

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-2-2024/grade/ekh3>

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

Submissions:

Submission Selection

1 Submission [active] 4/17/2024 1:07:12 AM ▾

Instructions

▲ COLLAPSE ▾

Implement the Milestone 2 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 Milestone2 branch
Create a pull request from Milestone2 to dev and keep it open until you get the output PDF from this assignment.
Gather the evidence of feature completion based on the below tasks.
Once finished, get the output PDF and copy/move it to your repository folder on your local machine.
Run the necessary git add, commit, and push steps to move it to GitHub
Complete the pull request that was opened earlier
Create and merge a pull request from dev to prod
Upload the same output PDF to Canvas

Branch name: Milestone2

Tasks: 29 Points: 10.00

● Define Appropriate Tables for Data (1 pt.)

▲ COLLAPSE ▾

Task #1 - Points: 1

Text: Screenshots of Table SQL

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Table(s) should have the 3 core columns we'll always be using (id, created, modified) plus additional columns for the incoming API data
#2	1	Columns should be logical and thought out (not valid to have a single field of JSON data or similar)
#3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a database application interface with a large table of game data. The table has columns such as id, name, developer, description, and many others. The data is presented in a grid format with numerous rows. The application has a standard Windows-style window with a toolbar at the top and a sidebar on the left containing navigation links.

Games database, contains all fields on the left side and has populated data on the right. Null data indicates lazy loading.

Checklist Items (3)

#1 Table(s) should have the 3 core columns we'll always be using (id, created, modified) plus additional columns for the incoming API data

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

#3 Clearly caption screenshots

The screenshot shows a database application interface with a large table of game data. The table has columns such as id, name, developer, description, and many others. The data is presented in a grid format with numerous rows. The application has a standard Windows-style window with a toolbar at the top and a sidebar on the left containing navigation links.

The screenshot shows two separate database interfaces. The left interface displays a table named 'Genres' with columns: id, name, description, opencritic_url, screenshot_url, square_image_url, release_date, top_critic_score, publisher, developer, is_api, is_active, date_created, and date_modified. The right interface displays a table named 'Platforms' with similar columns. Both tables have rows corresponding to the genres and platforms listed in the accompanying text.

Tables for Genres and Platforms with their relation tables

Checklist Items (3)

#1 Table(s) should have the 3 core columns we'll always be using (id, created, modified) plus additional columns for the incoming API data

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

#3 Clearly caption screenshots

Task #2 - Points: 1 Text: Explain the design

Checklist

*The checkboxes are for your own tracking

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

Response:

The fields used for the game include the id, name, description, OpenCritic URL, in-game screenshot URL, square image URL, release date, top critic score, publisher, and developer from the API. The data such as the name, description, URL, and square image are mainly going to be used for single item view as a way for the user to see more about a game. Information like the top critic score, release date, publisher, and developer are going to be used for the game part of the app as a way to give hints about which game it is. Other fields such as is_api, is_active, date_created, and date_modified are used for housekeeping purposes such as seeing if it's a custom game, soft deleting, and records. Everything would be easily pulled from the API except the image URLs and companies related to the game, which were embedded under a layer in the JSON. As the game data can only be obtained using an ID, the popular games request would fill some of the games with their API IDs and scores first, and then later the rest of the data would be lazy-loaded.

loaded.

There were also the genres and platforms correlated to the game. Due to the want of comparing the data later on and attempting to make the data third normal form, these would be put in their own tables. They would just have the API ID and a name. These would have relation tables built with the game which would correlate the games to their genres and the games to their platforms. As a result, the data from the JSON would need to be parsed and separated into the genre and platforms where they would further be split into the 4 additional tables.

Task #3 - Points: 1

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

Details:

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

URL #1

<https://github.com/WokFriedE/ekh3-it202-008/pull/52>

Data Creation Page (2 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the creation page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show potentially valid data filled in for the custom creation page
<input type="checkbox"/> #2	1	Show how the API data is fetched for API data (must be server-side)
<input type="checkbox"/> #3	1	Show examples of validation messages
<input type="checkbox"/> #4	1	Show an example of successful creation message
<input type="checkbox"/> #5	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
<input type="checkbox"/> #6	1	Make sure the heroku dev url is visible in the address bar
<input type="checkbox"/> #7	1	Clearly caption screenshots

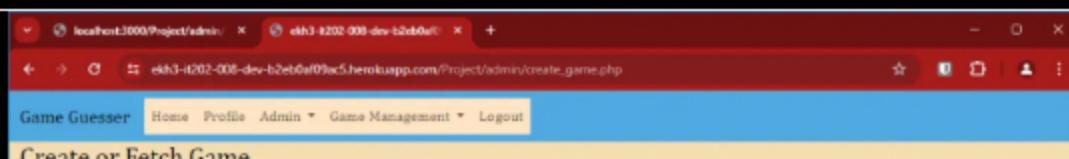
Task Screenshots:

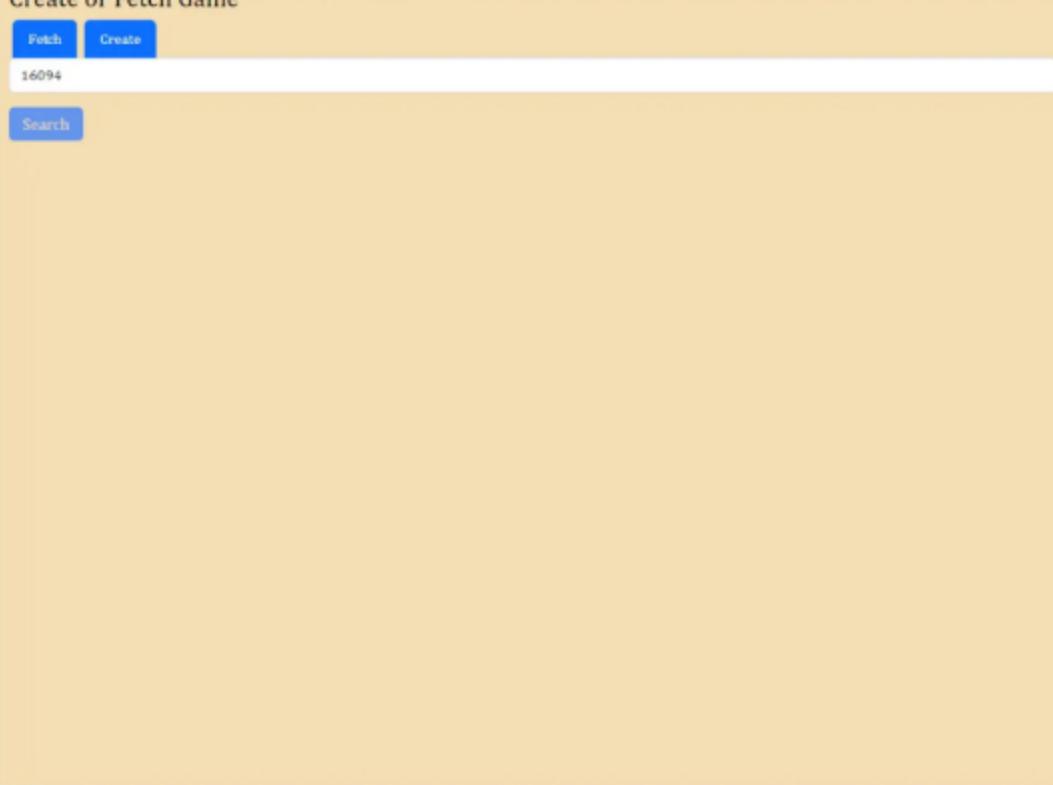
Gallery Style: Large View

Small

Medium

Large





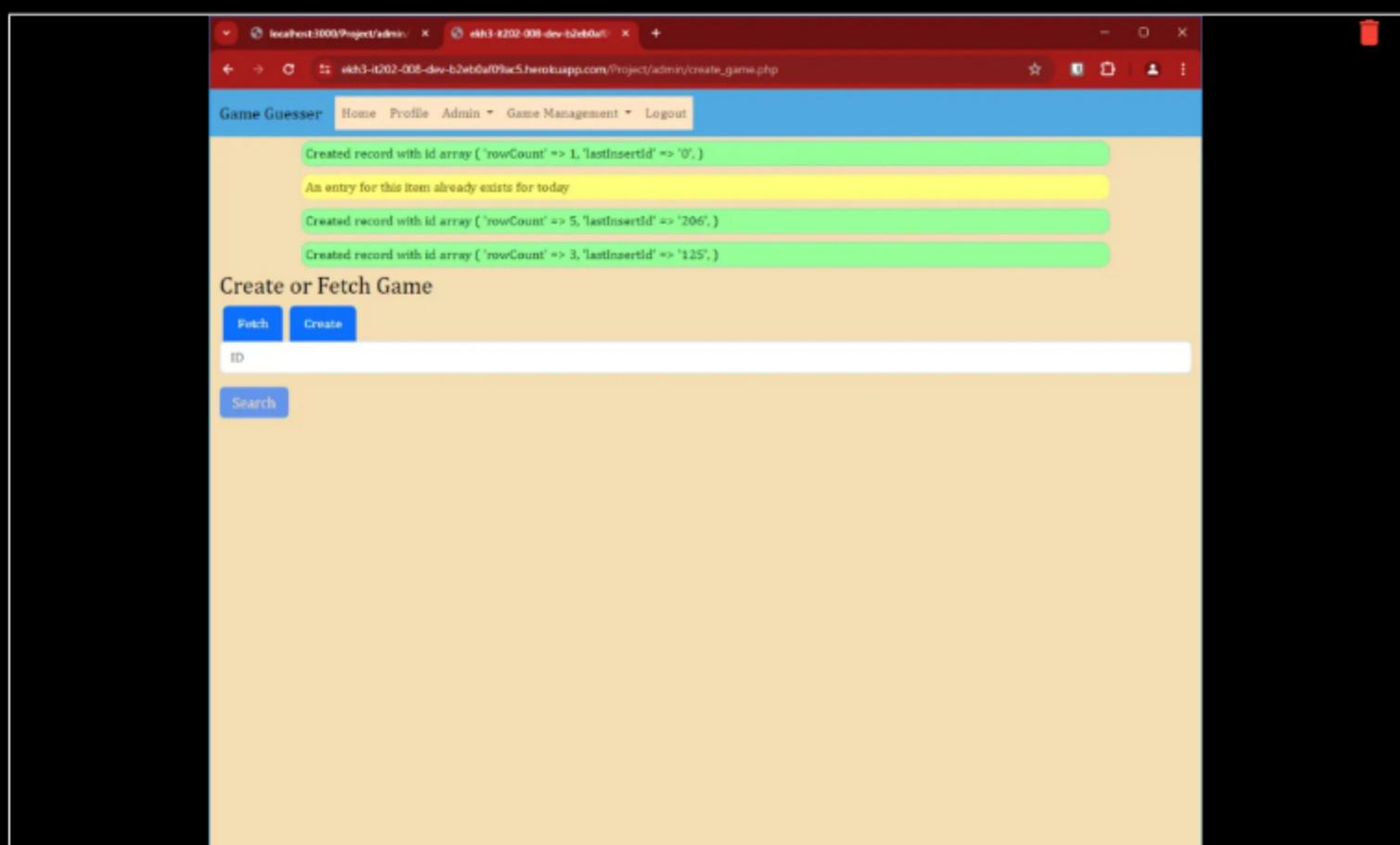
Fetch example - before

Checklist Items (3)

