

Submission Worksheet

CLICK TO GRADE

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

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

Submissions:

Submission Selection

1 Submission [active] 4/16/2024 9:49:53 PM

Instructions

[▲ COLLAPSE ▲](#)

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

Branch name: Milestone2

Tasks: 29 **Points:** 10.00

 Define Appropriate Tables for Data (1 pt.)

[▲ COLLAPSE ▲](#)

 Task #1 - Points: 1

Text: Screenshots of Table SQL

Checklist

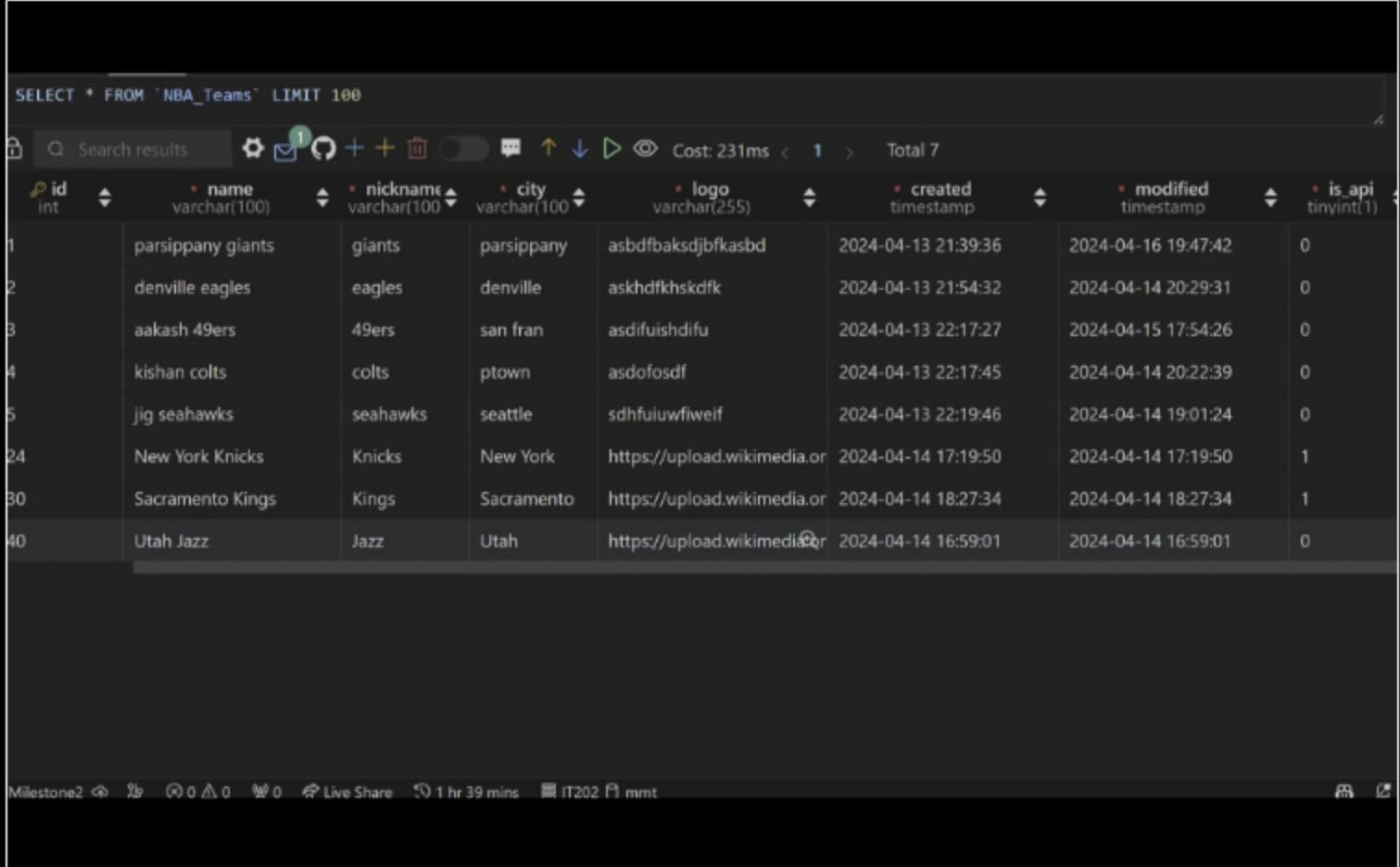
*The checkboxes are for your own tracking

