        alert("Rated is required");
  113        return false;
  114    }
  115
  116    let imdb_rating = form.imdb_rating.value;
  117
  118    if (imdb_rating === "") {
  119        alert("IMDb Rating is required");
  120        return false;
  121    }
  122
  123    let description = form.description.value;
  124
  125    if (description === "") {
  126        alert("Description is required");
  127        return false;
  128    }
  129
  130    return true;
  131 }

```

```
 97 let genres = form.genres.value;
 98 let rated = form.rated.value;
 99 let imdb_rating = form.imdb_rating.value;
100 let description = form.description.value;
101 let valid = true;
102
103 if(title.length == 0) {
104     flash("Must include title", "warning");
105     valid = false;
106 }
107
108 if(genres.length == 0) {
109     flash("Must include genre", "warning");
110     valid = false;
111 }
112
113 if(imdb_rating.length == 0) {
114     flash("Must include rating", "warning");
115     valid = false;
116 }
117
118 if(isNaN(imdb_rating)) {
119     flash("Rating must be a number", "warning");
120     valid = false;
121 }
122
123 if(description.length == 0) {
124     flash("Must include description", "warning");
125     valid = false;
126 }
127
128 return valid;
129 } //-- #125-129 function validation(form)
130 </script>
131
132 </p>
```

JS VALIDATION

Checklist Items (0)

The screenshot shows a Microsoft Visual Studio Code interface with two tabs open: `edit_show.php` and `register.php`. The left sidebar displays a file tree for a project named `JN426-IT202-008`, specifically navigating through the `public_html/project/admin` directory. The `edit_show.php` file contains PHP code for updating a show record, including validation logic for title, genres, rating, and description fields. The `register.php` file is partially visible at the top right. The bottom status bar shows the file path as `IT202Server/JN426`, the current file as `edit_show.php`, and other details like line numbers and file size.

```
public_html > project > admin > edit_show.php > ..
```

```
132     </php>
133     if (isset($_POST["title"]) && isset($_POST["genres"]) && isset($_POST["rated"]) && isset($_POST["description"])) {
134         $title = se($_POST, "title", "", false);
135         $genres = se($_POST, "genres", "", false);
136         $rated = se($_POST, "rated", "", false);
137         $imdb_rating = se($_POST, "imdb_rating", "", false);
138         $description = se($_POST, "description", "", false);
139         // [000] 3
140         $hasError = false;
141         //jn426 4/26/20
142         if (empty($title)) {
143             flash("Title must not be empty", "danger");
144             $hasError = true;
145         }
146         if (empty($genres)) {
147             flash("Genre must not be empty", "danger");
148             $hasError = true;
149         }
150         if (empty($rated)) {
151             flash("Audience rating must not be empty", "danger");
152             $hasError = true;
153         }
154         if (empty($imdb_rating)) {
155             flash("Rating must not be empty", "danger");
156             $hasError = true;
157         }
158         if (empty($description)) {
159             flash("Description must not be empty", "danger");
160             $hasError = true;
161         }
162         if ($hasError) {
163             // [000] 4
164             $params = [
165                 "title" => $title,
166                 "genres" => $genres,
167                 "rated" => $rated,
168                 "imdb_rating" => $imdb_rating,
169                 "description" => $description
170             ]; // #167-173 $params =
171             $db = getDB();
172             $stmt = $db->prepare("UPDATE Shows SET title = :title, genres = :genres, rated = :rated, imdb_rating = :imdb_rating, description = :description");
173             $stmt->execute($params);
174             flash("Successfully updated!", "success");
175         } catch (PDOException $e) {
176             users_check_duplicate($e->errorInfo);
177         }
178     } // #165-183 if (!$hasError)
179 } // #183-184 if (isset($_POST["title"]) && isset($_POST["genres"]) && isset($_POST["rated"]) && isset($_POST["description"]))
180 // [000] 4
181 </div class="container-fluid">
```

php validation

Checklist Items (0)

A screenshot of a web browser window. The address bar shows a local file path: 'public_html > project > admin > edit_show.php'. The main content area displays a code editor with PHP code for 'edit_show.php'. The code includes a header, a database connection, and a query to select data from a table named 'tbl_products'. A red box highlights the file name 'edit_show.php' in both the address bar and the code editor.

