

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-452-M2024/it202-api-project-milestone-2-2024-m24/grade/mrs43>

IT202-452-M2024 - [IT202] API Project Milestone 2 2024 M24

Submissions:

Submission Selection

1 Submission [active] 7/25/2024 7:22:08 PM

Instructions

[^ COLLAPSE ^](#)

Implement the Milestone 2 features from the project's proposal document:

<https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88EWVwfo/view>

Make sure you add your ucid/date as code comments where code changes are done All code changes should reach the Milestone2 branch Create a pull request from Milestone2 to dev and keep it open until you get the output PDF from this assignment. Gather the evidence of feature completion based on the below tasks. Once finished, get the output PDF and copy/move it to your repository folder on your local machine. Run the necessary git add, commit, and push steps to move it to GitHub Complete the pull request that was opened earlier Create and merge a pull request from dev to prod Upload the same output PDF to Canvas

Branch name: Milestone2

Tasks: 23 Points: 10.00

 Define Appropriate Tables for Data (1 pt.)

[^ COLLAPSE ^](#)

 Task #1 - Points: 1

Text: Screenshots of Table SQL

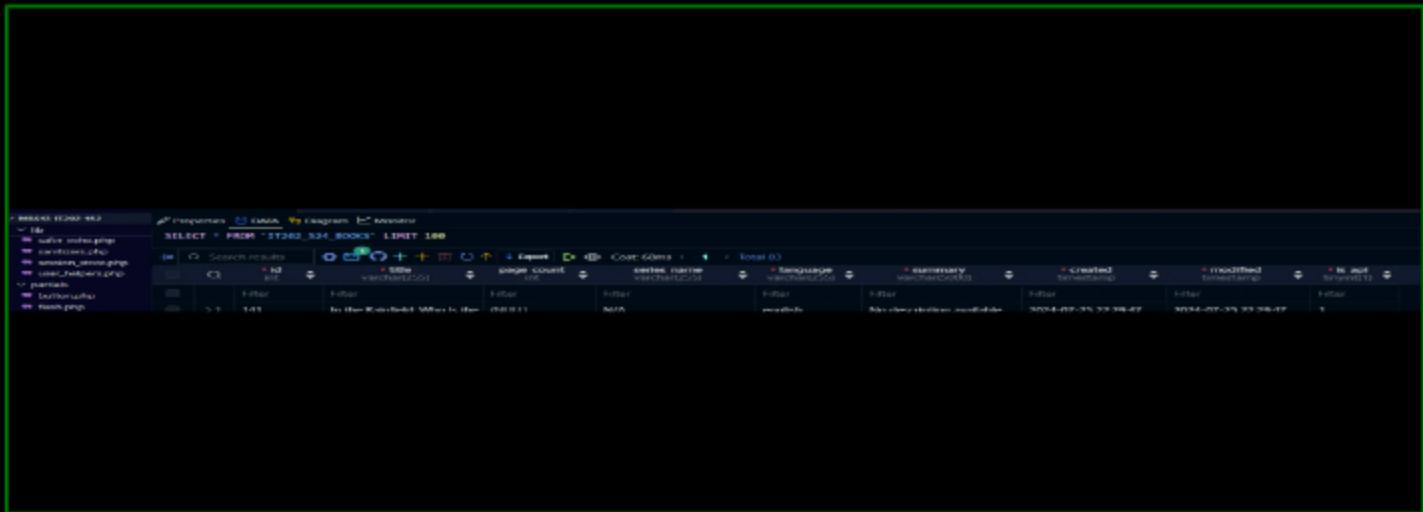
Checklist

*The checkboxes are for your own tracking

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

#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
#2	1	Columns should be logical and thought out (not valid to have a single field of JSON data or similar)

#1) Screenshot of the table (you can duplicate this subtask as needed)



Caption (required) ✓

Describe/highlight what's being shown

Screenshot from VSCode of SQL table

Explanation (required) ✓

Explain the columns and what data they represent (briefly), also note any normalization that may have been necessary

PREVIEW RESPONSE

id: identifier of record

title: title of book

page_count: number of pages

series_name: if book is part of series, this is the name

language: language book is written in

summary: summary of book

created: date and time record was created

modified: date and time record was last modified

is_api: 1 if data from API and 0 if manually entered

Task #2 - Points: 1

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

Details:

URL #1

<https://github.com/m-sansone/mrs43-IT202-452/pull/70>

ht

<https://github.com/m-sansone/mrs43-IT202-452>



URL #2

<https://github.com/m-sansone/mrs43-IT202-452/pull/71>

ht

<https://github.com/m-sansone/mrs43-IT202-452>



URL #3

<https://github.com/m-sansone/mrs43-IT202-452/pull/73>

ht

<https://github.com/m-sansone/mrs43-IT202-452>



URL #4

<https://github.com/m-sansone/mrs43-IT202-452/pull/75>

ht

<https://github.com/m-sansone/mrs43-IT202-452>



+ ADD ANOTHER URL

● Data Creation Page (2 pts.)

