

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-3-2024/grade/bns24>

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

1.

Submissions:

Submission Selection

2.

1 Submission [active] 4/27/2024 7:28:51 PM

3.

4.

5.

6.

Instructions

7.

8.

▲ COLLAPSE ▲

9.

10.

Implement the Milestone 3 features from the project's proposal

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

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

All code changes should reach the Milestone3 branch

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

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

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

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

Complete the pull request that was opened earlier

Create and merge a pull request from dev to prod

Upload the same output PDF to Canvas

Branch name: Milestone3

Tasks: 26 **Points:** 10.00

 API (1 pt.)

▲ COLLAPSE ▲

 Task #1 - Points: 1

Text: Data Related to Users

Checklist

*The checkboxes are for your own tracking

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

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

Response:

My app right now is a list of years, along with a fun fact about that year. The association that I am doing allows users to favorite any year facts that are on the website. It is a many to many relationship, because multiple people can favorite the same year, and a user can favorite as many years as they want.

Task #2 - Points: 1

Text: Updating Entities

Checklist

*The checkboxes are for your own tracking

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

Response:

When a year fact is edited manually by an admin, the the associated data is also affected. Users will be able to automatically see the new data. However, once a year fact has been favorited by at least one user, you cannot delete the parent year fact because it is already associated with a user. Only years that are not associated to any user are able to be deleted.

Handle Data Association (1 pt.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



```
<!-- bns24 04/30/24 -->
<?php if (!isset($data["user_id"]) || $data["user_id"] === "N/A") : ?>
    <?php
        $id = isset($data["id"]) ? $data["id"] : (isset($_GET["id"]) ? $_GET["id"] : -1);
    ?>
    <?php if($tabledata != []) :?>
        <a href = "<?php echo get_url('api/addFavorites.php?year_id=' . $data["id"]);?>" class = "card-link">Add To Favorites</a>
    <?php endif; ?>
<?php endif; ?>
```

This section of the code is in the card file in partials. It is used throughout various places in the application.

Checklist Items (0)

public_html > project > api > addFavorites.php > ...

```
1  <?php
2  //no nav here because this is a temp file, not a user-facing file bns24 04/30/24
3  require(__DIR__ . "/../../lib/functions.php");
4  session_start();
5  if(isset($_GET["year_id"])) && is_logged_in(){
6      $db = getDB();
7      $query = "INSERT INTO `UserFavorites` (user_id, year_id) VALUES (:user_id, :year_id)";
8      try{
9          $stmt = $db->prepare($query);
10         $stmt->execute([":user_id"=>get_user_id(), ":year_id"=>$_GET["year_id"]]);
11         flash("Added to Favorites", "success");
12         redirect("myFavorites.php");
13     }catch(PDOException $e){
14         //probably shouldn't happen cause many to many
15         if($e->errorInfo[1] == 1062){
16             flash("You already have favorited this year.", "danger");
17         }else{
18             flash("Unhandled error occurred", "danger");
19         }
20         error_log("Error handling year: " . var_export($e, true));
21     } <- #13-21 catch(PDOException $e)
22 } <- #5-22 if(isset($_GET["year_id"])) && is_logged_in()
23
24 redirect("userListYears.php");
```

The code that actually inserts into the association table the data. This is not a separate page, but it is used in the card

Checklist Items (0)**Task #2 - Points: 1****Text: Screenshot of the association table(s)****Checklist*****The checkboxes are for your own tracking**

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the table(s) you made to handle the associations
<input checked="" type="checkbox"/> #2	1	Should have some example data
<input checked="" type="checkbox"/> #3	1	Make sure the heroku dev url is visible in the address bar
<input checked="" type="checkbox"/> #4	1	Clearly caption screenshots

Task Screenshots:**Gallery Style: Large View****Small****Medium****Large**

The screenshot shows a web application titled "YEAR FACTOIDS" with a URL of https://bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/admin/year_associations.php. The page is titled "Associated Years". It features a search bar with fields for "Username", "Year Number", and "Year Info", and dropdowns for "Sort" (Year), "Order" (+), and "Limit" (10). Filter and clear buttons are also present. Below the search bar, it says "Results: 10/10". The main content area displays four cards, each representing a year and its details:

- 1972**: Favorited By: bruce12. Year: 1972. Text: My mom's birthday year. Type: year. Buttons: View, Unfavorite.
- 1776**: Favorited By: bruce12. Year: 1776. Text: Declaration of Independence is Signed. Type: year. Buttons: View, Unfavorite.
- 2024**: Favorited By: bruce12. Year: 2024. Text: IT202 Class. Type: year. Buttons: View, Unfavorite.
- 1914**: Favorited By: newtesting12. Year: 1914. Text: the city of Irving, Texas is incorporated. Type: year. Buttons: View, Unfavorite.

Screenshot of the UserFavorites table which handles the association, with example data and the heroku dev url in the address bar.

Checklist Items (0)

```
SELECT * FROM `UserFavorites` LIMIT 100
```

	Q	id int	user_id int	year_id int	is_active tinyint(1)	created timestamp	modified timestamp
	> 1	3	18	26	1	2024-04-23 02:54:47	2024-04-23 02:54:47
	> 2	5	18	9	1	2024-04-23 03:19:02	2024-04-23 03:19:02
	> 3	11	18	1	0	2024-04-25 03:22:15	2024-04-25 03:22:19
	> 4	17	19	44	1	2024-04-26 01:06:34	2024-04-26 01:06:34
	> 5	18	19	43	1	2024-04-26 01:06:39	2024-04-26 01:06:39
	> 6	21	21	46	1	2024-04-26 01:47:35	2024-04-26 01:47:35
	> 7	34	8	9	1	2024-04-27 03:23:44	2024-04-27 03:23:44
	> 8	36	8	30	1	2024-04-27 03:23:51	2024-04-27 03:23:51
	> 9	40	21	44	1	2024-04-27 03:24:46	2024-04-27 03:24:46
	> 10	41	8	44	1	2024-04-27 03:58:39	2024-04-27 03:58:39

Screenshot of the UserFavorites table from MySQL, showing the more appropriate columns which handle the associations.

Checklist Items (0)

Task #3 - Points: 1

Text: Explain solution

Checklist

*The checkboxes are for your own tracking

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

Response:

My application handles association by option 1, where it adds onto the existing pages with an "Add To Favorites" link. There is only one association table in my application. It has the primary key column "id", which is simply the id of the association in the table. There is also a foreign key column "user_id" which references the id column in the Users table. There is another foreign key column called "year_id" which references the id column in the Numbers table. Finally, there is an is_active column, which while it is not used in my application, I initially implemented it into the table in the case I wanted to remove an association without hard deleting it. There is also the created and modified column, which states when the association was created and when it was last modified.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://bns24-it202-008-prod-4d42e45e31d0.herokuapp.com/project/user_list_years.php

URL #2

<https://github.com/BruceSiaNJIT/bns24-it202-008/pull/54>

● Current User's Association Page (2 pts.)