File Edit Selection View Go Run ...

```

12 $Id = se($_GET, "id", -1, false);
13 //TODO handle show fetch, JN426 4/26/24
14 if (isset($_POST["title"])) {
15     foreach ($_POST as $k => $v) {
16         if (!in_array($k, ["title", "release_date", "imdb_id", "description", "rated", "imdb_rating", "genres"])) {
17             unset($_POST[$k]);
18         }
19     }
20     $quote = $_POST;
21     error_log("Cleaned up POST: " . var_export($quote, true));
22     //insert data
23     $db = getDB();
24     $query = "UPDATE 'Shows' SET ";
25
26     $params = [];
27     //per record
28     foreach ($quote as $k => $v) {
29
30         if ($params) {
31             $query .= ",";
32         }
33         //be sure $k is trusted as this is a source of sql injection
34         $query .= "$k=$v";
35         $params[$k] = $v;
36     }
37
38     $query .= " WHERE id = :id";
39     $params[":id"] = $Id;
40     error_log("Query: " . $query);
41     error_log("Params: " . var_export($params, true));
42     try {
43         $stmt = $db->prepare($query);
44         $stmt->execute($params);
45         flash("Updated record", "success");
46     } catch (PDOException $e) {
47         error_log("Something broke with the query" . var_export($e, true));
48         flash("An error occurred", "danger");
49     }
50 } <- #14-50 If (isset($_POST["title"]))
51
52 $show = [];
53 if ($Id > -1) {
54     //fetch
55     $db = getDB();
56     $query = "SELECT title, release_date, imdb_id, description, rated, imdb_rating,
57     try {
58         $stmt = $db->prepare($query);
59         $stmt->execute([":id" => $Id]);
60         $r = $stmt->fetch();
61         if ($r) {
62             $show = $r;
63         }
64     } catch (PDOException $e) {
65         error_log("Error fetching record: " . var_export($e, true));
66         flash("Error fetching record", "danger");
67     }
68 } else {
69     flash("Invalid id passed", "danger");
70     redirect("admin/list_shows.php");
71 }
72 if ($show) { //JN426 4/26/24
73     $form = [
74         ["type" => "text", "name" => "title", "placeholder" => "Show Title", "label" => "Title"],
75         ["type" => "date", "name" => "release_date", "placeholder" => "Release Date", "label" => "Release Date"]
76     ];

```

You, 4 days ago | 1 author (You)

File Edit Selection View Go Run ...

37°F Mostly clear

12:47 AM 4/27/2024

success and error messages

Checklist Items (0)

File Edit Selection View Go Run ...

```

EXPLORER JN426-IT202-008
public_html > project > admin > edit_show.php ...
You, 12 minutes ago | 1 author (You)
<?php
//note we need to go up 1 more directory
require(__DIR__ . "/../../../../partials/nav.php");

if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    redirect("home.php");
}

<?php
$id = se($_GET, "id", -1, false);
//TODO handle show fetch, JN426 4/26/24
if (isset($_POST["title"])) {
    foreach ($_POST as $k => $v) {
        if (!in_array($k, ["title", "release_date", "imdb_id", "description", "rated", "imdb_rating", "genres"])) {
            unset($_POST[$k]);
        }
    }
    $quote = $_POST;
    error_log("Cleaned up POST: " . var_export($quote, true));
} <- #15-21 foreach ($_POST as $k => $v)
//insert data
$db = getDB();
$query = "UPDATE 'Shows' SET ";

$params = [];
//per record
foreach ($quote as $k => $v) {

    if ($params) {
        $query .= ",";
    }
}

```

You, 4 days ago | 1 author (You)

File Edit Selection View Go Run ...

