

# Submission Worksheet

**CLICK TO GRADE**

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-3-2024/grade/njz5>

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

### Submissions:

Submission Selection

1 Submission [active] 4/29/2024 1:22:13 PM

### Instructions

[^ COLLAPSE ^](#)

Implement the Milestone 3 features from the project's proposal document: <https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88E>  
Make sure you add your ucid/date as code comments where code changes are done  
All code changes should reach the Milestone3 branch  
Create a pull request from Milestone3 to dev and keep it open until you get the output PDF from this assignment.  
Gather the evidence of feature completion based on the below tasks.  
Once finished, get the output PDF and copy/move it to your repository folder on your local machine.  
Run the necessary git add, commit, and push steps to move it to GitHub  
Complete the pull request that was opened earlier  
Create and merge a pull request from dev to prod  
Upload the same output PDF to Canvas

**Branch name:** Milestone3

**Tasks:** 26 **Points:** 10.00

 API (1 pt.)

[^ COLLAPSE ^](#)

 Task #1 - Points: 1

**Text:** Data Related to Users

**Checklist**

\*The checkboxes are for your own tracking

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

#	Points	Details
<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:

The concept that I am using is that users can save any book they want in order to view it later. Each book has a unique id (the one from the default id, created, modified) in the Books table, and the UserBooks table maps these book ids to the respective user ids.

The relationship is many to many because many users can save many books. One user can save as many books as they want, and many users can save the same books. This means that books are not unique per user, so anyone can save any book they want.

### 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:

When a user fetches an already stored book from the api, the only thing that changes is the book id in the books table. To do this, I did "ON DUPLICATE KEY UPDATE bookID = VALUES(bookID)". Book information shouldn't change much, if at all, so I only did this to avoid duplicate entries in the database. When a user does a manual update for a book, only the information for the book with the respective id is changed.

Because users reference books in the UserBooks table through userID and bookID, they immediately receive the new data. No matter what happens in the books table, because they reference through id, they will always be able to access the exact, latest book that they want.

### Handle Data Association (1 pt.)

### 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

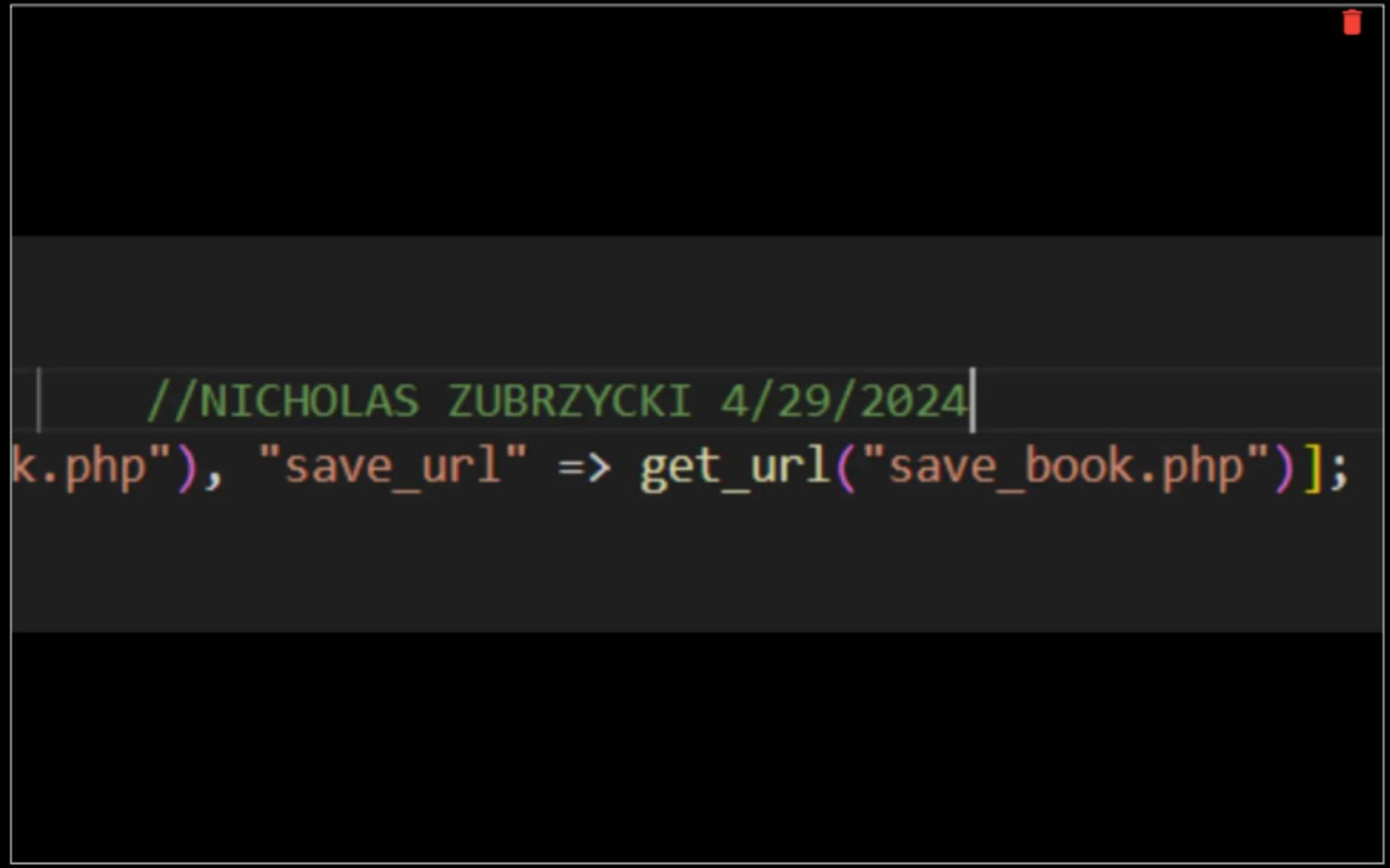
## Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



```
//NICHOLAS ZUBRZYCKI 4/29/2024
k.php"), "save_url" => get_url("save_book.php")];
```

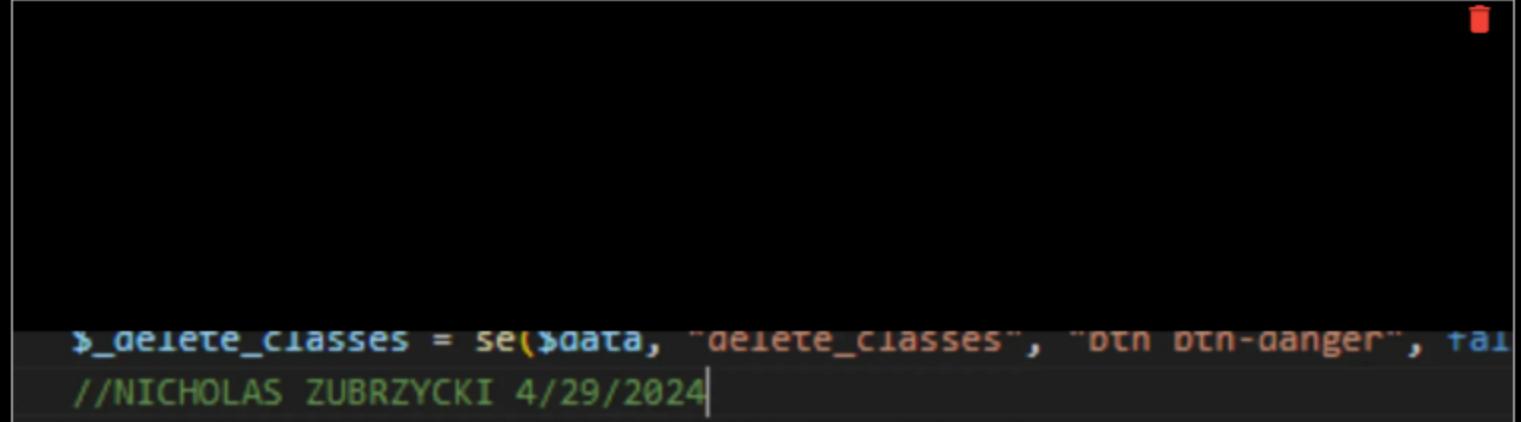
Code to render the save button on the books listing page in the table actions.

## 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



```
$delete_classes = self::data, "delete_classes", "btn btn-danger", true
//NICHOLAS ZUBRZYCKI 4/29/2024
$
```

```
$_save_url = se($data, "save_url", "", false);
$_save_label = se($data, "save_label", "Save", false);
$_save_classes = se($data, "save_classes", "btn btn-info", false);
$_unsave_url = se($data, "unsave_url", "", false);
```

### Save button options in table.

#### Checklist Items (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)

```
//NICHOLAS ZUBRZYCKI 4/29/2024
<?php if ($_save_url) : ?>
    <a href="php se($_save_url); ?&gt;&lt;?php se($_primary_key_column); ?&gt;=?&lt;?php se($row, $_primary_key_column); ?&gt;" class="<?php se($_save_classes); ?&gt;"php se($_save_label); ?&gt;&lt;/a&gt;
&lt;?php endif; ?&gt;</pre
```

### Render save button in table.

#### Checklist Items (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)

```
...
1  <?php
2 //NICHOLAS ZUBRZYCKI 4/29/2024
3 session_start();
4 require(__DIR__ . "/../../lib/functions.php");
5 /* if (!has_role("Admin")) {
6     flash("You don't have permission to view this page", "warning");
```

```

7     exit(header("Location: $BASE_PATH" . "/home.php"));
8 } */
9
10 $id = se($_GET, "id", -1, false);
11 if ($id < 1) {
12     flash("Invalid id passed to save", "danger");
13     //exit(header("Location: " . get_url("list_books.php")));
14     redirect("list_books.php");
15 }
16
17 $db = getDB();
18 try {
19     $stmt = $db->prepare("INSERT INTO `UserBooks` (`userID`, `bookID`) VALUES(:userID, :bookID)");
20     $stmt->execute([":userID" => get_user_id(), ":bookID" => $id]);
21     flash("Saved record with id $id", "success");
22 } catch (PDOException $e) {
23     if ($e->errorInfo[1] === 1062) {
24         flash("You already saved this book, please try another", "warning");
25     } else {
26         error_log("Error saving book $id" . var_export($e, true));
27         flash("Error saving book", "danger");
28     }
29 }
30
31 //exit(header("Location: " . get_url("list_books.php")));
32 redirect("list_books.php");

```

Code with SQL query that is done to save the book for the user into the database.