[COLLAPSE ^](#)

● Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a web browser displaying the 'My Favorites' page of the 'YEAR FACTOIDS' application. The URL in the address bar is <https://bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/myFavorites.php>. The page has a blue header with the title 'YEAR FACTOIDS' and navigation links for Home, Profile, See Years, My Favorites, Unclaimed Years, Roles, Year Management, and Logout. Below the header, there is a section titled 'My Favorites' with a red 'Remove All Favorites' button. There are three input fields: 'Year Number', 'Year Info', and 'Sort'. Underneath these fields are 'Filter' and 'Clear' buttons. At the bottom, it says 'Results: 3/3' and shows three cards, each with the text 'Favorited By: bruce12'.

1972

Year: 1972
Text: My mom's birthday year
Type: year

View Unfavorite

1776

Year: 1776
Text: Declaration of Independence is Signed
Type: year

View Unfavorite

2024

Year: 2024
Text: IT202 Class
Type: year

View Unfavorite

Screenshot of application showing the current user's associations. Screenshot shows relevant information per entity. There is a single view button and an unfavorite (delete) button which only deletes the association and not the parent year. There is also a remove all favorites button which removes all associations.

Checklist Items (0)

YEAR FACTOIDS Home Profile See Years My Favorites Unclaimed Years Roles Year Management

Year: 1776

Back

1776

Year: 1776
Text: Declaration of Independence is Signed
Type: year
Created: 2024-04-20 02:24:32
Modified: 2024-04-20 02:24:32

What appears when the single view button is clicked.

Checklist Items (0)

YEAR FACTOIDS Home Profile See Years My Favorites Unclaimed Years Roles Year Management Logout

My Favorites

Remove All Favorites

Year Number	Year Info	Sort	Order
Year Number	d	Year	+

[Filter](#) [Clear](#)

Results: 2/3

Favorited By: [bruce12](#)

1776

Year: 1776

Text: Declaration of
Independence is Signed

Type: year

[View](#)

[Unfavorite](#)

Favorited By: [bruce12](#)

1972

Year: 1972

Text: My mom's birthday year

Type: year

[View](#)

[Unfavorite](#)

A variation of the number of shown items, when applied with the filter "d". Shows only 2 of the 3 associated records.

Checklist Items (0)

← → ⌂ bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/myFavorites.php?d

YEAR FACTOIDS Home Profile See Years My Favorites Unclaimed Years R

My Favorites

[Remove All Favorites](#)

Year Number	Year Info	Sort
Year Number	hello	Year

[Filter](#) [Clear](#)

Results: 0/3
No results to show

Variation showing no results, with a proper message being displayed.

Checklist Items (0)

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



```
//bns24 04/27/24
$query = "SELECT username, n.id, text, year, user_id, type FROM `Numbers` n
JOIN `UserFavorites` f ON n.id = f.year_id LEFT JOIN Users u ON u.id = f.user_id
WHERE user_id = :user_id";
$params = [":user_id" => get_user_id()];
$session_key = $_SERVER["SCRIPT_NAME"];
$is_clear = isset($_GET["clear"]);
$session_data = session_load($session_key);
```

The query that fetches the user's associations.

Checklist Items (0)



```
<div class = "container-fluid">
    <h3>My Favorites</h3>
    <div>
        <a href = "?remove" onclick = "if(confirm('Are you sure you want to unfavorite all your years?')){event.preventDefault();}" class = "btn btn-danger">Remove All Favorites</a>
    </div>
    <form method = "GET">
        <div class = "row mb-3" style = "align-items: flex-end;">
            <?php foreach($form as $k=>$v) : ?>
                <div class = "col">
                    <?php render_input($v);?>
                </div>
            </?php endforeach;?>
        </div>
        <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
        <a href = "?clear" class = "btn btn-secondary">Clear</a>
    <?php render_result_counts(count($results), $total_records); ?>
    <div class = "row row-cols-sm-1 row-cols-md-3 row-cols-lg-4 g-4">
        <?php foreach($results as $yr):?>
            <?php $testarray = []; ?>
            <?php $testarray[0] = $yr?>
            <!-- bns 04/27/24 -->
            <?php // $table = ["data"=>$testarray, "ignored_columns" => ["id", "text", "year", "type"], "view_url"=>get_url("user_view_year.php"), "favorite_url"=>get_url("fa
            <?php $table = ["data"=>$testarray, "deleteyear" => ""] ;?>
            <div class = "col">
                <?php render_card($yr, $table, false); ?>
            </div>
        </?php endforeach; ?>
        <?php if(count($results) === 0):?>
            <div class = "col">
                No results to show
            </div>
        </?php endif; ?>
    </div>
</div>
```

Code related to the display of results. It goes through the result array and renders the card for each.

Checklist Items (0)



```
<div class="card-body">
    <!-- bns24 04/27/24 -->
    <?php if (isset($data["id"]) && $stabledata != []): ?>
        <?php if(!isset($stabledata["adminyear"])) : ?>
            <a class="btn btn-primary" href="<?php echo get_url("user_view_year.php?id" . $data["id"]); ?>">View</a>
        <?php endif; ?>
        <?php if(isset($stabledata["deleteyear"])) : ?>
            <?php if (isset($data["user_id"])) : ?>
                <a class="btn btn-danger" href="<?php echo get_url("admin/delete_favorite.php?id" . $data["id"] . "&user_id" . $data["user_id"]); ?>">Unfavorite</a>
            <?php else: ?>
                <a class="btn btn-danger" href="<?php echo get_url("admin/delete_year.php?id" . $data["id"]); ?>">Delete</a>
            <?php endif; ?>
        <?php endif; ?>
    <?php endif; ?>
    <?php if (!isset($data["user_id"]) || $data["user_id"] === "N/A") : ?>
        <?php
            $id = isset($data["id"]) ? $data["id"] : (isset($_GET["id"]) ? $_GET["id"] : -1);
        ?>
        <?php if($stabledata != []) : ?>
            <a href = "<?php echo get_url('api/addFavorites.php?year_id=' . $data["id"]); ?>" class = "card-link">Add To Favorites</a>
        <?php endif; ?>
    <?php endif; ?>
</div>
</div>
</div>
```

Code that displays the view button for each association. It also displays the delete button, named "Unfavorite", which

only removes the association and not the parent year.

Checklist Items (0)

```
//remove all associations bns24 04/27/24
if(isset($_GET["remove"])){
    $query = "DELETE FROM UserFavorites WHERE user_id = :user_id";
    try{
        $stmt = $db->prepare($query);
        $stmt->execute([":user_id"=>get_user_id()]);
        flash("Successfully unfavorited all years.", "success");
    }catch(PDOException $e){
        error_log("Error removing year associations" . var_export($e, true));
        flash("Error removing year associations", "danger");
    }
    redirect("my_favorites.php");
} <- #6-17 if(isset($_GET["remove"]))
```

Code that removes all associations for the current user.

Checklist Items (0)

```
$total_records = get_total_count(`Numbers` n
JOIN `UserFavorites` f ON n.id = f.year_id
WHERE user_id = :user_id", [":user_id"=>get_user_id()]);
```

//bns24 04/27/24

Screenshot 1, showing how I got the total records to display.