35°F Mostly clear

12:50 AM 4/27/2024

admin role guard

Checklist Items (0)

Task #3 - Points: 1

COLLAPSE

Text: Screenshot of records from DB

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show a before and after screenshot of the record
<input type="checkbox"/> #2	1	Note what differs
<input type="checkbox"/> #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Id	Title	Release Date	IMDb ID	Created	Modified	Description	Poster	Rated	IMDb Rating
1	Breaking Bad	2008-01-20	tt0903747	2024-04-15 20:08:44	2024-04-26 03:37:07	Walter White, a New Mexico	(NULL)	TV-MA	8.9
2	Game of Thrones	2011-04-17	tt0944947	2024-04-17 17:54:37	2024-04-26 04:06:36	Seven noble families fight for	(NULL)	TV-MA	8.4
3	Abbott Elementary	2021-12-07	tt14218830	2024-04-19 03:52:45	2024-04-26 03:38:58	In this workplace comedy, a	(NULL)	TV-PG	7.724
4	The Sopranos	1999-01-10	tt0141842	2024-04-26 16:43:08	2024-04-26 16:43:08	The story of New Jersey-bar	(NULL)	(NULL)	(NULL)
5	Custom Show	2003-01-02	(NULL)	2024-04-27 03:46:30	2024-04-27 03:46:30	Edited	(NULL)	d	(NULL)
6	Custom Show 2	1999-01-11	(NULL)	2024-04-27 03:40:40	2024-04-27 03:40:40	This is a custom data	(NULL)	PG	(NULL)
7	Kamp Koral: SpongeBob's Under Years	2021-03-04	tt10436284	2024-04-27 03:44:22	2024-04-27 03:44:22	(NULL)	(NULL)	(NULL)	(NULL)
8	SpongeBob SquarePants	1999-05-01	tt0206512	2024-04-27 03:44:22	2024-04-27 03:44:22	(NULL)	(NULL)	(NULL)	(NULL)
9	The SpongeBob SquarePants Movie	2020-07-23	tt12895414	2024-04-27 03:44:22	2024-04-27 03:44:22	(NULL)	(NULL)	(NULL)	(NULL)

Before

Checklist Items (0)

Id	Title	Release Date	IMDb ID	Created	Modified	Description	Poster	Rated	IMDb Rating
1	Breaking Bad	2008-01-20	tt0903747	2024-04-15 20:08:44	2024-04-26 03:37:07	Walter White, a New Mexico	(NULL)	TV-MA	8.9
2	Game of Thrones	2011-04-17	tt0944947	2024-04-17 17:54:37	2024-04-26 04:06:36	Seven noble families fight for	(NULL)	TV-MA	8.4
3	Abbott Elementary	2021-12-07	tt14218830	2024-04-19 03:52:45	2024-04-26 03:38:58	In this workplace comedy, a	(NULL)	TV-PG	7.724
4	The Sopranos	1999-01-10	tt0141842	2024-04-26 16:43:08	2024-04-27 04:16:33	The story of New Jersey-bar	(NULL)	TV-MA	8.6
5	Custom Show	2003-01-02	(NULL)	2024-04-27 03:46:30	2024-04-27 04:25:34	Edited	(NULL)	d	10

```

    PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITHUB
    remote: Resolving deltas: 100% (7/7), completed with 7 local objects.
    To github.com:jennetinguyen/2N426-IT282-008.git
      4a6f793..e36bd85  MS2-Fest-Create -> MS2-Fest-Create
    D:\jenninguyen\2N426-IT282-008 (MS2-Fest-Create)
    $ 
  
```

After, description of custom show is changed from d to edited and the rating is from null to 10

Checklist Items (0)

Task #4 - Points: 1

Text: Explain the process

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Provide a high-level step-by-step of how the fetching of the record, populating the form, and the update works and gets changed in your DB
<input type="checkbox"/> #2	1	Briefly describe the validations for the applicable fields

Response:

The code starts by checking the user's role and redirects if they don't have admin permissions. Then, it retrieves the ID parameter from the URL and attempts to fetch the corresponding record from the database. If the record exists, it populates the form fields with the retrieved data. When the form is submitted, the PHP script handles the update process by sanitizing and validating the input fields. It constructs an SQL query to update the relevant fields in the database based on the provided data. If the update is successful, a success message is flashed; otherwise, an error message is displayed. JavaScript validation is also included to ensure that essential fields such as title, genre, audience rating, IMDb rating, and description are not empty, and that the IMDb rating is a valid number. The PHP script performs server-side validation for the same fields, ensuring that they are not empty before updating the database.