### 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



```
SELECT * FROM `UserBooks` LIMIT 100
```

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for search, refresh, add, delete, and navigation.
- Cost:** 126ms
- Total:** 5 rows
- Table Structure:**
  - Columns:** id (int), created (timestamp), modified (timestamp), userID (int), bookID (int), is\_active (tinyint(1)).
  - Primary Key:** id
  - Foreign Keys:** bookID references Books(id), userID references Users(id).
  - Unique Key:** saved\_book(userID, bookID)
- Data Rows:** 5 rows with IDs 1, 2, 3, 4, 5. Book IDs 1, 2, 3 are associated with user IDs 1, 2, 3 respectively. The is\_active column values are 1, 1, 1, 0, 0.

UserBooks table mapping userID to bookID, as well as is\_active used for toggle/assign. As shown, multiple users can have the same book.

#### Checklist Items (3)

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

#2 Should have some example data

#3 Clearly caption screenshots

```
CREATE TABLE `UserBooks` ( You, 6 days ago • Created UserBooks table for saved as
  `id` int NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `created` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `modified` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `userID` int,
  `bookID` int,
  FOREIGN KEY (`bookID`) REFERENCES `Books` (`id`),
  FOREIGN KEY (`userID`) REFERENCES `Users` (`id`),
  UNIQUE KEY `saved_book` (`userID`, `bookID`)
)
```

## UserBooks creation query, added is\_active later.

### 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 used option 1 for the association. On the books listing page, I added a save button where users can save the books that they want. However, option 2 also seems to be a separate requirement (Admins can associate any entity with any user) so I did that as well. This is done on the assign books page where you can toggle the books that users have.

In the UserBooks table, id, created, and modified are the three default fields that all tables have for this project.

The first column that I added is userID, which maps to the id field in Users that maps to each specific user. I used the get\_user\_id() helper function for these values, which retrieves the user id from the session data.

The next column I added is bookID, which maps to the unique ids in the Books table. This also maps to the default, unique id field, not the bookID that is returned from the api data

The last column is the is\_active field, where 1 means the book is active for the user and they're able to see it, and 0 means the book is inactive and the user is unable to see it in their personal books list. This column is only affected by the assign/toggle page, where toggle switches between 0 and 1. On any other page where the book is deleted, is\_active is not used since I used a hard delete.

#### 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 /#. Same pull request shouldn't be used for each feature

#### URL #1

[https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/list\\_books.php](https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/list_books.php)

#### URL #2

<https://github.com/NickZub/njz5-it202-008/pull/62>

#### URL #3

<https://github.com/NickZub/njz5-it202-008/pull/63>

#### URL #4

<https://github.com/NickZub/njz5-it202-008/pull/64>

 Current User's Association Page (2 pts.)

[^COLLAPSE ^](#)

#### Task #1 - Points: 1

[Text: Screenshots of this page](#)

#### Checklist

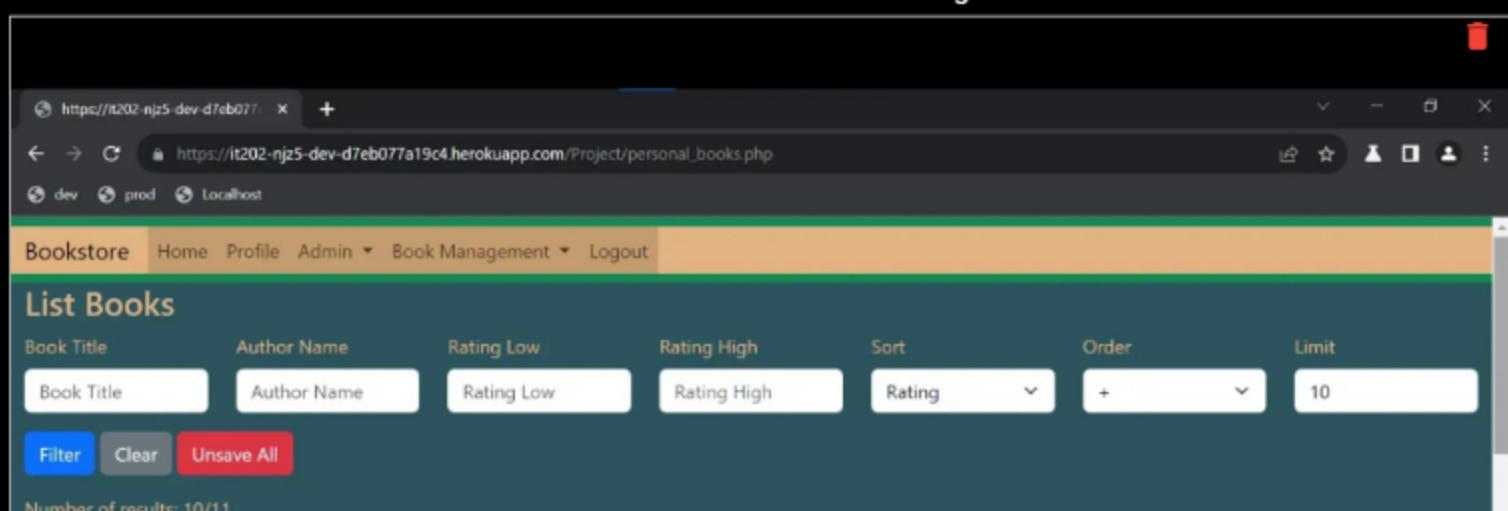
\*The checkboxes are for your own tracking

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

#### Task Screenshots:

##### Gallery Style: Large View

Small      Medium      Large



## Personal Books

title	author	image	rating	Actions
TitleZeroView	AuthorZero		0.00	<button>View</button> <button>Edit</button> <button>Unsave</button>
The Girls of Atomic City: The Untold Story of the Women Who Helped Win World War II	Denise Kiernan	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1352912866/15801668._SY500_.jpg	3.71	<button>View</button> <button>Edit</button>

Default personal books listing page with some saved books. 10 results by default. Unsave is the personal delete option.

### Checklist Items (6)

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

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

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

#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

#6 Clearly caption screenshots

https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/personal\_books.php?book\_title=&author\_name=&rating\_low=4&rating\_high=&sort=r...

title	author	image	rating	Actions
The Adventures of Tom Sawyer and Adventures of Huckleberry Finn	Mark Twain	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1597509702l/24580._SY500_.jpg	4.10	<button>View</button> <button>Edit</button> <button>Unsave</button>
The Adventures of Huckleberry Finn	Oliver Ho	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1388464215l/141315._SY500_.jpg	4.30	<button>View</button> <button>Edit</button> <button>Unsave</button>
The Bastard Brigade: The	Sam Kean	https://images-na.ssl-images-	4.32	<button>View</button>

**Checklist Items (4)**

#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

#6 Clearly caption screenshots

title	author	image	rating	Actions
Tvoj sin Huckleberry Finn	Bekim Sejranović	<a href="https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1446304155l/27400046._SY500_.jpg">https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1446304155l/27400046._SY500_.jpg</a>	4.37	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Unsave</a>

**Checklist Items (4)**

#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

#6 Clearly caption screenshots

← → C https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/personal\_books.php?book\_title=&author\_name=&rating\_low=5&rating\_high=&sort=... 🔍 ☆ 🌐

dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

## List Books

Book Title	Author Name	Rating Low	Rating High	Sort	Order	Limit
Book Title	Author Name	5	Rating High	Created	-	1

Filter Clear Unsave All

Number of results: 0/11

### Personal Books

No records to show

Personal books filter no records to show.

#### Checklist Items (3)

#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

#6 Clearly caption screenshots

← → C https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/view\_personal\_book.php?id=1 🔍 ☆ 🌐

dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

### Book: TitleZeroView

Back Edit



TitleZeroView

Author Name: AuthorZero
Rating: 0.00

Redirect to single view.

## Checklist Items (1)

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

Unsaved record with id: 1

### List Books

Book Title	Author Name	Rating Low	Rating High	Sort	Order	Limit
Book Title	Author Name	Rating Low	Rating High	Rating	+	10

Filter Clear Unsave All

Number of results: 2/2

### Personal Books

title	author	image	rating	Actions
TitleOne	AuthorOne	URLOne	0.00	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Unsave</a>
TitleTwo	AuthorTwo	URLTwo	0.02	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Unsave</a>

After delete.

## Checklist Items (1)

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

Unsaved every personal book.

### List Books

Book Title	Author Name	Rating Low	Rating High	Sort	Order	Limit
Book Title	Author Name	Rating Low	Rating High	Rating	+	10

Filter Clear Unsave All

Number of results: 0/0

## Personal Books

No records to show

After delete all.

### Checklist Items (1)

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

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

### Task Screenshots:

Gallery Style: Large View

Small      Medium      Large



```
//NICHOLAS ZUBRZYCKI 4/29/2024
$query = "SELECT id, bookID, title, author, image, rating FROM `Books` WHERE id IN
(SELECT bookID FROM `UserBooks` WHERE userID = :userID AND is_active=1)"; //ORDER BY created DESC LIMIT 25
```

Query to fetch user associations.

