

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-3-2024/grade/ekh3>

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

Submissions:

Submission Selection

1 Submission [active] 4/28/2024 10:16:44 PM

Instructions

[^ COLLAPSE ^](#)

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

Branch name: Milestone3

Tasks: 26 Points: 10.00

 API (1 pt.)

[^ COLLAPSE ^](#)

 Task #1 - Points: 1

Text: Data Related to Users

Checklist

*The checkboxes are for your own tracking

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

Response:

The relation is to see which challenges were completed by the user and the association will be the information related to completing it. The table used for the relation will keep track of the User's ID, challenge ID, attempts, and time taken. In this case the relationship is a many to many as many users can complete many different challenges. The challenges are unique to the day and can be any game from the games list as long as there's set info or pulled from the API.

Task #2 - Points: 1

Text: Updating Entities

Checklist

*The checkboxes are for your own tracking

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

Response:

In the current implementation, when an entity is updated, the relationship would not need to be changed. As all the relationships are built using a separate table and automatically generated numerical IDs, the data would propagate on the queries without needing to change other entities. However, the data shown to the user will be the new data. Under the presumption that the game itself is not completely altered from the original.

Handle Data Association (1 pt.)



Task #1 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Small

Medium

Large

Code for the association uses Option 2 which assigns the user the relation → other PHP functions were used in previous milestones PHP processing info and saving it to cookie and insertion query

Checklist Items (3)

#1 Option 1: Related pages will have a button to do association (like favorites or similar), Option 2: a separate page will be used to associate entities to a user by some other user (like assignment of entities)

#2 Include ucid/date comments for each code screenshot

#3 Clearly caption screenshots

Code for the association, having the live search API and the HTML + JS for the page

Checklist Items (2)

#2 Include ucid/date comments for each code screenshot

#3 Clearly caption screenshots

Task #2 - Points: 1

Text: Screenshot of the association table(s)

Checklist

*The checkboxes are for your own tracking

#	Points	Details
■ #1	1	Show the table(s) you made to handle the associations
■ #2	1	Should have some example data
■ #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows two MySQL Workbench windows side-by-side. Both windows have a title bar with the application name and a database name (playchallenge). The left window displays a SQL editor with the following query:

```
SELECT * FROM Completed_Games LIMIT 100;
```

The right window also displays a SQL editor with the following query:

```
SELECT * FROM DailyLogs LIMIT 100;
```

Both windows show the results of their respective queries in a grid format. The columns for the 'Completed_Games' table include: id, user_id, dailyScore, attempts, dailyLog, completed, b_active, created, and modified. The columns for the 'DailyLogs' table include: id, generated, b_active, dailyScore, created, and modified.

Association tables. Left - Relationship between user and challenges Right - Challenge Table

Checklist Items (3)

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

#2 Should have some example data

#3 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain solution

Checklist

*The checkboxes are for your own tracking

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

Response:

In the application, the user would be associated with a challenge when they successfully guess a challenge. That being, it would be option 2 assignment. The table DailyGame is used for storing the challenges. It has a ID which is used to refer to the challenge, gameId to indicate the original game, dailyDate for the date it was intended for. The Completed_Games table is the association relationship between the User and the challenge. It has the user ID, challenge ID, number of attempts taken, time taken for seconds for leaderboards (if later implemented), and the normal active status for "deletion." The create and modify exist in both tables for housekeeping.

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

URL #1

https://ekh3-it202-008-prod-e9bbcd10cf36.herokuapp.com/Project/play_challenge.php?start&id=1

URL #2

<https://github.com/WokFriedE/ekh3-it202-008/pull/75>

Current User's Association Page (2 pts.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