#	Points	Details
#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 interface displaying the results of a query: `SELECT * FROM 'NBA_Teams' LIMIT 100`. The results are presented in a table with the following columns: id, name, nickname, city, logo, created, modified, and is_api. The data includes various NBA teams like the New York Knicks, Sacramento Kings, and Utah Jazz, along with their respective names, nicknames, cities, logos, creation and modification dates, and an is_api flag. The interface includes search, filter, and sort functions at the top.

	id	name	nickname	city	logo	created	modified	is_api
1		parsippany giants	giants	parsippany	asbdfbaksdjbfkasbd	2024-04-13 21:39:36	2024-04-16 19:47:42	0
2		denville eagles	eagles	denville	askhdfkhsksdkf	2024-04-13 21:54:32	2024-04-14 20:29:31	0
3		aakash 49ers	49ers	san fran	asdifuishdifu	2024-04-13 22:17:27	2024-04-15 17:54:26	0
4		kishan colts	colts	ptown	asdofosdf	2024-04-13 22:17:45	2024-04-14 20:22:39	0
5		jig seahawks	seahawks	seattle	sdhfuiuwfiweif	2024-04-13 22:19:46	2024-04-14 19:01:24	0
24		New York Knicks	Knicks	New York	https://upload.wikimedia.or	2024-04-14 17:19:50	2024-04-14 17:19:50	1
30		Sacramento Kings	Kings	Sacramento	https://upload.wikimedia.or	2024-04-14 18:27:34	2024-04-14 18:27:34	1
40		Utah Jazz	Jazz	Utah	https://upload.wikimedia.or	2024-04-14 16:59:01	2024-04-14 16:59:01	0

Milestone2 Live Share 1 hr 39 mins IT202 mmr

Table includes the following fields: id, name, nickname, city, and logo of an NBA team. Also added an is_api column which displays 1 if the data that was added is from the api or not

Checklist Items (0)

Task #2 - Points: 1

Text: Explain the design

Checklist		*The checkboxes are for your own tracking
#	Points	Details
#1	1	Note the different fields and their purpose

#2

1

Note if you needed to normalize the data into separate tables or if one table fit your needs

Response:

- The name field represents the name of an nba team (ex: golden state warriors)
- The nickname field represents the nickname of an nba team (ex: warriors)
- The city field represents the city the team is from (ex: San francisco)
- The logo field represents the logo of an nba team
- I did not need to normalize my data into separate tables

Task #3 - Points: 1

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

1 Details:

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

URL #1

<https://github.com/mananthakore/mmt-it202-008/pull/57>

URL #2

<https://github.com/mananthakore/mmt-it202-008/pull/63>

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
#1	1	Show potentially valid data filled in for the custom creation page
#2	1	Show how the API data is fetched for API data (must be server-side)
#3	1	Show examples of validation messages
#4	1	Show an example of successful creation message
#5	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)

#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 the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php. The page title is "Add NBA Team". The form fields are as follows:

- Name: NJIT Highlanders
- Nickname: Highlanders
- City: Newark
- Logo: https://en.wikipedia.org/wiki/New_Jersey_Institute_of_Technology#/media/File:New_Jersey_IT_seal.svg

A blue button labeled "Add Team" is at the bottom of the form.

screenshot of create team fields being filled out

Checklist Items (0)

The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php. The page title is "Add NBA Team". A message box displays "Team added successfully". The form fields are identical to the previous screenshot:

- Name: Enter team name
- Nickname: Enter team nickname
- City: Enter team city

Enter team city

Logo

Enter team logo URL

Add Team

screenshot of success message after team was added

Checklist Items (0)

The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php. The page title is "Add NBA Team". The "Name" field is empty and highlighted with a red border. A validation message "[js] Team name is required" is displayed in a yellow box below the input field.

Name

Enter team name

Nickname

b

City

c

Logo

d

js validation for empty team name

Checklist Items (0)

The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php. The page title is "Add NBA Team". The "Name" field is empty and highlighted with a red border. A validation message "[server] Team name is required" is displayed in a yellow box below the input field.

Name

Enter team name

Nickname

Enter team nickname

City

Enter team city

Please fill out this field.