## Checklist Items (2)

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

#9 Include ucid/date comments for each code screenshot

```
//NICHOLAS ZUBRZYCKI 4/29/2024
$stable = ["data" => $results, "title" => "Personal Books", "ignored_columns" => [*id*, "bookID"], "edit_url" => get_url("admin/edit_personal_book.php"), "view_url" => get_url("view_personal
?>


### List Books:


<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>
    <a href="<?php echo get_url("delete_every_personal_book.php"); ?>" class="btn btn-danger">Unsave All</a>
</form>
<p>Number of results: <?php se($count); ?>/<?php se($total); ?></p>
<?php render_table($stable); ?>
</div>

<?php
//note we need to go up 1 more directory
require_once(__DIR__ . "/../../partials/flash.php");
?>


```

Displaying results, as well as the unsave all button.

## Checklist Items (2)

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

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

```
//NICHOLAS ZUBRZYCKI 4/29/2024
], "edit_url" => get_url("admin/edit_personal_book.php"), "view_url" => get_url("view_personal_book.php"), "unsave_url" => get_url("delete_personal_book.php")];
```

Edit, single view, and personal unsave buttons.

## 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

```
//NICHOLAS ZUBRZYCKI 4/29/2024
$db = getDB();
$stmt1 = $db->prepare($query);
$stmt2 = $db->prepare("SELECT * FROM `UserBooks` WHERE userID = :userID AND is_active=1");
$results = [];
$count = 0;
$total = 0;
try {
    //Get query data and filtered count
    $stmt1->execute($params);
    $r1 = $stmt1->fetchAll();
    if ($r1) {
        $results = $r1;
        $count = count($results);
    }
    //Get total count associated to user before filter
    $stmt2->execute([':userID' => $user_id]);
    $r2 = $stmt2->fetch();
    $total = $r2['count'];
}
```

```

$stmt2->execute(":userId" => get_user_id());
$r2 = $stmt2->fetchAll();
if ($r2) {
    $total = count($r2);
}
} catch (PDOException $e) {
    error_log("Error fetching stocks " . var_export($e, true));
    flash("Uncaught error occurred", "danger");
}

```

Logic for displaying the count of the number of results. \$count is the count with the filter applied, \$total is the general amount without the filter.

## 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)

```

//NICHOLAS ZUBRZYCKI 4/29/2024
if (count($_GET) > 0) {
    session_save($session_key, $_GET);
    $keys = array_keys($_GET);

    foreach ($form as $k => $v) {
        if (in_array($v["name"], $keys)) {
            $form[$k]["value"] = $_GET[$v["name"]];
        }
    }
    //book title
    $book_title = se($_GET, "book_title", "", false);
    if (!empty($book_title)) {
        $query .= " AND title like :title";
        $params[":title"] = "%$book_title%";
    }
    //author name
    $author_name = se($_GET, "author_name", "", false);
    if (!empty($author_name)) {
        $query .= " AND author like :author";
        $params[":author"] = "%$author_name%";
    }
    //rating low
    $rating_low = se($_GET, "rating_low", "-1", false);
    if (!empty($rating_low) && $rating_low > -1) {
        $query .= " AND rating >= :rating_low";
        $params[":rating_low"] = $rating_low;
    }
    //rating high
    $rating_high = se($_GET, "rating_high", "-1", false);
    if (!empty($rating_high) && $rating_high > -1) {
        $query .= " AND rating <= :rating_high";
        $params[":rating_high"] = $rating_high;
    }
    //sort and order
    $sort = se($_GET, "sort", "created", false);
    if (!in_array($sort, ["rating", "created"])) {
        $sort = "created";
    }
}

```

Filtering and sorting logic.

## Checklist Items (1)

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

```

}
//NICHOLAS ZUBRZYCKI 4/29/2024
$order = se($_GET, "order", "desc", false);
if (!in_array($order, ["asc", "desc"])) {
    $order = "desc";
}

```

```

//IMPORTANT make sure you fully validate/trust $sort and $order (sql injection possibility)
$query .= " ORDER BY $sort $order";
//limit
try {
    $limit = (int)se($_GET, "limit", "10", false);
} catch (Exception $e) {
    $limit = 10;
}
if ($limit < 1 || $limit > 100) {
    $limit = 10;
}
//IMPORTANT make sure you fully validate/trust $limit (sql injection possibility)
$query .= " LIMIT $limit";
}
else {
    $query .= " LIMIT 10";
}

```

Filtering and sorting logic continued.

#### Checklist Items (1)

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

#### 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:

To get all the results associated to a single user, I fetch all bookIDs from the UserBooks table where the userID is equal to the current user's id (from `get_user_id()`) and `is_active` is 1. I query for all books from the Books table where the book's id is in the bookIDs that are returned from the first statement. The filtering logic then appends the specific sort or filter for each sorting field only if the respective input is filled out. For example, if rating low is set to 4, then "`AND rating > :rating_low`" is appended to the query, and 4 would be passed as a parameter.

After running the query with the sorting and filtering options available, I take the returned results array (stored in `$results` from the `fetchAll`) and pass it to the PHP `count` function, which counts the number of elements. This returns the number of results that are displayed by using the available sorting inputs. I then created another general query that returns the total number of results associated to the user, regardless of the sorting and filtering. If the userIDs match and the bookID `is_active` is set to 1, the book is returned and again, I use the `count` function.

The single delete button redirects to the personal unsaving page with the bookID as the GET URL parameter. Using `get_user_id()` and the passed parameter for book id, the matching result in the UserBooks table is hard deleted. The all

get\_user\_id() and the passed parameter for book id, the matching result in the UserBooks table is hard deleted. The all associations delete button is almost exactly the same, except it doesn't take a book id as a parameter. It deletes all records with the matching userID from the UserBooks table.

#### 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 /#. Same pull request shouldn't be used for each feature

URL #1

[https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/list\\_books.php](https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/list_books.php)

URL #2

<https://github.com/NickZub/njz5-it202-008/pull/64>

URL #3

<https://github.com/NickZub/njz5-it202-008/pull/65>

URL #4

<https://github.com/NickZub/njz5-it202-008/pull/69>

All Users Association Page (likely an admin 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
<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

https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/all\_associations.php

dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

## Associated Books

Username	Book Title	Author Name	Rating Low	Rating High	Sort	Order	Limit
<input type="text" value="Username"/>	<input type="text" value="Book Title"/>	<input type="text" value="Author Name"/>	<input type="text" value="Rating Low"/>	<input type="text" value="Rating High"/>	<input type="button" value="Rating"/>	<input type="button" value="+"/>	<input type="text" value="10"/>
<input type="button" value="Filter"/>	<input type="button" value="Clear"/>						

Number of results: 9/9

Title	Author	Rating	Users	Actions
TitleZeroView	AuthorZero	0.00	<a href="#">email3</a> <a href="#">Delete</a> <a href="#">heroku Delete</a> <a href="#">email2</a> <a href="#">Delete</a>	<input type="button" value="View"/>
TitleTwo	AuthorTwo	0.02	<a href="#">email2</a> <a href="#">Delete</a>	<input type="button" value="View"/>
The Winter Fortress: The Epic Mission to Sabotage Hitlers Atomic Bomb	Neal Bascomb	4.24	<a href="#">email2</a> <a href="#">Delete</a>	<input type="button" value="View"/>
video	video	0.00	<a href="#">video</a> <a href="#">Delete</a>	<input type="button" value="View"/>

Default associated books page. The delete link is to delete the association between the user and the book. The view under actions is for the single view of the book. Making the username clickable was done after recording.

### 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

https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/all\_associations.php?username=&book\_title=title&author\_name=&rating\_low=...  
dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

## Associated Books

Username	Book Title	Author Name	Rating Low	Rating High	Sort	Order	Limit
<input type="text" value="Username"/>	<input type="text" value="title"/>	<input type="text" value="Author Name"/>	<input type="text" value="Rating Low"/>	<input type="text" value="Rating High"/>	<input type="button" value="Rating"/>	<input type="button" value="+"/>	<input type="text" value="10"/>
<input type="button" value="Filter"/>	<input type="button" value="Clear"/>						