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

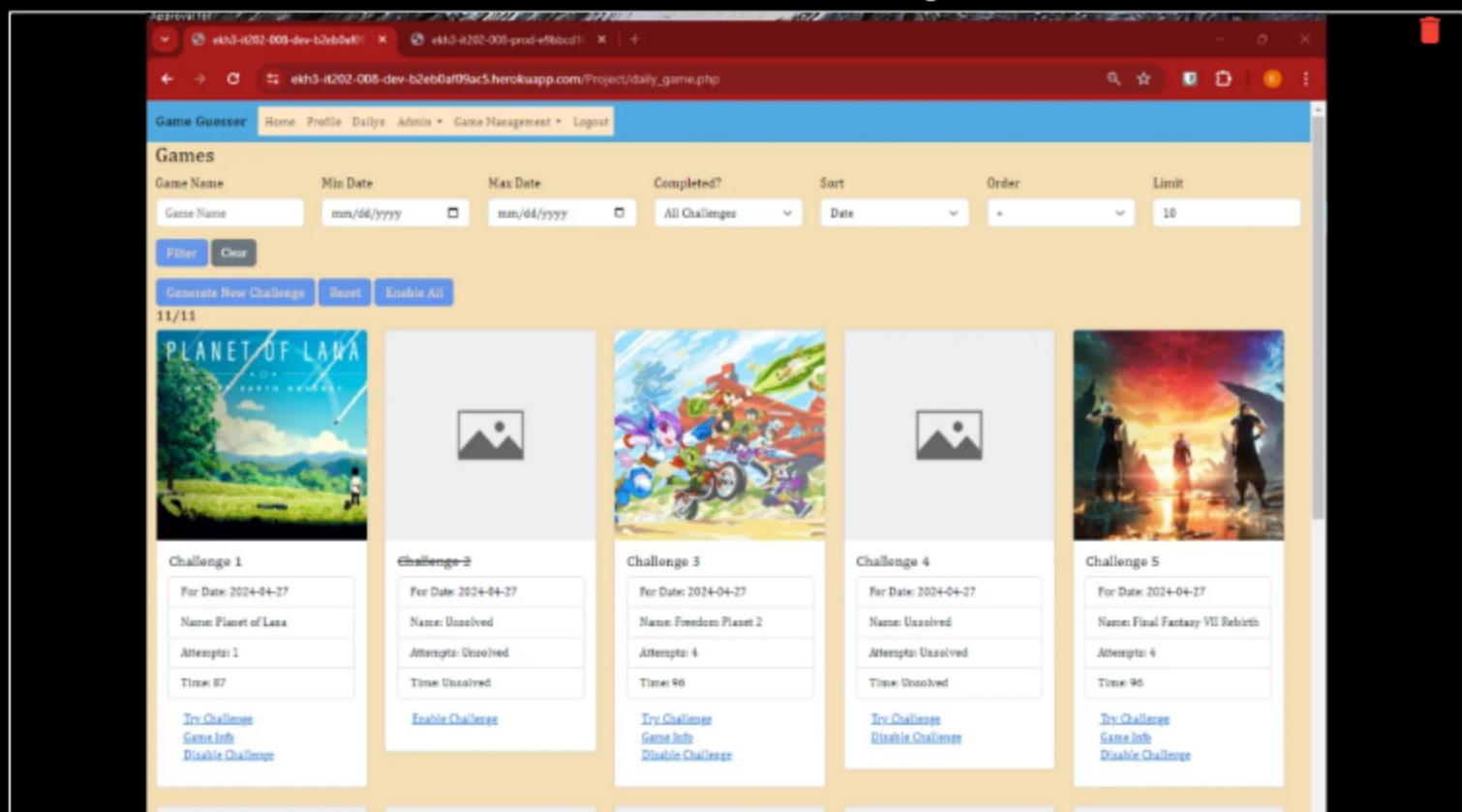
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large





Current user associations. Unsolved will have placeholders. Shows a summary of solved and unsolved challenges. The challenges can be sorted to have only solved or all. Each item has buttons for single view - game info, try challenge - to play, and enable - to disable if admin

Checklist Items (4)

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

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

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

#6 Clearly caption screenshots

The screenshot shows a web application interface for managing challenges. At the top, there's a navigation bar with links for Home, Profile, Dailyrs, Admin, Game Management, and Logout. Below the navigation, a yellow banner displays the message "Cannot sort by name with unknowns". The main area is titled "Games" and contains a table with columns: Game Name, Min Date, Max Date, Completed?, Sort, Order, and Limit. There are buttons for Filter, Clear, Generate New Challenge, Reset, and Enable All. Below the table, it says "5/11". The challenges are listed in cards:

- Challenge 5**
For Date: 2024-04-27
Name: Final Fantasy VII Rebirth
Attempts: 4
Time: 96
[Try Challenge](#) [Game Info](#) [Disable Challenge](#)
- Challenge 1**
For Date: 2024-04-27
Name: Planet of Lava
Attempts: 1
Time: 87
[Try Challenge](#) [Game Info](#) [Disable Challenge](#)
- Challenge 3**
For Date: 2024-04-27
Name: Freedom Planet 2
Attempts: 4
Time: 96
[Try Challenge](#) [Game Info](#) [Disable Challenge](#)
- Challenge 6**
For Date: 2024-04-27
Name: Unsolved
Attempts: Unsolved
Time: Unsolved
[Enable Challenge](#)

Variation for sort and filter. Due to not having only solved items, name cannot be sorted.