Task #5 - 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 this page

#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/80>**URL #2**https://jn426-it202-prod-a94514875a20.herokuapp.com/project/admin/edit_show.php?id=1**Delete Handling (1 pt.)**[COLLAPSE ^](#)**Task #1 - Points: 1****Text: Screenshots related to delete****Checklist****The checkboxes are for your own tracking*

#	Points	Details
<input type="checkbox"/> #1	1	Show the success message of a delete
<input type="checkbox"/> #2	1	Show any error messages of a failed delete (like id not being passed)
<input type="checkbox"/> #3	1	Show the code related to the delete processing

Task Screenshots:**Gallery Style: Large View**[Small](#)[Medium](#)[Large](#)

```
<?php
session_start();
require(__DIR__ . "/../../../../lib/functions.php");
if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    redirect("home.php");
}
$id = se($_GET, "id", -1, false);
if ($id < 1) {
    flash("Invalid id passed to delete", "danger");
    redirect("admin/list_shows.php");
}
$db = getDB();
$query = "DELETE FROM `Shows` WHERE id = :id";
try {
    $stmt = $db->prepare($query);
    $stmt->execute([":id" => $id]);
    flash("Deleted record with id $id", "success");
} catch (Exception $e) {
    error_log("Error deleting stock $id" . var_export($e, true));
    flash("Error deleting record", "danger");
}
redirect("admin/list_shows.php");
```

The screenshot shows the Microsoft Visual Studio Code interface with the file 'delete_show.php' open in the editor. The code implements a delete functionality for a show record in a MySQL database. It checks if the user has Admin role, gets the id from GET, and then tries to execute a DELETE query on the 'Shows' table. If successful, it flashes a success message. If an exception occurs, it logs the error and flashes a danger message. Finally, it redirects to the 'admin/list_shows.php' page.

Checklist Items (0)

Deleted record with id: 71

List Shows

Show Title	Show Rating	Show Audience	Genre(s)	Sort	Order	Limit