Number of results: 2/9

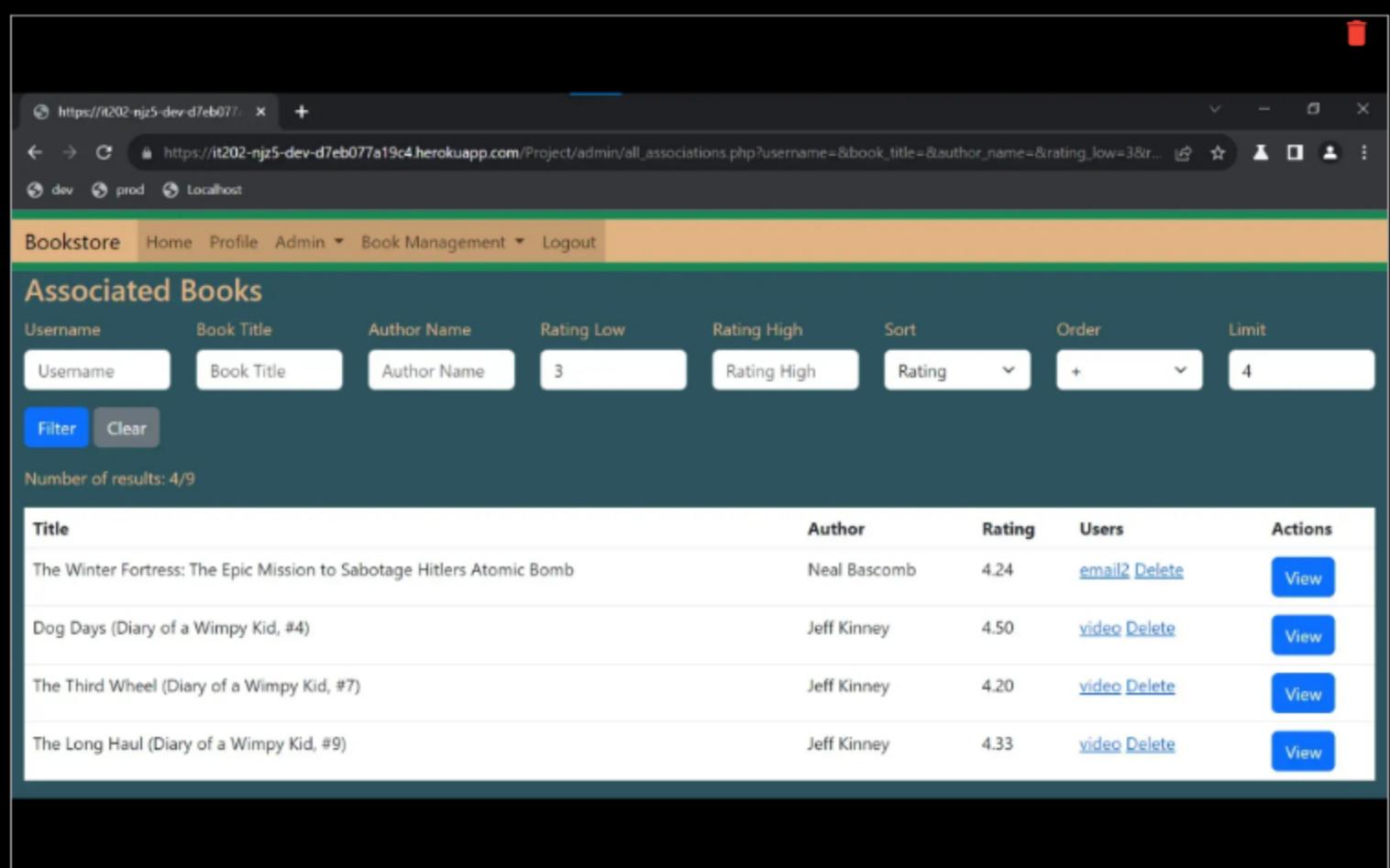
Title	Author	Rating	Users	Actions
TitleZeroView	AuthorZero	0.00	<a href="#">email3</a> <a href="#">Delete</a> <a href="#">heroku</a> <a href="#">Delete</a> <a href="#">email2</a> <a href="#">Delete</a>	<a href="#">View</a>
TitleTwo	AuthorTwo	0.02	<a href="#">email2</a> <a href="#">Delete</a>	<a href="#">View</a>

Sort by book title including title.

### Checklist Items (2)

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

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



The screenshot shows a web browser window with the URL <https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/>. The page title is "Project/admin/all\_associations.php?username=&book\_title=&author\_name=&rating\_low=3&rating\_high=4&sort=Rating&order=+&limit=4". The main content area is titled "Associated Books" and contains a search form with fields for Username, Book Title, Author Name, Rating Low (3), Rating High (Rating), Sort (Rating), Order (+), and Limit (4). Below the form, a message says "Number of results: 4/9". A table lists four books: "The Winter Fortress: The Epic Mission to Sabotage Hitlers Atomic Bomb" by Neal Bascomb with a rating of 4.24; "Dog Days (Diary of a Wimpy Kid, #4)" by Jeff Kinney with a rating of 4.50; "The Third Wheel (Diary of a Wimpy Kid, #7)" by Jeff Kinney with a rating of 4.20; and "The Long Haul (Diary of a Wimpy Kid, #9)" by Jeff Kinney with a rating of 4.33. Each row has "View" and "Delete" links.

Title	Author	Rating	Users	Actions
The Winter Fortress: The Epic Mission to Sabotage Hitlers Atomic Bomb	Neal Bascomb	4.24	<a href="#">email2</a> <a href="#">Delete</a>	<a href="#">View</a>
Dog Days (Diary of a Wimpy Kid, #4)	Jeff Kinney	4.50	<a href="#">video</a> <a href="#">Delete</a>	<a href="#">View</a>
The Third Wheel (Diary of a Wimpy Kid, #7)	Jeff Kinney	4.20	<a href="#">video</a> <a href="#">Delete</a>	<a href="#">View</a>
The Long Haul (Diary of a Wimpy Kid, #9)	Jeff Kinney	4.33	<a href="#">video</a> <a href="#">Delete</a>	<a href="#">View</a>

Sort by rating low of 3 limit 4.

### Checklist Items (2)

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

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

https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/all\_associations.php?username=&book\_title=title&author\_name=&rating\_low=... ↗ ☆

dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

## Associated Books

Username	Book Title	Author Name	Rating Low	Rating High	Sort	Order	Limit
Username	title	Author Name	3	Rating High	Rating	+	4

Filter Clear

Number of results: 0/9

Title	Author	Rating	Users	Actions
No Saved Books				

No sorted results.

### Checklist Items (2)

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

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

https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/view\_associated\_book.php?id=1

dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

## Book: TitleZeroView

Back Edit



TitleZeroView

Author Name: AuthorZero
Rating: 0.00

## Single view.

### Checklist Items (1)

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

The screenshot shows a web browser window with the URL <https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/>. The page title is "Project/admin/profile.php?username=heroku". The main content area displays a user profile for "Username: heroku" who joined on "2024-04-01 23:46:34". There is a "Back" button at the top left and a red trash can icon in the top right corner of the main content area.

[View profile.](#)

### Checklist Items (1)

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

The screenshot shows a web browser window with the URL <https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/>. The page title is "Project/admin/all\_associations.php". The main content area displays a message "Unsaved record with id 1 for email2" in a green bar. Below it, there is a section titled "Associated Books" with various search filters and buttons. At the bottom, it shows "Number of results: 9/9".

Title	Author	Rating	Users	Actions

TitleZeroView	AuthorZero	0.00	<a href="#">email3</a> <a href="#">Delete</a>	<a href="#">View</a>
TitleTwo	AuthorTwo	0.02	<a href="#">email2</a> <a href="#">Delete</a>	<a href="#">View</a>
The Winter Fortress: The Epic Mission to Sabotage Hitlers Atomic Bomb	Neal Bascomb	4.24	<a href="#">email2</a> <a href="#">Delete</a>	<a href="#">View</a>

Delete association.

### Checklist Items (1)

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

#### Task #2 - Points: 1

Text: Screenshot the code

### Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the code related to fetching all associations (including the query)
<input type="checkbox"/> #2	1	Show the code related to the display of the results
<input type="checkbox"/> #3	1	Each record should have a button/link for single view
<input type="checkbox"/> #4	1	Each record should have a username field that is clickable to go to the user's profile page
<input type="checkbox"/> #5	1	Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity
<input type="checkbox"/> #6	1	Show the logic related to the count of all associated items (even the ones not shown in the filtered results)
<input type="checkbox"/> #7	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
<input type="checkbox"/> #8	1	Show the logic related to filter/sort (should include a partial match for username ) (limit should be constrained to 1-100 otherwise default to 10)
<input type="checkbox"/> #9	1	Include ucid/date comments for each code screenshot
<input type="checkbox"/> #10	1	Clearly caption screenshots

### Task Screenshots:

Gallery Style: Large View

Small      Medium      Large



```

23 //NICHOLAS ZUBRZYCKI 4/29/2024
24 $query = "SELECT b.id, b.bookID, b.title, b.author, b.image, b.rating, b.created, GROUP_CONCAT(u.username SEPARATOR ',') AS Users FROM `Books` b
25     JOIN `UserBooks` ub ON b.id = ub.bookID
26     JOIN `Users` u ON ub.userID = u.id WHERE is_active=1"; //ORDER BY created DESC LIMIT 25
27

```

### Query to fetch associations (GROUP BY is added later on)

#### Checklist Items (3)

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

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

```

//NICHOLAS ZUBRZYCKI 4/30/2024   You, 13 seconds ago * Uncommitted changes
$ignored_columns = ["id", "bookID", "image", "created"];


### Associated Books


<!--Form method="GET" style="margin-bottom: 10px;">



| Title                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Author | Rating | Users | Actions |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|--------|-------|---------|
| <!-- \$php render_table(\$stable); --> RENDER THE TABLE BELOW --                     <p>Number of results: <\$php se(\$count); ></p>                     <table class="table">                         <thead>                             <tr>                                 <th>Title</th>                                 <th>Author</th>                                 <th>Rating</th>                                 <th>Users</th>                                 <th>Actions</th>                             </thead>                             <tbody>                                 <tr>                                     <td>If is_array(\$results) AND count(\$results) > 0 </td>                                     <td>                                         <!-- \$php foreach (\$results as \$row); -->                                         <tr>                                             <td>If (\$in_array(\$t, \$ignored_columns)) && </td>                                             <td>If (\$k != "Users") && </td>                                                 <td><\$php se(\$v); ></td>                                                 <!-- If we are iterating username, add the view and delete association options -->                                                 <td>                                                     <!-- \$php \$eachUser = explode(",", \$v); -->                                                     <td>                                                         <!-- \$php foreach (\$eachUser as \$currUser); -->                                                            <div><a href="profile.php?username=<\$php se(\$currUser); >></a></div>                                                            <div><a href="delete_associated_book.php?username=<\$php se(\$currUser); >&id=<\$php se(\$row, "id"); >></a></div>                                                         <td>                                                     </td>                                                 <td>                                                     <!-- \$php endif; -->                                                     <td>                                                         <!-- \$php endforeach; -->                                                         <td>                                                             <!-- \$php href="view_associated_book.php?id=<\$php se(\$row, "id"); >" class="btn-primary" -->                                                         </td>                                                     </td>                                                 </td>                                             </td>                                         </tr>                                     </td>                                 </tr>                             </tbody>                         </table>                     <!-- \$php endforeach(); --> |        |        |       |         |


```

Displaying the results with links for viewing profile, deleting association, and single view.