Checklist Items (0)



```
] <- #58-107 if(count($_GET) > 0)
//bns24 04/27/24
$stmt = $db->prepare($query);
$results = [];
try{
    $stmt->execute($params);
    $r = $stmt->fetchAll();
    if($r){
        $results = $r;
    }
} <- #111-117 try

catch(PDOException $e){
    error_log("Error fetching year " . var_export($e, true));
    flash("Unhandled error occurred", "danger");
}

//stable = ["data"=>$results, "title" => "Your Year List", "ignored_columns" => ["id"], "view_url"=>get_url("user_view_year.php"));
//stable = ["data"=>$results, "title" => "Your Year List", "ignored_columns" => ["id", "text", "year", "type"], "view_url"=>get_url("user_view_year.php")];
?>

<div class = "container-fluid">
    <h3>My Favorites</h3>
    <div>
        <a href = "?remove" onclick = "if(confirm('Are you sure you want to unfavorite all your years?'))event.preventDefault();"" class = "btn btn-danger">Remove All Fav
    </div>
    <form method = "GET">
        <div class = "row mb-3" style = "align-items: flex-end;">
            <?php foreach($form as $k=>$v) : ?>
                <div class = "col">
                    <?php render_input($v);?>
                </div>
            <?php endforeach;?>
        </div>
        <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
        <a href = "?clear" class = "btn btn-secondary">Clear</a>
        <?php render_result_counts(count($results), $total_records); ?>
    </div>

```

Code showing how I obtained the count of the items currently displayed. \$results came from the filtered query, and render_result_count is a helper function which displays those two variables.

Checklist Items (0)



```
foreach($form as $k=>$v){
    /*if($v["name"] == "number"){
        $form[$k]["name"] = "year";
        $form[$k]["value"] = $_GET[$v["number"]]; bns24 04/27/24
    }*/
    if(in_array($v["name"], $keys)){
        $form[$k]["value"] = $_GET[$v["name"]];
    }
} <- #63-71 foreach($form as $k=>$v)

$yr = se($_GET, "number", "-1", false);
if(!empty($yr) && $yr > -1){
    $query .= " AND year like :year";
    $params[':year'] = "%$yr%";
}
$text = se($_GET, "text", "", false);
if(!empty($text)){
    $query .= " AND text like :text";
    $params[':text'] = "%$text%";
}

$sort = se($_GET, "sort", "year", false);
if (!in_array($sort, ["year", "text"])){
    $sort = "year";
}
$order = se($_GET, "order", "desc", false);
if (!in_array($order, ["asc", "desc"])){
    $order = "desc";
}

$query .= " ORDER BY $sort $order";

try{
    $limit = (int)se($_GET, "limit", "10", false);
} catch(Exception $e){
    $limit = 10;
}

if($limit < 1 || $limit > 100){
    $limit = 10;
}
```

```
$query .= " LIMIT $limit";
} <- #58-107 IF(count($_GET) > 0)
//line 26 04/14/24
```

Code showing the logic for the filter.

Checklist Items (0)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

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

Response:

The way that I got the result list was through a query which gets the number id, text, year, user_id, and type columns from the Numbers table joined with the UserFavorites table on the number id matching the UserFavorites "year_id" column where the user_id is equal to the current user's id. This essentially grabs all the associated years for the current user, and grabs all the appropriate data for the year. Next, we use a very similar filter query as we used in Milestone 2, where we have a form and build onto the query with the data from the form for parameters like the year, text, order by, etc.

In order to display the two counts, I first got the total account by using a db_helper function. I pass in a query where it selects all associations where user_id is equal to the current user id (which comes from another helper function). We store the result in a variable called \$total_records. Then, in order to the visible count result, we just use the count method and pass in the \$result variable, which is either the full unfiltered number, or the filtered query. We pass in these two values into a helper function called render_result_counts. This function first checks to see if the two values are set, then it just prints to the screen the visible result count out of the total count.

The way that I handled the single delete for the association might be a little weird, and is probably not the most efficient way to do it, but it works. In the render_card function, which is what is used to display each of the associations, I also pass a table. The table holds two variables, one being the data for a specific card, and an extra variable which holds the string "deleteyear". In the render_card function, I check to see if "deleteyear" is in the table. If it is, then we check to see if "user_id" is in the data for the specific card. This only shows up for the associations, because those are the only places where it is selected in the query. If it is, then we render delete button, which leads to the delete_favorite.php file, which deletes the association where the year_id is the id in the url and the user_id is the current user.

Finally, to remove all associations, there is a button that, once clicked, passes "remove" to the url. However before it does this, it alerts the user saying whether they are sure they want to actually delete all their favorites, and they have an option to accept or decline. If they decline, nothing happens. If they accept, "remove" is passed to the url, and towards the top of the code we check to see if "remove" is in the url. If it is, then we delete everything from UserFavorites where the user_id column is equal to the current user.

Task #4 - Points: 1

Text: Add related links

Checklist

***The checkboxes are for your own tracking**

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

URL #1

URL #2

<https://github.com/BruceSiaNJIT/bns24-it202-008/pull/55>



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

[COLLAPSE](#)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show the summary of the results with relevant information per entity
#2	1	Show the single view buttons/links, delete button/links
#3	1	Show the username related to the specific entity and that it's clickable
#4	1	Show variations of the number of shown items count and show the count of total number of associated items
#5	1	Show variations of the filter/sort including no results (proper message should be visible)
#6	1	Make sure the heroku dev url is visible in the address bar
#7	1	Clearly caption screenshots

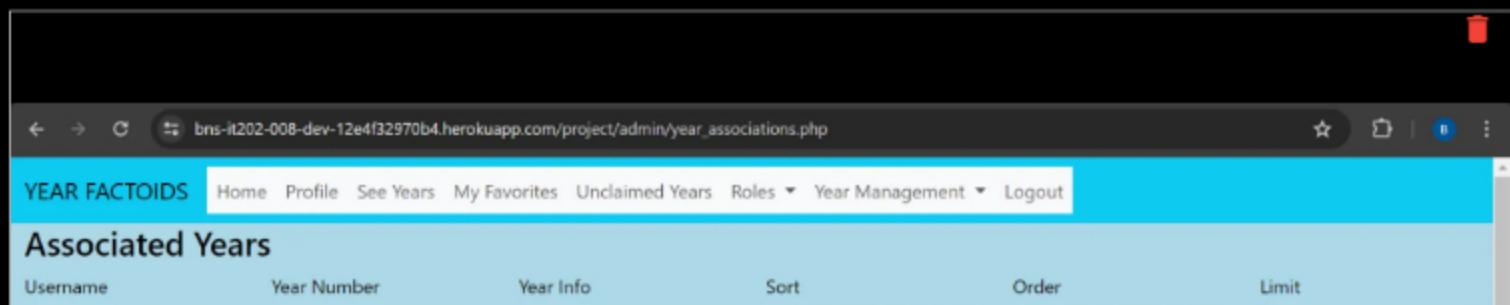
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Username

Year Number

Year Info

Year

Filter Clear

Results:

10/10

Favorited By: bruce12

1972

Year: 1972

Text: My mom's birthday year

Type: year

[View](#) [Unfavorite](#)

Favorited By: bruce12

1776

Year: 1776

Text: Declaration of Independence is Signed

Type: year

[View](#) [Unfavorite](#)

Favorited By: bruce12

2024

Year: 2024

Text: IT202 Class

Type: year

[View](#) [Unfavorite](#)

Favorited By: newtesting12

1914

Year: 1914

Text: the city of Irving, Texas is incorporated

Type: year

[View](#) [Unfavorite](#)

Screenshot showing the associated years, with a summary of relevant information per entity, single view button, unfavorite button, and the username of who it is favorited by.