Checklist Items (3)

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

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

#6 Clearly caption screenshots

Game Guesser

Home Profile Dailyrs Admin Game Management Logout

Games

Game Name	Min Date	Max Date	Completed?	Sort	Order	Limit
Game Name	04/10/2024	04/23/2024	All Challenges	Date	ASC	5

Filter Clear Generate New Challenge Reset Enable All

0/11 No Results Available

If the sort returns nothing

Checklist Items (3)

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

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

#6 Clearly caption screenshots

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the code related to fetching the user's associations (including the query)
<input checked="" type="checkbox"/> #2	1	Show the code related to the display of the results
<input checked="" type="checkbox"/> #3	1	Each record should have a button/link for single view
<input checked="" type="checkbox"/> #4	1	Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity
<input checked="" type="checkbox"/> #5	1	Show the logic for deleting all associations for the user (this may be admin-only but should be present for the specific role)

#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

Code for user association. Contains, generating a new daily, form information and query for getting information (left) also reset and enable redirects (right) the second part of the main query and filters/sort

Checklist Items (4)

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

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

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

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

Query for the main card display. Function to get total count and HTML including the count of results for count (left) Right card display for render

Checklist Items (4)

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

#4 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

The screenshot displays a Windows desktop environment with multiple open windows. At the top, there is a browser window titled 'd0d-4282-008' showing a complex PHP code editor with numerous files listed on the left side. Below this, there are several other browser windows and file explorers, all showing different parts of a web application. One window shows a login form with fields for 'username' and 'password'. Another window shows a list of users with columns for 'id', 'username', 'name', 'email', and 'status'. A third window shows a detailed view of a user's associations, listing 'Game', 'Category', and 'Character' entries. The overall layout suggests a developer is working on a large-scale project involving user management and data associations.

```

1 // User.php
2 if ($id > 0) {
3     if (!empty($associations)) {
4         foreach ($associations as $association) {
5             if ($association['challenge_id'] == $challenge_id) {
6                 $query = "DELETE FROM `user_associations` WHERE challenge_id = ? AND user_id = ? AND user_id = ?";
```

The screenshot shows a code editor with several tabs open. The main tab displays PHP code related to user associations. The code includes logic for updating user associations based on challenge IDs and user IDs. It also includes comments for checklist items 5, 6, and 7.

Top left get total count for the table. Top middle - render helper Top right - used to get URL for missing image Bottom left - delete the specific association Bottom middle - enable all associations for a user Bottom right - disable all associations for a user

Checklist Items (5)

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

#6 Show the logic related to the count of all associated items to the user (even the ones not shown in the filtered results)

#7 Show the logic related to the count of the items on the page (this value should change based on the filter applied)

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

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:

For the current user association page, would utilize the tables established for the challenges and the completed challenges. It would utilize a query that would pull all the daily game information and join it with the information from the completed challenges table for the specific user. This allows for information regarding challenges like attempts and the time taken to get pulled. To get all the information related, the Games table is also merged to get items such as the game name. To be more specific, the query would pull the daily games list to get all the games (completed or not) and then left join it with the information from the completed table based on the challengeID. There would be one custom field that would give a 1 if the user is associated with the challenge, else a 0. For the filters, it would be similar to milestone 2 where the filters would be added to the where clause of the query. It would sort by minimum and maximum dates and make sure the date that the challenge was made was in between that. The partial name would

only be searchable if the sort only allows games that are completed. There's a filter to only allow games that are completed and any other ones will be unset. The user would also be able to sort using order by date, time taken, attempts, completion, and name. Lastly, there is the limit which would limit the results. All the information would be passed through GET and saved within the session cookie.

For the two counts, the total of all items would use a count query for the daily_games tables which gets all the challenges. Inactive challenges would be disabled and crossed off for now. In the future, I hope to keep them in that state only if the user already completed the item. For the results total, it would just use PHP's count function to get the records returned from the initial query.

For the single delete, there would be a relation deletion page that would take the user ID and challenge ID via GET. It would then toggle that specific record and relation to off or on accordingly. The update would occur in the relation table, Completed_Games, and utilize the UPDATE SQL command. For nonexistent relations, the query would work, but nothing is done. When deleting all the associations for the specific user, a PHP page would take the user ID via GET and disable all the relations in the relationship table by setting all of them to inactive. Therefore, there is also another button to reenable all those relations as well, which does the opposite. Both deletes would only be shown if logged in as an admin which is checked by checking if the current user has the admin role.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://ekh3-it202-008-prod-e9bbcd10cf36.herokuapp.com/Project/daily_game.php

URL #2

<https://github.com/WokFriedE/ekh3-it202-008/pull/63>

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
<input type="checkbox"/> #1	1	Show the summary of the results with relevant information per entity
<input type="checkbox"/> #2	1	Show the single view buttons/links, delete button/links

#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

The screenshot shows a web browser window with two tabs open. The main tab displays the 'Game Guesser' application's 'all_user_associations.php' page. The page has a header with navigation links like Home, Profile, Daily's, Admin, Game Management, and Logout. Below the header is a search/filter section with fields for Username, Min Date, Max Date, Completed?, Sort, Order, and Limit (set to 10). There are also Filter and Clear buttons, along with Generate New Challenge, Reset, and Enable All buttons. The main content area shows a grid of challenges. Each challenge card includes a thumbnail image, the challenge name, the date it was created, the number of successful users, and a list of those users with 'Remove' and 'Hello' buttons. The challenges listed are: Challenge 1 (Planet of Lava), Challenge 2 (TestWorld), Challenge 3 (Freedom Planet 2), Challenge 4 (yala), and Challenge 5 (Final Fantasy VII Rebirth).