▲ COLLAPSE ▲

● Task #1 - Points: 1

Text: Screenshots of the creation page



▼ EXPAND ▼

● Task #2 - Points: 1

Text: Screenshots of creation page code



▼ EXPAND ▼

● Task #3 - Points: 1

Text: Screenshot of records from DB



▼ EXPAND ▼

● Task #4 - Points: 1

Text: Add related links



▼ EXPAND ▼

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

▲ COLLAPSE ▲

● Task #1 - Points: 1

Text: Screenshots of the list page



▲ COLLAPSE ▲

i Details:

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



The screenshot shows a web application interface for managing books. At the top, there's a navigation bar with links for 'To Be Read', 'Home', 'Profile', 'Admin', 'Book Management', and 'Logout'. Below the navigation is a search/filter form with fields for 'Title', 'Number of Pages', 'Series', 'Language', and 'Summary', along with dropdowns for 'Sort' (Title, Order), 'Order' (ASC, DESC), and 'Limit' (15). Below the form is a section titled 'Latest Searched Books' containing a table of book entries. Each entry includes the title, language (all listed as 'english'), and three action buttons: 'View', 'Edit', and 'Delete'. The table also has columns for 'language' and 'Actions'.

Title	language	Actions
#13: The Planet of Tear: Editors	english	View Edit Delete
A Great Day for Pup!	english	View Edit Delete
A History of US: War, Peace, and All That Jazz	english	View Edit Delete
Adore	english	View Edit Delete
An Introduction to Multilevel Modeling Techniques	english	View Edit Delete
Basketball Talk	english	View Edit Delete
Boards of Directors: Their Changing Roles, Structure, and Information Needs	english	View Edit Delete

Caption (required) ✓

Describe/highlight what's being shown

Screenshot from Heroku of list page

Explanation (required) ✓

Briefly describe the page

PREVIEW RESPONSE

There is a list of books from the SQL table with basic information and view, edit, delete buttons. There are also fields to filter, sort, and limit the results.

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



This screenshot is identical to the one above, showing the same 'List Books' page. It features the same navigation bar, search/filter form, and 'Latest Searched Books' table. The table lists the same seven books with their details and actions. The filter/sort form at the top is also identical.

Caption (required) ✓

Caption (required) ✓*Describe/highlight what's being shown*

Screenshot of Heroku list page with filters

Explanation (required) ✓*Briefly mention what's available to the user***[PREVIEW RESPONSE**

The user can filter the list based on title, number of pages, series name, language, and summary. They can sort (ascending/descending) the data by the title, number of pages, or language. The user can also limit the number of results. For each result, the user can view more information, edit the information, or delete the record.

#3) Demonstrate a few varied filters/sorts

To Be Read Home Profile Alerts Book Management Logout

List Books

Title	Number of Pages	Series	Language	Summary	Sort	Order	Limit
<input type="text" value="Cat"/>	<input type="button" value="Number of Pages"/>	<input type="button" value="Series"/>	<input type="button" value="Language"/>	<input type="button" value="Summary"/>	<input type="button" value="Title"/>	<input type="button" value="Number of Pages"/>	<input type="button" value="100"/>
<input type="button" value="View"/>	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>					

Latest Searched Books

Title	Language	Action
El Cat Has No Name	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Knowledge Spaces: Theories, Empirical Research, and Applications	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Advanced Analysis: Techniques and Applications	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
New Developments in Categorical Data Analysis for the Social and Behavioral Sciences	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Perspectives on Nuclear War and Peace Education	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
The 100 Cats of the Cat in the Hat	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

To Be Read Home Profile Alerts Book Management Logout

List Books

Title	Number of Pages	Series	Language	Summary	Sort	Order	Limit
<input type="text" value="Fever"/>	<input type="button" value="Number of Pages"/>	<input type="button" value="Series"/>	<input type="button" value="Language"/>	<input type="button" value="Summary"/>	<input type="button" value="Title"/>	<input type="button" value="Number of Pages"/>	<input type="button" value="100"/>
<input type="button" value="View"/>	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>					

Latest Searched Books

Title	Language	Action
A Fever of Ten	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
A Great Day for Puff!	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
A History of 200 War, Peace, and All That Jazz	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Adolescence	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

To Be Read Home Profile Alerts Book Management Logout

List Books

Title	Number of Pages	Series	Language	Summary	Sort	Order	Limit
<input type="text" value="Eggs"/>	<input type="button" value="Number of Pages"/>	<input type="button" value="Series"/>	<input type="button" value="Language"/>	<input type="button" value="Summary"/>	<input type="button" value="Title"/>	<input type="button" value="Number of Pages"/>	<input type="button" value="100"/>
<input type="button" value="View"/>	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>					

Latest Searched Books

Title	Language	Action
Should You Rather Be a Polygamist?	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
What Is a Fish?	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Why Can't Why Are Deserts Dry?	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Who Hatches the Eggs?	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
What's Inside? What's Below?	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
What Was the Devil Doing First?	english	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Caption (required) ✓*Describe/highlight what's being shown*

Screenshots from Heroku of the data being filtered in different ways

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

To Be Read Home Profile Admin Book Management Logout

List Books

Title	Number of Pages	Series	Language	Summary	Sort	Order
war and peace part 1	Number of Pages	Series	Language	Summary	Sort	Number of Pages

Filter Clear

Latest Searched Books

No records to show

Caption (required) ✓

Describe/highlight what's being shown

Screenshot of filter with no results and appropriate message shown

#5) 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, include examples from different user roles)



To Be Read Home Profile Admin Book Management Logout

List Books

Title	Language	Actions
Rethinking War: The Electoral Incentive in the Democratic Politics of War and Peace	english	View Edit Delete
Gandhi on War and Peace	english	View Edit Delete
Leo Tolstoy's War and Peace	english	View Edit Delete
Perspectives on Nuclear War and Peace Education	english	View Edit Delete

Caption (required) ✓

Describe/highlight what's being shown

Each list item has a view, edit, and delete button

Explanation (required) ✓

Mention which users can interact with the view, edit, and delete links

PREVIEW RESPONSE

Admin will be able to view, edit, and delete links. Other users will only be able to view them.