Checklist Items (0)

The screenshot shows a web browser window with the URL bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/profile.php?id=8. The page title is "YEAR FACTOIDS". The main content area displays the user profile for "bruce12", showing the username, joining date (2024-03-20 17:40:55), and an "Edit" button. Below the profile is a large, empty black rectangular area representing the checklist items.

Example of when you click on username. This is my own profile so it is editable.

Checklist Items (0)

The screenshot shows a web browser window with the URL bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/admin/year_associations.php?username=it202&num=10. The page title is "YEAR FACTOIDS". The main content area displays the "Associated Years" section for the user "it202", listing four years with their details and favoriting buttons.

Username: it202 Year Number: Year Info Sort: Year

Filter Clear

Results: 2/10

Favorited By: [it202](#)

1628

Year: 1628

Text: the Siege of La Rochelle ends with the surrender of the Huguenots

Type: year

[View](#) [Unfavorite](#)

Favorited By: [it202](#)

2024

Year: 2024

Text: IT202 Class

Type: year

[View](#) [Unfavorite](#)

Variation in the number of items displayed, along with the count of the total number of items and what is currently visible.

Checklist Items (0)

YEAR FACTOIDS Home Profile See Years My Favorites Unclaimed Years Roles Year Management Logout

Associated Years

Username: it2022 Year Number: Year Info Sort: Year Order: +

Filter Clear

Results: 0/10

No results to show

Screenshot of no results being shown, along with an appropriate message.

Checklist Items (0)

Task #2 - Points: 1

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large



```
//bns24 04/27/24
$query = "SELECT u.username, n.id, text, year, user_id, type FROM `Numbers` n
JOIN `UserFavorites` f ON n.id = f.year_id JOIN Users u on u.id = f.user_id";
$params = [];
$session_key = $_SERVER["SCRIPT_NAME"];
$is_clear = isset($_GET["clear"]);
$session_data = session_load($session_key);
```

Code related to fetching all associations, including the query.

Checklist Items (0)

```
<div class = "container-fluid">
    <h3>Associated Years</h3>
    <form method = "GET">
        <div class = "row mb-3" style = "align-items: flex-end;">
            <?php foreach($form as $k=>$v) : ?>
                <div class = "col">
                    <?php render_input($v);?>
                </div>
            </?php endforeach;?>
        </div>
        <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
        <a href = "?clear" class = "btn btn-secondary">Clear</a>
    <?php render_result_counts(count($results), $total_records); ?>
<div class = "row row-cols-sm-1 row-cols-md-3 row-cols-lg-4 g-4">
    <?php foreach($results as $yr):?>
        <?php $testarray = []; ?>
        <?php $testarray[0] = $yr?>
        <?php // $table = ["data"=>$testarray, "ignored_columns" => ["id", "text", "year", "type"], "view_url"=>get_url("user_view_year")];
        <?php $table = ["data"=>$testarray, "deleteyear" => ""] ;?>
        <!-- bns24 04/27/24 -->
        <div class = "col">
            <?php render_card($yr, $table, false); ?>
        </div>
    </?php endforeach; ?>
    <?php if(count($results) === 0):?>
        <div class = "col">
            No results to show
        </div>
    </?php endif; ?>
```

Code used to display the results by rendering the cards.

Checklist Items (0)

```
</div>
<div class="card-body">
    <?php if (isset($data["id"]) && $stabledata != []) : ?>
        <!-- bns24 04/27/24 -->
        <?php if(!isset($stabledata["adminyear"])) :?>
            <a class="btn btn-primary" href=<?php echo get_url("user_view_year.php?id=" . $data["id"]); ?>>View</a>
        <?php endif; ?>
        <?php if(isset($stabledata["deleteyear"])) : ?>
            <?php if (isset($data["user_id"])) : ?>
                <a class="btn btn-danger" href=<?php echo get_url("admin/delete_favorite.php?id=" . $data["id"] . "&user_id=" . $data["user_id"]); ?>>Unfavorite</a>
            <?php else: ?>
                <a class="btn btn-danger" href=<?php echo get_url("admin/delete_year.php?id=" . $data["id"]); ?>>Delete</a>
            <?php endif; ?>
        <?php endif; ?>
        <?php if (!isset($data["user_id"]) || $data["user_id"] === "N/A") : ?>
            <?php
                $id = isset($data["id"]) ? $data["id"] : (isset($_GET["id"]) ? $_GET["id"] : -1);
            ?>
            <?php if($stabledata != []) :?>
                <a href = "<?php echo get_url('api/addFavorites.php?year_id=' . $data["id"]); ?>" class = "card-link">Add To Favorites</a>
            <?php endif; ?>
        <?php endif; ?>
    </?php endif; ?>
```

Code for the single view and delete button (only deletes associations)

Checklist Items (0)



```
?php if(isset($data)):>
<div class="card mx-auto" style="width: 18rem;">
    <?php if(isset($data["username"])):>
        <!-- bns24 04/27/24 -->
        <div class = "card-header">
            Favorited By: <a href=<?php echo get_url("profile.php?id=" . $data["user_id"]); ?>><?php se($data, "username", "N/A"); ?>
        </div>
    <?php endif;?>
    <div class="card-body">
        <h5 class="card-title"><?php se($data, "year", "Unknown"); ?></h5>
        <div class="card-text">
            <ul class="list-group">
                <li class="list-group-item">Year: <?php se($data, "year", "Unknown"); ?></li>
                <li class="list-group-item">Text: <?php se($data, "text", "Unknown"); ?></li>
                <li class="list-group-item">Type: <?php se($data, "type", "Unknown"); ?></li>
                <?php if($includesCreate):?>
                    <li class="list-group-item">Created: <?php se($data, "created", "Unknown"); ?></li>
            </ul>
        </div>
    </div>
</div>
```

Code for the header of the card which displays the user name and has a clickable link to their profile.

Checklist Items (0)



```
$total_records = get_total_count(`Numbers` n
JOIN `UserFavorites` f ON n.id = f.year_id");
```

```
//bns24 04/27/24
```

Code to get the total records/total items on the page

Checklist Items (0)

```
$results = [];
try{
    $stmt->execute($params);
    $r = $stmt->fetchAll();
    if($r){
        $results = $r;
    }
} <- #110-116 try

catch(PDOException $e){
    error_log("Error fetching year " . var_export($e, true));
    flash("Unhandled error occurred", "danger");
}
//bns242 04/27/24
//$table = ["data"=>$results, "title" => "Your Year List", "ignored_columns" => ["id"], "view_url"=>get_url("user_view_year.php")];
//$table = ["data"=>$results, "title" => "Your Year List", "ignored_columns" => ["id", "text", "year", "type"], "view_url"=>get_url("u
?>

<div class = "container-fluid">
    <h3>Associated Years</h3>
    <form method = "GET">
        <div class = "row mb-3" style = "align-items: flex-end;">
            <?php foreach($form as $k=>$v) : ?>
                <div class = "col">
                    <?php render_input($v);?>
                </div>
            <?php endforeach;?>
        </div>
        <?php render_button([["text" => "Search", "type" => "submit", "text" => "Filter"]]); ?>
        <a href = "?clear" class = "btn btn-secondary">Clear</a>
        <?php render_result_counts(count($results), $total_records); ?>
```

Code related to getting the total number of visible records. Also includes the render function to display the total number of visible records, and total number of records.