Basic view of the all associations page, each user is clickable and are associated with specific challenges

Checklist Items (5)

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

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

#3 Show the username related to the specific entity and that it's clickable

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

#7 Clearly caption screenshots

The screenshot shows a web-based application titled "Game Guesser". The main navigation bar includes links for Home, Profile, Daily's, Admin, Game Management, and Logout. The "Game Management" section is active, displaying a table of user associations. The search bar at the top has "hi" entered. The table columns are: Username, Min Date, Max Date, Completed?, Sort, Order, and Limit. A dropdown menu for "Completed?" is set to "All Challenges". The "Sort" dropdown is set to "Date" and the "Order" dropdown is set to "ASC". The "Limit" input field is set to 2. Below the table are buttons for "Filter", "Clear", "Generate New Challenge", "Reset", and "Enable All". The table displays two rows of data:

Successful Users	(1)
hi_thars	Remove

Below the table, there are two challenge cards:

Challenge 2
For Date: 2024-04-27
Name: TestWorld
[Enable Challenge](#)

Challenge 4
For Date: 2024-04-27
Name: Testing Refl
[Try Challenge](#)
[Game Info](#)
[Disable Challenge](#)

All associations sorted with a username partial and limit

Checklist Items (5)

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

#3 Show the username related to the specific entity and that it's clickable

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

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

#7 Clearly caption screenshots

The screenshot shows the same "Game Guesser" application interface. The search bar now contains "hiddd". The table below shows 0 results.

No Results Available

All association no return

Checklist Items (2)

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

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

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Similar functions to previous on left side - enforce admin role (right) contains the query and the filter sort logic

Checklist Items (5)

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

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

#8 Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

left - query for filter , query to get total and disable all logic right - HTML and counts

Checklist Items (5)

#6 Show the logic related to the count of all associated items (even the ones not shown in the filtered results)

#7 Show the logic related to the count of the items on the page (this value should change based on the filter applied)

#8 Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

The screenshot shows a dual-monitor setup. The left monitor displays a code editor with multiple tabs open, including 'index.php', 'associates.php', and 'associates_update.php'. The right monitor displays a terminal window titled 'git bash-4382-008' showing command-line output related to file operations.

```
git bash-4382-008
$ cd /var/www/html
$ ls
index.php  associates.php  associates_update.php
$ cd associates_update.php
$ ls
index.php  index.html  index.html.old  index.html.recover  index.html~  index.html~.old  index.html~.recover
```

The association card used to display each item. The delete buttons exist here and the code for those are in the previous.

Checklist Items (5)

Checklist Items (6)

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

#4 Each record should have a username field that is clickable to go to the user's profile page

#5 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

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

Response:

To get the list a large merge would need to occur. In a similar query to the current user association, however, the all the users would be saved as a string with their UID's tacked onto their usernames. The query entails joining the game, challenge, challenge relation, and user tables together via several joins. However, this time since it is all associations, it would ignore any challenges that have no users associated with it. For the total count, a select query is done to get all the DailyGames where there is an association involved. The results count would be counting the length of the results. The single delete would use delete_user_associations.php which takes the UID and challenge ID as get items and then update the relationship to disable. The username partial search would be integrated into the original query, pulling only the challenges with the associated usernames. For clicking on the username, it would work by exploding out the information from the query and then directing the users who click to the profile page of the username via their UID.

