

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-2-2024/grade/njz5>

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

Submissions:

Submission Selection

1 Submission [active] 4/15/2024 7:48:54 PM

Instructions

[^ COLLAPSE ^](#)

1. Implement the Milestone 2 features from the project's proposal document: <https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88E>
2. Make sure you add your ucid/date as code comments where code changes are done
3. All code changes should reach the Milestone2 branch
4. Create a pull request from Milestone2 to dev and keep it open until you get the output PDF from this assignment.
5. Gather the evidence of feature completion based on the below tasks.
6. Once finished, get the output PDF and copy/move it to your repository folder on your local machine.
7. Run the necessary git add, commit, and push steps to move it to GitHub
8. Complete the pull request that was opened earlier
9. Create and merge a pull request from dev to prod
10. Upload the same output PDF to Canvas

Branch name: 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
---	--------	---------

#	Points	Details
<input checked="" 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 checked="" type="checkbox"/> #2	1	Columns should be logical and thought out (not valid to have a single field of JSON data or similar)
<input checked="" type="checkbox"/> #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

ted imp	modified timestamp	userID int	bookID varchar(200)	title varchar(200)	author varchar(50)	image varchar(200)	rating decimal(3,2)
1:40:05	2024-04-15 22:50:24	1	(NULL)	TitleZeroView	AuthorZero		0.00
1:42:25	2024-04-13 01:42:25	2	(NULL)	TitleOne	AuthorOne	URLOne	0.00
1:47:39	2024-04-13 01:47:39	1	(NULL)	TitleTwo	AuthorTwo	URLTwo	0.02
0:56:57	2024-04-13 20:56:57	1	15801668-the-girls-of-atom	The Girls of Atomic City: Th	Denise Kiernan	https://images-na.ssl-image	3.71
0:56:57	2024-04-14 01:23:16	1	35069547-the-atomic-city-5	The Atomic City Girls	Janet Beard	https://images-na.ssl-image	3.35
0:56:57	2024-04-13 20:56:57	1	53360084-atomic-love	Atomic Love	Jennie Fields	https://images-na.ssl-image	3.64
0:56:57	2024-04-13 20:56:57	1	58684535-atomic-anna	Atomic Anna	Rachel Barenbaum	https://images-na.ssl-image	3.80

Screenshot of database extension (id and created not shown)

Checklist Items (3)

#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

#2 Columns should be logical and thought out (not valid to have a single field of JSON data or similar)

#3 Clearly caption screenshots

1 CREATE TABLE IF NOT EXISTS `Books` (You, 4 days ago • Created books table.

```

2   `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
3   `created` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
4   `modified` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
5   `userID` INT NOT NULL,
6   `bookID` VARCHAR(200) DEFAULT NULL,
7   `title` VARCHAR(200) NOT NULL,
8   `author` VARCHAR(50) NOT NULL,
9   `image` VARCHAR(200) DEFAULT NULL,
10  `rating` DECIMAL(3,2) DEFAULT NULL,
11  UNIQUE(`userID`, `bookID`)
12 )

```

Screenshot of table query.

Checklist Items (3)

#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

#2 Columns should be logical and thought out (not valid to have a single field of JSON data or similar

#3 Clearly caption screenshots

Task #2 - Points: 1

Text: Explain the design

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Note the different fields and their purpose
<input checked="" type="checkbox"/> #2	1	Note if you needed to normalize the data into separate tables or if one table fit your needs

Response:

BookID, title, author, image, and rating all come straight from the api response. The api returns id for the unique identifier of each book in their database, and I mapped this to BookID because id already existed. Title is the book title, author is the name of the person who wrote the book, image is the url address of the book cover, and rating is a decimal scale from 0-5 on how good the book is.

I also created another column called UserID, so the api fetch or book creation result knows which user to relate the response to. When a user goes to the books listing page, only database entries that have their personal UserID will be returned. I believe this is also a milestone 3 feature, so I may end up removing this feature to create a new table that maps UserIDs to BookIDs, or something similar.

^COLLAPSE ^

Task #3 - Points: 1

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

Details:

Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature.

URL #1

<https://github.com/NickZub/njz5-it202-008/pull/43>

^COLLAPSE ^

Data Creation Page (2 pts.)

^COLLAPSE ^

Task #1 - Points: 1

Text: Screenshots of the creation page

Checklist

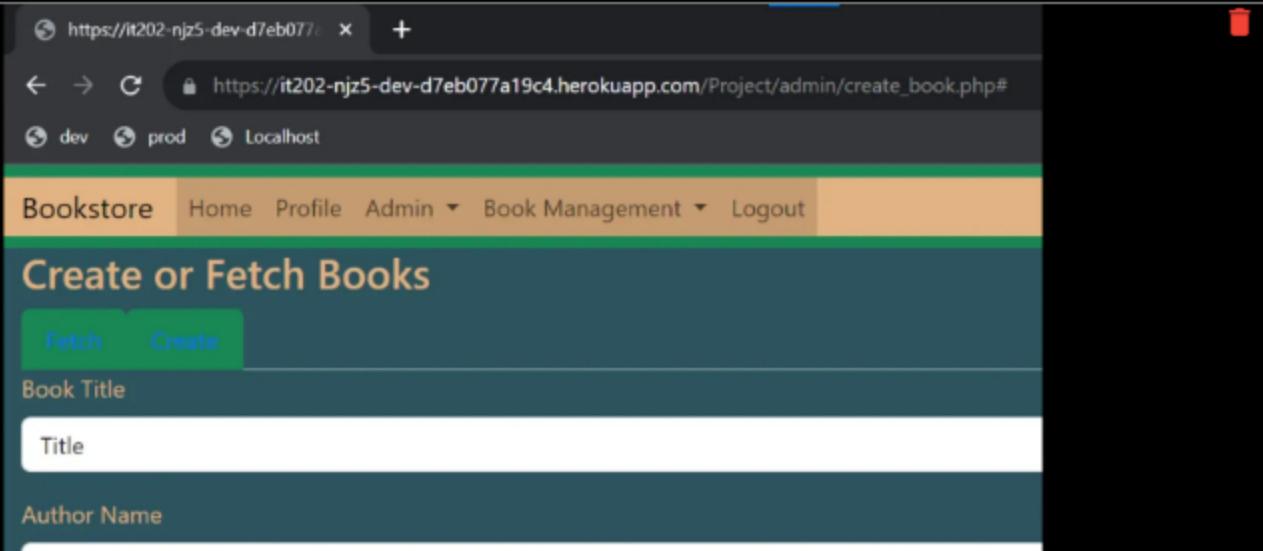
*The checkboxes are for your own tracking

#	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



Author

Image URL

Image URL

Book Rating [0-5]

Book Rating [0-5]

Create

Minimal amount of data needed for book creation.

Checklist Items (4)

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

#5 Design/Style should be considered (i.e., bootstrap, custom css, etc)

#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/create_book.php#

dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

Create or Fetch Books

Fetch Create

Book Title

Title

Author Name

Author

Image URL

url

Please enter a URL.

Book Rating [0-5]

Book Rating [0-5]

Create

URL validation.

Checklist Items (1)

#3 Show examples of validation messages

← → C https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/create_book.php#

dev prod Localhost

Bookstore Home Profile Admin ▾ Book Management ▾ Logout

Create or Fetch Books

Fetch Create

Book Title

Title

Author Name

Author

Image URL

Image URL

Book Rating [0-5]

1000000

Create

! Value must be less than or equal to 5.

Rating validation.

Checklist Items (1)

#3 Show examples of validation messages

https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/create_book.php#

dev prod Localhost

Bookstore Home Profile Admin ▾ Book Management ▾ Logout

Inserted record 128

Create or Fetch Books

Fetch Create

Book Title

Search

Checklist Items (1)

#4 Show an example of successful creation message

The screenshot shows a web browser window with the URL https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/create_book.php. The page title is "Create or Fetch Books". A green button labeled "Fetch" is selected. Below it is a text input field containing "Book Title". A blue button labeled "Search" is visible. At the bottom of the page, there is a message: "Search for book title to fetch from api."

Checklist Items (1)

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

The screenshot shows a web browser window with the URL https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/create_book.php#. The page title is "Create or Fetch Books". A green button labeled "Fetch" is selected. Below it is a text input field containing "Book Title". A blue button labeled "Search" is visible. At the bottom of the page, there is a message: "Please fill out this field".

Author Name

Please fill out this field.

Title validation.

Checklist Items (1)

#3 Show examples of validation messages

https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/create_book.php#

dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

Create or Fetch Books

Fetch Create

Book Title

Title

Author Name

Author Name

Image URL

Please fill out this field.

Image URL

Author validation.