<input type="text" value="Show Title"/>	<input type="text" value="Show Rating"/>	<input type="text" value="Show Audience"/>	<input type="text" value="Genre(s)"/>	<input type="button" value="Title"/>	<input type="button" value="+"/>	10
<input type="button" value="Filter"/> <input type="button" value="Clear"/>						

Latest Shows

title	genres	imdb_id	imdb_rating	rated	Actions
Breaking Bad	Drama, Crime, Thriller, Western, Suspense	tt0903747	8.9	TV-MA	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Game of Thrones	Fantasy, Science-Fiction, Drama, Action, Adventure, Mystery, War, Science-fiction	tt0944947	8.4	TV-MA	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Abbott Elementary	Comedy	tt14218830	7.724	TV-PG	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Type here to search 35°F Mostly clear 12:55 AM 4/27/2024

delete success

Checklist Items (0)

Invalid id passed to delete

List Shows

Show Title	Show Rating	Show Audience	Genre(s)	Sort	Order	Limit
<input type="text" value="Show Title"/>	<input type="text" value="Show Rating"/>	<input type="text" value="Show Audience"/>	<input type="text" value="Genre(s)"/>	<input type="button" value="Title"/>	<input type="button" value="+"/>	10
<input type="button" value="Filter"/> <input type="button" value="Clear"/>						

Latest Shows

title	genres	imdb_id	imdb_rating	rated	Actions
Breaking Bad	Drama, Crime, Thriller, Western, Suspense	tt0903747	8.9	TV-MA	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Game of Thrones	Fantasy, Science-Fiction, Drama, Action, Adventure, Mystery, War, Science-fiction	tt0944947	8.4	TV-MA	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Abbott Elementary	Comedy	tt14218830	7.724	TV-PG	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Type here to search 35°F Mostly clear 12:55 AM 4/27/2024

Checklist Items (0)

▲ COLLAPSE ▾
Task #2 - Points: 1
Text: Screenshots of the data

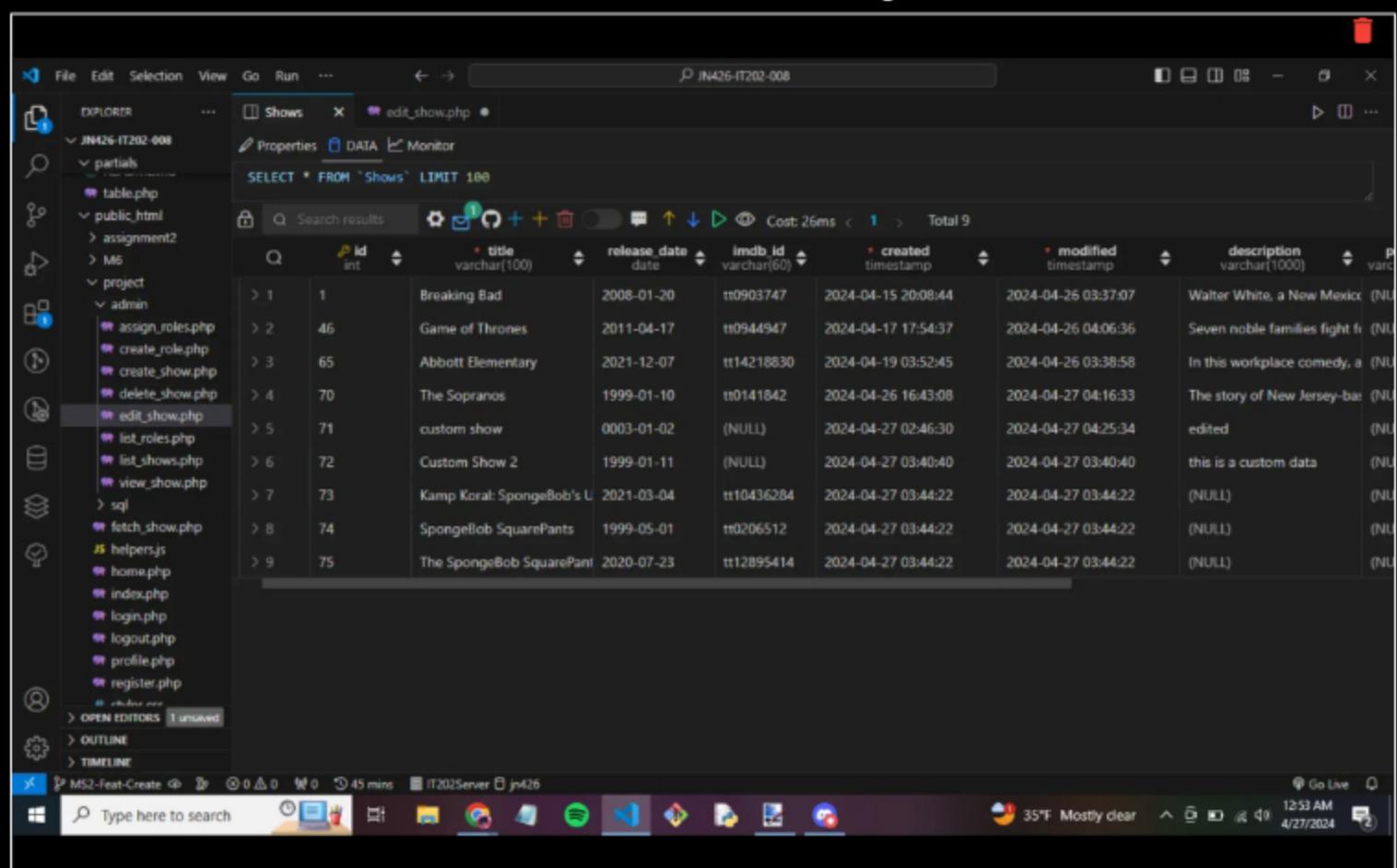
Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show a before and after screenshot of the DB data
<input type="checkbox"/> #2	1	Note what changed (i.e., record removed or soft delete value changed)