#6) Each list item should have a summary of the entity (likely won't be the entire entity data)



To Be Read Home Profile Admin Book Management Logout

List Books

Title Number of Pages Series Language Summary Sort Order Limit

war and peace Number of Pages Series Language Summary Title Order Limit

Filter Clear

Latest Searched Books

Title	Language	Actions
Elections and War: The Electoral Incentive in the Democratic Politics of War and Peace	english	View Edit Delete
Gandhi on War and Peace	english	View Edit Delete
Leo Tolstoy's War and Peace	english	View Edit Delete
Perspectives on Nuclear War and Peace Education	english	View Edit Delete

Caption (required) ✓

Describe/highlight what's being shown

Only the title and language are listed in the summary

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



To Be Read Home Profile Admin Book Management Logout

List Books

Title Number of Pages Series Language Summary Sort Order Limit

war and peace Number of Pages Series Language Summary Title Order Limit

Filter Clear

Latest Searched Books

Title	Language	Actions
Elections and War: The Electoral Incentive in the Democratic Politics of War and Peace	english	View Edit Delete
Gandhi on War and Peace	english	View Edit Delete
Leo Tolstoy's War and Peace	english	View Edit Delete
Perspectives on Nuclear War and Peace Education	english	View Edit Delete

Caption (required) ✓

Describe/highlight what's being shown

Screenshot of list page in Heroku

Explanation (required) ✓

Briefly explain your design choices

PREVIEW RESPONSE

I primarily used bootstrap (and have a green and white/grey theme) because I liked the clean look. It seems fresh and modern, while also earthy, which I believe fits the idea of searching for books well.

Task #2 - Points: 1

Text: Screenshots of the list page code

COLLAPSE

i Details:

Include ucid/date comments for each code screenshot

#1) Show the filter/sort form generation



```
//mri43 7-25-2024
//title
$Form = [
    "type" => "text", "name" => "title", "placeholder" => "Title", "label" => "Title", "include_margin" => false,
    "type" => "checkbox", "name" => "page_count", "placeholder" => "Number of Pages", "label" => "Number of Pages", "include_margin" => false,
    "type" => "checkbox", "name" => "language", "placeholder" => "Language", "label" => "Language", "include_margin" => false,
    "type" => "checkbox", "name" => "summary", "placeholder" => "Summary", "label" => "Summary", "include_margin" => false,
    "type" => "radio", "name" => "sport", "options" => ["title" => "titles", "page_count" => "Number of Pages", "language" => "Language", "include_margin" => false],
    "type" => "radio", "name" => "category", "options" => ["titles" => "titles", "page_count" => "Number of Pages", "language" => "Language", "include_margin" => false]
];
$p = $Form;
$Form = array_map(function($item) {
    return $item['name'];
}, $Form);
$p = $Form;
$p = $Form;
```

Caption (required) ✓

Describe/highlight what's being shown
Code from VSC of building search form

Explanation (required) ✓

Briefly explain how the code works (i.e., some stuff may be dynamic)

PREVIEW RESPONSE

A form is created with associative arrays for each item to filter by.

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



```
error_log("Form data: " . var_export($Form, true));
//mri43 7-25-2024
$query = "SELECT id, title, language FROM `TT200_S2A_BOOKS` WHERE 1=1";
$params = [];
$session_key = $_SERVER["SCRIPT_NAME"];
$_SESSION['clear'] = isset($_GET['clear']);
if ($_SESSION['clear']) {
    session_delete($session_key);
    unset($_GET['clear']);
    die(header("Location: " . $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 && !isset($session_data) && count($session_data) > 0) {
    if (!$_SESSION['data']) {
        $session_data = $_GET;
    }
}
foreach ($Form as $k => $v) {
    if (in_array($k, $keys)) {
        $form[$k]['value'] = $_GET[$k['name']];
    }
}
```

```
//mri43 7-25-2024
//title
$title = se($_GET, "title", "", false);
if (!empty($title)) {
    $query .= " AND title like :title";
    $params[":title"] = "%$title%";
}

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

//sort and order
$sort = se($_GET, "sort", "title", false);
if (!in_array($sort, ["title", "page_count", "language"])) {
    $sort = "title";
}
$order = se($_GET, "order", "desc", false);
if (!in_array($order, ["asc", "desc"])) {
    $order = "desc";
}
```

Caption (required) ✓

Describe/highlight what's being shown

Code of database query and filter/sort implementation

Explanation (required) ✓

Briefly explain the related logic

PREVIEW RESPONSE

A query selects fields from the table. The user's session data is analyzed to see if there are current filters, then the query is filtered/sorted/limited according to the user's selections. The data is then fetched.