Checklist Items (1)

#3 Show examples of validation messages

Task #2 - Points: 1

Text: Screenshots of creation page code

Checklist

*The checkboxes are for your own tracking

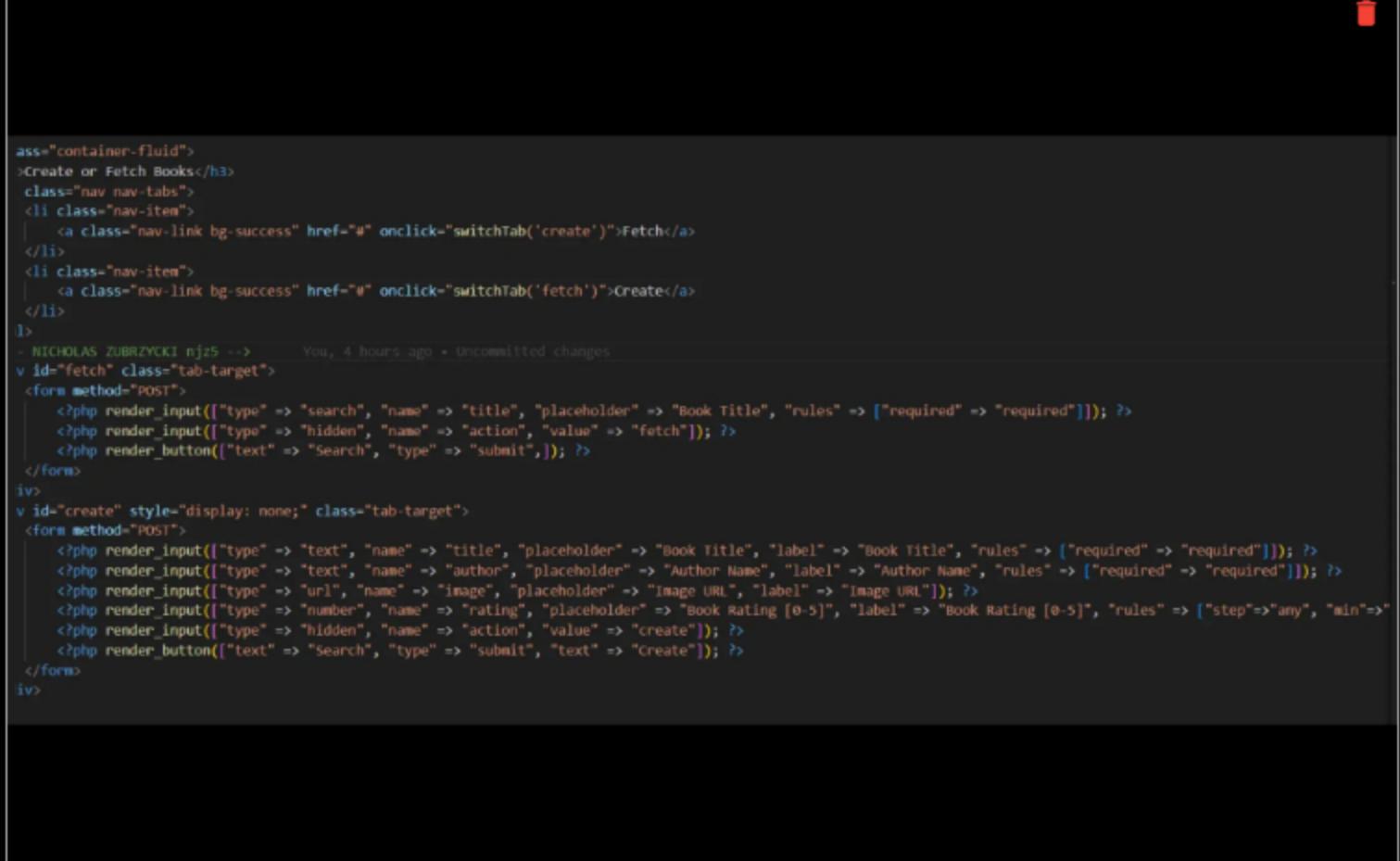
#	Points	Details
<input checked="" type="checkbox"/> #1	1	Form should have correct data types for each property being requested
<input checked="" type="checkbox"/> #2	1	Form should have correct validation for each field (HTML, JS, and PHP)
<input checked="" type="checkbox"/> #3	1	Successful creation should have a user friendly message

<input checked="" type="checkbox"/>	#3	Successful creation should have a user-friendly message
<input checked="" type="checkbox"/>	#4	Any errors should have user-friendly messages
<input checked="" type="checkbox"/>	#5	Include the form/process for fetching API data
<input checked="" type="checkbox"/>	#6	Include some indicator between custom data and API data
<input checked="" type="checkbox"/>	#7	Include any other rules like role guards and login checks
<input checked="" type="checkbox"/>	#8	Include ucid/date comments for each code screenshot
<input checked="" type="checkbox"/>	#9	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



```

<div>
  <h3>Create or Fetch Books</h3>
  <div>
    <ul class="nav nav-tabs">
      <li class="nav-item">
        <a class="nav-link bg-success" href="#" onclick="switchTab('create')>Fetch</a>
      </li>
      <li class="nav-item">
        <a class="nav-link bg-success" href="#" onclick="switchTab('fetch')>Create</a>
      </li>
    </ul>
    - NICHOLAS ZUBRZYCKI njz5 -- You, 4 hours ago + Uncommitted changes
    <div id="fetch" class="tab-target">
      <form method="POST">
        <?php render_input(["type" => "search", "name" => "title", "placeholder" => "Book Title", "rules" => ["required" => "required"]]); ?>
        <?php render_input(["type" => "hidden", "name" => "action", "value" => "fetch"]); ?>
        <?php render_button(["text" => "Search", "type" => "submit"]); ?>
      </form>
    </div>
    <div id="create" style="display: none;" class="tab-target">
      <form method="POST">
        <?php render_input(["type" => "text", "name" => "title", "placeholder" => "Book Title", "label" => "Book Title", "rules" => ["required" => "required"])); ?>
        <?php render_input(["type" => "text", "name" => "author", "placeholder" => "Author Name", "label" => "Author Name", "rules" => ["required" => "required"])); ?>
        <?php render_input(["type" => "url", "name" => "image", "placeholder" => "Image URL", "label" => "Image URL"]); ?>
        <?php render_input(["type" => "number", "name" => "rating", "placeholder" => "Book Rating [0-5]", "label" => "Book Rating [0-5]", "rules" => ["step"=>"any", "min"=>0, "max"=>5]); ?>
        <?php render_input(["type" => "hidden", "name" => "action", "value" => "create"]); ?>
        <?php render_button(["text" => "Search", "type" => "submit", "text" => "Create"]); ?>
      </form>
    </div>
  </div>
</div>

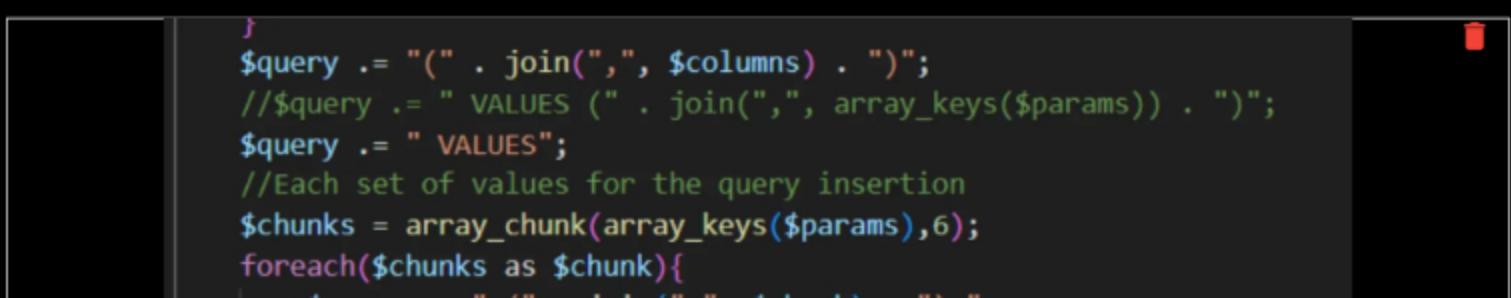
```

Form for requested data.

Checklist Items (2)

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

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



```

}
$query .= "(" . join(", ", $columns) . ")";
// $query .= " VALUES (" . join(", ", array_keys($params)) . ")";
$query .= " VALUES";
// Each set of values for the query insertion
$chunks = array_chunk(array_keys($params), 6);
foreach($chunks as $chunk){
  $query .= " (" . join(", ", $chunk) . ")";
}

```

```

        $query .= "(" . join(", ", $chunk) . ")";
    }
    $query = rtrim($query, ',');
    $query .= " ON DUPLICATE KEY UPDATE `bookID` = VALUES(`bookID`)";
    error_log("Query: " . $query . "<br>");
    error_log("Params: " . var_export($params, true) . "<br>");
    try {
        $stmt = $db->prepare($query);
        $stmt->execute($params);
        flash("Inserted record " . $db->lastInsertId(), "success");
    } catch (PDOException $e) {
        echo("Something broke with the query" . var_export($e, true));
        flash("An error occurred", "danger");
    }
    //NICHOLAS ZUBRZYCKI njz5
}

```

User messages.

Checklist Items (2)

#3 Successful creation should have a user-friendly message

#4 Any errors should have user-friendly messages

```

//insert data
$db = getDB();
$query = "INSERT INTO `Books` ";
$columns = [];
$params = [];
//per record
$index = 0;
foreach($books as $currBook){
    $currBook["userID"] = get_user_id();
    foreach($currBook as $k => $v) {
        if(!in_array("`$k`", $columns, true)){
            array_push($columns, "`$k`");
        }
        //Sanitize rating before query. Set 0 if empty or negative, or round to 2 decimal places.
        //NICHOLAS ZUBRZYCKI njz5
        if($k == "rating"){
            if($v == ''){
                $v = 0.00;
            } else
                $v = sprintf("%.2f", $v);
        }
        $params[":$k$index"] = $v;
    }
    $index += 1;
}
$query .= "(" . join(", ", $columns) . ")";

```

Further php validation for rating.