**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**

```
//NICHOLAS ZUBRZYCKI 4/29/2024
$db = getDB();
$stmt1 = $db->prepare($query);
$stmt2 = $db->prepare("SELECT b.id, b.bookID, b.title, b.author, b.image, b.rating, b.created, GROUP_CONCAT(u.username SEPARATOR ',') AS Users FROM `Books` b
JOIN `UserBooks` ub ON b.id = ub.bookID
JOIN `Users` u ON ub.userID = u.id
WHERE is_active=1
GROUP BY b.id, b.bookID, b.title, b.author, b.image, b.rating, b.created");
$results = [];
$count = 0;
$total = 0;
try {
    //Get query data and filtered count
    $stmt1->execute($params);
    $r1 = $stmt1->fetchAll();
    if ($r1) {
        $results = $r1;
        $count = count($results);
    }
    //Get total count before filtering results
    $stmt2->execute();
    $r2 = $stmt2->fetchAll();
    if ($r2) {
        $total = count($r2);
    }
} catch (PDOException $e) {
    error_log("Error fetching stocks " . var_export($e, true));
    flash("Unhandled error occurred $query", "danger");
}

```

Count number of items before and after filtering. \$count is filtered count, \$total is total count before filter.

**Checklist Items (4)****#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**

```
//NICHOLAS ZUBRZYCKI 4/30/2024      You, 9 minutes ago + Uncommitted changes
$params = [];
$session_key = $_SERVER["SCRIPT_NAME"];
$is_clear = isset($_GET["clear"]);
if ($is_clear) {
    session_delete($session_key);
    unset($_GET["clear"]);
    //exit(header("Location: " . $session_key));
    redirect($session_key);
} else {
```

```

        $session_data = session_load($session_key);
    }

    if (count($_GET) == 0 && !isset($session_data) && count($session_data) > 0) {
        if ($session_data) {
            $_GET = $session_data;
        }
    }

    $username = "";
    if (count($_GET) > 0) {
        session_save($session_key, $_GET);
        $keys = array_keys($_GET);

        foreach ($form as $k => $v) {
            if (!in_array($v["name"], $keys)) {
                $form[$k]["value"] = $_GET[$v["name"]];
            }
        }
    }

    //username
    $username = se($_GET, "username", "", false);
    if (!empty($username)) {
        $query .= " AND u.username like :username";
        $params[":username"] = "%$username%";
    }

    //book title
    $book_title = se($_GET, "book_title", "", false);
    if (!empty($book_title)) {
        $query .= " AND title like :title";
        $params[":title"] = "%$book_title%";
    }

    //author name
    $author_name = se($_GET, "author_name", "", false);
    if (!empty($author_name)) {
        $query .= " AND author like :author";
        $params[":author"] = "%$author_name%";
    }

    //rating low
    $rating_low = se($_GET, "rating_low", "-1", false);
    if (!empty($rating_low) && $rating_low > -1) {
        $query .= " AND rating > :rating_low";
        $params[":rating_low"] = $rating_low;
    }
}

```

## Filtering and sorting.

### 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)

```

}
//NICHOLAS ZUBRZYCKI 4/29/2024| You, 2 minutes ago * Uncommitted changes
//rating high
$rating_high = se($_GET, "rating_high", "-1", false);
if (!empty($rating_high) && $rating_high > -1) {
    $query .= " AND rating <= :rating_high";
    $params[":rating_high"] = $rating_high;
}

//sort and order
$sort = se($_GET, "sort", "b.created", false);
if (!in_array($sort, ["rating", "created"])) {
    $sort = "b.created";
}
$order = se($_GET, "order", "desc", false);
if (!in_array($order, ["asc", "desc"])) {
    $order = "desc";
}

//IMPORTANT make sure you fully validate/trust $sort and $order (sql injection possibility)
$query .= " GROUP BY b.id, b.bookID, b.title, b.author, b.image, b.rating, b.created ORDER BY $sort $order";
//limit
try {
    $limit = (int)se($_GET, "limit", "10", false);
} catch (Exception $e) {
    $limit = 10;
}
if ($limit < 1 || $limit > 100) {
    $limit = 10;
}

//IMPORTANT make sure you fully validate/trust $limit (sql injection possibility)
$query .= " LIMIT $limit";
}

else{
    $query .= " GROUP BY b.id, b.bookID, b.title, b.author, b.image, b.rating, b.created LIMIT 10";
}

```

## Filtering and sorting continued.

### Checklist Items (1)

#8 Show the logic related to filter/sort (should include a partial match for username ) (limit should be constrained to

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

**Response:**

To get the resulting list, I join all three of the main tables that I use into one query (Users, Books, UserBooks). I map the User's table id to UserBooks userID column, and I map the Book's table id to the UserBooks bookID column. Using these other tables, I query for each book's information from the Books table along with the users who saved each respective book. For example, if userID is 1 and bookID is 1, I will have all the information from the users table where the id is 1 (such as username) and I will have all the book information where the id is 1 (such as title and author) in a combined record. The filtering logic is the same as all the other pages, where the specific sort or filter is appended to the query only if the respective input field is filled out on the page.

To get the count of the number of visible results, I just pass the resulting array from the query to the count() function to count the number of records. I do the same thing to get the count of the total items, except I created another query without any sorting or filtering to keep the fixed total. Both queries are the same except for the fact that one has filtering and one does not.

For the single delete feature, I have each Delete button shown on the page redirect to another page called delete\_associated\_book with the respective username of the user and id of the book to be deleted. This page gathers the id and username variables from the GET URL. There is then a delete query which joins Users and UserBooks together to delete the matching UserBooks record. I map username to userID, then delete the UserBooks record with the userID and bookID. In order to get each username, I split the GROUP CONCATENATED usernames using explode('.'). Then, when displaying the results, if I am displaying a username, I append references to the delete or profile view with the user's username and bookID if needed.

To handle the profile view for each user, I created an almost identical profile page to the earlier one, but I query for username rather than user id. Using the username passed through GET, I query from Users where the username column is equal to that username.

**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 <code>vpitli/#</code> . Same pull request shouldn't be used for each feature

URL #1

[https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/admin/all\\_associations.php](https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/admin/all_associations.php)

URL #2

<https://github.com/NickZub/njz5-it202-008/pull/68>

URL #3

<https://github.com/NickZub/njz5-it202-008/pull/69>

URL #4

<https://github.com/NickZub/njz5-it202-008/pull/70>

URL #5

<https://github.com/NickZub/njz5-it202-008/pull/73>

● Unassociated Page (2 pts.)

▲COLLAPSE▲

● Task #1 - Points: 1

▲COLLAPSE▲ Text: Screenshots of this page

### Checklist

\*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Filter Clear

Number of results: 10/27

## Unsaved Books

title	author	image	rating	Actions
The Atomic Weight of Love	Elizabeth J. Church	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1459102377l/25810606._SY500_.jpg	3.00	<button>View</button> <button>Edit</button>
Tokyo Ghost, Vol. 1: Atomic Garden	Rick Remender	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1456502365l/27753938._SY500_.jpg	4.20	<button>View</button> <button>Edit</button>
Countdown 1945: The	Chris	https://images-na.ssl-images-	4.25	<button>View</button>

Default unassociated books page.

## Checklist Items (2)

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

#2 Show the single view buttons/links

A screenshot of a web application interface. At the top, there is a header bar with a trash icon. Below it is a browser-like address bar showing the URL <https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/>. Underneath the address bar are navigation buttons for dev, prod, and Localhost. The main content area has a teal header with the title "Unsaved Books". Below the header is a search form with fields for Book Title, Author Name, Rating Low (set to 4.5), Rating High (set to Rating High), Sort (set to Rating), Order (set to +), and Limit (set to 10). There are also "Filter" and "Clear" buttons. Below the search form is a message "Number of results: 3/27". A table titled "Unsaved Books" lists three items: "The Hunger Games: Official Illustrated Movie Companion" (Rating 4.50), "Catching Fire: The Official Illustrated Movie Companion (2) (The Hunger Games)" (Rating 4.52), and "Book5" (Rating 5.00). Each item has "View" and "Edit" buttons. At the bottom of the page is a message "Filter by rating low of 4.5."

title	author	image	rating	Actions
The Hunger Games: Official Illustrated Movie Companion	Author Name	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1363678804l/11742691._SY500_.jpg	4.50	<button>View</button> <button>Edit</button>
Catching Fire: The Official Illustrated Movie Companion (2) (The Hunger Games)	Author Name	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1378827944l/17623910._SY500_.jpg	4.52	<button>View</button> <button>Edit</button>
Book5	Author5		5.00	<button>View</button> <button>Edit</button>

## 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)

https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/no\_associations.php?book\_title=&author\_name=&rating\_low=&rating\_high=3...

Bookstore Home Profile Admin Book Management Logout

## Unassociated Books

Book Title	Author Name	Rating Low	Rating High	Sort	Order	Limit
Book Title	Author Name	Rating Low	3	Created	-	10

**Filter** **Clear**

Number of results: 5/27

## Unsaved Books

title	author	image	rating	Actions
Huck New New	NEW		1.00	<b>View</b> <b>Edit</b>
Title	Author		0.00	<b>View</b> <b>Edit</b>
Empty	Empty		0.00	<b>View</b>

Filter by rating high of 3, sort by created descending.

### Checklist Items (4)

#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)

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

#6 Clearly caption screenshots

No records to show

No records to show.