#3 Show how the output is generated and displayed



```
//mrs43 7-25-2024
$stable = ["data" -> $results, "title" -> "Latest Searched Books", "ignored_columns" -> ["id"], "view_url" -> get_url("index/view_book.php")];
if(has_role("Admin")){
    $stable["edit_url"] = get_url("admin/edit_books.php");
    $stable["delete_url"] = get_url("admin/delete_book.php");
}


### List Books


<form method="GET">
    <div class="row mb-5" 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="#" class="btn btn-secondary">Clear</a>
</form>
<?php render_table($stable); ?>
</div>


```

Caption (required) ✓

Describe/highlight what's being shown

The code of results being added to a table

Explanation (required) ✓

Briefly explain the related logic

PREVIEW RESPONSE

The results of the query are added to a table, then rendered.

#4 Show any restrictions like role guard or login checks



```
//mrs43 7-25-2024
if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    die(header("Location: $BASE_PATH" . "/home.php"));
```

// Build search form

Caption (required) ✓

Describe/highlight what's being shown

Screenshot of admin requirement in VSCode

Explanation (required) ✓

Briefly explain the logic/reasoning

PREVIEW RESPONSE

The code checks if the user is assigned the role "Admin". If not, then the user is redirected to the home page with an error message.

Task #3 - 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 /pull/#. Same pull request shouldn't be used for each feature

URL #1

https://mrs43-prod-5f0e0e79560e.herokuapp.com/project/admin/list_books.php

tt

[https://mrs43-prod-5f0e0e79560e.herokuapp.co](https://mrs43-prod-5f0e0e79560e.herokuapp.com/)



URL #2

<https://github.com/m-sansone/mrs43-IT202-452/pull/72>

tt

<https://github.com/m-sansone/mrs43-IT202-452>



URL #3

<https://github.com/m-sansone/mrs43-IT202-452/pull/73>

tt

<https://github.com/m-sansone/mrs43-IT202-452>



ADD ANOTHER URL

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

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the details page

COLLAPSE

Details:

Heroku dev url must be visible in all relevant screenshots

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



```
//mrs43 7-25-2024
$id = $_GET['id', -1, false];
$book = [];
if ($id > -1) {
    //fetch
    $db = getDB();
    $query = "SELECT title, page_count, series_name, language, summary, is_apl FROM TT202_S24_BOOKS WHERE id = :id";
    try {
        $stmt = $db->prepare($query);
        $stmt->execute([':id' => $id]);
        $r = $stmt->fetch(PDO::FETCH_ASSOC);
        if ($r) {
            $book = $r;
        }
    } catch (PDOException $e) {
        error_log("Error fetching record: " . var_export($e, true));
        flash("Error fetching record", "danger");
    }
} else {
    flash("Invalid id passed", "danger");
    die(header("Location:" . get_url("admin/list_books.php")));
}
```

Caption (required) ✓

Describe/highlight what's being shown

Record is fetched based on id from VSCode

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



```
else {
    //mrs43 7-25-2024
    flash("Invalid id passed", "danger");
    die(header("Location:" . get_url("admin/list_books.php")));
}
```

Caption (required) ✓

Describe/highlight what's being shown

Invalid id redirects back to list page

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



The screenshot shows a web application interface for managing books. At the top, there's a navigation bar with links like 'To Be Read', 'Home', 'Profile', 'Admin', 'Book Management', and 'Logout'. Below the navigation is a green header bar with the title 'View Book'. Underneath are three buttons: 'Back' (grey), 'Edit' (green), and 'Delete' (red). The main content area is a white box containing several data entries:

- Title:** The Great Chicago Fire
- Series Name:** Code Red
- Language:** english
- Number of Pages:** (empty)
- Summary:** No description available
- User or API Data:** API

Caption (required) ✓

Describe/highlight what's being shown

Screenshot of view page from Heroku

Explanation (required) ✓

Briefly explain your design choices

PREVIEW RESPONSE

I primarily used bootstrap (and have a green and white/grey theme) because I liked the clean look. It seems fresh and modern, while also earthy, which I believe fits the idea of searching for books well.

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



This screenshot is identical to the one above, showing the 'View Book' page for 'The Great Chicago Fire'. The layout, data entries, and overall design are the same, demonstrating that the application provides more detailed information than a simple summary view.

Caption (required) ✓

Describe/highlight what's being shown

Screenshot of more fields on view page then list page

Explanation (required) ✓

Briefly explain the details shown and how it differs from the list view

PREVIEW RESPONSE

The list view only has the title of the book and the language. This view page has the title, series name, language, number of pages, summary, and whether the data came from a user or the API.

#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)



Caption (required) ✓

Describe/highlight what's being shown

Edit button on screen

#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)



Caption (required) ✓

Describe/highlight what's being shown

Delete button on screen

 COLLAPSE

Task #2 - Points: 1

Text: Screenshots of the details page code

Details:

Include ucid/date comments for each code screenshot

#1) Show how id is fetched



```
//mrs45 7-25-2024
$id = se($_GET, "id", -1, false);
$book = [];
if ($id > -1) {
    //fetch
    $db = getDB();
    $query = "SELECT title, page_count, series_name, language, summary, is_api FROM `IT202_S24_BOOKS` WHERE id = :id";
    try {
        $stmt = $db->prepare($query);
        $stmt->execute([":id" => $id]);
        $r = $stmt->fetch(PDO::FETCH_ASSOC);
        if ($r) {
            $book = $r;
        }
    } catch (PDOException $e) {
        error_log("Error fetching record: " . var_export($e, true));
        flash("Error fetching record", "danger");
    }
} else {
    flash("Invalid id passed", "danger");
    die(header("Location: " . get_url("admin/list_books.php")));
}
```

Caption (required) ✓

Describe/highlight what's being shown

Code from VSCode showing how id is fetched

Explanation (required) ✓

Briefly explain the logic and how it's handled when the property is missing or not "valid"

 PREVIEW RESPONSE

The id is retrieved and then validated. If the id is not valid, there's a flash message and the user is redirected back to the list page.