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

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

#7 Clearly caption screenshots



Fetch game - after fetch (in this case it was an update as the ID existed)

Checklist Items (3)

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

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

#7 Clearly caption screenshots

localhost:3003/Project/admin/ × ekh3-it202-008-dev-b2eb0af09ac5.herokuapp.com/Project/admin/create_game.php#

Game Guesser Home Profile Admins Game Management Logout

Create or Fetch Game

Fetch Create

Game ID
10

Name
A New Game

Publisher
Publisher (optional)

Developer
Ethan Ho

Description
It's a game for the gods, gameogenesis one night city

Critic Score
98

Release Date
04/13/2024

Square Image
Square Image URL (optional)

Screenshot Image
Screenshot Image URL (optional)

Game Page URL
Game Page (URL (optional))

Platforms

Xbox Series X/S (inactive)
 PlayStation 5 (inactive)
 PlayStation 4 (inactive)

Genres

Fighting (inactive)
 Action (inactive)
 Simulation (inactive)

Game creation - potentially valid data part 1 with CSS

Checklist Items (4)

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

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

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

#7 Clearly caption screenshots

localhost:3003/Project/admin/ × ekh3-it202-008-dev-b2eb0af09ac5.herokuapp.com/Project/admin/create_game.php#

Game Guesser Home Profile Admins Game Management Logout

Create or Fetch Game

Fetch Create

Publisher
Publisher (optional)

Developer
Ethan Ho

Description

Make a game for the games_guesser.com website

Critic Score
40

Release Date
2023-01-01

Square Image
Square Image URL (optional)

Thumbnail Image
Thumbnail Image URL (optional)

Game Page URL
Game Page URL (optional)

Platforms

- Xbox Series X/S (optional)
- PlayStation 5 (optional)
- PlayStation 4 (optional)
- Xbox One (optional)
- VR/VR (optional)
- PC (optional)
- Nintendo Switch (optional)
- PlayStation Vita (optional)
- Nintendo 3DS (optional)
- Google Stadia (optional)

Genres

- Fighting (optional)
- Action (optional)
- Simulation (optional)
- Adventure (optional)
- RPG (optional)
- Turn-Based Strategy (optional)
- Platformer (optional)
- Racing (optional)
- Real-Time Strategy (optional)
- Sports (optional)
- Strategy (optional)
- Third-Person Shooter (optional)
- Turn-Based Strategy (optional)
- Vehicle Combat (optional)
- Art (optional)
- Interactive Story (optional)
- Arcade (optional)
- Horror (optional)
- Roguelike (optional)
- Puzzles (optional)
- Indie (optional)
- MMORPG (optional)
- Survival (optional)
- Casual (optional)

Submit

Game creation - potentially valid data part 2

Checklist Items (4)

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

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

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

#7 Clearly caption screenshots

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

- Address Bar:** location:12000/Project/admin (localhost) | eth3-0202-008-dev-b2eb0f09c5.herokuapp.com/Project/admin/create_game.php#
- Page Title:** Game Guesser
- Navigation:** Home, Profile, Admin, Game Management, Logout
- Form Fields (with validation errors):**
 - Name: Name (Client) invalid score please enter a number
 - Publisher: Publisher (Client) Developer is required
 - Developer: Developer (Client) Developer is required
 - Description: Description (Client) Description is required
 - Release Date: Release Date (Client) invalid date please enter one like YYYY-MM-DD
- Buttons:** Reck, Create

Screenshot Image
Screenshot Image URL (optional)
Game Page URL

Game creation - validation examples

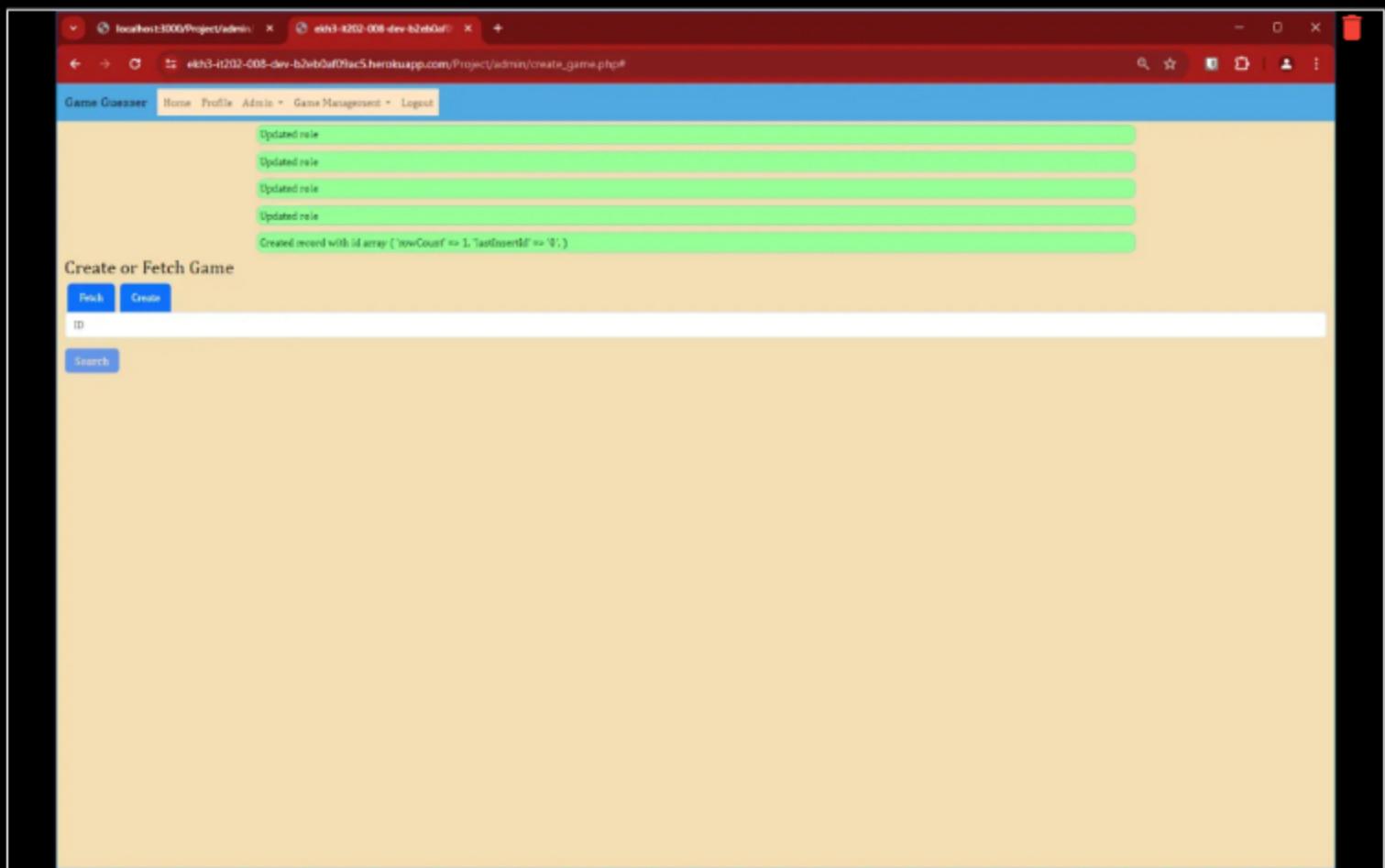
Checklist Items (4)

#3 Show examples of validation messages

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

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

#7 Clearly caption screenshots



Game creation - completed insert

Checklist Items (4)

#4 Show an example of successful creation message

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

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

#7 Clearly caption screenshots

Task #2 - Points: 1

Text: Screenshots of creation page code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Form should have correct data types for each property being requested
#2	1	Form should have correct validation for each field (HTML, JS, and PHP)
#3	1	Successful creation should have a user-friendly message
#4	1	Any errors should have user-friendly messages
#5	1	Include the form/process for fetching API data
#6	1	Include some indicator between custom data and API data
#7	1	Include any other rules like role guards and login checks
#8	1	Include ucid/date comments for each code screenshot
#9	1	Clearly caption screenshots