Checklist Items (0)

```
//username bns24 04/27/24
$username = se($_GET, "username", "", false);
if(!empty($username)){
    $query .= " AND u.username like :username";
    $params[":username"] = "%$username%";
}

$yr = se($_GET, "number", "-1", false);
if(!empty($yr) && $yr > -1){
    $query .= " AND year like :year";
    $params[":year"] = "%$yr%";
}

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

$sort = se($_GET, "sort", "year", false);
if (!in_array($sort, ["year", "text"])){
    $sort = "year";
}

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

$query .= " ORDER BY $sort $order";

try{
    $limit = (int)se($_GET, "limit", "10", false);
    $offset = (int)se($_GET, "offset", "0", false);
    $stmt = $db->prepare($query);
    $stmt->execute([':username' => $username, ':year' => $yr, ':text' => $text, ':sort' => $sort, ':order' => $order]);
    $r = $stmt->fetchAll();
    $results = $r;
    $total_records = count($r);
    $total_items = $total_records;
    $total_pages = ceil($total_items / $limit);
    $current_page = ($offset / $limit) + 1;
    $prev_page = $current_page - 1;
    $next_page = $current_page + 1;
    $last_page = $total_pages;
    $first_page = 1;
    $page_links = [
        "first" => $first_page,
        "prev" => $prev_page,
        "current" => $current_page,
        "next" => $next_page,
        "last" => $last_page
    ];
    $page_links = array_filter($page_links, function($link) {
        return $link != null;
    });
    $page_links = array_values($page_links);
    $page_links = array_map(function($link) {
        return $link;
    }, $page_links);
}
```

```

}catch(Exception $e){
    $limit = 10;
}

if($limit < 1 || $limit > 100){
    $limit = 10;
}

$query .= " LIMIT $limit";

```

Code related to the logic for the filtering, with the limit and the partial match for the username.

Checklist Items (0)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Mention how you determine the result list (include the association logic and filters)
<input checked="" type="checkbox"/> #2	1	Mention the logic behind the two counts (total items and visible result count)
<input checked="" type="checkbox"/> #3	1	Mention the logic for single delete
<input checked="" type="checkbox"/> #4	1	Mention the logic for handling the username requirements

Response:

We have a select query which gets the id, text, year, type from the Numbers table, user_id from the Favorites table, and the username from the Users table using a JOIN. Then the form for the filters builds onto the query, with a new username form field to filter by the username from the Users table.

The logic behind the two counts is very similar to the last one, except the total records come from the total count of the table Numbers joined with UserFavorites on the number id matching with the UserFavorites year id. This essentially gets all associations for all users, instead of limiting it to just the current user. The filtered count is the same as the previous one, where we take the count of the \$results variable (which comes as a result of the executed query). To display these counts, we use the render_result_counts helper function.

The logic for the single delete is also the exact same as before, where we pass in a table into the render_card function with the data for each record and the "deleteyear" string. In the card partial file, we check to see if there is a "deleteyear" string in the passed in table, and if there is, we check to see if user_id is passed in the data. Because we are dealing with associations, it is, and so it displays an Unfavorite button for the single delete.

Finally, for the username requirement, we checked to see if "username" was passed in. If it was, then we create a header at the top of the card that says "Favorited By", followed by the username. The username is also clickable and redirects them to the profile.php page with the id being the user_id, which was also passed in.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/admin/year_associations.php

URL #2

<https://github.com/BruceSiaNJIT/bns24-it202-008/pull/55>

● Unassociated Page (2 pts.)

▲COLLAPSE▲

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a web application interface for managing years. At the top, there's a navigation bar with links for Home, Profile, See Years, My Favorites, Unclaimed Years, Roles, Year Management, and Logout. Below the navigation is a header titled "YEAR FACTOIDS". The main content area is titled "Unclaimed Years" and contains a message: "These years have not been favorited by anyone. Be the first!". Below this, there are search and filter controls. The search section includes fields for "Year Number" and "Year Info", and dropdowns for "Sort" (set to "Year") and "Order" (set to "+"). A "Limit" dropdown is set to "10". There are "Filter" and "Clear" buttons. At the bottom left, it says "Results: 6/6". The main table displays six rows, each containing a year number (1914) and a corresponding year info field (Year: 1914). The last row shows a year number of 1981 with a year info field of "Year: 1981".

Type: year

[View](#) [Add To Favorites](#)

Type: year

[View](#) [Add To Favorites](#)

Type: year

[View](#) [Add To Favorites](#)

Screenshot of the unassociated page, with relevant information for each entity and a single view button.

Checklist Items (0)

← → ⌂
bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/unclaimed_years.php?number=1914&text=&sort=year&order=asc&limit=10
✖

YEAR FACTOIDS [Home](#) [Profile](#) [See Years](#) [My Favorites](#) [Unclaimed Years](#) [Roles](#) [Year Management](#) [Logout](#)

Unclaimed Years

These years have not been favorited by anyone. Be the first!

Year Number

Year Info

Sort

Year

Order

+ ↴

Filter
Clear

Results: 3/6

1914

Year: 1914

Type: year

[View](#) [Add To Favorites](#)

1914

Year: 1914

Type: year

[View](#) [Add To Favorites](#)

1914

Year: 1914

Type: year

[View](#) [Add To Favorites](#)

Variation of the number of items shown, with the number of items shown and the total number of items being displayed.

Checklist Items (0)

← → ⌂
bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/unclaimed_years.php?number=19141&text=&sort=year&order=asc&limit=10
✖

YEAR FACTOIDS
[Home](#) [Profile](#) [See Years](#) [My Favorites](#) [Unclaimed Years](#) [Roles](#) [Year Management](#) [Logout](#)

Unclaimed Years

These years have not been favorited by anyone. Be the first!

Year Number

Year Info

Sort

Year

Order

+ ↴

Limit

10

Filter
Clear

Results: 0/6

No results to show

Screenshot showing no results, along with an appropriate message.

Checklist Items (0)

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large



//bns24 04/27/24

```
$query = "SELECT n.id, text, year, type FROM `Numbers` n WHERE n.id NOT IN (SELECT year_id FROM UserFavorites)";
```

```
WHERE u.id NOT IN (SELECT year_id FROM userFavorites) ,  
$params = [];  
$session_key = $_SERVER["SCRIPT_NAME"];  
$is_clear = isset($_GET["clear"]);  
$session_data = session_load($session_key);
```

Code related to fetching all unassociated entities, including the query.