<input type="checkbox"/> #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

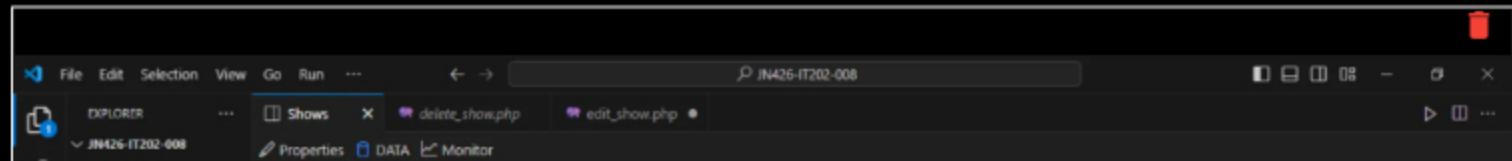
Small**Medium****Large**


The screenshot shows a Microsoft Visual Studio Code (VS Code) window with the following details:

- Explorer:** Shows a project structure for "JN426-IT202-008" containing files like "edit_show.php", "table.php", "public_html", "assignment2", "M6", "project", "admin", and various PHP scripts.
- Properties:** Shows the current file "edit_show.php" selected.
- SQL Editor:** Displays the SQL query: "SELECT * FROM 'Shows' LIMIT 100".
- Results Table:** A large table showing data from the "Shows" table. The columns are: id, title, release_date, imdb_id, created_timestamp, modified_timestamp, and description. The table contains 9 rows of data, including entries for "Breaking Bad", "Game of Thrones", "Abbott Elementary", "The Sopranos", "custom show", "Custom Show 2", "Kamp Koral: SpongeBob's Li", "SpongeBob SquarePants", and "The SpongeBob SquarePant".
- Status Bar:** Shows the file path "JN426-IT202-008", the status "45 mins", and the date "4/27/2024".

Before

Checklist Items (0)



The screenshot shows a Microsoft Visual Studio Code (VS Code) window with the following details:

- Explorer:** Shows a project structure for "JN426-IT202-008" containing files like "delete_show.php", "edit_show.php", "table.php", "public_html", "assignment2", "M6", and various PHP scripts.
- Properties:** Shows the current file "edit_show.php" selected.
- SQL Editor:** Displays the SQL query: "SELECT * FROM 'Shows' LIMIT 100".
- Results Table:** A large table showing data from the "Shows" table. The columns are: id, title, release_date, imdb_id, created_timestamp, modified_timestamp, and description. The table contains 9 rows of data, including entries for "Breaking Bad", "Game of Thrones", "Abbott Elementary", "The Sopranos", "custom show", "Custom Show 2", "Kamp Koral: SpongeBob's Li", "SpongeBob SquarePants", and "The SpongeBob SquarePant".
- Status Bar:** Shows the file path "JN426-IT202-008", the status "45 mins", and the date "4/27/2024".