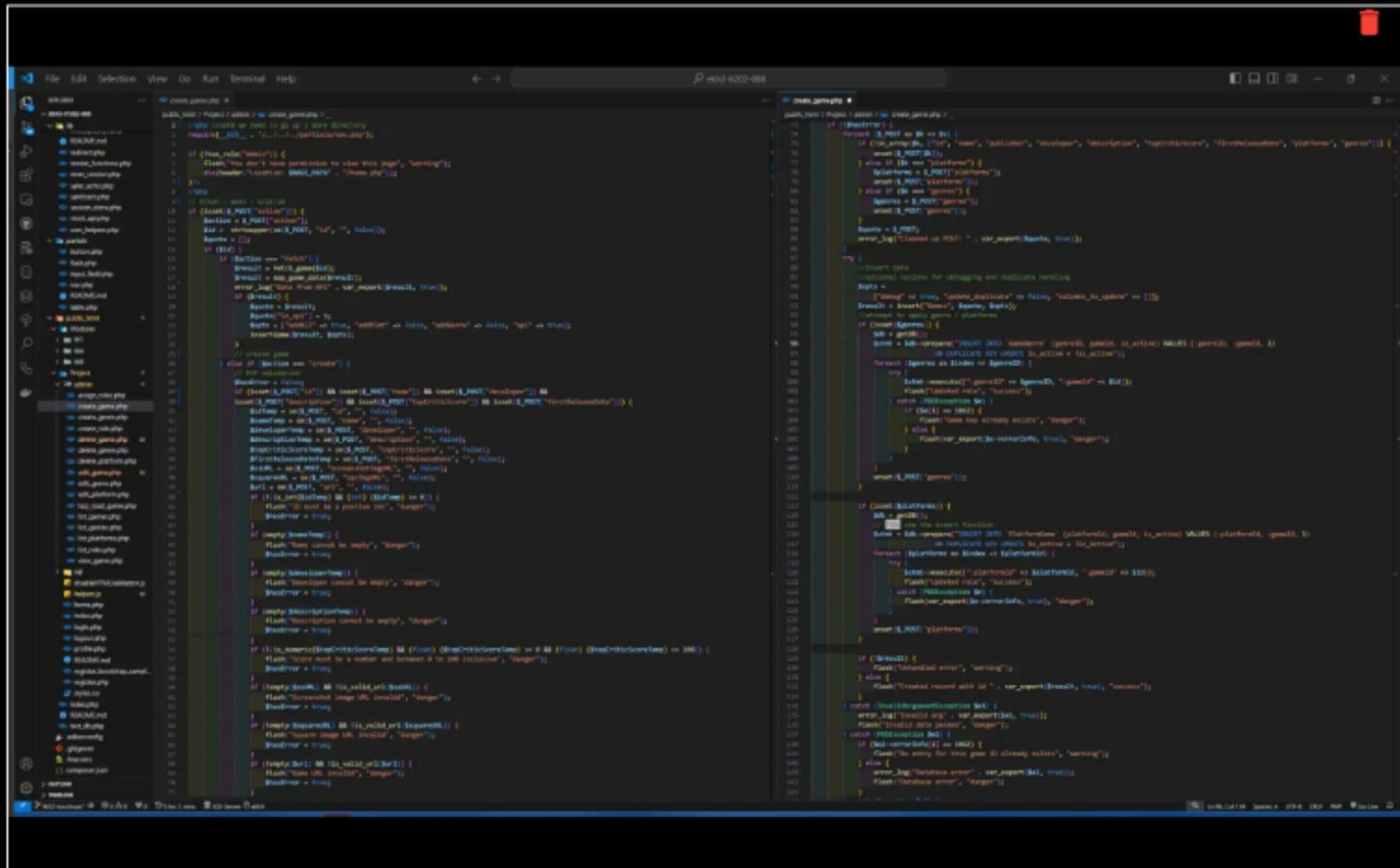
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



The image shows a code editor with two tabs open, both displaying PHP code related to game creation logic. The left tab contains a large block of PHP code with numerous conditional statements, loops, and database interaction code. The right tab contains a smaller block of PHP code, likely a controller or service layer, which interacts with the database and handles validation logic. Both tabs show syntax highlighting and code completion features.

PHP code for create game - includes admin check, PHP validations, and DB interaction, including the status messages for fail and complete on right side

Checklist Items (4)

#6 Include some indicator between custom data and API data

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

#8 Include ucid/date comments for each code screenshot

#9 Clearly caption screenshots

Create game - HTML + form + JS validation (please refer to list view code for PHP code for generating partials)

Checklist Items (4)

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

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

#8 Include ucid/date comments for each code screenshot

#9 Clearly caption screenshots

PHP API fetch functions

Checklist Items (3)

#5 Include the form/process for fetching API data

#8 Include ucid/date comments for each code screenshot

#9 Clearly caption screenshots

Game mapping used for insertion and load api key on bottom right

Checklist Items (3)

#5 Include the form/process for fetching API data

#8 Include ucid/date comments for each code screenshot

#9 Clearly caption screenshots

DB insertion helper (provided)

Checklist Items (3)

#5 Include the form/process for fetching API data

#8 Include ucid/date comments for each code screenshot

#9 Clearly caption screenshots

```

    if (isMySQLError($error)) {
        $msg = "MySQL error: " . $error->getMessage();
        $log = "MySQL error: " . $error->getLine() . " - " . $error->getFile();
        $logger->warning($msg);
        $logger->error($log);
    } else {
        $msg = "Database error: " . $error->getMessage();
        $log = "Database error: " . $error->getLine() . " - " . $error->getFile();
        $logger->warning($msg);
        $logger->error($log);
    }
}

// Insertion logic
function insert($table, $data, $log = false) {
    $query = "INSERT INTO $table ($data) VALUES ('" . implode("', '", $data) . "')";
    $result = mysqli_query($connection, $query);
    if (!$result) {
        $error = mysqli_error($connection);
        $msg = "MySQL error: " . $error;
        $log = "MySQL error: " . $error . " - " . $connection->getLine() . " - " . $connection->getFileName();
        $logger->warning($msg);
        $logger->error($log);
    } else {
        $msg = "MySQL record inserted successfully";
        $log = "MySQL record inserted successfully";
        $logger->info($msg);
        $logger->info($log);
    }
}

// Insertion logic
function insertSubsystems($table, $data, $log = false) {
    $query = "INSERT INTO $table ($data) VALUES ('" . implode("', '", $data) . "')";
    $result = mysqli_query($connection, $query);
    if (!$result) {
        $error = mysqli_error($connection);
        $msg = "MySQL error: " . $error;
        $log = "MySQL error: " . $error . " - " . $connection->getLine() . " - " . $connection->getFileName();
        $logger->warning($msg);
        $logger->error($log);
    } else {
        $msg = "MySQL record inserted successfully";
        $log = "MySQL record inserted successfully";
        $logger->info($msg);
        $logger->info($log);
    }
}

```

DB helper which houses insertion subsets for creation page

Checklist Items (1)

#5 Include the form/process for fetching API data

```

function verifyEmail($email) {
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $msg = "Email is not valid";
        $log = "Email is not valid";
        $logger->warning($msg);
        $logger->error($log);
    } else {
        $msg = "Email is valid";
        $log = "Email is valid";
        $logger->info($msg);
        $logger->info($log);
    }
    return $msg;
}

function verifyPassword($password) {
    if (!preg_match('/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[\W\w]).{8,}$/', $password)) {
        $msg = "Password must be at least 8 characters, have 1 uppercase, 1 number, and 1 special character";
        $log = "Password must be at least 8 characters, have 1 uppercase, 1 number, and 1 special character";
        $logger->warning($msg);
        $logger->error($log);
    } else {
        $msg = "Password is valid";
        $log = "Password is valid";
        $logger->info($msg);
        $logger->info($log);
    }
    return $msg;
}

function verifyName($name) {
    if (!is_string($name)) {
        $msg = "Name must be a string";
        $log = "Name must be a string";
        $logger->warning($msg);
        $logger->error($log);
    } else {
        $msg = "Name is valid";
        $log = "Name is valid";
        $logger->info($msg);
        $logger->info($log);
    }
    return $msg;
}

function verifyUcid($ucid) {
    if (!is_string($ucid)) {
        $msg = "UCID must be a string";
        $log = "UCID must be a string";
        $logger->warning($msg);
        $logger->error($log);
    } else {
        $msg = "UCID is valid";
        $log = "UCID is valid";
        $logger->info($msg);
        $logger->info($log);
    }
    return $msg;
}

function verifyDate($date) {
    if (!is_string($date)) {
        $msg = "Date must be a string";
        $log = "Date must be a string";
        $logger->warning($msg);
        $logger->error($log);
    } else {
        $msg = "Date is valid";
        $log = "Date is valid";
        $logger->info($msg);
        $logger->info($log);
    }
    return $msg;
}

function verifyMobile($mobile) {
    if (!is_string($mobile)) {
        $msg = "Mobile must be a string";
        $log = "Mobile must be a string";
        $logger->warning($msg);
        $logger->error($log);
    } else {
        $msg = "Mobile is valid";
        $log = "Mobile is valid";
        $logger->info($msg);
        $logger->info($log);
    }
    return $msg;
}

```

Validations for JS and PHP

Checklist Items (3)

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

#8 Include ucid/date comments for each code screenshot

Task #3 - Points: 1

Text: Screenshot of records from DB

Checklist

*The checkboxes are for your own tracking

#	Points	Details
■ #1	1	Show at least one record fetched from the API
■ #2	1	Show at least one record created via the creation form
■ #3	1	Note what differs
■ #4	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

API fetched data - filled data is pulled directly with a fetch, those with null are pulled from a button on list view for lazy loading. API is indicated by is_api field with 1 being yes.