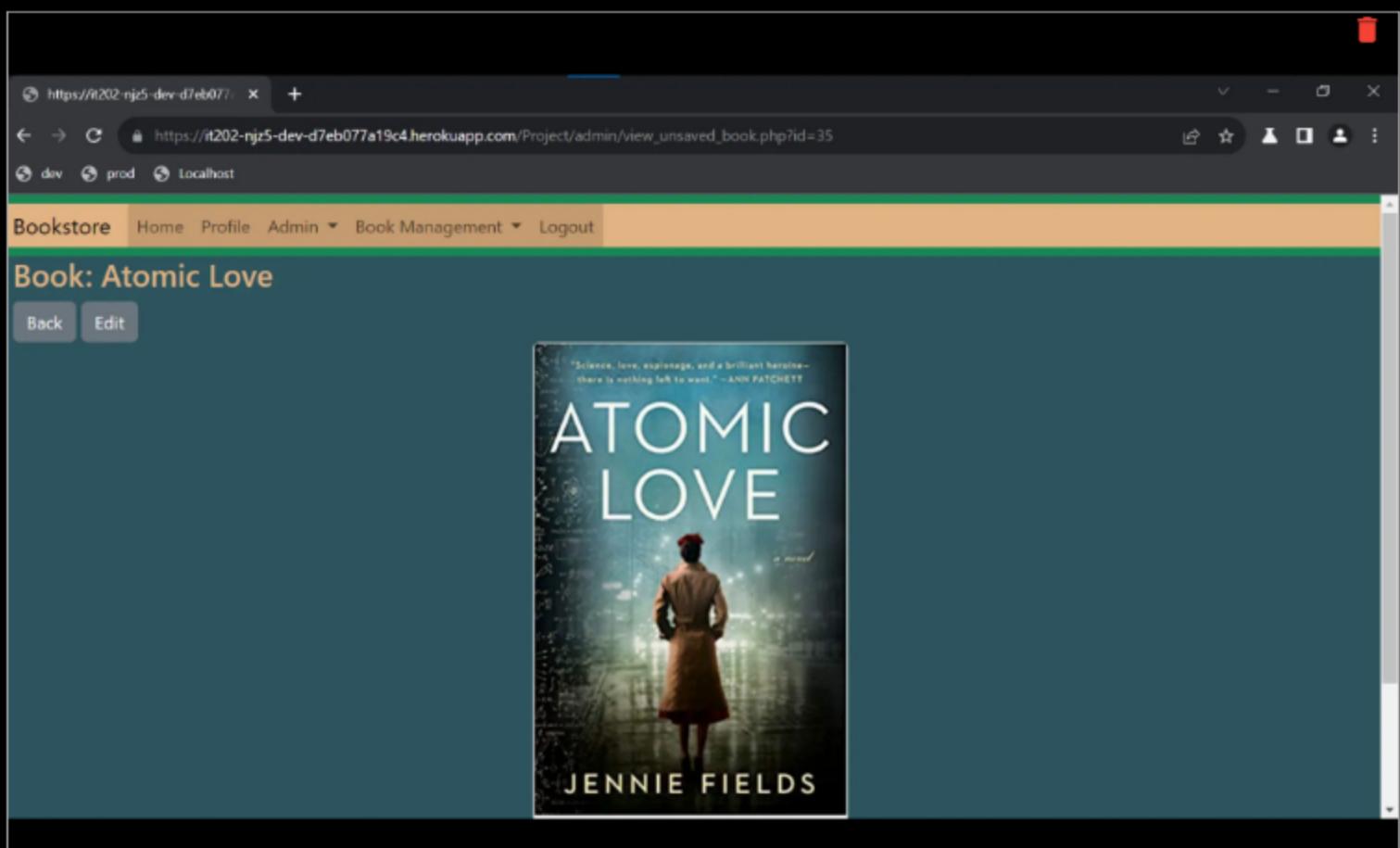
#### Checklist Items (4)

#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)

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

#6 Clearly caption screenshots



Single view page.

#### Checklist Items (1)

#2 Show the single view buttons/links

Task #2 - Points: 1

ACOLLAPSE

## Text: Screenshot the code

## Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the code related to fetching all unassociated entities (including the query)
<input type="checkbox"/> #2	1	Show the code related to the display of the results
<input type="checkbox"/> #3	1	Each record should have a button/link for single view
<input type="checkbox"/> #4	1	Show the logic related to the count of all unassociated items (even the ones not shown in the filtered results)
<input type="checkbox"/> #5	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
<input type="checkbox"/> #6	1	Show the logic related to filter/sort (should include a partial match for username ) (limit should be constrained to 1-100 otherwise default to 10)
<input type="checkbox"/> #7	1	Include ucid/date comments for each code screenshot
<input type="checkbox"/> #8	1	Clearly caption screenshots

## Task Screenshots:

## Gallery Style: Large View

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

//NICHOLAS ZUBRZYCKI 4/29/2024

```
$query = "SELECT id, bookID, title, author, image, rating FROM `Books` WHERE id NOT IN
(SELECT bookID FROM `UserBooks`)//ORDER BY created DESC LIMIT 25
```

Query to fetch all unassociated entities.

## Checklist Items (3)

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

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

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots

```
$table = ["data" => $results, "title" => "Unsaved Books", "ignored_columns" => ["id", "bookID"], "edit_url" => get_url("admin/edit_"
//NICHOLAS ZUBRZYCKI 4/29/2024]
?>
<div class="container-fluid">
    <h3>Unassociated Books</h3>
    <form method="GET">
        <div class="row mb-3" style="align-items: flex-end;">
            <?php foreach ($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>
    <p>Number of results: <?php se($count); ?>/<?php se($total); ?></p>
    <?php render_table($table); ?>
</div>

<?php
//note we need to go up 1 more directory
require_once(__DIR__ . "/../../../../partials/flash.php");
?>
```

Displaying the results.

Checklist Items (1)

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

```
"edit_url" => get_url("admin/edit_book.php"), "view_url" => get_url("admin/view_unsaved_book.php")];
//NICHOLAS ZUBRZYCKI 4/29/2024 You, 1 minute ago • Uncommitted changes
```

Render buttons for edit and single view.

#### Checklist Items (1)

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

```
//NICHOLAS ZUBRZYCKI 4/29/2024| You, 2 minutes ago • Uncommitted changes
$db = getDB();
$stmt1 = $db->prepare($query);
$stmt2 = $db->prepare("SELECT id, bookID, title, author, image, rating FROM `Books` WHERE id NOT IN (SELECT bookID FROM `UserBooks`)");
$results = [];
$count = 0;
$total = 0;
try {
    $stmt1->execute($params);
    $r1 = $stmt1->fetchAll();
    if ($r1) {
        $results = $r1;
        $count = count($results);
    }
    $stmt2->execute();
    $r2 = $stmt2->fetchAll();
    if ($r2) {
        $total = count($r2);
    }
} catch (PDOException $e) {
    error_log("Error fetching stocks " . var_export($e, true));
    flash("Unhandled error occurred", "danger");
}
```

Count the number of total and visible items. \$count is the number of visible items, \$total is the number of total items.

#### Checklist Items (4)

#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

```
//NICHOLAS ZUBRZYCKI 4/29/2024|
$params = [];
$session_key = $_SERVER["SCRIPT_NAME"];
$is_clear = isset($_GET["clear"]);
if ($is_clear) {
    session_delete($session_key);
    unset($_GET["clear"]);
    //exit(header("Location: " . $session_key));
```

```

        redirect($session_key);
    } else {
        $session_data = session_load($session_key);
    }

    if (count($_GET) == 0 && isset($session_data) && count($session_data) > 0) {
        if ($session_data) {
            $_GET = $session_data;
        }
    }

    if (count($_GET) > 0) {
        session_save($session_key, $_GET);
        $keys = array_keys($_GET);

        foreach ($form as $k => $v) {
            if (in_array($v["name"], $keys)) {
                $form[$k]["value"] = $_GET[$v["name"]];
            }
        }
    }

    //book title
    $book_title = se($_GET, "book_title", "", false);
    if (!empty($book_title)) {
        $query .= " AND title like :title";
        $params[":title"] = "%$book_title%";
    }

    //author name
    $author_name = se($_GET, "author_name", "", false);
    if (!empty($author_name)) {
        $query .= " AND author like :author";
        $params[":author"] = "%$author_name%";
    }

    //rating low
    $rating_low = se($_GET, "rating_low", "-1", false);
    if (!empty($rating_low) && $rating_low > -1) {
        $query .= " AND rating >= :rating_low";
        $params[":rating_low"] = $rating_low;
    }

    //rating high
    $rating_high = se($_GET, "rating_high", "-1", false);

```

## Filtering and sorting.

### Checklist Items (1)

#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)

```

}
//NICHOLAS ZUBRZYCKI 4/29/2024      You, 15 seconds ago * Uncommitted changes
//rating high
$rating_high = se($_GET, "rating_high", "-1", false);
if (!empty($rating_high) && $rating_high > -1) {
    $query .= " AND rating <= :rating_high";
    $params[":rating_high"] = $rating_high;
}

//sort and order
$sort = se($_GET, "sort", "created", false);
if (!in_array($sort, ["rating", "created"])) {
    $sort = "created";
}
$order = se($_GET, "order", "desc", false);
if (!in_array($order, ["asc", "desc"])) {
    $order = "desc";
}

//IMPORTANT make sure you fully validate/trust $sort and $order (sql injection possibility)
$query .= " ORDER BY $sort $order";
//limit
try {
    $limit = (int)se($_GET, "limit", "10", false);
} catch (Exception $e) {
    $limit = 10;
}
if ($limit < 1 || $limit > 100) {
    $limit = 10;
}

//IMPORTANT make sure you fully validate/trust $limit (sql injection possibility)
$query .= " LIMIT $limit";
}

else {
    $query .= " LIMIT 10";
}

```

## Filtering and sorting continued.

### Checklist Items (1)

## 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:

I query for all values of bookID in UserBooks, meaning some user has that specific book's id saved. I then query for all books from Books where their id is not in that list, so only unsaved books are returned. The filtering and sorting is the same as all the other pages. I append filtering strings to the query only if the respective filtering option is selected or filled out on the page.