Task #4 - Points: 1

Text: Add related links

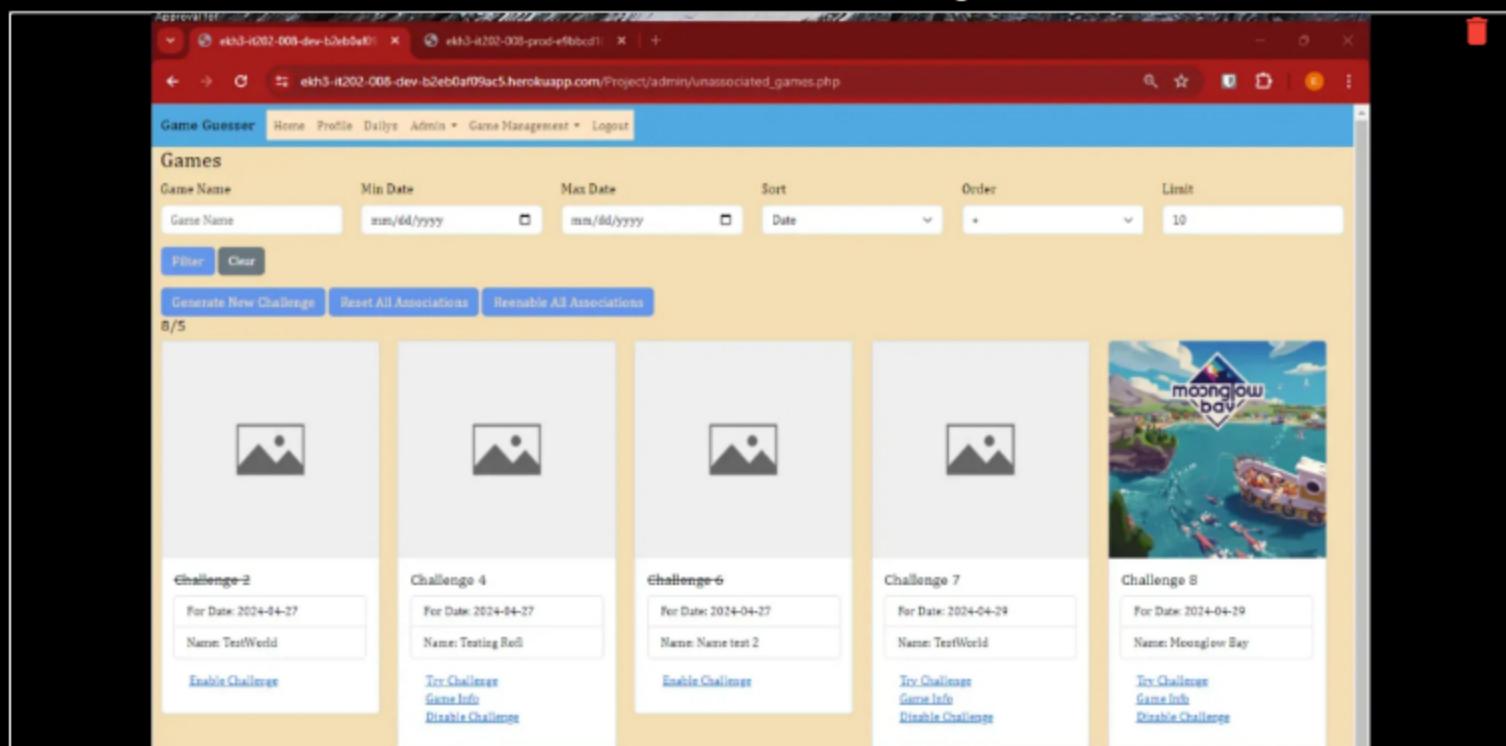
Checklist

*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

URL #1https://ekh3-it202-008-prod-e9bbcd10cf36.herokuapp.com/Project/admin/all_user_associations.php**URL #2**<https://github.com/WokFriedE/ekh3-it202-008/pull/74>**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 checked="" 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**



Basic view for no association with no filter Each item has a game info which is a single view and disable clear all associations (rest all associations) and reenable all associations The max total seems to be broken

Checklist Items (3)

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

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

#6 Clearly caption screenshots

The screenshot shows a web browser window with the URL https://ekh3-it202-008-dev-b2eb0af09ac5.herokuapp.com/project/admin/unassociated_games.php?name=&date_min=2024-04-27&date_max=2024-04-27&sort=date&order=desc&limit=10. The page title is "Game Guesser". The main content area displays three challenges:

- Challenge 4**: For Date: 2024-04-27, Name: Testing Roll. Buttons: Try Challenge, Game Info, Disable Challenge.
- Challenge 6**: For Date: 2024-04-27, Name: Name test 2. Buttons: Enable Challenge.
- Challenge 2**: For Date: 2024-04-27, Name: TestWorld. Buttons: Enable Challenge.

No associations with filter

Checklist Items (0)

The screenshot shows a web browser window with the URL https://ekh3-it202-008-dev-b2eb0af09ac5.herokuapp.com/project/admin/unassociated_games.php?name=ddddd&date_min=&date_max=&sort=date&order=desc&limit=10. The page title is "Game Guesser". The main content area shows a search form with empty fields and a message:

Game Name: Min Date: Max Date: Sort: Order: Limit:

Buttons: Filter, Clear, Generate New Challenge, Reset All Associations, Reenable All Associations.