Checklist Items (3)

#1 Show at least one record fetched from the API

#3 Note what differs

#4 Clearly caption screenshots

	id	name	version	publisher	developer	description	topCollectionVersion	imgURL	screenshotURL	url	firstReleaseDate	is_api	is_active	createdTimestamp	modifiedTimestamp
1	1	Testing Test	1	Ethan	test	1	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-20	0	1	2024-04-18 00:02:50.00	2024-04-18 00:11:29
2	2	NODONODON	2	Ethan	something random	2	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-21	0	1	2024-04-21 03:16:25	2024-04-21 03:16:25
3	3	Name Test	3	Ethan	Test	3	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-21	0	1	2024-04-21 03:17:52	2024-04-21 03:17:52
4	4	Name Test 2	4	Ethan	Sequel to Name Test 3	7	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-21	0	0	2024-04-21 03:18:18	2024-04-21 03:18:18
5	5	Name Test 3	5	Ethan	Sequel to Name Test 4	7	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-21	0	1	2024-04-21 03:19:08	2024-04-21 03:19:08
6	11	Hello Friends	80	Ethan	Its a game about friends	80	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-12	0	1	2024-04-21 03:23:37	2024-04-21 03:23:37
7	85213	Game Guesser	No-one	Ethan	Its a game for guessing per	100	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-30	0	1	2024-04-17 00:11:13	2024-04-17 00:11:13
8	90000000	TestWorld	Ethan	Ethan	Its not a game	10	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-17	0	1	2024-04-17 11:05:01	2024-04-17 11:05:01
9	11111111	TestWorld	NOT	Ethan	Its a game	10	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-16	0	1	2024-04-17 00:45:24	2024-04-17 00:45:24
10	709086705	wklo	Randomser	Ethan	holleache	10	(NULL)	(NULL)	(NULL)	(NULL)	2024-04-17	0	1	2024-04-17 03:29:36	2024-04-17 03:29:36

Custom data - non API is indicated by **is_api** field with 0 being no. Some fields are missing as image urls are null (as I have not chosen any images in this snapshot)

Checklist Items (3)

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

#3 Note what differs

#4 Clearly caption screenshots

Task #4 - Points: 1

Text: Explain the process

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Provide a high-level step-by-step of how fetching API data works and gets added to your DB (include how duplicates are handled)
<input checked="" type="checkbox"/> #2	-	Provide a high-level step-by-step of how creating custom data works and gets added to your DB

<input checked="" type="checkbox"/> #2	1	added to your DB (include how duplicates are handled)
<input checked="" type="checkbox"/> #3	1	Briefly describe the validations for the applicable fields
<input checked="" type="checkbox"/> #4	1	Describe how duplicate data is handled

Response:

To get the information from the database, when the admin calls for the popular games, the database would input the information for the API IDs (the primary key), names, and top critic score. No parsing is needed. It would also generate information such as the fact that it is an API and timestamps. Later when the admin views one of the games the rest of the data will be pulled using the id from the popular pull. This would take in the rest of the information as an associative array where all the images are in one location, the simple data as just tags, the companies associated, the genres, and the platforms. Any simple data such as release and description will be pulled directly. For the release date, the date would need to be parsed to just get the date and turn it into a simple date format by sub-stringing it. For the images, I would follow the JSON and pull out the original image link for the square image and a screenshot. These would be concatenated with a static URL which provides the image. The companies would be parsed by the developer and the publisher. By following the JSON, under companies, there exists a list of the companies with their type.

Therefore, under the assumption that there is only one of each, the attributes are populated that way. Lastly, for the genres and platforms, the calculations were roughly the same where they would have their association that has a list of items with the names and IDs. Each item would need to be parsed and put into the respective databases with their relations. It is also possible to pull all the platforms or genres directly which would provide lists that can be iterated to get the names and ids. When getting the data needed from the API, it would search for the items provided and ignore or unset the rest.

Creating custom data would allow the user to put in all the info they would like from the database, some of which will not be required and are marked in their placeholders as so. The user would also be able to choose the platforms and genres the game applies to. The same queries used to insert the API data would insert this data but set the is_api flag to false.

In this case, not much validation was needed as they were just pulled directly or had the information already needed. If the length of an attribute is exceeded, despite giving much room, then a PDO exception would occur to prevent the insert. Any duplicate would be treated as an update and provide newer information to the database. The primary key is only the API ID, so the games can have similar names.

For client-side input validation, it would enforce that some fields are required and not blank such as name, description, developer, date, top critic score, and ID. The form would guarantee the date is in YYYY-MM-DD format for SQL. For the critic score, it would check if it is a valid number. The ID would make sure it is a positive number and below SQL's integer limit.

Task #5 - Points: 1

Text: Add related links

Checklist			*The checkboxes are for your own tracking
#	Points	Details	
<input checked="" type="checkbox"/> #1	1	Include the heroku prod link for this page	
<input checked="" type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature	

URL #1

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

URL #2

<https://github.com/WokFriedE/ekh3-it202-008/pull/53>

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

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the list page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show the page of your entities listed (have a reasonable number shown)
<input type="checkbox"/> #2	1	Show the filter/sort form based on your data and the required limit field
<input type="checkbox"/> #3	1	Demonstrate a few varied filters/sorts
<input type="checkbox"/> #4	1	Demonstrate a filter that doesn't have any records (should show an appropriate message)
<input type="checkbox"/> #5	1	Each list item should have a link of single view (i.e., details), edit, and delete (some of which may only be visible to admin users)
<input type="checkbox"/> #6	1	Each list item should have a summary of the entity (likely won't be the entire entity data)
<input type="checkbox"/> #7	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
<input type="checkbox"/> #8	1	Make sure the heroku dev url is visible in the address bar
<input type="checkbox"/> #9	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a web-based game management system. At the top, there's a navigation bar with links for Home, Profile, Admin, Game Management, and Logout. Below the navigation is a search bar and a filter section with dropdowns for Game ID, Game Name, Publisher, Developer, Score Min, Score Max, Min Date, Max Date, Sort, Order, Set Disabled, and Limit (set to 10). There are also buttons for Filter and Clear. Below the filter is a row of buttons for Full Popular Games, Full Details, and Full Platforms. The main area is titled "List Games" and contains a table with columns: id, name, publisher, developer, top score, release date, Is API, Active, created, modified, and Actions. The table lists several games, such as "Testing Roff" (id 1, developer Ethan, top score 1, release date 2020-05-20), "NOMONONO" (id 2, developer Ethan, top score 2, release date 2024-04-21), "Name test" (id 3, developer Ethan, top score 5, release date 2024-04-21), "Name test 3" (id 5, developer Ethan, top score 7, release date 2024-04-21), "Hello Friends" (id 11, developer Ethan, top score 60, release date 2024-04-12), "Hi-Fi Radio" (id 16227, developer Bethesda Softworks, top score 88.7562, release date 2023-01-25), "Loretta" (id 14303, developer DANCE! Entertainment, top score 76.25, release date 2023-02-16), and "ArcRanger" (id 14910, developer PO Ashe, top score 76.75, release date 2023-04-27). Each game entry has a "View", "Edit", and "Toggle Active" button in the Actions column.

id	name	publisher	developer	top score	release date	Is API	Active	created	modified	Actions
1	Testing Roff		Ethan	1	2020-05-20	No	Active	2024-04-18 02:59:29	2024-04-18 03:31:29	View Edit Toggle Active
2	NOMONONO		Ethan	2	2024-04-21	No	Active	2024-04-21 07:10:25	2024-04-21 07:10:25	View Edit Toggle Active
3	Name test		Ethan	5	2024-04-21	No	Active	2024-04-21 07:17:52	2024-04-21 07:17:52	View Edit Toggle Active
5	Name test 3		Ethan	7	2024-04-21	No	Active	2024-04-21 07:19:00	2024-04-21 07:19:00	View Edit Toggle Active
11	Hello Friends		Ethan	60	2024-04-12	No	Active	2024-04-21 06:23:37	2024-04-21 06:23:37	View Edit Toggle Active
16227	Hi-Fi Radio	Bethesda Softworks	Tango Gameworks	88.7562	2023-01-25	Yes	Active	2024-04-15 22:26:47	2024-04-18 00:29:45	View Edit Toggle Active
14303	Loretta	DANCE! Entertainment	Yakov Batszoff	76.25	2023-02-16	Yes	Active	2024-04-15 22:26:47	2024-04-16 22:02:37	View Edit Toggle Active
14910	ArcRanger	PO Ashe	Trickster Games Ltd	76.75	2023-04-27	Yes	Active	2024-04-16 00:16:29	2024-04-18 00:29:45	View Edit Toggle Active

ID	Name	Publisher	Developer	Score Min	Score Max	Min Date	Max Date	Sort	Order	See Disabled	Last	
14935	Inbound	Shiny Shoe	Shiny Shoe	81,4286	2023-05-23	Yes	Active	2024-04-15 22:26:47	2024-04-16 22:31:21	View	Edit	Toggle Active
14938	The Outlast Trials	Red Barrels	Red Barrels	72,0263	2023-05-10	Yes	Active	2024-04-15 22:26:47	2024-04-16 22:03:43	View	Edit	Toggle Active
14952	Planet of Lava	Thunderful Publishing	Witchify	81,1585	2023-05-23	Yes	Active	2024-04-16 18:56:52	2024-04-16 22:31:40	View	Edit	Toggle Active
15136	Final Fantasy VII Remake	Square Enix	Square Enix	92,5455	2024-02-29	Yes	Active	2024-04-15 22:26:47	2024-04-16 07:43:47	View	Edit	Toggle Active
15151	Bare the River	MINTROCKET	MINTROCKET	09,5595	2023-06-20	Yes	Active	2024-04-15 22:26:47	2024-04-16 00:29:45	View	Edit	Toggle Active
15236	Let's! Revolution!	Hawthorn Games	Astfield	86,6	2023-07-19	Yes	Active	2024-04-15 22:26:47	2024-04-17 01:58:58	View	Edit	Toggle Active
15497	Princess Peach Showtime!			75,2034		Yes	Active	2024-04-15 22:26:47	2024-04-18 03:26:57	View	Edit	Toggle Active

List page for Games with no sort or filter

Checklist Items (6)

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

#5 Each list item should have a link of single view (i.e., details), edit, and delete (some of which may only be visible to admin users)

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

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

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

#9 Clearly caption screenshots

List games view - when the result has no records

Checklist Items (1)

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

ID	Name	Publisher	Developer	Top Score	Release Date	Is API	Active	Created	Modified	Actions
11	Hello Friends	Ethan		80	2024-04-12	No	Active	2024-04-15 06:22:37	2024-04-21 09:22:37	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
14910	ArchRunner	FQuake	TrickJump Games Ltd	78.75	2023-04-27	Yes	Active	2024-04-16 00:16:29	2024-04-18 09:29:45	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
14958	The Outlast Trials	Red Barrels	Red Barrels	72.0283	2025-05-18	Yes	Active	2024-04-15 22:26:47	2024-04-16 22:05:43	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
16557	Planet Zoo Console Edition	Frontier Developments	Frontier Developments	78.125	2024-03-26	Yes	Active	2024-04-15 22:26:47	2024-04-21 07:42:44	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
16574	Spirit City: Lost Sessions	Mooncube Games	Mooncube Games	78.5	2024-04-03	Yes	Active	2024-04-15 22:26:47	2024-04-17 01:57:11	<button>View</button> <button>Edit</button> <button>Toggle Action</button>

List games sort example

Checklist Items (3)