#2) Show the DB query to get the record



```
//mrs45 7-25-2024
$id = se($_GET, "id", -1, false);
$book = [];
if ($id > -1) {
    //fetch
    $db = getDB();
    $query = "SELECT title, page_count, series_name, language, summary, is_api FROM `IT202_S24_BOOKS` WHERE id = :id";
    try {
        $stmt = $db->prepare($query);
        $stmt->execute([":id" => $id]);
        $r = $stmt->fetch(PDO::FETCH_ASSOC);
        if ($r) {
            $book = $r;
        }
    } catch (PDOException $e) {
        error_log("Error fetching record: " . var_export($e, true));
        flash("Error fetching record", "danger");
    }
} else {
    flash("Invalid id passed", "danger");
    die(header("Location: " . get_url("admin/list_books.php")));
}
```

```
        .+$r) {
        $book = $r;
    }
} catch (PDOException $e) {
    error_log("Error fetching record: " . var_export($e, true));
    flash("Error fetching record", "danger");
}
} else {
    You, 3 hours ago + added filters, delete page, view page, apt vs m...
    flash("Invalid id passed", "danger");
    die(header("Location:" . get_url("admin/list_books.php")));
}
```

Caption (required) ✓

Describe/highlight what's being shown

DB query on view page from VSCode

Explanation (required) ✓

Briefly explain the logic

 PREVIEW RESPONSE

A query selects data from a table based on the id retrieved. The query is then executed and the results are fetched.

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



```
<!--mrs43 7-25-2024-->
<div class="container-fluid">
    <h3>View Book</h3>
    <div>
        <a href=<?php echo get_url("admin/list_books.php");?>" class="btn btn-secondary">Back</a>
        <?php if (has_role("Admin")):>
            <a href=<?php echo get_url("admin/edit_books.php?id=" . $id);?>" class="btn btn-success">Edit</a>
            <a href=<?php echo get_url("admin/delete_book.php?id=" . $id);?>" class="btn btn-danger">Delete</a>
        <?php endif; ?>
        <ul class="list-group list-group-flush mt-3">
            <li class="list-group-item">
                <strong>Title: </strong><?php echo $title; ?>
            </li>
            <li class="list-group-item list-group-item-success">
                <strong>Series Name: </strong><?php echo $series_name; ?>
            </li>
            <li class="list-group-item">
                <strong>Language: </strong><?php echo $language; ?>
            </li>
            <li class="list-group-item list-group-item-success">
                <strong>Number of Pages: </strong><?php echo $page_count; ?>
            </li>
            <li class="list-group-item">
                <strong>Summary: </strong><?php echo $summary; ?>
            </li>
            <li class="list-group-item list-group-item-success">
                <strong>User or API Data: </strong><?php echo $is_api ? 'API' : 'User'; ?>
            </li>
        </ul>
    </div>
</div>
```

Caption (required) ✓

Describe/highlight what's being shown

The structure of the data from VSCode

Explanation (required) ✓

Briefly explain the logic

 PREVIEW RESPONSE

The data is structured as a list using bootstrap. The links are added using buttons.

Task #3 - Points: 1

Text: Add related links

 COLLAPSE

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://mrs43-prod-5f0e0e79560e.herokuapp.com/view_book.php?id=145

UR

<https://mrs43-prod-5f0e0e79560e.herokuapp.co>



URL #2

<https://github.com/m-sansone/mrs43-IT202-452/pull/73>

UR

<https://github.com/m-sansone/mrs43-IT202-452>



+ ADD ANOTHER URL

Edit Data Page (2 pts.)

COLLAPSE ^

Task #1 - Points: 1

Text: Screenshots of the edit page

Details:

Heroku dev url must be visible in all relevant screenshots

#1) Show before and after screenshots of data you'll edit (two screenshots required)



The screenshot shows a web application interface for managing books. At the top, there's a navigation bar with links like 'Home', 'Profile', 'Admin', 'Book Management', and 'Logout'. Below the navigation, a section titled 'View Book' contains the following information:
- Title: The cat in the hat.
- Series Name: Dr Seuss Classics
- Language: English
- Number of Pages: 123
- Summary: There was a cat with a hat.
- User or API Data: User

This screenshot shows the same 'View Book' page, but with different data. The book title is now 'The hat in the cat'. The other details remain the same: Series Name: Dr Seuss Classics, Language: English, Number of Pages: 111, and Summary: There was a hat that had a cat. The User or API Data field still shows 'User'.

This screenshot shows the same 'View Book' page again. A message at the bottom of the page reads 'Updated record'. The book details are identical to the previous screenshots.

Edit Book

Back

Title	The hat in the cat
Number of Pages	111
Series	Dr Seuss Classics
Language	English
Summary	There was a hat that had a cat.

Update **Delete**

Caption (required) ✓

Describe/highlight what's being shown

Before and after editing in the view page, as well as edit confirmation

Explanation (required) ✓

Briefly explain what you changed

PREVIEW RESPONSE

I changed the title, number of pages, and summary.