Checklist Items (0)



```
<div class = "container-fluid">  
    <h3>Unclaimed Years</h3>  
    <p>These years have not been favorited by anyone. Be the first!</p>  
    <form method = "GET">  
        <div class = "row mb-3" style = "align-items: flex-end;">  
            <?php foreach($form as $k->$v) : ?>  
                <div class = "col">  
                    <?php render_input($v);?>  
                </div>  
            <?php endforeach;?>  
        </div>  
        <?php render_button(["text" -> "Search", "type" -> "submit", "text" -> "Filter"]); ?>  
        <a href = "?clear" class = "btn btn-secondary">Clear</a>  
        <?php render_result_counts(count($results), $total_records); ?>  
        <div class = "row row-cols-sm-1 row-cols-md-3 row-cols-lg-4 g-4">  
            <?php foreach($results as $yr):?>  
                <?php $testarray = []; ?>  
                <?php $testarray[0] = $yr?>  
                <?php //bn24 04/27/24 $table = ["data"=>$testarray, "ignored_columns" => ["id", "text", "year", "type"], "view_url"=>get_url("user_view_year.php"), "favorite_url"=>get_url("user_favorite_year.php")];?>  
                <?php $table = ["data"=>$testarray, "ignored_columns" => ["id", "text", "year", "type"], "view_url"=>get_url("user_view_year.php")];?>  
                <div class = "col">  
                    <?php render_card($yr, $table, false); ?>  
                </div>  
            <?php endforeach; ?>  
            <?php if(count($results) === 0):?>  
                <div class = "col">  
                    No results to show  
                </div>  
            <?php endif; ?>  
        </div>  
    </div>
```

Code related to displaying the results of the query.

Checklist Items (0)



```
<div class="card-body">
    <!-- bns 04/27/24 -->
    <?php if (isset($data["id"]) && $tabledata != []) : ?>
        <?php if(!isset($tabledata["adminyear"])) :?>
            <a class="btn btn-primary" href="<?php echo get_url("user_view_year.php?id=" . $data["id"]); ?>">View</a>
        <?php endif; ?>
```

Code related to displaying a single view button per record

Checklist Items (0)



```
$total_records = get_total_count(`^Numbers` n
WHERE n.id NOT IN (SELECT year_id FROM UserFavorites));
```

Code related to the logic for getting the total count for all unassociated records

Checklist Items (0)



```
95 $results = [];
96 try{
97     $stmt->execute($params);
98     $r = $stmt->fetchAll();
99     if($r){
100         $results = $r;
101     }
102 } <- #96-102 try
103
104 catch(PDOException $e){
105     error_log("Error fetching year " . var_export($e, true));
106     flash("Unhanlded error occurred", "danger");
107 }
```

```

108 //bns 04/27/24
109 //stable = ["data"=>$results, "title" => "Your Year List", "ignored_columns" => ["id"], "view_url"=>get_url("user_view_year.php")];
110 //stable = ["data"=>$results, "title" => "Your Year List", "ignored_columns" => ["id", "text", "year", "type"], "view_url"=>get_url("user_view_year.php")];
111 ?>
112 <div class = "container-fluid">
113     <h3>Unclaimed Years</h3>
114     <p>These years have not been favorited by anyone. Be the first!</p>
115     <form method = "GET">
116         <div class = "row mb-3" style = "align-items: flex-end;">
117             <?php foreach($form as $k=>$v) : ?>
118                 <div class = "col">
119                     <?php render_input($v);?>
120                 </div>
121             </?php endforeach;?>
122         </div>
123         <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
124         <a href = "?clear" class = "btn btn-secondary">Clear</a>
125     <?php render_result_counts(count($results), $total_records); ?> 
126     <div class = "row row-cols-sm-1 row-cols-md-3 row-cols-lg-4 g-4">
127         <?php foreach($results as $yr):?>
128             <?php $testarray = [] ;?>
129             <?php $testarray[0] = $yr;>
130             <?php //table = ["data"=>$testarray, "ignored_columns" => ["id", "text", "year", "type"], "view_url"=>get_url("user_view_year.php"), "favorite_url"=>get_url("fav...
131             <?php //stable = ["data"=>$testarray, "ignored_columns" => ["id", "text", "year", "type"], "view_url"=>get_url("user_view_year.php")];?>
132             <div class = "col">
133                 <?php render_card($yr, $table, false); ?>

```

Code showing how it counts the number of items on the page based on the filter. Also shows function which displays both of these counts.

Checklist Items (0)

```

$yr = se($_GET, "number", "-1", false);
if(!empty($yr) && $yr > -1){
    $query .= " AND year like :year";
    $params[':year'] = "%$yr%";
}
$text = se($_GET, "text", "", false);
if(!empty($text)){
    $query .= " AND text like :text";
    $params[':text'] = "%$text%";
}

$sort = se($_GET, "sort", "year", false);
if (!in_array($sort, ["year", "text"])){
    $sort = "year";
}
$order = se($_GET, "order", "desc", false);
if (!in_array($order, ["asc", "desc"])){
    $order = "desc";
}

$query .= " ORDER BY $sort $order";

try{
    $limit = (int)se($_GET, "limit", "10", false);
} catch(Exception $e){
    $limit = 10;
}

if($limit < 1 || $limit > 100){
    $limit = 10;
}
//bns24 04/27/2024

$query .= " LIMIT $limit";

```

Code for the filter logic, including the limit constraint.

Checklist Items (0)

Task #3 - Points: 1

Text: Explain the solution



Checklist

*The checkboxes are for your own tracking

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

Response:

The query which is used to find unassociated years first selects the year_id column from UserFavorites. It then selects the id, text, type, year column from the Numbers table where id is not in the year_id column from UserFavorites. This essentially gets all the year data for years that have not been favorited by anybody. Next, the filter comes from form fields, which builds onto the query using the passed in data.

The total number of items comes from getting the total count from the same query as above. get_total_count is a helper function which is used to do just this. To get the visible results, we use the result which comes from the executed query mentioned before, and take the count of that variable. We then use another helper function called render_result_counts, which displays these two counts.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/unclaimed_years.php

URL #2

<https://github.com/BruceSiaNJIT/bns24-it202-008/pull/55>

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

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the search form with valid data
<input checked="" type="checkbox"/> #2	1	Show the results of the search
<input checked="" type="checkbox"/> #3	1	Show the result of entities and users being associated and unassociated

<input type="checkbox"/> #4	1	Make sure the heroku dev url is visible in the address bar
<input type="checkbox"/> #5	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a web browser window with the URL `localhost:3000/project/admin/assign_year.php`. The page title is "Assign Years". At the top, there are two search input fields: "Username Search" and "Year Search", followed by a blue "Search" button. Below these is a table with two columns: "Users" and "Years to Assign". The "Users" column lists several user IDs with checkboxes next to them. The "Years to Assign" column contains a long list of year IDs from 1 to 46, each preceded by a checkbox. A "Toggle Roles" button is located at the bottom left of the table area.