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

#3 Demonstrate a few varied filters/sorts

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

ID	Name	Publisher	Developer	Top Score	Release Date	Is API	Active	Created	Modified	Actions
15724	Tekken 8	Bandai Namco Entertainment	Bandal Namco Studios Inc.	89.7469	2024-01-25	Yes	Active	2024-04-21 09:04:04	2024-04-21 09:04:04	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
16094	Persona 5 Reload	SEGA	ATLUS	87.7812	2024-02-02	Yes	Active	2024-04-21 09:16:22	2024-04-21 09:16:22	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
16309	Balstro	Playstack	LocalThunk	91.375	2024-02-20	Yes	Active	2024-04-21 09:08:03	2024-04-21 09:08:03	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
15136	Final Fantasy VII Rebirth		Square Enix	92.5455	2024-02-29	Yes	Active	2024-04-15 22:26:47	2024-04-16 09:43:47	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
16429	Star Wars: Battlefront Classic Collection	Aspyr		69.8909	2024-05-14	Yes	Active	2024-04-15 22:26:47	2024-04-16 21:48:43	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
16050	Alone in the Dark (2024)	THQ Nordic		61.8278	2024-03-20	Yes	Active	2024-04-15 22:26:47	2024-04-18 06:35:55	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
16051	Dragon's Dogma 2	Capcom U.S.A., Inc.	CAPCOM Co., Ltd.	87.3704	2024-03-22	Yes	Active	2024-04-15 22:26:47	2024-04-16 22:03:14	<button>View</button> <button>Edit</button> <button>Toggle Action</button>
16489	South Park: Snow Day	THQ Nordic	Cartoon Network	59.68	2024-05-26	Yes	Active	2024-04-15 22:26:47	2024-04-16 01:15:34	<button>View</button> <button>Edit</button> <button>Toggle Action</button>

ID	Product Name	Developer	Publisher	Release Date	Last Update	Status	Activation Count	Last Activation	View	Edit	Action	
16557	Frostbit Loco Session Edition	Frontier Developments	Frontier Developments	2024-03-29	2024-03-29	Yes	Active	2024-04-15 22:26:47	2024-04-21 01:42:14	<button>View</button>	<button>Edit</button>	<button>Toggle Active</button>
16574	Spirit City: Lofi Sessions	Mooncycle Games	Mooncycle Games	2024-04-09	2024-04-09	Yes	Active	2024-04-15 22:26:47	2024-04-17 01:51:12	<button>View</button>	<button>Edit</button>	<button>Toggle Active</button>
16554	Gigantic Rampage Edition	Gearbox Publishing	Abstractive Games	2024-04-09	2024-04-09	Yes	Active	2024-04-18 00:26:45	2024-04-18 00:26:45	<button>View</button>	<button>Edit</button>	<button>Toggle Active</button>
11	Hello Friends	Ethan	Ethan	2024-04-12	2024-04-12	No	Active	2024-04-21 00:23:37	2024-04-21 00:23:37	<button>View</button>	<button>Edit</button>	<button>Toggle Active</button>
11111111	TestWorld	NJIT	Ethan	2024-04-16	2024-04-16	No	Active	2024-04-17 00:45:24	2024-04-17 00:45:24	<button>View</button>	<button>Edit</button>	<button>Toggle Active</button>
789098765	hello	Customer	Ethan	2024-04-17	2024-04-17	No	Disabled	2024-04-17 18:29:38	2024-04-21 20:31:32	<button>View</button>	<button>Edit</button>	<button>Toggle Active</button>
9999999	TestWorld		Ethan	2024-04-17	2024-04-17	No	Active	2024-04-17 17:42:01	2024-04-17 17:42:01	<button>View</button>	<button>Edit</button>	<button>Toggle Active</button>

List games - Filter sort by release date and allow disabled games

Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

Task #2 - Points: 1

Text: Screenshots of the list page code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large

PHP for list games, shows admin requirement, buttons for fetching general data from the api, filter field inputs, and beginning of the query

Checklist Items (4)

#1 Show the filter/sort form generation

#4 Show any restrictions like role guard or login checks

#5 Include ucid/date comments for each code screenshot

#6 Clearly caption screenshots

List games - PHP part 2, includes second half of query with filter and sorts, execution of the DB query, and HTML + PHP processing of data

Checklist Items (5)

#1 Show the filter/sort form generation

#2 Show the PR query and how the filter/cort is handled (including the restriction on the limit field).

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

#3 Show how the output is generated and displayed

#5 Include ucid/date comments for each code screenshot

#6 Clearly caption screenshots

PHP partials that help generate the buttons and inputs

Checklist Items (1)

#3 Show how the output is generated and displayed

```

    // gameList.php
    $query = "SELECT * FROM games ORDER BY id ASC";
    $result = mysqli_query($conn, $query);
    if ($result) {
        $games = mysqli_fetch_all($result, MYSQLI_ASSOC);
        echo json_encode($games);
    } else {
        echo "Error: " . mysqli_error($conn);
    }
}

// filterByDeveloper.php
$developer = $_GET['developer'];
$query = "SELECT * FROM games WHERE developer = '$developer' ORDER BY id ASC";
$result = mysqli_query($conn, $query);
if ($result) {
    $games = mysqli_fetch_all($result, MYSQLI_ASSOC);
    echo json_encode($games);
} else {
    echo "Error: " . mysqli_error($conn);
}

// search.php
$searchTerm = $_GET['search'];
$query = "SELECT * FROM games WHERE name LIKE '%$searchTerm%' ORDER BY id ASC";
$result = mysqli_query($conn, $query);
if ($result) {
    $games = mysqli_fetch_all($result, MYSQLI_ASSOC);
    echo json_encode($games);
} else {
    echo "Error: " . mysqli_error($conn);
}

```

```

    // gameList.php
    $query = "SELECT * FROM games ORDER BY id ASC";
    $result = mysqli_query($conn, $query);
    if ($result) {
        $games = mysqli_fetch_all($result, MYSQLI_ASSOC);
        echo json_encode($games);
    } else {
        echo "Error: " . mysqli_error($conn);
    }
}

// filterByDeveloper.php
$developer = $_GET['developer'];
$query = "SELECT * FROM games WHERE developer = '$developer' ORDER BY id ASC";
$result = mysqli_query($conn, $query);
if ($result) {
    $games = mysqli_fetch_all($result, MYSQLI_ASSOC);
    echo json_encode($games);
} else {
    echo "Error: " . mysqli_error($conn);
}

// search.php
$searchTerm = $_GET['search'];
$query = "SELECT * FROM games WHERE name LIKE '%$searchTerm%' ORDER BY id ASC";
$result = mysqli_query($conn, $query);
if ($result) {
    $games = mysqli_fetch_all($result, MYSQLI_ASSOC);
    echo json_encode($games);
} else {
    echo "Error: " . mysqli_error($conn);
}

```

PHP helper to help build the table and its outputs

Checklist Items (1)

#3 Show how the output is generated and displayed

Task #3 - Points: 1

Text: Explain how the page works

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Provide the high-level steps of how the filter/sorting works and how data is displayed
<input checked="" type="checkbox"/> #2	1	Summarize what you're showing on the screen
<input checked="" type="checkbox"/> #3	1	Mention your design/style choice
<input checked="" type="checkbox"/> #4	1	Mention which users can interact with the view, edit, and delete links

Response:

The sort and filtering would be based on a single query. This query would allow the user to put the data in specified fields and would tack on more filters or aspects of the query. Most of the attributes would check if there were any data specific in the URL from a GET request. Otherwise, the field would be ignored, or a default value would be set. For fields like the name, publisher, or developer they would use like clauses to show games that contain the text sent. The score could have a minimum and maximum set to limit the score value. Similarly, the date would be set as so. The user can also sort by score or release date in either an ascending or descending order. The user can view the games that have been soft disabled or hide them by checking for the active flag. The user would be able to limit the number of responses from the query and notify them if the amount is exceeded. On the screen, there would be the fields that can be edited, a few buttons for pulling new data and searching, and the table. The table would include the name, publisher, developer, top critic score, release date if it's an API, if it's active, and the creation/modification date. Each record returned is given its own row and has a view, edit, and delete button which sends the user to the page with the game ID. As this page is only visible to the admin, those pages are integrated for the admin only for now.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

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

URL #2

<https://github.com/WokFriedE/ekh3-it202-008/pull/53>

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

▲COLLAPSE ▲

● Task #1 - Points: 1

Text: Screenshots of the details page

Checklist

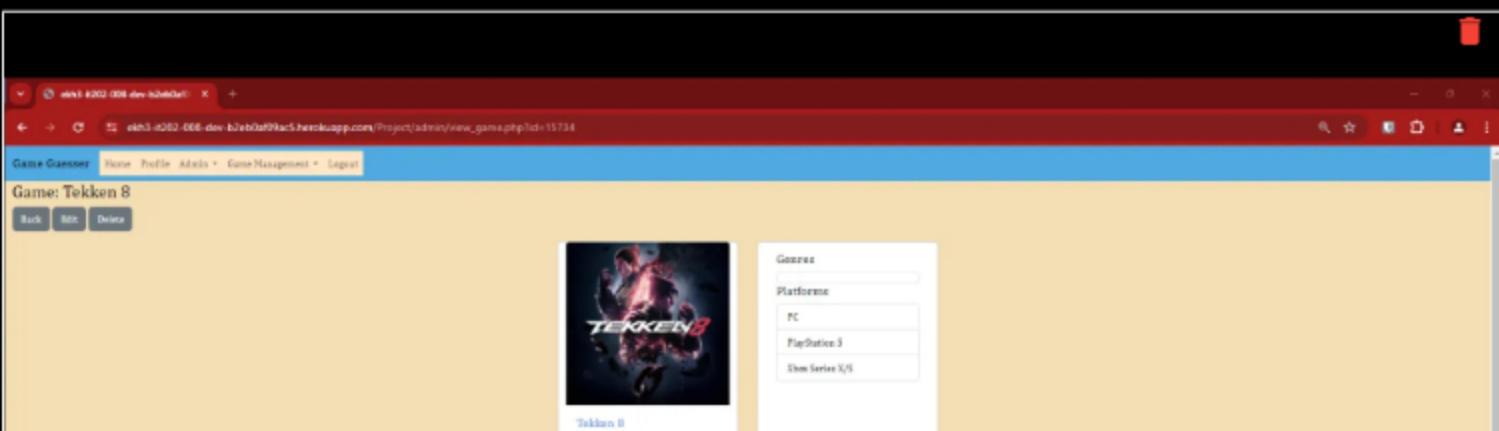
*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Entity should be fetch by id (via the url)
<input type="checkbox"/> #2	1	A missing id should redirect back to the list page with an applicable message
<input type="checkbox"/> #3	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
<input type="checkbox"/> #4	1	Data shown should be more detailed/inclusive than the summary view
<input type="checkbox"/> #5	1	There should be a link to edit the entity (this may be an admin-only thing, but it should be present for the respective role)
<input type="checkbox"/> #6	1	There should be a link to delete the entity (this may be an admin-only thing, but it should be present for the respective role)
<input type="checkbox"/> #7	1	Make sure the heroku dev url is visible in the address bar
<input type="checkbox"/> #8	1	Clearly caption screenshots