#2 Show examples of validation messages

Edit Book

Back

Title	The hat in the cat
Number of Pages	<input type="text" value="4"/>
Series	<input type="text" value="Dr Seuss Classics"/> <small>(Please enter a number.)</small>
Language	English
Summary	There was a hat that had a cat.

Update **Delete**

List Books

Title	Number of Pages	Series	Language	Summary
war and peace	Number of Pages	Series	Language	

Filter **Clear**

Latest Searched Books

title

- Elections and War: The Electoral Incentive in the Democratic Politics of War and Peace
- Gandhi on War and Peace
- Leo Tolstoy's War and Peace

Caption (required) ✓

Describe/highlight what's being shown

Builtin validation as well as invalid id, which redirects to list page

#3 Show an example of successful edit messages

Edit Book

Back

Updated record

Title	The hat in the cat
-------	--------------------

Number of Pages
111

Series
Dr Seuss Classics

Language
English

Summary
There was a hat that had a cat

[Update](#) [Delete](#)

Caption (required) ✓

Describe/highlight what's being shown
successful record edit message



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

Heroku THD Mobile THD Desktop Library Think Python Self Service Software Development Block Problem Web3 Annotated Bibliographies Portfolio Image pdf

To Be Read Home Profile Admin Book Management Logout

Edit Book

Back

Title
Perspectives on Nuclear War and Peace Education

Number of Pages
Number of Pages

Series
N/A

Language
english

Summary
No description available

[Update](#) [Delete](#)

Caption (required) ✓

Describe/highlight what's being shown
Edit page on Heroku

Explanation (required) ✓

Briefly explain your design choices

PREVIEW RESPONSE

I primarily used bootstrap (and have a green and white/grey theme) because I liked the clean look. It seems fresh and modern, while also earthy, which I believe fits the idea of searching for books well.

Task #2 - Points: 1

Text: Screenshots of edit page code

Details:

Include ucid/date comments for each code screenshot

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



Caption (required) ✓

Describe/highlight what's being shown
form with data types from VSCode

Explanation (required) ✓

Briefly explain the data type choices



title: I used text because it is the name of the book, which is a string

`page_count`: I used `number` because this is the number of pages, which is a number

`series_name`: I used `text` because it is the name of the book series, which is a string

language: I used text because it is the language the book is written in (ex. english)

summary: I used text because this is a description of what the book is about, which is provided as a long string

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



Caption (required) ✓

Describe/highlight what's being shown

Validation for edit page from VSCode

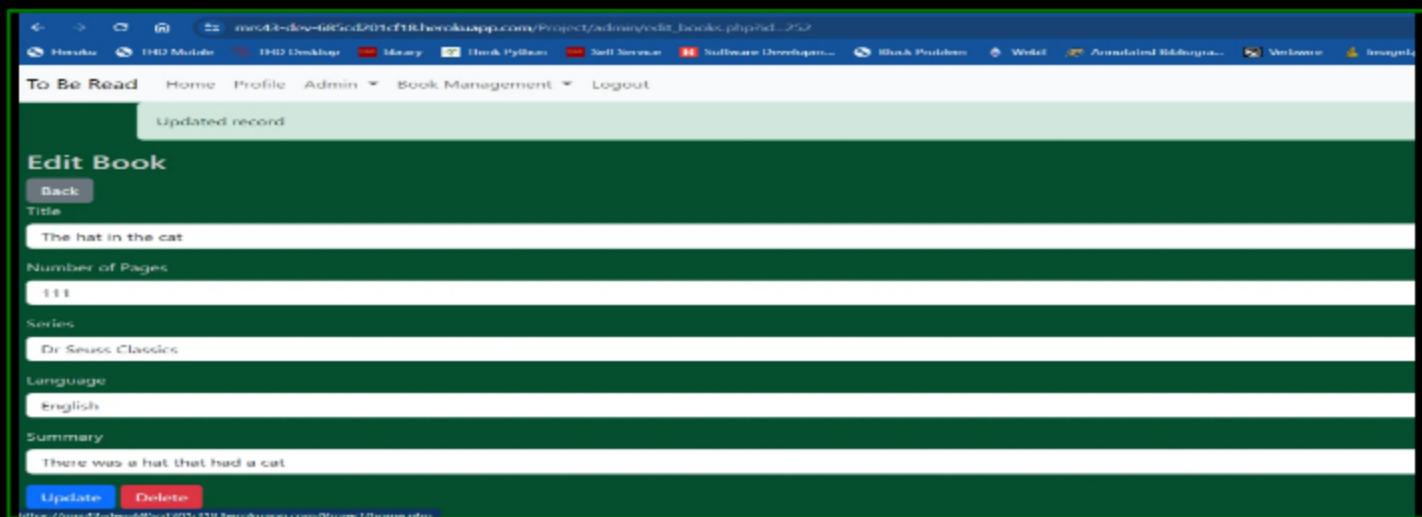
Explanation (required) ✓

Briefly explain the validations

 PREVIEW RESPONSE

.There is built in HTML validation based on the data types and whether the fields are required. The data is validation and sanitized in PHP

#3 Successful edit should have a user-friendly message



The screenshot shows a web application interface for editing a book record. At the top, there is a navigation bar with links for 'Home', 'Profile', 'Admin', 'Book Management', and 'Logout'. Below the navigation, a success message 'Updated record' is displayed. The main form is titled 'Edit Book' and contains fields for 'Title' (value: 'The hat in the cat'), 'Number of Pages' (value: '111'), 'Series' (value: 'Dr. Seuss Classics'), 'Language' (value: 'English'), and 'Summary' (value: 'There was a hat that had a cat'). At the bottom of the form are two buttons: 'Update' (blue) and 'Delete' (red).