YEAR FACTOIDS Home Profile See Years My Favorites Unclaimed Years Roles Year Management Logout

Assign Years

Username Search

Year Search

Search

Users	Years to Assign
ID: [1]	1914 <input type="checkbox"/>
ID: [2]	1914 <input type="checkbox"/>
ID: [4]	1914 <input type="checkbox"/>
ID: [5]	1914 <input type="checkbox"/>
ID: [6]	1981 <input type="checkbox"/>
ID: [7]	1822 <input type="checkbox"/>
ID: [8]	1953 <input type="checkbox"/>
ID: [9]	1972 <input type="checkbox"/>
ID: [26]	2005 <input type="checkbox"/>
ID: [30]	1776 <input type="checkbox"/>
ID: [43]	1452 <input type="checkbox"/>
ID: [44]	2024 <input type="checkbox"/>
ID: [46]	1628 <input type="checkbox"/>

Toggle Roles

Search form with valid form fields data, and valid data.

Checklist Items (0)

This screenshot is identical to the one above it, showing the same web browser interface and data for the "Assign Years" page. It displays the same search fields, table structure, and list of years for assignment.

YEAR FACTOIDS Home Profile See Years My Favorites Unclaimed Years Roles Year Management Logout

Assign Years

12

19

Search

Users	Years to Assign
bruce12 <input type="checkbox"/>	ID: [9] 1972 <input type="checkbox"/>
newtesting12 <input type="checkbox"/>	ID: [8] 1953 , ID: [9] 1972 <input type="checkbox"/>
	ID: [1] 1914 <input type="checkbox"/>
	ID: [2] 1914 <input type="checkbox"/>
	ID: [4] 1914 <input type="checkbox"/>
	ID: [5] 1914 <input type="checkbox"/>
	ID: [6] 1981 <input type="checkbox"/>
	ID: [7] 1822 <input type="checkbox"/>
	ID: [8] 1953 <input type="checkbox"/>
	ID: [9] 1972 <input type="checkbox"/>
	ID: [26] 2005 <input type="checkbox"/>
	ID: [30] 1776 <input type="checkbox"/>

Toggle Roles

Results of filtering by usernames containing "12" with years favorites with "19" in it.

Checklist Items (0)

YEAR FACTOIDS Home Profile See Years My Favorites Unclaimed Years Roles ▾ Year Management ▾ Logout

Updated role

Assign Years

12

Year Search

Search

Users	Years to Assign
bruce12 □	ID: [6] 1981 , ID: [9] 1972 , ID: [30] 1776 , ID: [44] 2024
newtesting12 □	ID: [8] 1953 , ID: [9] 1972 , ID: [26] 2005

Associated year 1981 with user bruce12

Checklist Items (0)

localhost:3000/project/admin/assign_year.php

YEAR FACTOIDS Home Profile See Years My Favorites Unclaimed Years Roles ▾ Year Management ▾ Logout

Updated role

Assign Years

12

Year Search

Search

Users	Years to Assign
bruce12 □	ID: [9] 1972 , ID: [30] 1776 , ID: [44] 2024
newtesting12 □	ID: [8] 1953 , ID: [9] 1972 , ID: [26] 2005

ID: [6] 1981 □
ID: [7] 1822 □
ID: [8] 1953 □
ID: [9] 1972 □
ID: [26] 2005 □
ID: [30] 1776 □
ID: [43] 1452 □
ID: [44] 2024 □
ID: [46] 1459 □

Unassociated year 1981 with user bruce12

Checklist Items (0)

Task #2 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```
<form method="POST">
  <?php render_input([ "type" => "search", "name" => "username", "placeholder" => "Username Search", "value" => $username]);/*lazy value to check if form submitted, no ideal*/ ?>
  <?php render_input([ "type" => "search", "name" => "year", "placeholder" => "Year Search", "value" => $year]);/*lazy value to check if form submitted, not ideal*/ ?>
  <?php render_button([ "text" => "Search", "type" => "submit"]); ?>
</form>
<form method="POST">
```

Search form field for finding a partial match for usernames and entities (year number).

Checklist Items (0)



```
$stmt = $db->prepare("SELECT Users.id, username,
(SELECT GROUP_CONCAT(' ID: [' , Numbers.id, '] ', year, ' ') from
UserFavorites uf JOIN Numbers on uf.year_id = Numbers.id WHERE uf.user_id = Users.id AND Numbers.year like :year LIMIT 25) as years
from Users WHERE username like :username LIMIT 25");
try {
$stmt->execute([":username" => "%$username%", ":year" => "%$year%"]);
$results = $stmt->fetchAll(PDO::FETCH_ASSOC);
if ($results) {
$users = $results;
}
foreach($users as $key => &$testarray){
foreach($testarray as $k=>$v){
if($k === "years" && is_null($v)){
unset($users[$key]);
}
} <- #105-109 foreach($testarray as $k=>$v)
} <- #104-110 foreach($users as $key => &$testarray)
} catch (PDOException $e) {
flash(var_export($e->errorInfo, true), "danger");
}
}if (!empty($year)){
$db = getDB();
$stmt = $db->prepare("SELECT Users.id, username,
(SELECT GROUP_CONCAT(' ID: [' , Numbers.id, '] ', year, ' ') from
UserFavorites uf JOIN Numbers on uf.year_id = Numbers.id WHERE uf.user_id = Users.id AND Numbers.year like :year LIMIT 25) as years
from Users LIMIT 25");
try {
$stmt->execute([":year" => "%$year%"]);
}
```

Code related to getting a maximum of 25 results from the query.

Checklist Items (0)



```
<tbody>
<tr>
<td>
<table class="table">
<?php foreach ($users as $user) : ?>
<tr>
<td>
<label for="user_<?php se($user, 'id'); ?>"><?php se($user, "username"); ?></label>
<input id="user_<?php se($user, 'id'); ?>" type="checkbox" name="users[]" value="<?php se($user, 'id'); ?>" />
</td>
<td><?php se($user, "years", "No Years"); ?></td>
</tr>
<?php endforeach; ?>
</table>
</td>
<td>
<?php foreach ($active_years as $year) : ?>
<div>
```

```

        <label for="year_<?php se($year, 'id'); ?>"><?php se("ID: [" . se($year, "id") . se("] ") . se($year, "year"); ?></label>
        <input id="year_<?php se($year, 'id'); ?>" type="checkbox" name="years[]" value="<?php se($year, 'id'); ?>" />
    </div>
</php endforeach; ?>
</td>

```

Code that generates the checkbox next to each list, both the users and the entities.

Checklist Items (0)