SELECT * FROM 'Shows' LIMIT 100

Q	Id	Title	Release Date	Imdb Id	Created	Modified	Description
> 1	1	Breaking Bad	2008-01-20	tt0903747	2024-04-15 20:08:44	2024-04-26 03:37:07	Walter White, a New Mexico (NU)
> 2	46	Game of Thrones	2011-04-17	tt0944947	2024-04-17 17:54:37	2024-04-26 04:06:36	Seven noble families fight for (NU)
> 3	65	Abbott Elementary	2021-12-07	tt14218830	2024-04-19 03:52:45	2024-04-26 03:38:58	In this workplace comedy, a (NU)
> 4	70	The Sopranos	1999-01-10	tt0141842	2024-04-26 16:43:08	2024-04-27 04:16:33	The story of New Jersey-bas (NU)
> 5	72	Custom Show 2	1999-01-11	(NULL)	2024-04-27 03:40:40	2024-04-27 03:40:40	this is a custom data (NU)
> 6	73	Kamp Koral: SpongeBob's U	2021-03-04	tt10436284	2024-04-27 03:44:22	2024-04-27 03:44:22	(NULL)
> 7	74	SpongeBob SquarePants	1999-05-01	tt0206512	2024-04-27 03:44:22	2024-04-27 03:44:22	(NULL)
> 8	75	The SpongeBob SquarePant	2020-07-23	tt12895414	2024-04-27 03:44:22	2024-04-27 03:44:22	(NULL)

MS2-Fest-Create 45 mins IT202Server jn426 Go Live 35°F Mostly clear 12:55 AM 4/27/2024

After (hard delete) custom show gone

Checklist Items (0)

Task #3 - Points: 1

Text: Explain the delete logic

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Is it a soft or hard delete
<input type="checkbox"/> #2	1	Are there any necessary roles or restrictions? (can only delete their data, can only be done by admin, etc)
<input type="checkbox"/> #3	1	Provide the high-level steps for handling the DB lookup and handling the delete

Response:

This is a hard delete only done by the admin. The code begins by starting a session and including necessary functions. It then checks if the user has admin permissions and redirects them if not. Following that, it retrieves the ID parameter from the URL and verifies if it's valid. If the ID is less than 1, indicating an invalid ID, an error message is flashed, and the user is redirected to the list of shows page. If the ID is valid, the script proceeds to connect to the database and prepares a SQL query to delete the record with the specified ID from the "Shows" table. The query is executed, and if successful, a success message is flashed indicating the deletion of the record. In case of any exceptions or errors during the deletion process, an error message is flashed. Finally, the user is redirected back to the list of shows page.

Task #4 - Points: 1

[COLLAPSE](#)

Text: Add the pull request link for the branch related to this feature

i Details:

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/69>