Checklist Items (1)

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

```
...
1 <?php
2 //note we need to go up 1 more directory
3 require(__DIR__ . "/../../../../partials/nav.php");
4
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 //NICHOLAS ZUBRZYCKI njz5
10 ?>
11
```

Admin privilege check

Checklist Items (1)

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

```
//TODO handle book fetch
//NICHOLAS ZUBRZYCKI njz5
if (isset($_POST["action"])) {
    $action = $_POST["action"];
    $title = se($_POST, "title", "", false);
    $author = se($_POST, "author", "", false);
    $books = [];
    if (($action === "fetch") || ($action === "create") && $title && $author) {
        if ($action === "fetch") {
            $result = fetch_book($title);
            error_log("Data from API" . var_export($result, true));
            if ($result) {
                $books = $result;
            }
        } else if ($action === "create") {
            foreach ($_POST as $k => $v) {
                if (!in_array($k, ["title", "author", "image", "rating", "bookID"])) {
                    unset($_POST[$k]);
                }
            }
            $_POST["bookID"] = null;
            $books = [$POST];
            var_export($books,true);
            error_log("Cleaned up POST: " . var_export($books, true));
        }
    } else {
        flash("You must provide a book title and author name", "warning");
    }
}
```

Fetch API data. BookID is null if user created.

Checklist Items (2)

#5 Include the form/process for fetching API data

#6 Include some indicator between custom data and API data

Task #3 - Points: 1

Text: Screenshot of records from DB

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

#	id int	created timestamp	modified timestamp	userID int	bookID varchar(200)	title varchar(200)	author varchar(50)	image varchar(200)	rating decimal(3,2)
1		2024-04-13 01:40:05	2024-04-15 22:50:24	1	(NULL)	TitleZeroView	AuthorZero		0.00
2		2024-04-13 01:42:25	2024-04-13 01:42:25	2	(NULL)	TitleOne	AuthorOne	URLOne	0.00
3		2024-04-13 01:47:39	2024-04-13 01:47:39	1	(NULL)	TitleTwo	AuthorTwo	URLTwo	0.02
20		2024-04-13 20:56:57	2024-04-13 20:56:57	1	15801668-the-girls-of-atom	The Girls of Atomic City: Thr	Denise Kiernan	https://images-na.ssl-image	3.71
21		2024-04-13 20:56:57	2024-04-14 01:23:16	1	35069547-the-atomic-city-g	The Atomic City Girls	Janet Beard	https://images-na.ssl-image	3.35
22		2024-04-13 20:56:57	2024-04-13 20:56:57	1	53360084-atomic-love	Atomic Love	Jennie Fields	https://images-na.ssl-image	3.64
23		2024-04-13 20:56:57	2024-04-13 20:56:57	1	58684535-atomic-anna	Atomic Anna	Rachel Barenbaum	https://images-na.ssl-image	3.80

Checklist Items (4)

#1 Show at least one record fetched from the API

#2 Show at least one record created via the creation form

#3 Note what differs

#4 Clearly caption screenshots

Task #4 - Points: 1**Text:** Explain the process**Checklist**

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" 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 checked="" 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 checked="" type="checkbox"/> #3	1	Briefly describe the validations for the applicable fields
<input checked="" type="checkbox"/> #4	1	Describe how duplicate data is handled

Response:

The fetching and creating forms both post their respective action, and this determines how the results are set. The only difference is when creating data, bookID is explicitly set to null. The results are then iterated over their keys and values to set up the insertion query into the database.

If the user is fetching from the API, the user only needs to enter the title of the book. The title text input has a required attribute. This title is then sent to the `fetch_book($title)` function, which is the helper function given for fetching API results. The result is an array of arrays, with each inner array holding the book information. Duplicate entries update the bookID for the respective book in the database.

If the user is creating the book, the user has to enter a title and author name. If the other fields are not entered, they can either be empty (author and image), null (bookID), or set to zero (rating). The title and author text inputs have a required attribute, the image is a url input so it only accepts url, and rating only accepts values from 0-5. This data is sent to POST, which is then turned into an array to keep the array of arrays format. Duplicate entries are not handled because bookID is null and null values don't follow constraints.

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

URL #1

https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/admin/create_book.php

URL #2

<https://github.com/NickZub/njz5-it202-008/pull/43>

● 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



Rating Low

Rating High

Rating

+

-

Filter

Clear

Saved Books

bookID	title	author	image	rating	Actions
	TitleZeroView	AuthorZero		0.00	<button>View</button> <button>Edit</button> <button>Delete</button>
	Title	Author		0.00	<button>View</button> <button>Edit</button> <button>Delete</button>
	TitleTwo	AuthorTwo	URLTwo	0.02	<button>View</button> <button>Edit</button> <button>Delete</button>

Default listing with limit changed from 10 to 3.

Checklist Items (1)

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

Rating Low Rating High Sort Order Limit

-1 6 Rating + -1

Value must be greater than or equal to 0.

bookID	title	author	image	rating	Actions
	TitleZeroView	AuthorZero		0.00	<button>View</button> <button>Edit</button> <button>Delete</button>
	Title	Author		0.00	<button>View</button> <button>Edit</button> <button>Delete</button>
	TitleTwo	AuthorTwo	URLTwo	0.02	<button>View</button> <button>Edit</button> <button>Delete</button>

Rating low validation.

Checklist Items (0)

Rating Low Rating High Sort Order Limit

0 6 Rating + -1

bookID	title	author	image	rating	Actions
	TitleZeroView	AuthorZero		0.00	<button>View</button> <button>Edit</button> <button>Delete</button>
	Title	Author		0.00	<button>View</button> <button>Edit</button> <button>Delete</button>
	TitleTwo	AuthorTwo	URLTwo	0.02	<button>View</button> <button>Edit</button> <button>Delete</button>

List Books

Rating Low Rating High Sort Order Limit

Rating Low	<input type="text" value="6"/>	Rating	+	-1
<input type="button" value="Filter"/> <input type="button" value="Clear"/>		! Value must be less than or equal to 5.		

Saved Books

bookID	title	author	image	rating	Actions
	TitleZeroView	AuthorZero		0.00	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Title	Author		0.00	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
	TitleTwo	AuthorTwo	URLTwo	0.02	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Rating high validation.

Checklist Items (0)

Rating Low Rating High Sort Order Limit

Rating Low	<input type="text" value="Rating High"/>	Rating	+	1000
<input type="button" value="Filter"/> <input type="button" value="Clear"/>		! Value must be less than or equal to 100.		

Saved Books

bookID	title	author	image	rating	Actions
	TitleZeroView	AuthorZero		0.00	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Title	Author		0.00	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
	TitleTwo	AuthorTwo	URLTwo	0.02	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Limit validation.

Checklist Items (0)

Rating Low Rating High Sort Order Limit

Rating Low	<input type="text" value="Rating High"/>	Rating	+	5
<input type="button" value="Filter"/> <input type="button" value="Clear"/>		! Value must be less than or equal to 5.		

Saved Books

bookID	title	author	image	rating	Actions
	TitleZeroView	AuthorZero		0.00	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
	Title	Author		0.00	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
	TitleTwo	AuthorTwo	URLTwo	0.02	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

List Books				Rating Low	Rating High	Sort	Order	Limit
	4				Rating High	Rating	+	5
Filter Clear								
<h2>Saved Books</h2>								
bookID	title	author	image		rating	Actions		
7260188-mockingjay	Mockingjay (The Hunger Games, #3)	Suzanne Collins	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1586722918l/7260188._SY500_.jpg		4.09	View	Edit	Delete
3474928-atomic-robo-and-the-fightin-scientists-of-tesladyne	Atomic Robo and the Fightin Scientists of Tesladyne (Atomic Robo, #1)	Brian Clevinger	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1347271973l/3474928._SY500_.jpg		4.14	View	Edit	Delete
27753938-	Tokyo Ghost, Vol.	Rick	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1586722918l/7260188._SY500_.jpg		4.20	View		

Rating low of 4.

Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

List Books				Rating Low	Rating High	Sort	Order	Limit
	Rating Low	3		Created	Rating	+	5	
Filter Clear								
<h2>Saved Books</h2>								
bookID	title	author	image		rating	Actions		
	Title	Author			0.00	View	Edit	Delete
	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	View	Edit	Delete
43431483-atomic-marriage	Atomic Marriage	Curtis Sittenfeld	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1546507643l/43431483._SY500_.jpg		2.85	View	Edit	

Sort by created descending, rating high of 3.

Checklist Items (0)

List Books

Rating Low

Rating High

Sort

Order

Limit

[Filter](#)

[Clear](#)

Saved Books

bookID	title	author	image	rating	Actions
43431483-atomic-marriage	Atomic Marriage	Curtis Sittenfeld	https://images-na.ssl-images-amazon.com/images/S/compressed.photo.goodreads.com/books/1546507643l/43431483._SY500_.jpg	2.85	View Edit Delete
	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	View Edit Delete

Sort by rating low, rating high, sort, order, limit.

Checklist Items (1)

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

No records to show

List Books

Rating Low

Rating High

Sort

Order

Limit

[Filter](#)

[Clear](#)

Saved Books

No records to show

No records to show.