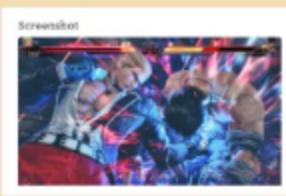
Task Screenshots:

Gallery Style: Large View

Small Medium Large



Publisher: Bandai Namco Entertainment
Developer: Bandai Namco Studios Inc.
Top Critic Score: 89.7444
Release Date: 2024-01-25



Description

GET READY FOR THE NEXT CHAPTER IN THE LEGENDARY FIGHTING GAME FRANCHISE, TEKKEN 8. • 32 FIGHTERS FOR A NEW GENERATION Completely redesigned character visuals. Elaborate, highly-detailed models built from the ground up and high-fidelity graphics break the limits of new-generation hardware by adding a new weight and atmosphere to TEKKEN's signature battles. Vivid environments and destructible stages combine to create an overwhelming sense of immersion, creating the ultimate play experience. • NEW GAME, NEW RIVALRY For Meets Rule In TEKKEN 8. Holding a record for the longest-running video game storyline, the TEKKEN series begins a new chapter! TEKKEN 8 contains the tragic saga of the Matsuda and Kazama blood loss, and their world-shaking father-and-son grudge matches starting from 8 months after the closure of the last match. The story of Jin Kazama's growth and determination marks a new chapter in the timeless saga. • THKILLING OVER THE TOP BATTLES EMBODDING AGGRESSION & DESTRUCTION The new battle system, Heat, dial up the aggressive nature of battles, while maintaining the play feel and tactis unique to the TEKKEN series. The intensity of battles is greatly enhanced by the destructible stages. Unleashing super moves like Rage Arts is sure to enthrall both

Game view - only can be seen by admins at the moment. Provided buttons for editing and soft deleting.

Checklist Items (7)

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

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

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

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

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

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

#8 Clearly caption screenshots

List Games												
Game ID	Game Name	Publisher	Developer	Score Min	Score Max	Min Date	Max Date	Sort	Order	See Disabled	Limit	
Game ID	Game Title	Publisher	Developer	Score Min	Score Max	01/01/2024	04/21/2024	Date	▼	Tes	25	
Filter Clear												
Put Popular Games Put Decons Put Platforms												
Current Games												
id	name	publisher	developer	top score	release date	Is API	Active	created	modified	Actions		
15734	Tekken 8	Bandai Namco Entertainment	Bandai Namco Studios Inc.	89.7444	2024-01-25	Yes	Active	2024-04-21 00:14:04	2024-04-21 00:14:04	View	Edit	Toggle Active
16094	Persona 5 Reload	SEGA	ATLUS	87.7812	2024-02-02	Yes	Active	2024-04-21 08:16:22	2024-04-21 08:16:22	View	Edit	Toggle Active
16300	Baltear	Playstack	LocalThunk	91.375	2024-02-29	Yes	Active	2024-04-21 09:48:03	2024-04-21 09:48:03	View	Edit	Toggle Active
15136	Final Fantasy VII Remake	Square Enix	Square Enix	92.5455	2024-02-29	Yes	Active	2024-04-15 22:26:47	2024-04-16 07:43:47	View	Edit	Toggle Active
16429	Star Wars: Battlefront Classic Collection	Aspyr	Aspyr	60.8409	2024-03-14	Yes	Active	2024-04-15 22:26:47	2024-04-16 01:48:43	View	Edit	Toggle Active
16050	Alone in the Dark (2024)	THQ Nordic	THQ Nordic	63.8378	2024-03-20	Yes	Active	2024-04-15 22:26:47	2024-04-18 06:35:55	View	Edit	Toggle Active
16051	Dragon's Dogma 2	Capcom U.S.A., Inc.	CAPCOM Co., Ltd.	87.3704	2024-03-22	Yes	Active	2024-04-15 22:26:47	2024-04-16 22:03:14	View	Edit	Toggle Active
16489	South Park: Stick of Truth	THQ Nordic	Quantic Dream	59.68	2024-03-26	Yes	Active	2024-04-15 22:26:47	2024-04-16 01:15:34	View	Edit	Toggle Active
16557	Planet Zee: Console Edition	Frontier Developments	Frontier Developments	78.125	2024-03-26	Yes	Active	2024-04-15 22:26:47	2024-04-21 07:42:44	View	Edit	Toggle Active
16574	Spirit City: Lost Sessions	Monolith Games	Monolith Games	78.5	2024-04-03	Yes	Active	2024-04-15 22:26:47	2024-04-17 01:57:12	View	Edit	Toggle Active
16554	Gigantic Rampage Edition	Gearbox Publishing	Gearbox Publishing	-1	2024-04-09	Yes	Active	2024-04-18 00:26:45	2024-04-18 00:26:56	View	Edit	Toggle Active

ID	Category	Sub-Category	Owner	Created At	Last Updated	Status	Created On	Last Modified On	Action
11	Hello Friends		Ethan	03	2024-04-12	No	Active	2024-04-21 00:23:37	2024-04-21 00:23:37
11111111	TestWorld	NJIT	Ethan	19	2024-04-16	No	Active	2024-04-17 00:53:04	2024-04-17 00:45:24
789098765	hello	Randome	Ethan	19	2024-04-17	No	Disabled	2024-04-17 18:29:38	2024-04-21 20:33:32

If an invalid URL is passed with an invalid ID, then it redirect back to list view

Checklist Items (3)

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

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

#8 Clearly caption screenshots

Task #2 - Points: 1

Text: Screenshots of the details page code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large

PHP and HTML for view game - please refer to DB_helper to get query

Checklist Items (4)

#1 Show how id is fetched

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

#4 Include ucid/date comments for each code screenshot

#5 Clearly caption screenshots

Query for record for game view

Checklist Items (3)

#2 Show the DB query to get the record

#4 Include ucid/date comments for each code screenshot

#5 Clearly caption screenshots

[^COLLAPSE ^](#)

Task #3 - Points: 1

Text: Explain how the page works

Checklist

*The checkboxes are for your own tracking

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

Response:

Due to the multiple forms, the query is quite complex to reduce multiple query requests. The main query would attempt to get all the game information but would also join the genre ID, genre name, platform ID, and platform name with the games. This would be done by left joining or merging, all the relationship tables with the 3 main tables to make a return for a specific game ID. Specifically, the Games table will be combined with the middle relation table between Genres and Games using the IDs of each one respectively as foreign keys and the join condition. The result would be left joined with the Genres table to get the name of the genre for the game. Similarly, the same would be done with the platforms, just on top of the previous table. The records returned would be all the game data with the genres and platforms that coexist with the game. The view page would also help lazy load data. If there is no release date for the data and it originates from an API fetch, then it will try to pull the missing data.

The query would then be parsed into the games and platforms. The result would be all the data being displayed for the user, including images, text about the game, genres, and platforms.

The page would have other buttons such as an edit and delete button which act similarly to those on the list. It would display all the information using cards in a grid layout, which Bootstrap provides.

[^COLLAPSE ^](#)

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://ekh3-it202-008-prod-e9bbcd10cf36.herokuapp.com/Project/admin/view_game.php?id=14227

URL #2

<https://github.com/WokFriedE/ekh3-it202-008/pull/54>



Edit Data Page (2 pts.)

[^COLLAPSE ^](#)



Task #1 - Points: 1

Text: Screenshots of the edit page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
■ #1	1	Show before and after screenshots of data you'll edit
■ #2	1	Show examples of validation messages
■ #3	1	Show an example of successful edit messages
■ #4	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
■ #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

Game Gauner | Home | Profile | Admin | Game Management | Logout

Edit Game

[Back](#)

Name
TextWorld

Publisher
SIFT

Developer
Ethan

Description
It's a game

Critic Score
10

Release Date
04/16/2024

Square Image

Screenshot Image

Game Page URL

Platforms

Xbox Series X/S (active)
 PlayStation 5 (active)
 PlayStation 4 (inactive)
 Xbox One (inactive)
 Wii U (inactive)

Genres

Fighting (inactive)
 Action (active)
 Simulation (active)
 Adventure (inactive)
 RPG (inactive)

Page before edit - Part 1

Checklist Items (4)

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

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

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

#6 Clearly caption screenshots

The screenshot shows a web-based application for managing games. On the left, there's a sidebar titled "Platforms" listing various gaming consoles and PC. On the right, there's a sidebar titled "Genres" listing various game genres. A large central area is mostly blank, with a single button labeled "Update info". Below the main area, a dark bar contains the text "Edit game before edit part 2".