0/5
No Results Available

No associations with nothing

Checklist Items (0)

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large

Left - shows reset and enable functions plus the generate new challenge right - shows the query to get all items plus the filter and sort

Checklist Items (6)

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

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

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

#6 Show the logic related to filter/sort (should include a

1-100 otherwise default to 10)

Include file/date comments for each code screenshot

#8 Clearly caption screenshots

Left - includes the getting max count and results count (within the HTML contains the display of both) & HTML Right- has date (is a repeat of image 1) The game_card and delete was same as the current user associations

Checklist Items (5)

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

#4 Show the logic related to the count of all unassociated items (even the ones not shown in the filtered results)

#5 Show the logic related to the count of the items on the page (this value should change based on the filter applied)

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

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

Response:

There is a query that acts the opposite of the all associations page. The query would merge the challenges table with the games table to get the information needed to be displayed. It would then try to neglect any challenges that have users associated with them. The filter and sort are the same as the other filters and sorts. The counts work the same as well. The max total is meant to take all the unassociated tables and get the count of that, but it seems to be having an issue. The count for the ones on the screen are the count of the results returned from the query.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://ekh3-it202-008-prod-e9bbcd10cf36.herokuapp.com/Project/admin/unassociated_games.php

URL #2

<https://github.com/WokFriedE/ekh3-it202-008/pull/73>

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

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the page

Checklist

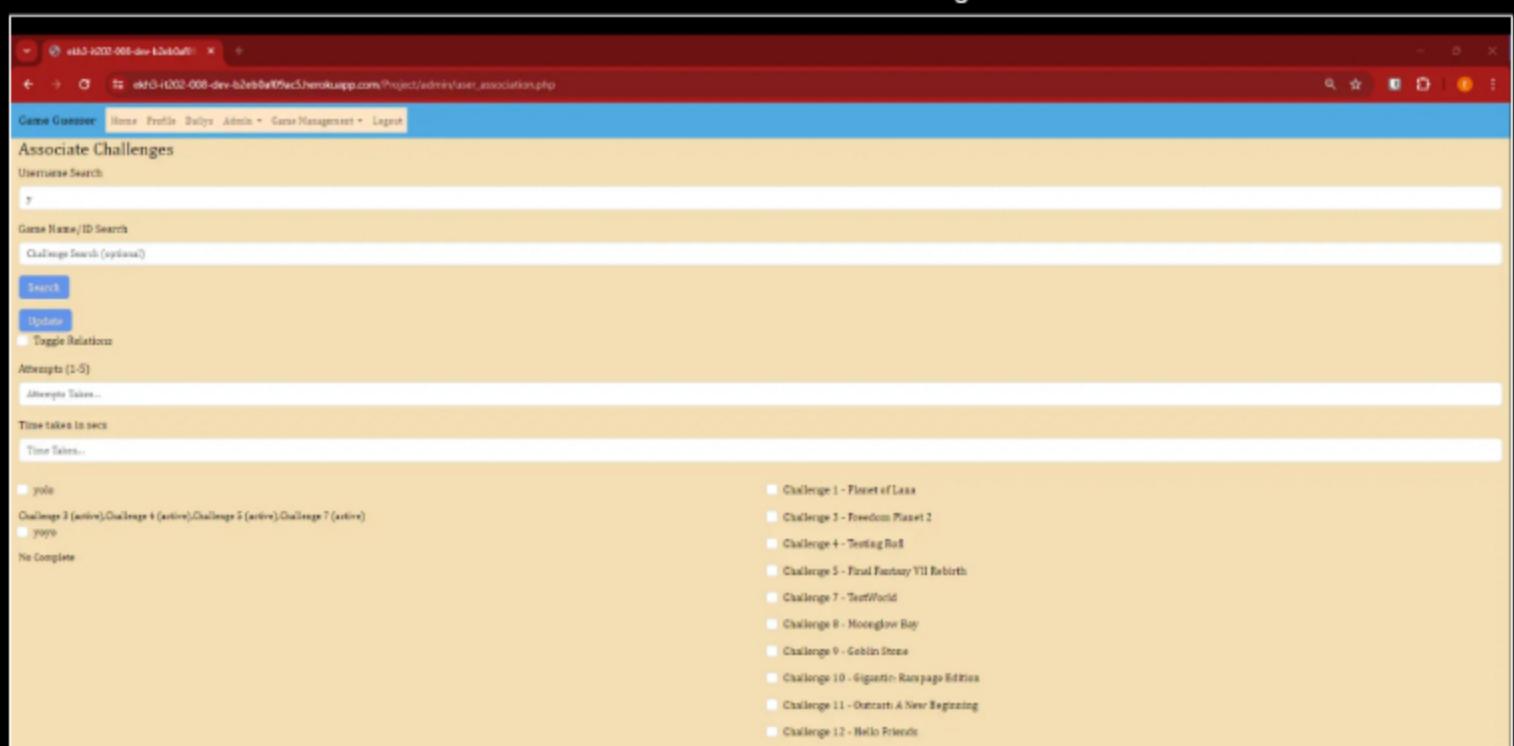
*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large