[COLLAPSE](#)

Misc (1 pt.)

[COLLAPSE](#)

Task #1 - Points: 1

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

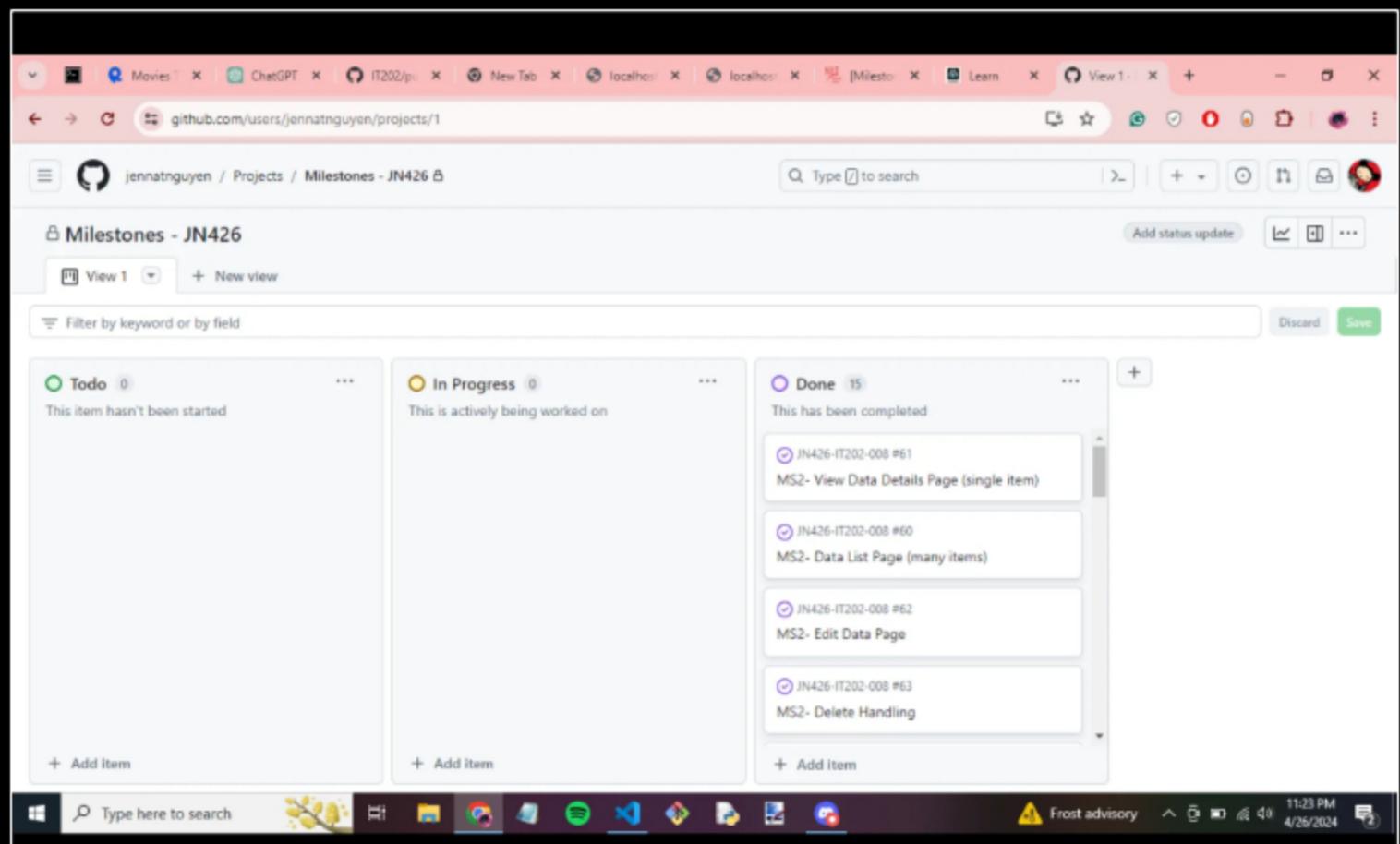
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



project board

[^COLLAPSE](#)

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>

[^COLLAPSE](#)

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

My biggest issue with the project was everything related to fetching from the API. I had trouble fetching from multiple endpoints and had to go through making a bunch of separate functions to assist with fetching, and when the multiple endpoint fetching worked, the fetching on the create page wasn't. Through solving this, I was able to learn more on how APIs work and how to retrieve data from them.

[^COLLAPSE](#)

Task #4 - Points: 1

Text: WakaTime Screenshot

Details:

Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows the WakaTime dashboard with a "Large" view selected. On the left, a sidebar lists navigation options: Upgrade, Dashboard, Insights, Teams, Projects (which is selected), Invoices, Goals, Shareables, Leaderboards, Integrations, and Supported IDEs. The main area displays a table of time spent across different categories. The "Files" column lists file paths, and the "Branches" column lists branches with their corresponding times. The browser's address bar shows the URL: wakatime.com/projects/JN426-IT202-008.

Files	Branches
3 hrs 58 mins ..project/admin/view_show.php	10 hrs 12 mins MS2-API-Integration-Admin
3 hrs 7 mins ..roject/admin/list_shows.php	1 hr 2 mins MS2-Feat-Create
2 hrs 32 mins ..roject/admin/create_show.php	55 mins Milestone2
55 mins ..project/admin/edit_show.php	40 mins MS2-Feat-Edit-Page
49 mins lib/show_api.php	
14 mins ..oject/admin/delete_show.php	
10 mins public_html/project/styles.css	
10 mins ..ic_html/project/testApi.php	
8 mins partials/nav.php	
8 mins lib/db_helpers.php	
7 mins ..html/project/fetch_show.php	
6 mins ...007_alter_table_shows.sql	
5 mins partials/input_field.php	
4 mins lib/redirect.php	
3 mins partials/table.php	
3 mins ..roject/admin/list_roles.php	
58 secs ...006_create_table_shows.sql	
52 secs lib/.env	

Blog
API Docs
Pricing
Community
FAQ

49 secs ...c_html/project/register.php
22 secs lib/api_helper.php
21 secs lib/functions.php
19 secs ...object/admin/create_role.php
17 secs .../0062_alter_table_shows.sql
16 secs ...ic_html/project/profile.php
12 secs public_html/project/home.php
9 secs ...ject/admin/assign_roles.php



WAKATIME SCREENSHOT

End of Assignment