Platforms

- Xbox Series X/S (active)
- PlayStation 5 (active)
- PlayStation 4 (inactive)
- Xbox One (active)
- Wii U (active)
- PC (inactive)
- Mata Quest (inactive)
- HTC Vive (inactive)
- PlayStation VR (inactive)
- Nintendo Switch (inactive)
- PlayStation Vita (inactive)
- Nintendo 3DS (inactive)
- Google Stadia (inactive)

Genres

- Fighting (inactive)
- Action (active)
- Simulation (active)
- Adventures (inactive)
- RPG (inactive)
- First-Person Shooter (active)
- Platformer (inactive)
- Racing (inactive)
- Real-Time Strategy (inactive)
- Sports (inactive)
- Strategy (inactive)
- Third-Person Shooter (inactive)
- Turn-Based Strategy (inactive)
- Vehicle Combat (inactive)
- Art (inactive)
- Interactive Story (inactive)
- Arcade (inactive)
- Horror (inactive)
- roguelike (inactive)
- Puzzle (inactive)
- Music (inactive)
- MMO (inactive)
- Survival (inactive)
- Creation (inactive)

Update info

Edit game before edit part 2

Checklist Items (3)

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

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

#6 Clearly caption screenshots

The screenshot shows a web-based application for managing games. At the top, there's a navigation bar with links for Home, Profile, Admin, Game Management, and Logout. Below the navigation, there's a list of validation errors for a form. The errors are:

- [Client] Invalid score please enter a number that is 0-100
- [Client] Name is required
- [Client] Developer is required
- [Client] Description is required
- [Client] Release date please enter one like YYYY-MM-DD
- [Client] Screenshot URL is not an image link
- [Client] Square URL is not an image link
- [Client] Game URL is not a link

Below the error list, there's a section titled "Edit Game" with fields for Name, Publisher, and Developer, each with its own validation message.

Game Gammer

Home Profile Admin Game Management Logout

[Client] Invalid score please enter a number that is 0-100
[Client] Name is required
[Client] Developer is required
[Client] Description is required
[Client] Release date please enter one like YYYY-MM-DD
[Client] Screenshot URL is not an image link
[Client] Square URL is not an image link
[Client] Game URL is not a link

Edit Game

Name:

Publisher: Publisher (optional)

Developer:

Developer...
Description
Description...
Critic Score
100%
Release Date
2021-01-15 00:00:00
Square Image
Avatar
Screenshot Image
Avatar


Edit Game JS validation messages

Checklist Items (1)

#2 Show examples of validation messages

Game Gaesser [Home](#) [Profile](#) [Admin](#) [Game Management](#) [Logout](#)

Score must be a number and between 0 to 100 inclusive.

Edit Game

Name

Publisher NFT 2

Developer Ethan

Description It's a game

Critic Score 10

Release Date 01/16/2026

Square Image Square Image URL (optional)

Screenshot Image Screenshot Image URL (optional)

Game Page URL Game Page URL (optional)

Platforms

Xbox Series X/S (active)
 PlayStation 5 (active)
 PlayStation 4 (inactive)
 Xbox One (active)

Genres

Fighting (inactive)
 Action (active)
 Simulation (active)
 Adventure (inactive)

Game edit - Example of PHP validation

Checklist Items (1)

#2 Show examples of validation messages

Game Gaesser [Home](#) [Profile](#) [Admin](#) [Game Management](#) [Logout](#)

Updated role
Updated role
Updated role
Updated record

Edit Game

Back

Name	TerrWorld
Publisher	MHT I
Developer	Huan
Description	It's a game
Critic Score	10
Release Date	04/16/2024
Square Image	<input type="text"/> Square Image URL (optional)
Screenshot Image	<input type="text"/> Screenshot Image URL (optional)
Game Page URL	<input type="text"/> Game Page URL (optional)
Platforms	<input type="checkbox"/> Xbox Series X/S (active)
Genres	<input type="checkbox"/> Fighting (inactive)

Edit game - on success messages

Checklist Items (0)

Task #2 - Points: 1

Text: Screenshots of edit page code

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

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```

    if ($request->has('name')) {
        $rules['name'] = 'required|max:255';
        $messages['name.required'] = 'Name cannot be empty.';
        $messages['name.max'] = 'Name must not exceed 255 characters.';
    }

    if ($request->has('description')) {
        $rules['description'] = 'required';
        $messages['description.required'] = 'Description cannot be empty.';
    }

    $validator = Validator::make($request->all(), $rules);
    $validator->validate();

    $game = Game::find($id);
    $game->name = $request->input('name');
    $game->description = $request->input('description');

    $game->update();

```

PHP for edit games - includes admin role requirement, PHP validation, and DB updates

Checklist Items (6)

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

#3 Successful edit should have a user-friendly message

#4 Any errors should have user-friendly messages

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

#6 Include ucid/date comments for each code screenshot

#7 Clearly caption screenshots

```

    if ($request->has('name')) {
        $rules['name'] = 'required|max:255';
        $messages['name.required'] = 'Name cannot be empty.';
        $messages['name.max'] = 'Name must not exceed 255 characters.';
    }

    if ($request->has('description')) {
        $rules['description'] = 'required';
        $messages['description.required'] = 'Description cannot be empty.';
    }

    $validator = Validator::make($request->all(), $rules);
    $validator->validate();

    $game = Game::find($id);
    $game->name = $request->input('name');
    $game->description = $request->input('description');

    $game->update();

```

edit game that has the form generation, HTML, and JS validation

Checklist Items (5)

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

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

#4 Any errors should have user-friendly messages

#6 Include ucid/date comments for each code screenshot

#7 Clearly caption screenshots

PHP sanitizers

Checklist Items (3)

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

#6 Include ucid/date comments for each code screenshot

#7 Clearly caption screenshots

[COLLAPSE](#)

Task #3 - Points: 1

Text: Screenshot of records from DB

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot displays three separate database interfaces side-by-side, each showing a table of records. The top interface shows a table with columns: id, name, publisher_id, developer_id, description, lastUpdate, averageRating, userCount, platform_id, releaseDate, inActive, created, and modified. The middle interface shows a table with columns: id, game_id, genre_id, in_active, and modified. The bottom interface shows a table with columns: id, platform_id, game_id, in_active, and modified. All three tables have rows corresponding to the data described in the text below.

Database before edit (note values are different from before on the webpage due to the success update earlier). The bottom left being genre relation and bottom right being platform relation. The top being the game table.

Checklist Items (2)

#1 Show a before and after screenshot of the record

#3 Clearly caption screenshots

Databases after the edit with the bottom left being genre relation and bottom right being platform relation. The is_active is updated for both relations, adding some like ID 234 for GameGenre and ID 276 in PlatformGame. The game data has been changed for publisher and topCriticScore.

Checklist Items (2)

#2 Note what differs

#3 Clearly caption screenshots

Task #4 - Points: 1

Text: Explain the process

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Provide a high-level step-by-step of how the fetching of the record, populating the form, and the update works and gets changed in your DB
<input checked="" type="checkbox"/> #2	1	Briefly describe the validations for the applicable fields

Response:

With the edit view, the ID would be passed as a GET item from the list view. When the page loads, the edit page pulls the ID and checks if it is a valid ID by checking if it's positive. If so, a similar query from the view page would run. It would use left joins on the Games table with the platforms and genres tables and their relation tables. The result would be several records of all the genres, genre IDs, platforms, platform IDs, and game information. Afterward, the information would be processed and returned as an associative array with all the game info with two associative arrays within it for the genre ID with genre Name and platform ID with platform name. The result would be sent to

POST and be processed field by field, filling in the applicable input boxes with the same name as the attribute. The user would be able to change the information about the game and once they click the update button it would change the data on the database and keep the changes. For validations, it would enforce that some fields are required and not blank. For the date, it would check if it is in YYYY-MM-DD format for the SQL parse. For the critic score, it would check if it is a valid number. For the URLs, there would be basic JS validation to make sure they are URLs with images. On the server side, the same validations will be made.

For the genres and platforms, the way they work is the same. Both would call queries for all the genres and platforms available and display them all as checkbox inputs. Using the IDs from the associative array of the game, it would suffix each item with active or inactive. The user would be able to check any ones they would like and click the update platforms or genres buttons to update the relationships.

Task #5 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://ekh3-it202-008-prod-e9bbcd10cf36.herokuapp.com/Project/admin/edit_game.php?id=14227

URL #2

<https://github.com/WokFriedE/ekh3-it202-008/pull/54>

Delete Handling (1 pt.)



Task #1 - Points: 1

Text: Screenshots related to delete

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



File Edit Selection View Go Run Terminal Help

public_html > Project > admin > delete_game.php > Create