Checklist Items (1)

#4 Demonstrate a filter that doesn't have any records (should show an appropriate message)

Task #2 - Points: 1

Text: Screenshots of the list page code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the filter/sort form generation
<input checked="" type="checkbox"/> #2	1	Show the DB query and how the filter/sort is handled (including the restriction on the limit field)
<input checked="" type="checkbox"/> #3	1	Show how the output is generated and displayed
<input checked="" type="checkbox"/> #4	1	Show any restrictions like role guard or login checks
<input checked="" type="checkbox"/> #5	1	Include ucid/date comments for each code screenshot
<input checked="" type="checkbox"/> #6	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

```
...
1 <?php
2 //note we need to go up 1 more directory
3 require(__DIR__ . "/../../../../partials/nav.php");
4
5 if (!has_role("Admin")) {
6     Flash("You don't have permission to view this page", "warning");
7     die(header("Location: $BASE_PATH" . "/home.php"));
8 }
//NICHOLAS ZUBRZYCKI
9 //build search form
10 $form = [
11     ["type" => "number", "name" => "rating_low", "placeholder" => "Rating Low", "label" => "Rating Low", "rules" => ["step"=>"any", "min"=>"0", "max"=>"5"], "include_margin" => true],
12     ["type" => "number", "name" => "rating_high", "placeholder" => "Rating High", "label" => "Rating High", "rules" => ["step"=>"any", "min"=>"0", "max"=>"5"], "include_margin" => true],
13     ["type" => "select", "name" => "sort", "label" => "Sort", "options" => ["rating" => "Rating", "created" => "Created"], "include_margin" => false],
14     ["type" => "select", "name" => "order", "label" => "Order", "options" => ["asc" => "+", "desc" => "-"], "include_margin" => false],
15     ["type" => "number", "name" => "limit", "label" => "Limit", "value" => "10", "rules" => ["min"=>"1", "max"=>"100"], "include_margin" => false],
16 ];
17 ];
18
19 $userID = get_user_id();
20 $query = "SELECT id, bookID, title, author, image, rating FROM `Books` WHERE userID = $userID"; //ORDER BY created DESC LIMIT 25
21
22 $params = [];
23 $session_key = $_SERVER["SCRIPT_NAME"];
24 $is_clear = isset($_GET["clear"]);
25 if ($is_clear) {
26     session_delete($session_key);
27     unset($_GET["clear"]);
28     die(header("Location: " . $session_key));
29 }
```

Checklist Items (4)

#1 Show the filter/sort form generation

#4 Show any restrictions like role guard or login checks

#5 Include ucid/date comments for each code screenshot

#6 Clearly caption screenshots

```
97 }
98 //NICHOLAS ZUBRZYCKI njz5
99 $table = ["data" -> $results, "title" -> "Saved Books", "ignored_columns" -> ["id"], "edit_url" -> get_url("admin/edit_book.php"), "delete_url" -> get_url("admin/delete_b
100 ?>
101 <div class="container-fluid">
102     <h3>List Books</h3>
103     <form method="GET">
104         <div class="row mb-3" style="align-items: flex-end;">
105             <?php foreach ($form as $k -> $v) : ?>
106             <div class="col">
107                 <?php render_input($v); ?>
108             </div>
109         </?php endforeach; ?>
110
111         </div>
112         <?php render_button(["text" -> "Search", "type" -> "submit", "text" -> "Filter"]); ?>
113         <a href="?clear" class="btn btn-secondary">Clear</a>
114     </form>
115     <?php render_table($table); ?>
116 </div>
117
118 </?php
119 //note we need to go up 1 more directory
120 require_once(__DIR__ . "/../../../../partials/flash.php");
121 ?>
```

Form generation continued and output rendering.

Checklist Items (4)

#1 Show the filter/sort form generation

#3 Show how the output is generated and displayed

#5 Include ucid/date comments for each code screenshot

#6 Clearly caption screenshots

```
18 //NICHOLAS ZUBRZYCKI
19 $userID = get_user_id();
20 $query = "SELECT id, bookID, title, author, image, rating FROM `Books` WHERE userID = $userID";
21
22 $params = [];
23 $session_key = $_SERVER["SCRIPT_NAME"];
24 $is_clear = isset($_GET["clear"]);
25 $is_update = isset($_GET["update"]);
```

```

25 if ($is_clear) {
26     session_delete($session_key);
27     unset($_GET["clear"]);
28     die(header("Location: " . $session_key));
29 } else {
30     $session_data = session_load($session_key);
31 }
32
33 if (count($_GET) == 0 && isset($session_data) && count($session_data) > 0) {
34     if ($session_data) {
35         $_GET = $session_data;
36     }
37 }
38
39 if (count($_GET) > 0) {
40     session_save($session_key, $_GET);
41     $keys = array_keys($_GET);
42
43     foreach ($form as $k => $v) {
44         if (in_array($v["name"], $keys)) {
45             $form[$k]["value"] = $_GET[$v["name"]];
46         }
47     }
48 }

```

Query setup.

Checklist Items (1)

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

```

48 //NICHOLAS ZUBRZYCKI
49 //rating low
50 $rating_low = se($_GET, "rating_low", "-1", false);
51 if (!empty($rating_low) && $rating_low > -1) {
52     $query .= " AND rating >= :rating_low";
53     $params[":rating_low"] = $rating_low;
54 }
55 //rating high
56 $rating_high = se($_GET, "rating_high", "-1", false);
57 if (!empty($rating_high) && $rating_high > -1) {
58     $query .= " AND rating <= :rating_high";
59     $params[":rating_high"] = $rating_high;
60 }
61 //sort and order
62 $sort = se($_GET, "sort", "created", false);
63 if (!in_array($sort, ["rating", "created"])) {
64     $sort = "created";
65 }
66 $order = se($_GET, "order", "desc", false);
67 if (!in_array($order, ["asc", "desc"])) {
68     $order = "desc";
69 }
70 //IMPORTANT make sure you fully validate/trust $sort and $order (sql injection possibility)
71 $query .= " ORDER BY $sort $order";
72 //limit
73 try {
74     $limit = (int)se($_GET, "limit", "10", false);
75 } catch (Exception $e) {
76     $limit = 10;
77 }
78 if ($limit < 1 || $limit > 100) {
79     $limit = 10;
80 }
81 //IMPORTANT make sure you fully validate/trust $limit (sql injection possibility)
82 $query .= " LIMIT $limit";
83 ]

```

Query continued.

Checklist Items (1)

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

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
<input type="checkbox"/> #4	1	Mention which users can interact with the view, edit, and delete links

Response:

Filtering and sorting is just a general select statement that appends filtering and sorting options based on conditional if statements. For example, if rating low is not empty and is valid, then "AND rating >= (input)" is appended to the query.

The screen is showing the results of the database query organized into a table format using the table rendering helper function. The result of the query is passed in as the data for the table, with other options for things like titles and buttons. Each table entry shown matches exactly what's in the database based on the filtering and sorting parameters.

I thought all the information was relevant for the user to see, namely title, author, image, and rating. BookID also shows what is and is not API data so I thought that was relevant to show as well. The options for sort can either be the rating of the books, or the order the books were entered into the database.

As of right now, only Admins can view, edit, and delete. Based on how I continue with milestone 3, I might make view available to anyone, and only admins can edit or delete.

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

URL #1

<https://github.com/NickZub/njz5-it202-008/pull/44>

URL #2

<https://github.com/NickZub/njz5-it202-008/pull/48>

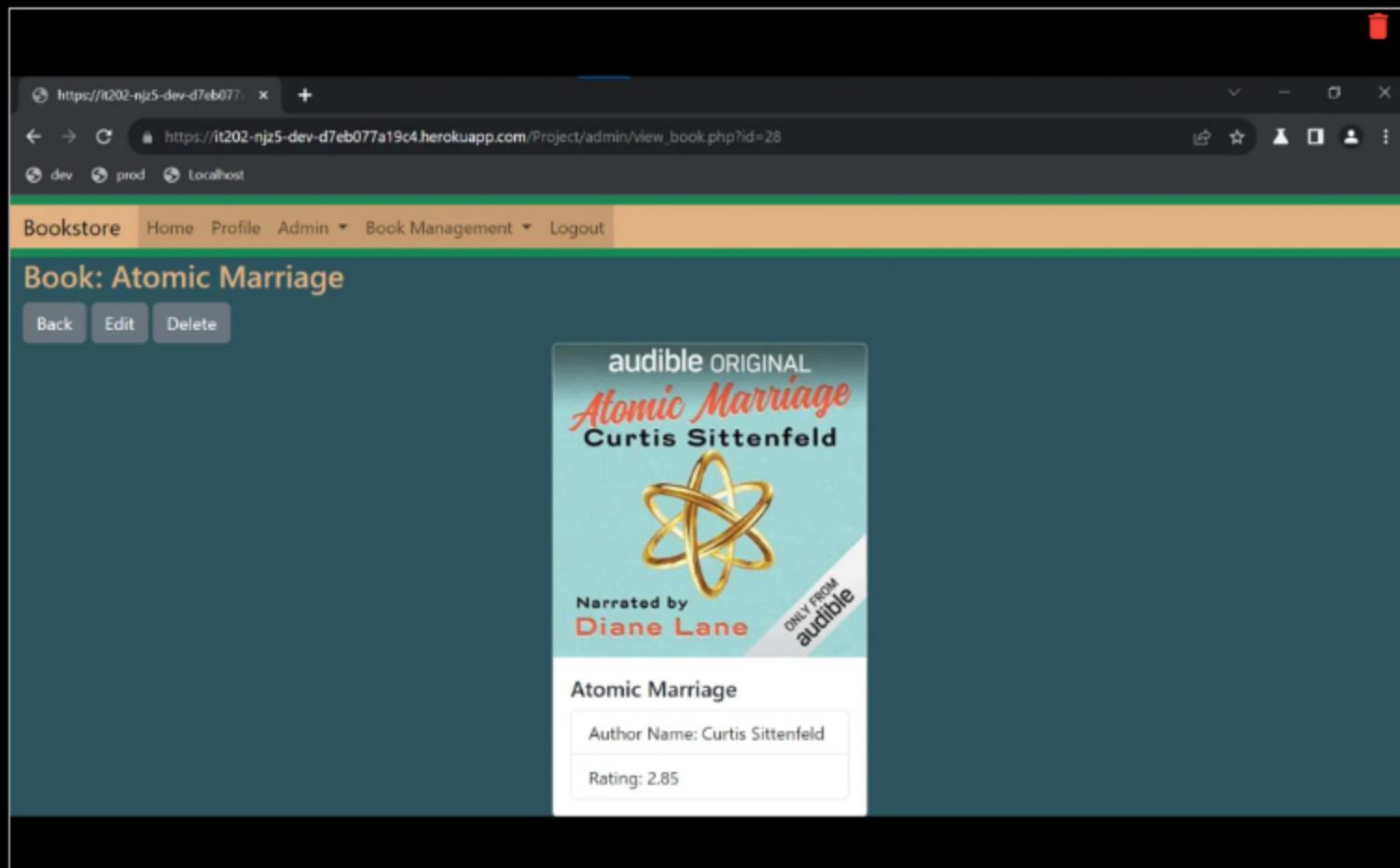
URL #3

https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/admin/list_books.php

ACOLLAPSE ^

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 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**