When typing in a username - a partial search occurs and shows the user's completed challenges

Checklist Items (5)

#1 Show the search form with valid data

#2 Show the results of the search

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

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

#5 Clearly caption screenshots

Task #2 - Points: 1

Text: Screenshots of the code

Checklist

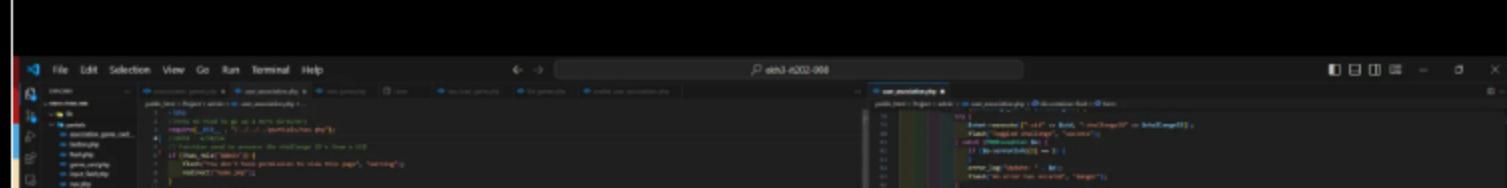
*The checkboxes are for your own tracking

#	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



```

// ...
foreach ($users as $user) {
    $row = [
        'id' => $user->id,
        'name' => $user->name,
        'email' => $user->email,
        'is_admin' => $user->is_admin,
        'is_entitled' => $user->is_entitled,
        'is_entitled' => $user->is_entitled,
        'is_entitled' => $user->is_entitled
    ];
    $rows[] = $row;
}
return $rows;
}

public function search($query) {
    $query = trim($query);
    if ($query === '') {
        return [];
    }
    $users = $this->getUsers();
    $entities = $this->getEntities();
    $rows = [];
    foreach ($users as $user) {
        if (stripos($user->name, $query) !== false || stripos($user->email, $query) !== false) {
            $row = [
                'id' => $user->id,
                'name' => $user->name,
                'email' => $user->email,
                'is_admin' => $user->is_admin,
                'is_entitled' => $user->is_entitled
            ];
            $rows[] = $row;
        }
    }
    foreach ($entities as $entity) {
        if (stripos($entity->name, $query) !== false || stripos($entity->description, $query) !== false) {
            $row = [
                'id' => $entity->id,
                'name' => $entity->name,
                'description' => $entity->description,
                'is_entitled' => $entity->is_entitled
            ];
            $rows[] = $row;
        }
    }
    return $rows;
}

```

```


| User ID | User Name  | User Email             | User Admin Status | User Entitled Status | Entity ID | Entity Name | Entity Description      | Entity Entitled Status |
|---------|------------|------------------------|-------------------|----------------------|-----------|-------------|-------------------------|------------------------|
| 1       | John Doe   | john.doe@example.com   | 1                 | 1                    | 1         | Entity A    | Description of Entity A | 1                      |
| 2       | Jane Smith | jane.smith@example.com | 0                 | 0                    | 2         | Entity B    | Description of Entity B | 0                      |


```

Left - indicates the admin requirements - also applies the partial match for the user and entities Queries shown on both sides Right -> PHP + html

Checklist Items (8)

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

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

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

#4 Code that generates the checkboxes next to each list (users and entities)

#5 Code related to submitting the checkbox lists

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

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#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 search would work by using queries with like clauses for both the challenges and the users. For the challenges it was optional, this worked by making the pattern for the search anything, and if the user put in something it would replace anything with that. For the users, it would be required and basically the query would take in all the information from the challenge and the relation from the user to the challenge and compile a list if the relationship exists or is active. Therefore due to the two lists, they would be checkboxes and side by side using bootstraps grid layout. For association, when the user does not click the toggle relation checkbox, it would result in the challenge ID and the user ID being inserted into the relationship table with the attempts taken and the time taken. The latter 2 pieces of information would be applied to all the challenges selected for the users selected. The query is a bit inefficient as it would conduct each query separately.

There would also be an admin role check.

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

URL #1

https://ekh3-it202-008-prod-e9bbcd10cf36.herokuapp.com/Project/admin/user_association.php

URL #2

<https://github.com/WokFriedE/ekh3-it202-008/pull/70>

Misc (1 pt.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a GitHub project board for the repository "Ethan IT202 Application". The board has three columns: "Todo", "In Progress", and "Done". The "Done" column contains the following items:

- ✓ vish-4200-008 #65 Handle the association of data to a user
- ✓ vish-4200-008 #64 API Data Association
- ✓ vish-4200-008 #66 Admin can associate any entity with any users
- ✓ vish-4200-008 #66 Logged in user's associated entities page
- ✓ vish-4200-008 #66 Create a page that shows data "that" is associated with any user
- ✓ vish-4200-008 #67 All Users association page

Github project board for only milestone 3

Task #2 - Points: 1

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

URL #1

<https://github.com/users/WokFriedE/projects/5/views/4>

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

There was a lot of things to learn about the assignment. For instance, seeing how AJAX works with the live search allows me to see how APIs work directly. The project as a whole taught me a lot with regards to PHP and SQL. The queries I implemented in this part of the assignment were complex due to all the relations and the tables being in somewhat third normal form.