The counts are also the same as the counts on the other pages. One query has the sorting and filtering options applied and the other does not. I take the count of the results returned from these queries using the count() function to get the number of visible and total counts.

## 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 <code>vpitli/#</code> . Same pull request shouldn't be used for each feature

## URL #1

[https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/admin/no\\_associations.php](https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/admin/no_associations.php)

## URL #2

<https://github.com/NickZub/njz5-it202-008/pull/66>

## URL #3

<https://github.com/NickZub/njz5-it202-008/pull/69>

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

## Task #1 - Points: 1

Text: Screenshots of the page

## Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the search form with valid data
<input checked="" type="checkbox"/> #2	1	Show the results of the search
<input 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

The screenshot shows a web browser window with the URL <https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/>. The page displays a search interface with fields for 'email' and 'title', and a 'Search' button. Below the search interface, there are two sections: 'Users' and 'Books to Assign'. The 'Users' section lists five entries: 'email', 'email2', 'email3', 'email4', and 'email5', each preceded by a checkbox. The 'Books to Assign' section lists four entries: 'TitleZeroView', 'AuthorZero', 'TitleOne', 'AuthorOne', 'TitleTwo', 'AuthorTwo', and 'Title', each preceded by a checkbox. A red trash can icon is visible in the top right corner of the browser window.

email

title

Search

Toggle Books

Users

email

email2

email3

email4

email5

Books to Assign

TitleZeroView

AuthorZero

TitleOne

AuthorOne

TitleTwo

AuthorTwo

Title

Author

Search form with results.

Checklist Items (2)

#1 Show the search form with valid data

#2 Show the results of the search

The screenshot shows a web browser window with the URL <https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/>. The page displays the results of a search query, showing a list of books and users. A red trash can icon is visible in the top right corner of the browser window.

email

title

Search

Toggle Books

Users

email

email2

email3

email4

email5

Books to Assign

TitleZeroView

AuthorZero

TitleOne

AuthorOne

TitleTwo

AuthorTwo

Title

Author

Updated book

## Assign Books

Username Search

Book Search

Search

Toggle Books

Users

Books to Assign

Result of association and unassociation.

### Checklist Items (3)

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

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

#5 Clearly caption screenshots

The screenshot shows a MySQL Workbench interface with a query editor at the top containing the SQL command:

```
SELECT * FROM `UserBooks` WHERE id=135 LIMIT 100
```

Below the query editor is a table titled "UserBooks". The table has the following columns and data:

	Q	id	created	modified	userID	bookID	is_active
	>1	135	2024-04-30 03:07:56	2024-04-30 03:07:56	1	1	1

Before toggle.

### Checklist Items (1)

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

The screenshot shows a MySQL query results page. The query is:

```
SELECT * FROM `UserBooks` WHERE id=135 LIMIT 100
```

The results table has the following columns and data:

	Q	id	created	modified	userID	bookID	is_active
	>1	135	2024-04-30 03:07:56	2024-04-30 03:10:16	1	1	0

After toggle.

### Checklist Items (1)

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

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

### Checklist

\*The checkboxes are for your own tracking

#	Points	Details
#1	1	Search form field for finding a partial match of usernames
#2	1	Search form field for finding a partial match of entities
#3	1	Code related to getting a max of 25 results for each field (i.e., 25 users and 25 entities limit)
#4	1	Code that generates the checkboxes next to each list (users and entities)
#5	1	Code related to submitting the checkbox lists
#6	1	Code related to applying the associations upon submission (i.e., add the relationship if it

		doesn't exist and remove the relationship if it does exist)
#7	1	Include ucid/date comments for each code screenshot
#8	1	Clearly caption screenshots

## Task Screenshots:

### Gallery Style: Large View

Small      Medium      Large

```
//NICHOLAS ZUBRZYCKI 4/29/2024    You, 11 hours ago + Uncommitted changes
//search for user by username
$users = [];
$username = "";
if (isset($_POST["username"])) {
    //Username given
    $username = se($_POST, "username", "", false);
    if (!empty($username)) {
        $db = getDB();
        $stmt = $db->prepare("SELECT Users.id, username from Users WHERE username like :username LIMIT 25");
        try {
            $stmt->execute([":username" => "%$username%"]);
            $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
            if ($results) {
                $users = $results;
            }
        } catch (PDOException $e) {
            flash(var_export($e->errorInfo, true), "danger");
        }
    } else {
        //Empty username
        //flash("Username must not be empty", "warning");
        $db = getDB();
        $stmt = $db->prepare("SELECT Users.id, username from Users LIMIT 25");
        try {
            $stmt->execute();
            $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
            if ($results) {
                $users = $results;
            }
        } catch (PDOException $e) {
            flash(var_export($e->errorInfo, true), "danger");
        }
    }
}
```

Code for partial match of username if given, otherwise, selects general username if empty. Limit of 25.

## Checklist Items (4)

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

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

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots

```
//NICHOLAS ZUBRZYCKI 4/29/2024
//search for books
$books = [];
$book = "";
if (isset($_POST["book"])) {
    //Book title given
    $book = se($_POST, "book", "", false);
    if (!empty($book)) {
        $db = getDB();
        $stmt = $db->prepare("SELECT id, title, author FROM Books WHERE title LIKE :title LIMIT 25");
        try {
```

```

        $stmt->execute([":title" => "%$book%"]);
        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
        if ($results) {
            $books = $results;
        }
    } catch (PDOException $e) {
        flash(var_export($e->errorInfo, true), "danger");
    }
} else {
    //Empty book title
    $db = getDB();
    $stmt = $db->prepare("SELECT id, title, author FROM Books LIMIT 25");
    try {
        $stmt->execute();
        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
        if ($results) {
            $books = $results;
        }
    } catch (PDOException $e) {
        flash(var_export($e->errorInfo, true), "danger");
    }
}
}

```

Code for partial match of books if given, otherwise, selects general books. Limit of 25.

#### Checklist Items (4)

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

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

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots

```

//NICHOLAS ZUBRZYCKI 4/29/2024
if (isset($_POST["users"]) && isset($_POST["books"])) {
    $user_ids = $_POST["users"]; //se() doesn't like arrays so we'll just do this
    $book_ids = $_POST["books"]; //se() doesn't like arrays so we'll just do this
    if (empty($user_ids) || empty($book_ids)) {
        flash("Both users and books need to be selected", "warning");
    } else {
        //for sake of simplicity, this will be a tad inefficient
        $db = getDB();
        $stmt = $db->prepare("INSERT INTO UserBooks (userID, bookID) VALUES (:uid, :bid) ON DUPLICATE KEY UPDATE is_active = !is_active");
        foreach ($user_ids as $uid) {
            foreach ($book_ids as $bid) {
                try {
                    $stmt->execute([":uid" => $uid, ":bid" => $bid]);
                    flash("Updated book", "success");
                } catch (PDOException $e) {
                    flash(var_export($e->errorInfo, true), "danger");
                }
            }
        }
    }
}

```

### Checklist Items (3)

#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

```
<div class="container-fluid">
<!-- NICHOLAS ZUBRZYCKI 4/29/2024 | -->
<h1>Assign Books</h1>
<form method="POST">
    <?php render_input(["type" => "search", "name" => "username", "placeholder" => "Username Search", "value" => $username]; /*lazy value to check if form submitted, not ideal*/ ?>
    <?php render_input(["type" => "search", "name" => "book", "placeholder" => "Book Search", "value" => $book]; /*lazy value to check if form submitted, not ideal*/ ?>
    <?php render_button(["text" => "Search", "type" => "submit"]); ?>
</form>
<form method="POST">
    <?php render_button(["text" => "Toggle Books", "type" => "submit", "color" => "secondary"]); ?>
    <br><br>
    <?php if (isset($username) && !empty($username)) : ?>
        <input type="hidden" name="username" value="<?php se($username, false); ?>" />
    <?php endif; ?>
    <?php if (isset($book) && !empty($book)) : ?>
        <input type="hidden" name="book" value="<?php se($book, false); ?>" />
    <?php endif; ?>
    <table class="table">
        <thead>
            <th>Users</th>
            <th>Books to Assign</th>
        </thead>
        <tbody>
            <tr>
                <td>
                    <table class="table">
                        <?php foreach ($users as $user) : ?>
                            <tr>
                                <td>
                                    <label for="user_<?php se($user, 'id'); ?>"><?php se($user, "username"); ?></label>
                                    <input id="user_<?php se($user, 'id'); ?>" type="checkbox" name="users[]" value="<?php se($user, 'id'); ?>" />
                                    <!-- <td><?php se($user, "roles", "No Roles"); ?></td> -->
                                </td>
                            </tr>
                        <?php endforeach; ?>
                    </table>
                <td>
                    <?php foreach ($books as $book) : ?>
                        <div>
                            <label for="role_<?php se($book, 'id'); ?>"><?php se($book, "title"); ?></label>
                            <input id="role_<?php se($book, 'id'); ?>" type="checkbox" name="books[]" value="<?php se($book, 'id'); ?>" />
                            <br>
                            <?php se($book, "author") ?>
                            <br>
                        </div>
                </td>
            </tr>
        </tbody>
    </table>
</div>
```

Generate the text boxes (bottom) and submit them (top).

### Checklist Items (4)

#5 Code related to submitting the checkbox lists

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

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain the solution

COLLAPSE ^

## Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<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:

On the given assign roles page used as a guide, you have to search for a username to assign the role to. My page is the same, except I allow not entering any information. If the username or book title is empty, the usernames and book titles are returned through a general query to the database without any parameters. Otherwise, the entered information is passed as a parameter for the database query.

I copied the UI from the assign roles page and added another field to search for books. I also changed them from requiring values upon entering to not requiring values.