Checklist Items (7)

#1 Entity should be fetch by id (via the url)

#3 Design/Style should be considered (i.e., bootstrap, custom css, etc)

#4 Data shown should be more detailed/inclusive than the summary view

#5 There should be a link to edit the entity (this may be an admin-only thing, but it should be present for the respective role)

#6 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)

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

#8 Clearly caption screenshots

bookID	title	author	image	rating	Actions
TitleZeroView	AuthorZero			0.00	<button>View</button> <button>Edit</button> <button>Delete</button>
TitleTwo	AuthorTwo	URLTwo		0.02	<button>View</button>

Checklist Items (1)

#2 A missing id should redirect back to the list page with an applicable message

Task #2 - Points: 1

Task #2 - Points: 1

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

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

public_html > Project > admin > view_book.php > ...

...

```

1  <?php
2  //note we need to go up 1 more directory
3  require(__DIR__ . "/../../../../partials/nav.php");
4
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
11 <?php
12 $id = se($_GET, "id", -1, false);
13 //NICHOLAS ZUBRZYCKI njz5

```

Role guard and id fetching.

Checklist Items (3)

#1 Show how id is fetched

#4 Include ucid/date comments for each code screenshot

#5 Clearly caption screenshots



```

//NICHOLAS ZUBRZYCKI njz5
$book = [];
if ($id > -1) {
    //fetch
    $db = getDB();
    $query = "SELECT bookID, title, author, image, rating, created, modified FROM `Books` WHERE id = :id";
    try {
        $stmt = $db->prepare($query);
        $stmt->execute([":id" => $id]);
        $r = $stmt->fetch();
        if ($r) {
            $book = $r;
        }
    } catch (PDOException $e) { You, yesterday + Single view page for books listing. Default ima...
        error_log("Error fetching record: " . var_export($e, true));
        flash("Error fetching record", "danger");
    }
} else {
    flash("Invalid id passed", "danger");
    exit(header("Location:" . get_url("admin/list_books.php")));
}

```

Database query to get record.

Checklist Items (3)

#2 Show the DB query to get the record

#4 Include ucid/date comments for each code screenshot

#5 Clearly caption screenshots

```

public_html > Project > admin > view_book.php > ...
34     foreach ($book as $key => &$value) {
35         if (is_null($value)) {
36             $book[$key] = "N/A";
37         }
38
39         if($key == "image"){
40             $url_validation_regex = "/^https?:\/\/((?:www\.)?[-a-zA-Z0-9@:._\+\-=]{1,256}\.\[a-zA-Z0-9()\]{1,6})\b(?:[-a-zA-Z0-9()@:_\+\.-#?&\=\"]*)$/";
41             if(!preg_match($url_validation_regex,$value)){
42                 $value = "https://clipartix.com/wp-content/uploads/2016/08/Asking-probing-questions-clipart.jpg";
43             }
44         }
45     }
46     unset($value);
47 //NICHOLAS ZUBRZYCKI njz5      You, 34 seconds ago + Uncommitted changes
48 //TODO handle manual create stock
49 ?>
50 <div class="container-fluid">
51     <h3>Book: <?php se($book, "title", "Unknown"); ?></h3>
52     <div>
53         <a href=<?php echo get_url("admin/list_books.php"); ?>" class="btn btn-secondary">Back</a>
54         <?php if (has_role("Admin")) : ?>
55             <a href=<?php echo get_url("admin/edit_book.php"); ?>?>id=<?php se($id,""); ?>" class="btn btn-secondary">Edit</a>
56             <a href=<?php echo get_url("admin/delete_book.php"); ?>?>id=<?php se($id,""); ?>" class="btn btn-secondary">Delete</a>
57         </?php endif; ?>
58     </div>
59     <!-- https://i.kym-cdn.com/entries/icons/original/000/020/959/Screen_Shot_2019-06-05_at_1.26.32_PM.jpg -->
60     <div class="card mx-auto" style="width: 18rem;">
61         <img src=<?php se($book, "image"); ?> class="card-img-top" alt="...">
62         <div class="card-body">
63             <h5 class="card-title"><?php se($book, "title", "Unknown"); ?></h5>
64             <div class="card-text">
65                 <ul class="list-group">

```

```

66
67
68
69
70
    <li class="list-group-item">Author Name: <?php se($book, "author", "Unknown"); ?></li>
    <li class="list-group-item">Rating: <?php se($book, "rating", "Unknown"); ?></li>
</div>

```

Presenting the data with action buttons.

Checklist Items (3)

#3 Show the code related to presenting the data and showing the links

#4 Include ucid/date comments for each code screenshot

#5 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain how the page works

Checklist

*The checkboxes are for your own tracking

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

Response:

Each record in the database has a unique identifier stored in the id column. When clicking on the single item view, all the relevant information is selected for the record where the id matches the selected id, and because each record has a unique id, one result is returned at most. This id is gathered from the \$_GET data in the beginning of the code. The result is then stored into a variable for later use.

When presenting the data, the buttons to go back, edit, and delete are first rendered onto the page. The same id is passed to each page (edit and delete) so the same book is being modified. Then, when rendering the image, author name, and rating in the card, the result variable from earlier is accessed through the safer echo functions. I used the same bootstrap card layout the professor used to style the information.

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 this page
<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/admin/view_book.php?id=1

URL #2

<https://github.com/NickZub/njz5-it202-008/pull/50>

URL #3

<https://github.com/NickZub/njz5-it202-008/pull/56>

● Edit Data Page (2 pts.)

[^COLLAPSE ^](#)

● Task #1 - Points: 1

Text: Screenshots of the edit page

[^COLLAPSE ^](#)

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

The screenshot shows a web browser window with the following details:

- Address Bar:** https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/list_books.php
- Toolbar:** Back, Forward, Stop, Reload, Home, Favorites, Dev Tools, Profile, Localhost.
- Header:** Bookstore, Home, Profile, Admin, Book Management, Logout.
- Section:** List Books
- Filters:** Rating Low, Rating High, Sort (Created), Order (Descending), Limit (10).
- Buttons:** Filter, Clear.
- Section:** Saved Books
- Table:** Displays a list of books with columns: bookID, title, author, image, rating, and Actions (View, Edit, Delete).

Data before edit.

Checklist Items (4)

#1 Show before and after screenshots of data you'll edit

#4 Design/Style should be considered (i.e., bootstrap, custom css, etc)

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

#6 Clearly caption screenshots

A screenshot of a web browser window. The address bar shows the URL <https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/>. The page title is "Edit Books". The main content area has a green header bar with the text "Updated record". Below it, there are input fields for "Book Title" (value: EditNew), "Author Name" (value: EditNew), "Image URL" (value: Image URL), and "Book Rating [0-5]" (value: 5.00). A blue "Update" button is visible. At the bottom of the page, there is a green footer bar with the text "Successful edit message."

Checklist Items (4)

#3 Show an example of successful edit messages

#4 Design/Style should be considered (i.e., bootstrap, custom css, etc)

#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/admin/list_books.php

dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

List Books

Rating Low Rating High Sort Order Limit

Rating Low Rating High Created - 10

Filter Clear

Saved Books

bookID	title	author	image	rating	Actions
	EditNew	EditNew		5.00	<button>View</button> <button>Edit</button> <button>Delete</button>

Data after edit.

Checklist Items (1)

#1 Show before and after screenshots of data you'll edit

https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/edit_book.php?id=129

dev prod Localhost

Bookstore Home Profile Admin Book Management Logout

Edit Books

Book Title
EditNew

Author Name
EditNew

Image URL
url

Book Rating [0-5]
5.00

Please enter a URL

Update

URL validation.

Checklist Items (0)

Bookstore Home Profile Admin Book Management Logout

Edit Books

Book Title

Author Name

Image URL

Book Rating [0-5]

10000

Update

Value must be less than or equal to 5.

Rating validation.

Checklist Items (4)

#2 Show examples of validation messages

#4 Design/Style should be considered (i.e., bootstrap, custom css, etc)

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

#6 Clearly caption screenshots

Bookstore Home Profile Admin Book Management Logout

Edit Books

Book Title

Author Name

Image URL

EditNew

Please fill out this field.

Book Rating [0-5]

1

Update

Title required.