The issues mainly correlate around time, as a lot of features hoped to be completed were not able to be implemented, especially with the game, but hopefully over the summer or sometime after I can work on it.

[COLLAPSE](#)

TASK #4 - Points: 1

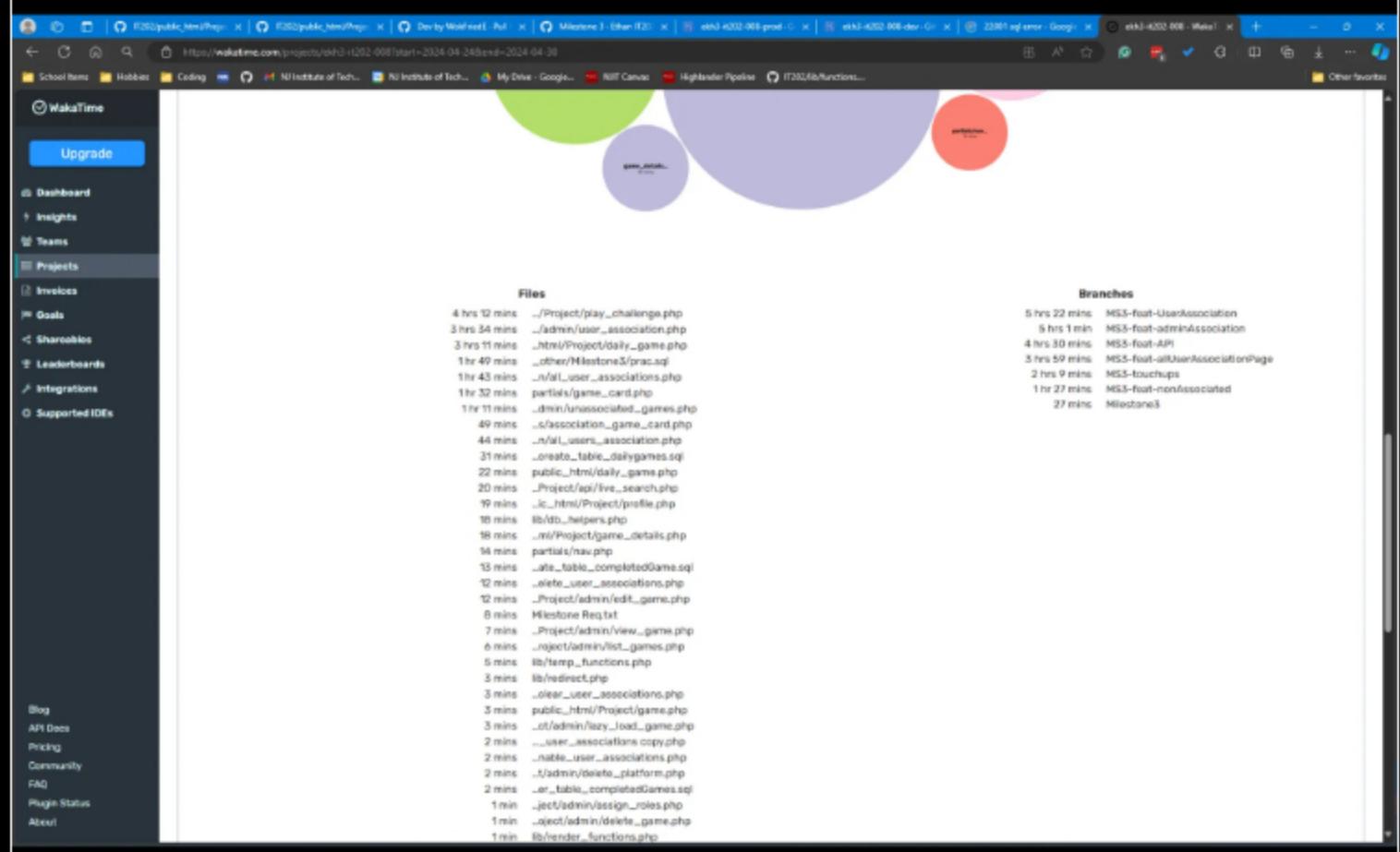
Text: WakaTime Screenshot

Details:

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Waka time screenshot

End of Assignment