```

//bs24 04/27/24
if (isset($_POST["username"])) {
    $username = se($_POST, "username", "", false);
    $year = se($_POST, "year", "", false);
    if (!empty($username)) {
        $db = getDB();
        $stmt = $db->prepare("SELECT Users.id, username,
        (SELECT GROUP_CONCAT(`ID: [' . Numbers.id, ']', year, ']') from
        UserFavorites uf JOIN Numbers on uf.year_id = Numbers.id WHERE uf.user_id = Users.id AND Numbers.year like :year LIMIT 25) as years
        from Users WHERE username like :username LIMIT 25");
        try {
            $stmt->execute([":username" => "%$username%", ":year" => "%$year%"]);
            $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
            if ($results) {
                $users = $results;
            }
            foreach($users as $key => $testarray){
                foreach($testarray as $k=>$v){
                    if($k === "years" && is_null($v)){
                        unset($users[$key]);
                    }
                }
            } <- #185-189 foreach($testarray as $k=>$v)
        } <- #184-190 foreach($users as $key => $testarray)
    } catch (PDOException $e) {
        flash(var_export($e->errorInfo, true), "danger");
    }
} elseif (!empty($year)){
    $db = getDB();
    $stmt = $db->prepare("SELECT Users.id, username,
    (SELECT GROUP_CONCAT(` ID: [' . Numbers.id, ']', year, ']') from
    UserFavorites uf JOIN Numbers on uf.year_id = Numbers.id WHERE uf.user_id = Users.id AND Numbers.year like :year LIMIT 25) as years
    from Users LIMIT 25");
    try {
        $stmt->execute([":year" => "%$year%"]);
        $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
        if ($results) {
            $users = $results;
        }
        foreach($users as $key => $testarray){
            foreach($testarray as $k=>$v){
                if($k === "years" && is_null($v)){
                    unset($users[$key]);
                }
            }
        } <- #127-131 foreach($testarray as $k=>$v)
    } <- #126-132 foreach($users as $key => $testarray)
} catch (PDOException $e) {
    flash(var_export($e->errorInfo, true), "danger");
}
} else {
    flash("Username and Year must not both be empty", "warning");
}
} <- #114-119 elseif (!empty($year))

```

Code related to the submitting the checkbox lists.

Checklist Items (0)

```

//attempt to apply
if (isset($_POST["users"]) && isset($_POST["years"])) {
    $user_ids = $_POST["users"]; //se() doesn't like arrays so we'll just do this
    $year_ids = $_POST["years"]; //se() doesn't like arrays so we'll just do this
    if (empty($user_ids) || empty($year_ids)) {
        flash("Both users and roles need to be selected", "warning");
    } else {
        //for sake of simplicity, this will be a tad inefficient
        $db = getDB();
        $stmt = $db->prepare("INSERT INTO UserFavorites (user_id, year_id, is_active) VALUES (:uid, :yid, 1)
        ON DUPLICATE KEY UPDATE is_active = !is_active
        ");
        foreach ($user_ids as $uid) {
            foreach ($year_ids as $yid) {
                try {
                    $stmt->execute([":uid" => $uid, ":yid" => $yid]);
                    flash("Updated role", "success");
                } catch (PDOException $e) {
                    flash(var_export($e->errorInfo, true), "danger");
                }
            }
        }
    }
}

```

```

} <- #22-23 foreach ($year_ids as $yid)
} <- #21-30 foreach ($user_ids as $uid)

//checks for duplicates then deletes them bns24 04/27/24
$query = "DELETE FROM `UserFavorites` WHERE is_active = 0";
try{
    $stmt = $db->prepare($query);
    $stmt->execute();
} catch(Exception $e){
    error_log("Error deleting stock $id" . var_export($e, true));
    flash("Error deleting record", "danger");
}

} <- #18-41 $stmt = $db->prepare

```

Code related to applying the associations upon submission.

Checklist Items (0)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

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

Response:

The UI for this is simply a form with two form fields, one for the username, and one for the entity (being the year number). Then, the UI for the association is the same as assign roles, where it displays the users, their associated years, and then the years I could assign them. Due to how my API works, there could be duplicate years, so the way I differentiated them was by putting the ID of the year in front of them. The user can then use checkboxes to assign or unassign years to a user.

When the form is submitted, there is a check to see if the username field is not empty. If it is not empty, it goes through a query where it gets all the necessary information, such as the user id, username, displaying the Number id followed by the year, specifically where username is like the passed in value from the username field on the form. When I tried this solution, it did this thing where it printed out values even though they were null, so I had a for loop go through the result and unset these whenever it happened. If the username field is empty, though, then it only checks for a partial match for the number field without checking username partial match. Both of these have a limit of 25, although admittedly it was sometimes off for reasons unknown and I worked too long on it. If both fields are empty, it flashes a warning message saying that both fields cannot be empty.

Then, the table is displayed, and we check to see if "users" and "years" have been submitted. If it is, we insert into the user association table the values user_id, year_id, and is_active. However, if it's a duplicate, we set is_active to whatever it's opposite is, which is always 0. Because the years function on a hard delete, I had to make a separate query that checked to see if any columns have is_active as 0. If they do, then we delete that column, essentially deleting the association for the user.

●
▲ COLLAPSE ▲

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://bns-it202-008-dev-12e4f32970b4.herokuapp.com/project/admin/assign_year.php

URL #2

<https://github.com/BruceSiaNJIT/bns24-it202-008/pull/56>

●
▲ COLLAPSE ▲

Misc (1 pt.)

Task #1 - Points: 1

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a GitHub project board with the following details:

- Project:** Bruce's Board IT202
- Columns:** Todo, In Progress, Done
- Todo:** This item hasn't been started.
- In Progress:** This is actively being worked on.
- Done:** This has been completed.
 - bns24-it202-008 #65 *** MS3-AdminAssociations
 - bns24-it202-008 #64 MS3-NotAssociated
 - bns24-it202-008 #63 MS3-All Users Association Page
 - bns24-it202-008 #62

+ Add item

+ Add item

+ Add item

Screenshot of project board from GitHub, tasks are in the proper column.

Task #2 - Points: 1

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

URL #1

<https://github.com/users/BruceSiaNJIT/projects/1/views/1>

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

This Milestone in particular wasn't as difficult as Milestone 2 in my opinion. Although that could just be because I feel I managed my time a lot better in this milestone (considering Milestone 2 I had to do most of it the day before it was due). I definitely learned a lot more about how to create associative tables and how to modify queries to get exactly what you want.

I guess I had only two major issues during this assignment, both of which I somewhat overcame. The first was the whole thing with buttons in the single view, or favorites page. The way that I rendered the button on the card was through a table, which wasn't the most optimal way. I ended up changing it to just do a button, but I don't know, I just found the rendering buttons on cards a little difficult. I think another thing that made the cards difficult was how different cases there was for the buttons on the cards. For example, for user_list_years, there could only be a single view button. For myFavorites, there needed to be a delete button, but that delete button must only work for the current user. Then for seeing all associations as an admin, the delete button needed to work for all users. I don't feel that the way that I coded the buttons was the most optimal way to do it, but it works and if I have time after the semester I might look back and try to fix it.

Another major issue was the Assign Year, where the admin can associate/unassociate whatever year to whatever user. It was a little different from Assign Role, because I needed to make it so that unassociating it would delete the association. I also needed to make a brand new search field. It was a lot of PHP query filtering, which really isn't really my strongsuit. I was able to get it to work, and I think the way I did it, while not optimal, was good by my own standards. The only issue is the LIMIT, which doesn't really work that well. Like I said in an earlier response, the query would work, but it would sometimes display all users relationships that didn't exist, and I handled this later in a for loop. My guess is that the LIMIT doesn't work sometimes because it's modifying the query itself and not the result that comes after the for loop. Doing these kinds of queries isn't my strongsuit, so I do believe that the query is the problem and that if I fixed it, it would resolve everything.

 COLLAPSE 

Task #4 - Points: 1
Text: WakaTime Screenshot

 Details:

Task Screenshots:

Gallery Style: Large View

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

Projects

bns24-it202-008

17 hrs 27 mins

[WakaTime Screenshot](#)

[End of Assignment](#)