Checklist Items (4)

#2 Show examples of validation messages

#4 Design/Style should be considered (i.e., bootstrap, custom css, etc)

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

#6 Clearly caption screenshots

The screenshot shows a web browser window with the URL https://it202-njz5-dev-d7eb077a19c4.herokuapp.com/Project/admin/edit_book.php?id=129. The page title is "Edit Books". The form fields include:

- Book Title: Input field containing "EditNew".
- Author Name: Input field containing "Author Name".
- Image URL: Input field containing "Image URL". A validation message "Please fill out this field." is displayed above this field.
- Book Rating [0-5]: Input field containing "1".

The browser's address bar shows the full URL. The status bar at the bottom of the browser window indicates "dev" and "prod" status.

Author required.

Checklist Items (4)

#2 Show examples of validation messages

#4 Design/Style should be considered (i.e., bootstrap, custom css, etc)

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

#6 Clearly caption screenshots

Task #2 - Points: 1

Text: Screenshots of edit page code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

```
78 //NICHOLAS ZUBRZYCKI njz5      You, 6 hours ago • Uncommitted changes
79 if ($book) {
80     if($book["bookID"] == ''){
81         $book["bookID"] = null;
82     }
83     $form = [
84         ["type" => "text", "name" => "title", "placeholder" => "Book Title", "label" => "Book Title", "rules" => ["required" => "required"]],
85         ["type" => "text", "name" => "author", "placeholder" => "Author Name", "label" => "Author Name", "rules" => ["required" => "required"]],
86         ["type" => "url", "name" => "image", "placeholder" => "Image URL", "label" => "Image URL"],
87         ["type" => "number", "name" => "rating", "placeholder" => "Book Rating [0-5]", "label" => "Book Rating [0-5]", "rules" => ["step"=>"any", "min"=>"0",
88             ["type" => "hidden", "name" => "bookID", "value" => $book["bookID"]],
89     ];
90     $keys = array_keys($book);
91
92     foreach ($form as $k => $v) {
93         if (in_array($v["name"], $keys)) {
94             $form[$k]["value"] = $book[$v["name"]];
95         }
96     }
97 }
98 //TODO handle manual create stock
99 ?>
100 <div class="container-fluid">
101     <h3>Edit Books</h3>
102     <form method="POST">
103         <?php foreach ($form as $k => $v) {
104             render_input($v);
105         } ?>
106         <?php render_button(["text" => "Search", "type" => "submit", "text" => "Update"]); ?>
107     </form>
108 </div>
```

Checklist Items (2)**#1** Form should have correct data types for each property being requested**#2** Form should have correct validation for each field (HTML, JS, and PHP)

```

43     $query .= " WHERE id = :id AND userID = $userID";
44     $params[":id"] = $id;
45     error_log("Query: " . $query);
46     error_log("Params: " . var_export($params, true));
47     try {
48         $stmt = $db->prepare($query);
49         $stmt->execute($params);
50         flash("Updated record", "success");
51     } catch (PDOException $e) {
52         error_log("Something broke with the query" . var_export($e, true));
53         flash("An error occurred", "danger");
54     }
55 }
56 //NICHOLAS ZUBRZYCKI njz5      You, 2 minutes ago * Uncommitted changes
57 $book = [];
58 if ($id > -1) {
59     //fetch
60     $db = getDB();
61
62     $query = "SELECT bookID, title, author, image, rating FROM `Books` WHERE id = :id AND userID = $userID";
63     try {
64         $stmt = $db->prepare($query);
65         $stmt->execute([":id" => $id]);
66         $r = $stmt->fetch();
67         if ($r) {
68             $book = $r;
69         }
70     } catch (PDOException $e) {
71         error_log("Error fetching record: " . var_export($e, true));
72         flash("Error fetching record", "danger");
73     }
74 } else {
75     flash("Invalid id passed", "danger");

```

User messages for success and errors.**Checklist Items (4)****#3** Successful edit should have a user-friendly message**#4** Any errors should have user-friendly messages**#6** Include ucid/date comments for each code screenshot**#7** Clearly caption screenshots

```

...
1  <?php
2  //note we need to go up 1 more directory
3  require(__DIR__ . "/../../../../partials/nav.php");
4

```

```

5 if (!has_role("Admin")) {
6     flash("You don't have permission to view this page", "warning");
7     die(header("Location: $BASE_PATH" . "/home.php"));
8 }
9 //NICHOLAS ZUBRZYCKI njz5
10 ?>
11
12 <?php
13 $id = se($_GET, "id", -1, false);
14

```

Role guard admin check.

Checklist Items (3)

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

#6 Include ucid/date comments for each code screenshot

#7 Clearly caption screenshots

Task #3 - Points: 1

Text: Screenshot of records from DB

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large



	Q	id int	created timestamp	modified timestamp	userID int	bookID varchar(200)	title varchar(200)	author varchar(50)	image varchar(200)	rating decimal(3,2)
	> 1	129	2024-04-16 01:58:02	2024-04-16 01:59:46	1	(NULL)	EditNew	EditNew		5.00

Data before edit.

Checklist Items (1)

- #1 Show a before and after screenshot of the record

	Q	id int	created timestamp	modified timestamp	userID int	bookID varchar(200)	title varchar(200)	author varchar(50)	image varchar(200)	rating decimal(3,2)
	> 1	129	2024-04-16 01:58:02	2024-04-16 02:14:38	1	(NULL)	EditNewNew	EditNewNew	https://new.new	0.00

Data after edit. Changed title, author, image, and rating. The modified field also changed.

Checklist Items (1)

- #1 Show a before and after screenshot of the record

Task #4 - Points: 1

COLLAPSE

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" 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 checked="" type="checkbox"/> #2	1	Briefly describe the validations for the applicable fields

Response:

Similar to the single page view, the id is obtained via URL parameters. First, a SELECT query is done against the database to retrieve the latest record for that specific id, and the result is stored in a variable. Then, a form is created for each field that can be updated. After the form is created, by using the variable that stored the previous SELECT record, the current values for each field are preloaded by iterating the form. Lastly, each input field is rendered in a POST form.

When the update button is pressed, that submits the post information to the page. If the proper fields are set, an UPDATE statement is ran using the new POST values as the values to be updated, and it makes sure to only update the record for that one id.

Task #5 - 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 this page
<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://github.com/NickZub/njz5-it202-008/pull/45>

URL #2

https://it202-njz5-prod-bedd6fae57b2.herokuapp.com/Project/admin/edit_book.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 checked="" type="checkbox"/> #1	1	Show the success message of a delete

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

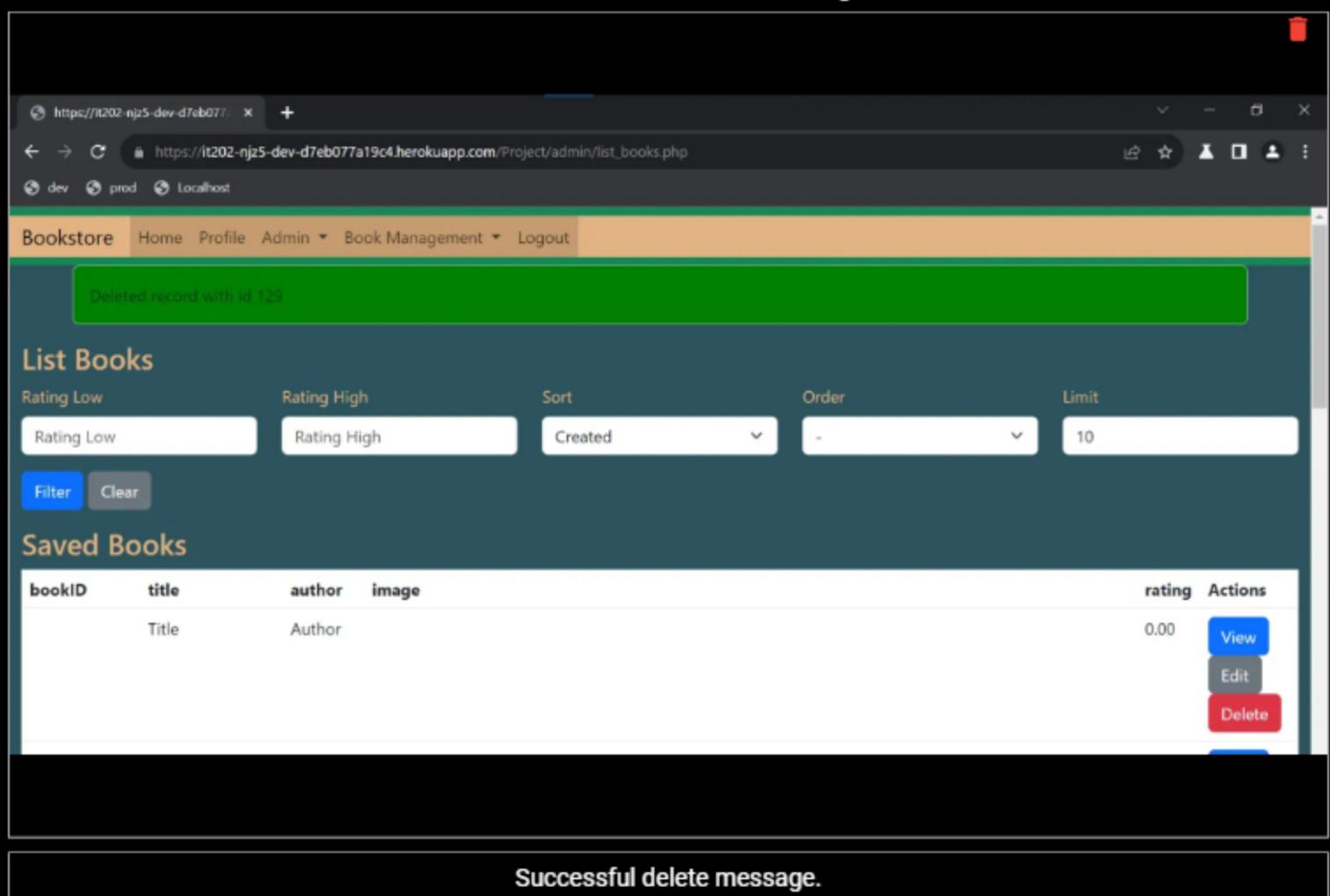
Task Screenshots:

Gallery Style: Large View

Small

Medium

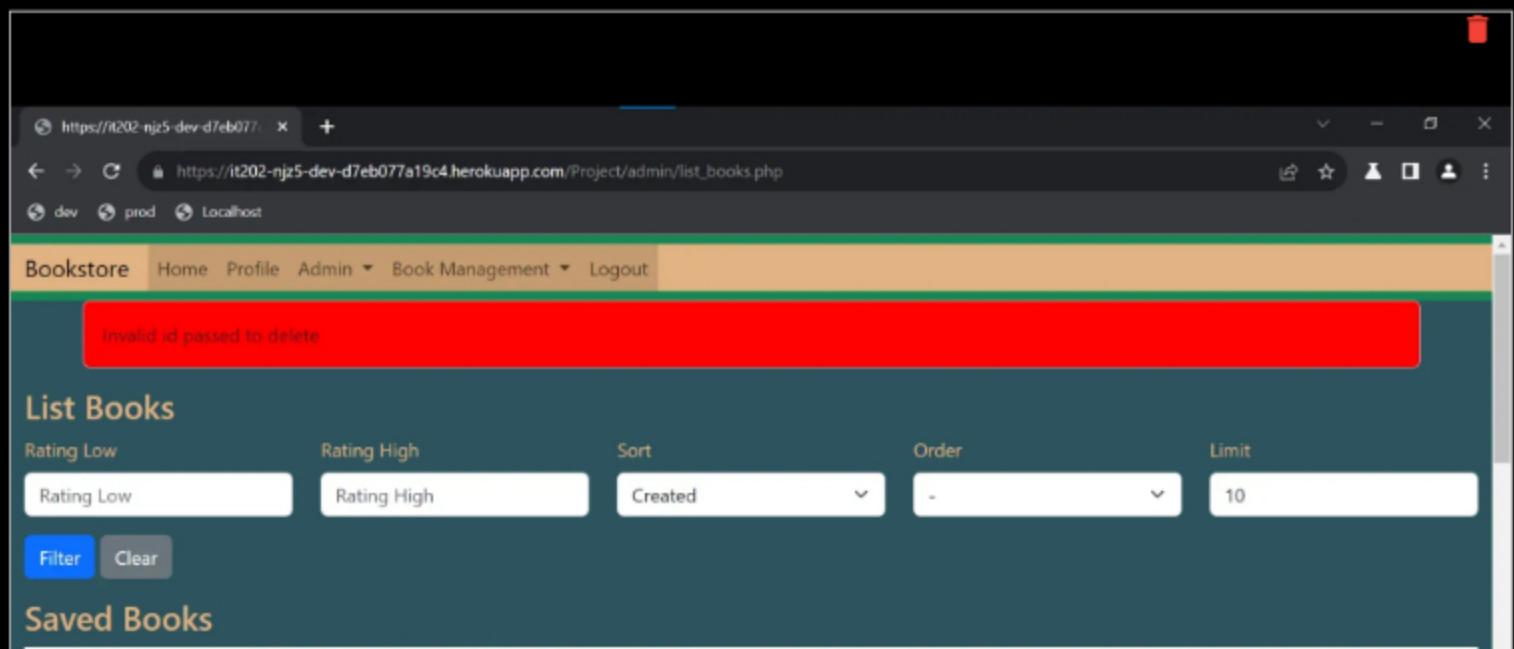
Large



The screenshot shows a web browser window with the URL <https://it202-njc5-dev-d7eb077a19c4.herokuapp.com/>. The page title is "Project/admin/list_books.php". The navigation bar includes "Bookstore", "Home", "Profile", "Admin", "Book Management", and "Logout". A green success message box at the top says "Deleted record with id 129". Below it, the "List Books" section has filtering options: "Rating Low" and "Rating High" dropdowns, "Sort" dropdowns for "Created" and "Order" (set to "-"), and a "Limit" input field set to "10". Buttons for "Filter" and "Clear" are present. The "Saved Books" section shows a table with columns: bookID, title, author, image, rating, and Actions. The first row contains "Title" and "Author" with a rating of "0.00" and three action buttons: "View" (blue), "Edit" (grey), and "Delete" (red). A large black area below the table is labeled "Successful delete message.".

Checklist Items (1)

#1 Show the success message of a delete



The screenshot shows a web browser window with the URL <https://it202-njc5-dev-d7eb077a19c4.herokuapp.com/>. The page title is "Project/admin/list_books.php". The navigation bar includes "Bookstore", "Home", "Profile", "Admin", "Book Management", and "Logout". A red error message box at the top says "Invalid id passed to delete". Below it, the "List Books" section has filtering options: "Rating Low" and "Rating High" dropdowns, "Sort" dropdowns for "Created" and "Order" (set to "-"), and a "Limit" input field set to "10". Buttons for "Filter" and "Clear" are present. The "Saved Books" section is partially visible at the bottom.

bookID	title	author	image	rating	Actions
	Title	Author		0.00	View Edit Delete

Invalid failed delete.

Checklist Items (1)

#2 Show any error messages of a failed delete (like id not being passed)

```

1   ...
2
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 delete", "danger");
13      exit(header("Location: " . get_url("admin/list_books.php")));
14  }
15
16  $db = getDB();
17  $query = "DELETE FROM `Books` WHERE id = :id";
18  try {
19      $stmt = $db->prepare($query);
20      $stmt->execute([":id" => $id]);
21      flash("Deleted record with id $id", "success");
22  } catch (Exception $e) {
23      error_log("Error deleting stock $id" . var_export($e, true));
24      flash("Error deleting record", "danger");
25  }
26  exit(header("Location: " . get_url("admin/list_books.php")));
27 //NICHOLAS ZUBRZYCKI njz5

```

Delete code.

Checklist Items (1)

#3 Show the code related to the delete processing

Task #2 - Points: 1

Text: Screenshots of the data

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show a before and after screenshot of the DB data

		SHOW DELTA AND DETAIL INFORMATION OF THE DB STATE
#2	1	Note what changed (i.e., record removed or soft delete value changed)
#3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a database interface with a query results table. The table has the following columns: id, created timestamp, modified timestamp, userID, bookID, title, author, image, and rating. The data for book ID 130 is shown in the first row.

	id	created timestamp	modified timestamp	userID	bookID	title	author	image	rating
	130	2024-04-16 02:35:06	2024-04-16 02:35:06	1	(NULL)	Delete	Delete	0.00	

Before delete.

Checklist Items (2)

#1 Show a before and after screenshot of the DB data

#3 Clearly caption screenshots

The screenshot shows a database interface with a query results table. The table has the same columns as the previous screenshot. The data for book ID 130 is shown in the first row. The 'Total' column shows a value of -1, indicating a change from the previous state.

	id	created timestamp	modified timestamp	userID	bookID	title	author	image	rating
	130	2024-04-16 02:35:06	2024-04-16 02:35:06	1	(NULL)	Delete	Delete	0.00	

After delete (hard delete).

Checklist Items (3)

#1 Show a before and after screenshot of the DB data

#2 Note what changed (i.e., record removed or soft delete value changed)

#3 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain the delete logic

Checklist

*The checkboxes are for your own tracking

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

Response:

The deletion is a hard delete, so the entire record is removed from the database.

Right now, only admins are able to delete records. This may change with milestone 3 depending on how I choose to do it.

The id for the record is passed in the GET variable through URL parameters. If the id is invalid, the user is redirected and nothing happens. Otherwise, the record with the matching id in the database is deleted. The user is then told if the deletion was a success or failure, and is redirected back to the book listing page.

Task #4 - Points: 1

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

Details:

Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

<https://github.com/NickZub/njz5-it202-008/pull/49>

Misc (1 pt.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

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

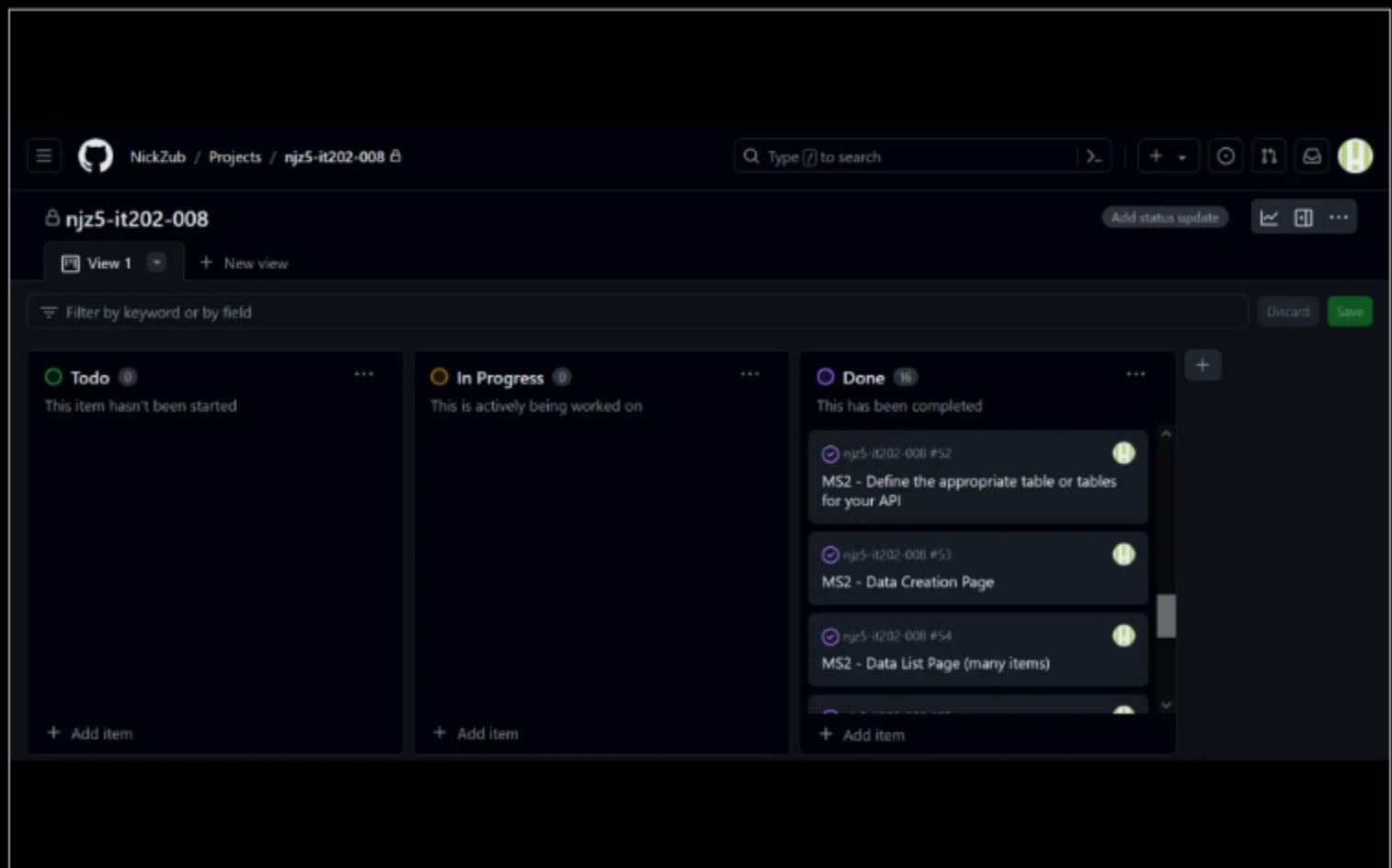
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Project board (not everything shown).

Task #2 - Points: 1

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

[^COLLAPSE ^](#)

URL #1

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

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

The creation page was the most difficult thing to work on. It was difficult to properly set up the query because I was working with arrays of arrays, so I had to do things a bit differently. Pretty much everything else was easier after getting that done.

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

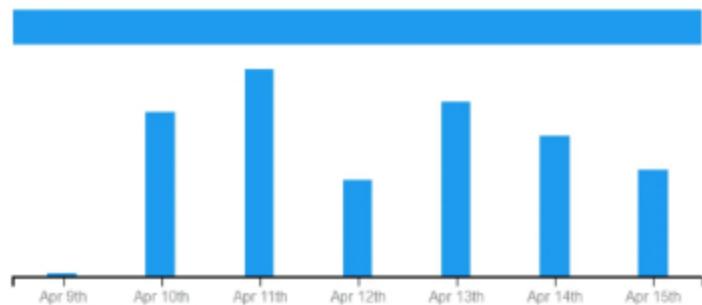
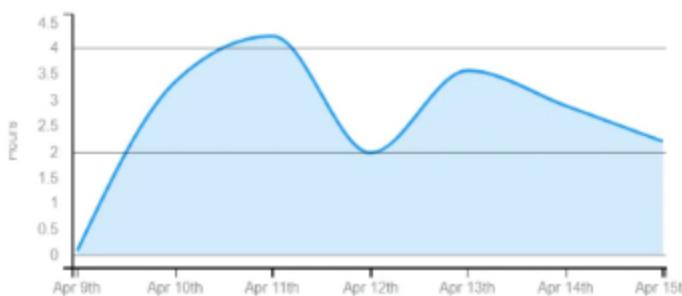
Medium

Large

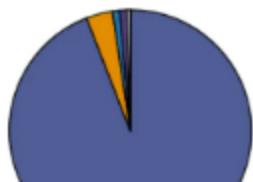
Projects • njz5-it202-008

total 29 hrs 48 mins

18 hrs 14 mins over the [Last 7 Days](#) in njz5-it202-008 under [all branches](#).



Languages



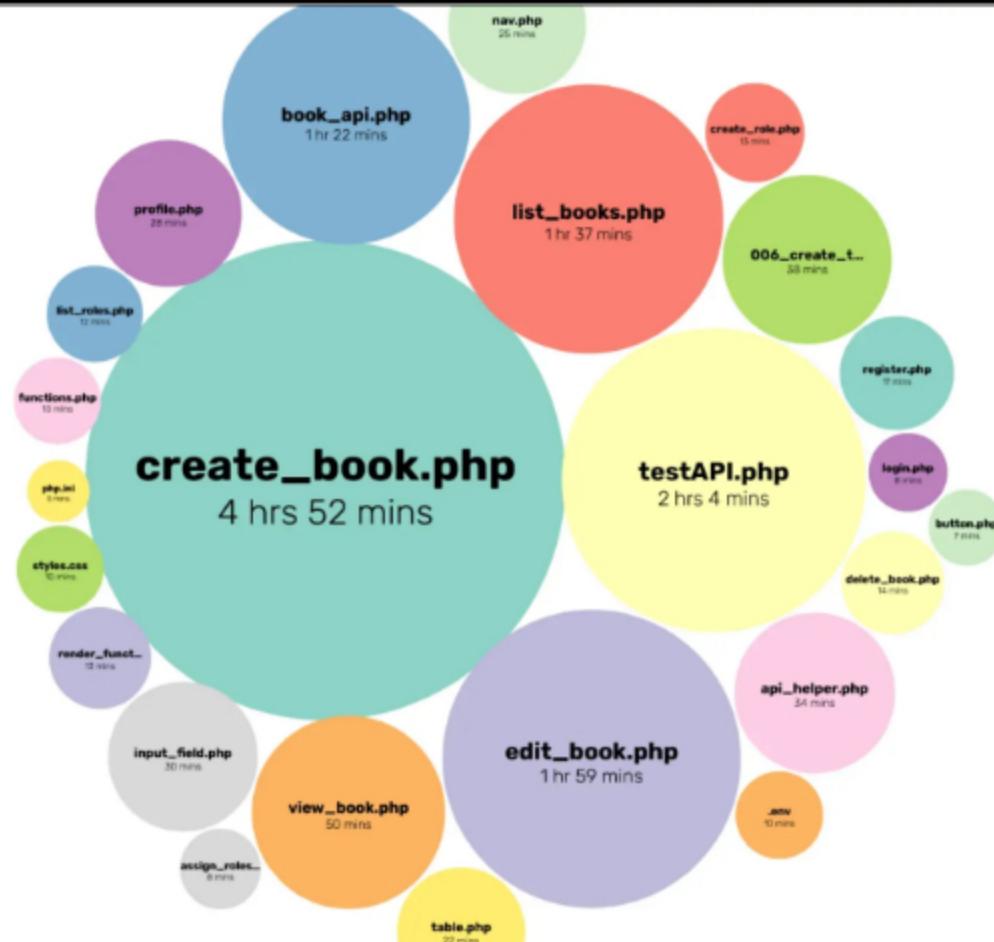
PHP - 17h 10m (94.11%)
SQL - 38m (3.52%)
Bash - 10m (0.94%)
CSS - 10m (0.93%)
INI - 5m (0.48%)
Markdown - 0m (0.01%)

Editors



VS Code - 18h 14m (100.00%)

Waka 1.



Waka 2.

Files

4 hrs 52 mins	...object/admin/create_book.php
2 hrs 4 mins	...ic_html/Project/testAPI.php
1 hr 59 mins	...Project/admin/edit_book.php
1 hr 37 mins	...roject/admin/list_books.php
1 hr 22 mins	lib/book_api.php
50 mins	...Project/admin/view_book.php
38 mins	.../006_create_table_books.sql
34 mins	lib/api_helper.php
30 mins	partials/input_field.php
28 mins	...ic_html/Project/profile.php
25 mins	partials/nav.php
22 mins	partials/table.php
17 mins	...c_html/Project/register.php
14 mins	...oject/admin/delete_book.php
13 mins	lib/render_functions.php
13 mins	...oject/admin/create_role.php
12 mins	...roject/admin/list_roles.php
10 mins	lib/.env
10 mins	public_html/Project/styles.css
10 mins	lib/functions.php
8 mins	...ject/admin/assign_roles.php
8 mins	public_html/Project/login.php
7 mins	partials/button.php
5 mins	C:/php8/php.ini
5 mins	lib/get_url.php

Branches

6 hrs 12 mins	MS2-Feat-APIIntegration
2 hrs 40 mins	MS2-Feat-APIUsage
2 hrs 8 mins	MS2-Feat-Bootstrap
2 hrs 5 mins	Milestone2
1 hr 23 mins	MS2-Feat-DataFiltering
1 hr 10 mins	MS2-Feat-DataListing
1 hr 9 mins	MS2-Feat-DataSingleView
52 mins	MS2-Feat-CleanUp
14 mins	MS2-Feat-DataEditing
7 mins	MS2-Feat-DataDeletion
5 mins	Milestone1
5 mins	Unknown

Waka 3.

End of Assignment