```

1  <?php
2
3  session_start();
4  require(__DIR__ . "/../../lib/functions.php");
5  if (!has_role("Admin")) {
6      die("You don't have permission to view this page", "warning");
7      die(header("Location: " . __DIR__ . "/home.php"));
8  }
9
10 $id = $_GET['id'];
11 if ($id < 1) {
12     echo("Invalid id passed to delete", "danger");
13     die(header("Location: " . get_url("admin/list_games.php")));
14 }
15
16 $id = getID();
17 $query = "UPDATE `Games` SET is_active = 0 WHERE id = $id";
18 try {
19     $stmt = $db->prepare($query);
20     $stmt->execute(["id" => $id]);
21     $class["tagged record with id $id", "success"];
22 } catch (Exception $e) {
23     error_log("Error deleting game $id" . var_export($e, true));
24     class["Error deleting record", "danger"];
25 }
26
27 // die(header("Location: " . get_url("admin/list_games.php")));
28 echo "script:history.back()</script>";
29
30
31 // echo $class["tagged record with id $id", "success"];
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
517
518
519
519
520
521
522
523
524
525
526
527
527
528
529
529
530
531
532
533
534
535
535
536
537
537
538
539
539
540
541
542
543
544
544
545
546
546
547
547
548
548
549
549
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1
```

Checklist Items (1)

- #1 Show the success message of a delete

The screenshot shows a web-based game management system. At the top, there's a navigation bar with links for Home, Profile, Admin, Game Management, and Logout. A red banner at the top of the main content area says "Game deleted". Below this, there's a search/filter section with fields for Game ID, Game Name, Publisher, Developer, Score Min, Score Max, N in Date, Max Date, Sort, Order, and a checkbox for "See Disabled". There are also buttons for "Filter" and "Clear". Below the search is a row of buttons: "Put Popular Games", "Put Games", and "Put Platforms". The main table area is titled "Current Games" and lists one game entry:

ID	Name	Publisher	Developer	Top Score	Release Date	Is API	Active	Created	Modified	Actions
11111111	TestWorld	NULL 2 refl	Ethan	50	2024-04-16	No	Disabled	2024-04-17 00:45:24	2024-04-21 23:09:45	View Edit Toggle Active

Delete game fail due to invalid ID pass in link

Checklist Items (1)

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

Task #2 - Points: 1

Text: Screenshots of the data

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show a before and after screenshot of the DB data
<input checked="" type="checkbox"/> #2	1	Note what changed (i.e., record removed or soft delete value changed)
<input checked="" type="checkbox"/> #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a database interface with a sidebar containing various schema and table options. The main area displays a table named 'Games'. A single row is selected, showing the following data:

	id	name	publisher	developer	description	topCriticScore	imgURL	screenShotURL	url	firstReleaseDate	is_api	is_active	created	modified
	1	1111111	TechWorld	NFTZ.io	Ethan	It's a game	50	(NULL)	(NULL)	(NULL)	0	1	2024-04-16 00:45:24	2024-04-21 23:07:29

Game before delete

Checklist Items (2)

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

#3 Clearly caption screenshots

This screenshot is identical to the one above, showing the 'Games' table with the same single row of data. The data is as follows:

	id	name	publisher	developer	description	topCriticScore	imgURL	screenShotURL	url	firstReleaseDate	is_api	is_active	created	modified
	1	1111111	TechWorld	NFTZ.io	Ethan	It's a game	50	(NULL)	(NULL)	(NULL)	0	1	2024-04-16 00:45:24	2024-04-21 23:08:45

Game in table after delete, the is_active attribute switched from 1 to 0 to indicate off

Checklist Items (3)

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

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

#3 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain the delete logic

Checklist

*The checkboxes are for your own tracking

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

Response:

The delete would act as a separate page that the user is directed to using the ID as a get field in the URL when sent there. It was decided to use a soft delete on the games, platforms, genres, game genre relation, and platform game relation. Each is implemented with an is_active tag that is either 0 or 1 for off or on. In the current state, the admin is the only user allowed to disable data via the list view for each major database entity.

When the page loads, it obtains the ID from GET and then passes an update query to the respective table is executed to toggle the status of the item.

Task #4 - Points: 1

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

i Details:

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

URL #1

<https://github.com/WokFriedE/ekh3-it202-008/pull/54>

Misc (1 pt.)

[COLLAPSE ^](#)

Task #1 - Points: 1

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

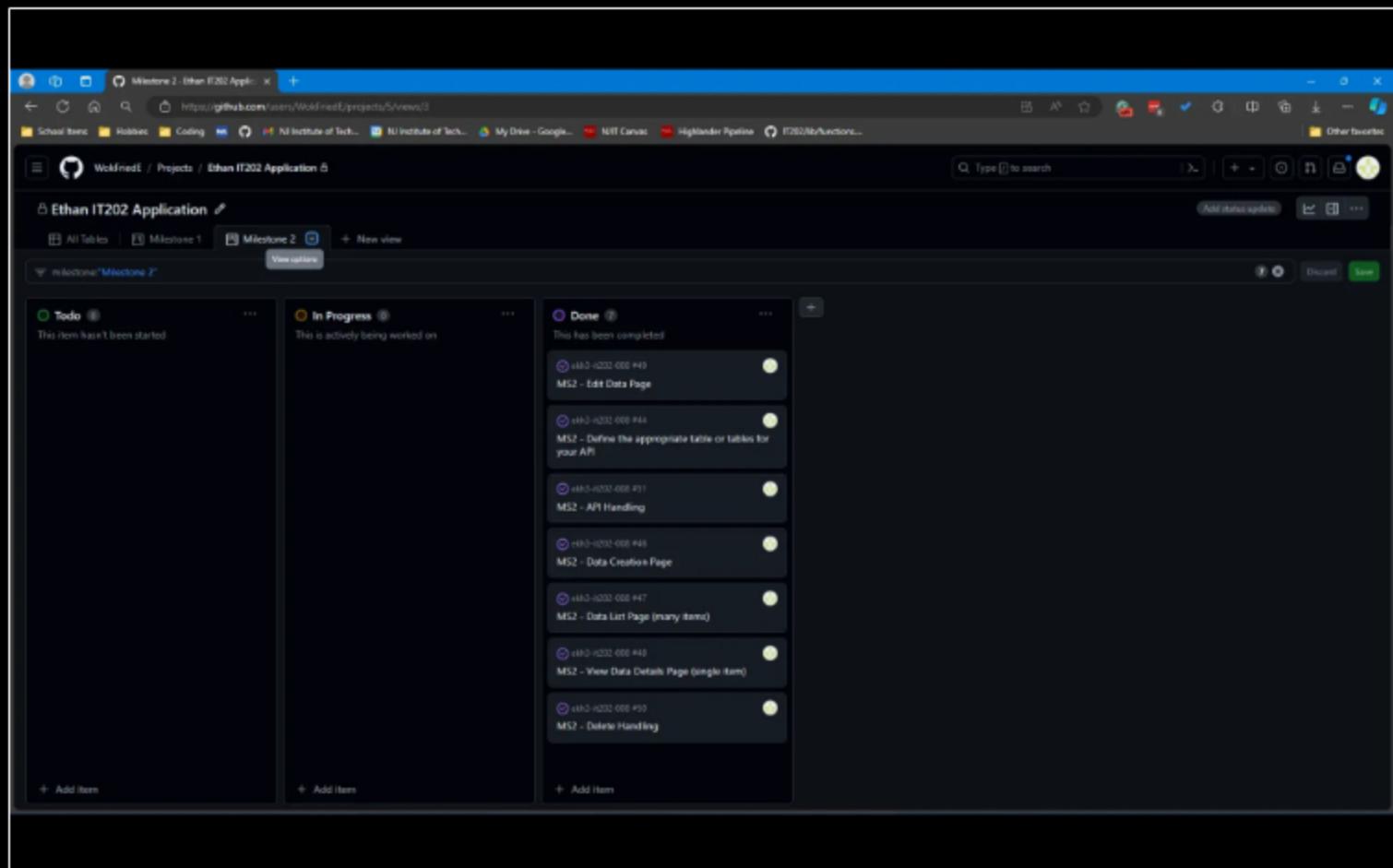
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Github Project board filtered with only Milestone 2 items

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/3>

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

One major issue was post data causing an issue when editing and creating new games. As there is a lot of data being handled around it was easy to make typos and forget specific attributes when sanitizing the data from POST.

Therefore it caused a huge issue when it came to figuring out why updating and creating games did not automatically add selected platforms and genres.

I decided to make the database closer to 3NF. As a result, a table was generated for both platforms, genres, and their relations back to each Game. The 4 additional tables added a lot more complexity and time, especially since platforms and genres needed similar actions to games.

Another huge issue was the time commitment to the assignment, in which the professor was notified of the late submission. Due to other classes overlapping their due dates with this assignment, it made it hard to juggle both.

Task #4 - Points: 1

Text: WakaTime Screenshot

Details:

Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved

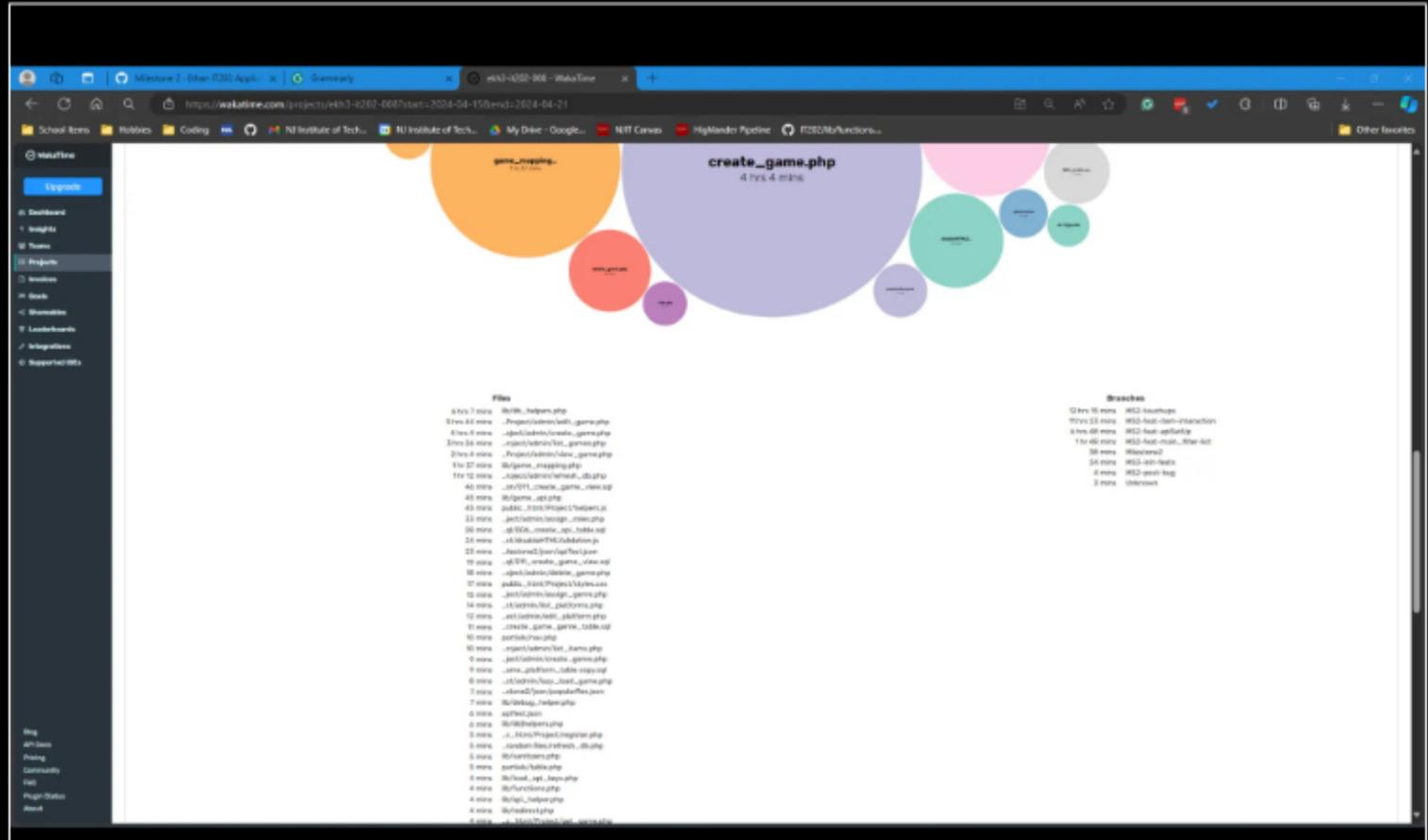
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



End of Assignment