Caption (required) ✓

Describe/highlight what's being shown

A success message for a record being edited on Heroku

Explanation (required) ✓

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

 PREVIEW RESPONSE

The code first checks if the user is an admin. If they are not, they are redirected to the home page with a flash message. The code then safely retrieves the id, which is set to -1 if not provided. The code then checks if the form has been submitted to ensure only necessary fields are used. A query is used to update the SQL table using the data that was just sanitized. The code tries to execute the query, but also catch any exceptions which would also prompt a flash message. The data is then fetched using the id. If the id is invalid, the user is redirected to the list page with a flash message. The form is set up with the current record data, then rendered with input fields and submit and delete buttons.

#4 Any errors should have user-friendly messages



The image contains two side-by-side screenshots of a web application. The left screenshot shows a search results page titled 'List Books' with a search bar and a table of results. An error message 'Invalid id passed' is displayed above the table. The right screenshot shows a simple 'Home' page with a message 'You don't have permission to view this page'.

Caption (required) ✓

Describe/highlight what's being shown

Error message because of invalid id

Explanation (required) ✓

Briefly explain the possible errors

PREVIEW RESPONSE

The user could be trying to access the page without being an admin, an invalid id could be used to fetch a record, if an error occurs with the SQL query, or an error with fetching the record.

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

```
//mrs43 7-25-2024
if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    die(header("Location: $BASE_PATH" . "/home.php"));
}
```

Caption (required) ✓

Describe/highlight what's being shown

Role validation for edit page

Explanation (required) ✓

Briefly explain the logic/reasoning

PREVIEW RESPONSE

The code checks if the user is assigned the role "Admin". If not, then the user is redirected to the home page with an error message.

Task #3 - Points: 1

Text: Screenshot of records from DB

#1) Show a before and after screenshot of the record (two screenshots)



This screenshot shows a 'View Book' page for a book titled 'The cat in the hat'. The page includes fields for Series Name (Dr Seuss Classics), Language (English), Number of Pages (121), and Summary ('There was a cat with a hat'). The URL is http://ms43-dev-685cd201cf18.herokuapp.com/Project/Admin/view_book.php?id=252.

This screenshot shows the same 'View Book' page for the book 'The cat in the hat'. The title has been changed to 'The hat in the cat'. The URL is http://ms43-dev-685cd201cf18.herokuapp.com/Project/admin/view_book.php?id=252.

This screenshot shows an 'Edit Book' page for the book 'The hat in the cat'. The title has been changed to 'The hat in the cat'. Other fields remain the same: Number of Pages (111), Series (Dr Seuss Classics), Language (English), and Summary ('There was a hat that had a cat'). The URL is http://ms43-dev-685cd201cf18.herokuapp.com/Project/Admin/edit_book.php?id=252.

Caption (required) ✓

Describe/highlight what's being shown

Before and after data on view page, as well as confirmation on edit page

Explanation (required) ✓

Explain what differs

I changed the title, number of pages, and summary.

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

URL #1

[https://mrs43-](https://mrs43-prod-5f0e0e79560e.herokuapp.com/)

[prod-5f0e0e79560e.herokuapp.com](#).[Project/admin/edit_books.php?](#)
id=252

URL #2

<https://github.com/m-sansone/mrs43-IT202-452/pull/72>

URL

<https://mrs43-prod-5f0e0e79560e.herokuapp.co>



URL

<https://github.com/m-sansone/mrs43-IT202-452>



[+ ADD ANOTHER URL](#)

Delete Handling (1 pt.)

[ACOLLAPSE ^](#)

Task #1 - Points: 1

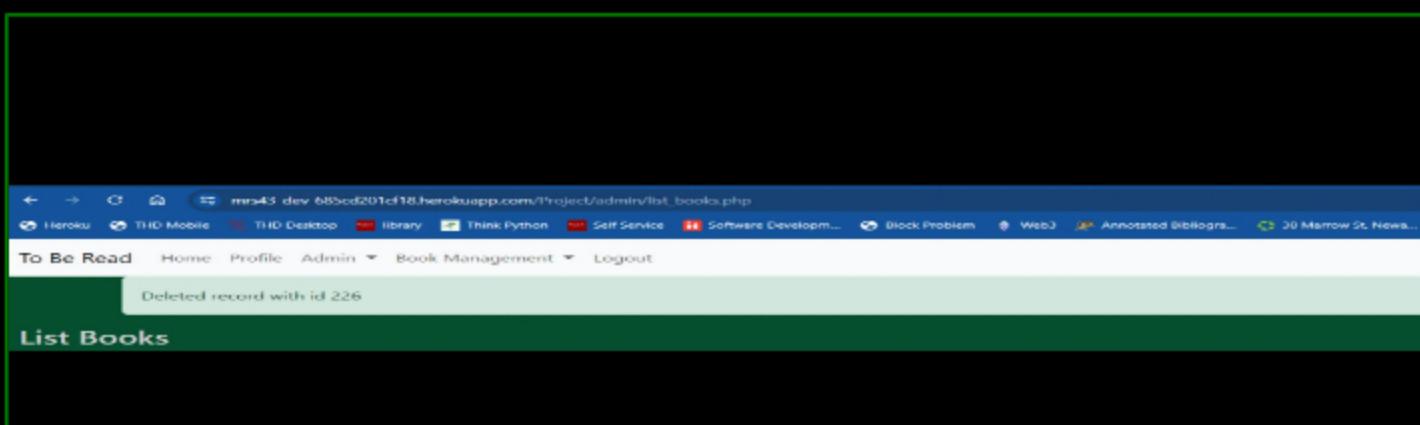
Text: Screenshots related to delete

Details:

Heroku dev url must be visible in all relevant screenshots

Include ucid/date comments for each code screenshot

#1) Show the success message of a delete

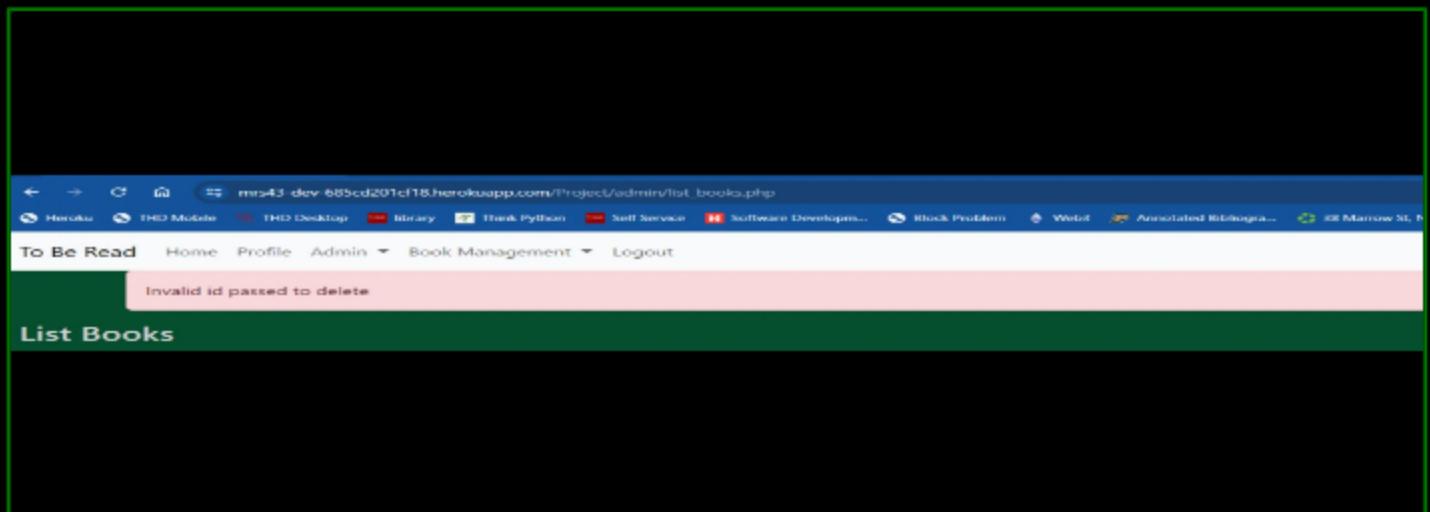


Caption (required) ✓

Describe/highlight what's being shown

Screenshot of successful delete message

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



Caption (required) ✓

Describe/highlight what's being shown

Screenshot of message saying invalid id to delete

Explanation (required) ✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The code tries to execute the query to delete the record, but the id was invalid. Another error that could have occurred is the code trying to execute the query, then catching an exception.

#3) Show the code related to the delete processing



```
//mrs43 7-25-2024
$db = getDB();
$query = "DELETE FROM `IT202_S24_BOOKS` WHERE id = :id";
try {
    $stmt = $db->prepare($query);
    $stmt->execute([":id" => $id]);
    flash("Deleted record with id $id", "success");
} catch (Exception $e) {
    error_log("Error deleting book $id" . var_export($e, true));
    flash("Error deleting record", "danger");
}
die(header("Location: " . get_url("admin/list_books.php")));
```

Caption (required) ✓

Describe/highlight what's being shown
code to delete a record from VSCode

Explanation (required) ✓

Explain in concise steps how this logically works

 PREVIEW RESPONSE

The code connects to the database, then prepares a query to delete a record from the table based on the id. It tries to execute the query, but also has a catch for any exceptions. If successful, the user will be redirected to the list page with a success flash message. If not, the user will be redirected to the list page with an error flash message.

#4) Explain the delete logic



Explanation (required) ✓

Is it a soft or hard delete? Are there any necessary roles or restrictions? (can only delete their data, can only be done by admin, etc)?

 PREVIEW RESPONSE

This is a hard delete that only admin are able to execute.

Task #2 - Points: 1

Text: Screenshots of the data

#1) Show a before and after screenshot of the DB data (two screenshots)



Caption (required) ✓

Describe/highlight what's being shown (note precisely what changed)

The record was deleted

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/m-sansone/mrs43-IT202-452/pull/73>

URL

<https://github.com/m-sansone/mrs43-IT202-452>

+ ADD ANOTHER URL

Misc (1 pt.)

▲ COLLAPSE ▲

Task #1 - Points: 1

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

▼ EXPAND ▼

Task #2 - Points: 1

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

▼ EXPAND ▼

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

▼ EXPAND ▼

Task #4 - Points: 1

Text: WakaTime Screenshot

End of Assignment