With the returned results, you select a username and book combination to either create or toggle their relationship. You can select many usernames and many books. If the association doesn't exist already, the bookID for the selected book(s) and userID corresponding to the username(s) is inserted into UserBooks with is\_active set to 1. Otherwise, is\_active is toggled between 0 and 1 for users that already have the association of userID and bookID.

### 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 <code>vpitli/#</code> . Same pull request shouldn't be used for each feature

URL #1

[https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/admin/assign\\_books.php](https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/admin/assign_books.php)

URL #2

<https://github.com/NickZub/njz5-it202-008/pull/67>

Misc (1 pt.)

[^COLLAPSE ^](#)

### Task #1 - Points: 1

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a GitHub project board with three columns: Todo, In Progress, and Done. The Todo column has one item: "This item hasn't been started". The In Progress column has one item: "This is actively being worked on". The Done column has three items: "njz5-it202-008 #74 MS3 - API Data Association", "njz5-it202-008 #75 MS3 - Handle the association of data to a user", and "njz5-it202-008 #76 MS3 - Logged in user's associated entities page". There are buttons to add new items to each column.

GitHub project board.

Task #2 - Points: 1

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

URL #1

<https://github.com/users/NickZub/projects/2>

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

I think the most difficult aspect of Milestone 3 was creating the queries for each page because there were so many of them and some were complicated, as well as using the results returned from the query for your features. Although, I guess that was pretty much the whole purpose of milestone 3. The all users association page was the most challenging for me to code. I had to create new copies of existing pages with slight workarounds to work with my logic. For example, I copied profile and made the new page use usernames rather than id.

A big issue I had on the all users association page was the check for an empty username. I thought I had an issue with

my query and kept printing it out to debug it. The query was correct all along. I was checking for "if (\$username = "")" rather than "if (\$username == "")" and it was causing my program to break.

## Task #4 - Points: 1

Text: WakaTime Screenshot

### Details:

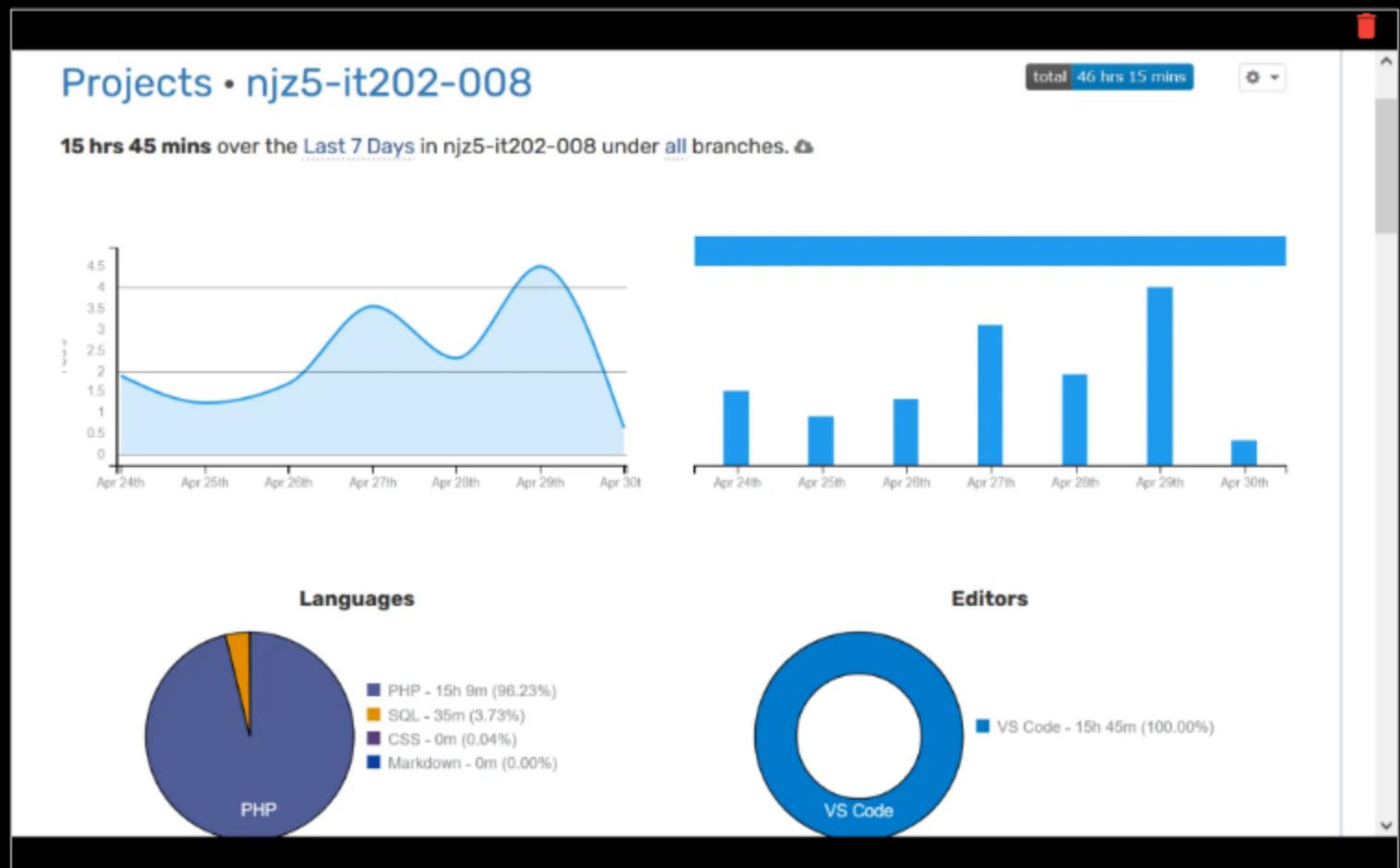
Task Screenshots:

Gallery Style: Large View

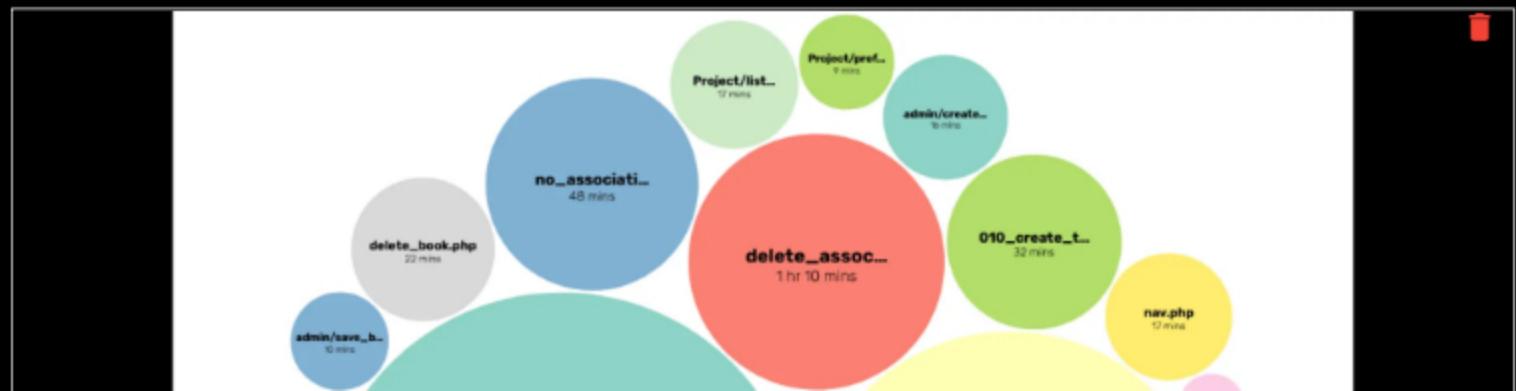
Small

Medium

Large



Waka 1.



## all\_associat...

4 hrs 12 mins

Project/crea...  
40 mins

table.php  
18 mins

admin/list\_b...  
42 mins

## assign\_books...

1 hr 38 mins

admin/profil...  
28 mins

Project/save...  
15 mins

delete\_perso...  
10 mins

## Waka 2.

### Files

4 hrs 12 mins	.../admin/all_associations.php
2 hrs 57 mins	.../Project/personal_books.php
1 hr 38 mins	...ject/admin/assign_books.php
1 hr 10 mins	.../delete_associated_book.php
48 mins	...t/admin/no_associations.php
42 mins	...roject/admin/list_books.php
32 mins	..._create_table_userbooks.sql
23 mins	...l/Project/admin/profile.php
22 mins	...ject/admin/delete_book.php
18 mins	partials/table.php
17 mins	...html/Project/list_books.php
17 mins	partials/nav.php
16 mins	...ject/admin/create_book.php
15 mins	...html/Project/save_book.php
11 mins	...html/Project/view_book.php
10 mins	...ct/delete_personal_book.php
10 mins	...Project/admin/save_book.php
10 mins	...ml/Project/create_book.php
9 mins	...ic_html/Project/profile.php
4 mins	...admin/view_unsaved_book.php
3 mins	...ete_every_personal_book.php
3 mins	...ject/admin/assign_roles.php
2 mins	...roject/admin/list_roles.php
2 mins	..._alter_table_userbooks.sql
2 mins	...ject/view_personal_book.php
1 min	...dmin/edit_personal_book.php

### Branches

4 hrs 56 mins	MS3-Feat-AllUsersAssociation
2 hrs 54 mins	MS3-Feat-GeneralCleanup
2 hrs 10 mins	Milestone3
1 hr 31 mins	MS3-Feat-HandleAssociation
1 hr 23 mins	MS3-Feat-LoggedInEntities
1 hr 16 mins	MS3-Feat-AdminAssociations
1 hr 1 min	MS3-Feat-APIAssociation
17 mins	MS3-Feat-NoDataAssociation
12 mins	MS3-Feat-PreCleanup

## Waka 3.

End of Assignment