Logo

Enter team logo URI

php validation for empty team name

Checklist Items (0)

The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php. The page title is "Add NBA Team". The "Nickname" input field is empty and highlighted in yellow, with the error message "[js] Team nickname is required" displayed above it. The browser's address bar shows the same URL. The navigation bar includes links for "Home", "Profile", "Admin", "Team Management", and "Logout".

[js] Team nickname is required

Add NBA Team

Name

b

Nickname

Enter team nickname

City

c

Logo

d

js validation for empty team nickname

Checklist Items (0)

The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php. The page title is "Add NBA Team". The "Nickname" input field is empty and highlighted in yellow, with the error message "[server] Team nickname is required" displayed above it. The browser's address bar shows the same URL. The navigation bar includes links for "Home", "Profile", "Admin", "Team Management", and "Logout".

[server] Team nickname is required

Add NBA Team

Name

Enter team name

Nickname

Enter team nickname

City

Please fill out this field.

Enter team city

Logo

Enter team logo URL

php validation for empty team nickname

Checklist Items (0)

← → ⌂

https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php

Getting Started Google YouTube

DunkData Home Profile Admin ▾ Team Management ▾ Logout

[js] Team city is required

Add NBA Team

Name

a

Nickname

b

City

Enter team city

js validation for empty team name

Checklist Items (0)

← → ⌂

https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php

Getting Started Google YouTube

DunkData Home Profile Admin ▾ Team Management ▾ Logout

[server] Team city is required

Add NBA Team

Name

Enter team name

Nickname

Enter team nickname

City

Please fill out this field.

Enter team city

Logo

Enter team logo URL

php validation for empty team city

Checklist Items (0)

The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php. The page title is "Add NBA Team". The navigation bar includes links for "Getting Started", "Google", and "YouTube". The main menu has items "DunkData", "Home", "Profile", "Admin", "Team Management", and "Logout". A yellow callout box displays the error message "[js] Team logo URL is required".

Name

a

Nickname

b

City

c

Logo

Enter team logo URI

js validation for empty logo url

Checklist Items (0)

The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/create_team.php. The page title is "Add NBA Team". The navigation bar includes links for "Getting Started", "Google", and "YouTube". A yellow callout box displays the error message "[server] Team logo URL is required".

Name

Enter team name

Nickname

Enter team nickname

City

Enter team city

Logo

Enter team logo URL

Add Team

php validation for empty logo url

Checklist Items (0)

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

```
<?php  
// Include navigation and role check logic, mmt 4/17/2024  
require(__DIR__ . "/../../../../partials/nav.php");
```

```
// Check if the user has admin role, redirect if not Manan Thakore, 4 days ago • started create
if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning");
    die(header("Location: $BASE_PATH" . "/home.php"));
}

// Initialize an array to store validation errors

if ($_SERVER["REQUEST_METHOD"] === "POST") {

    $name = se($_POST, "name", "", false);
    $nickname = se($_POST, "nickname", "", false);
    $city = se($_POST, "city", "", false);
    $logo = se($_POST, "logo", "", false);

    $error = false;
    // Fetch team information from form submission

    // Validate input
    if (empty($name)) {
        flash("[server] Team name is required", "warning");
        $error = true;
    }
}
```

screenshot of create page code

Checklist Items (0)

```
if (empty($nickname)) { // mmt 4/17/2024
    flash("[server] Team nickname is required", "warning");
    $error = true;
}

if (empty($city)) {
    flash("[server] Team city is required", "warning");
    $error = true;
}

if (empty($logo)) {
    flash("[server] Team logo URL is required", "warning");
    $error = true;
}

// If there are no errors and the form was submitted, proceed to insert into the database
if(!$error) {
    $db = getDB();
    $query = "INSERT INTO NBA_Teams (name, nickname, city, logo) VALUES (:name, :nickname, :city, :logo)";
    try {
        $stmt = $db->prepare($query);
        $stmt->execute([
            ":name" => $name,
            ":nickname" => $nickname,
            ":city" => $city,
            ":logo" => $logo
        ]); <- #50-55 $stmt->execute
        flash("Team added successfully", "success");
    } catch(PDOException $e) {
```

screenshot of create page code

Checklist Items (0)

```
public_html > Project > admin > . > create_team.php > ...
58     if($e->errorInfo[1] === 1062) {
59         flash("A team with this info already exists", "warning");
60     }
61     else {
62         flash("An unexpected error occurred", "danger");
63         error_log("Database error: " . $e->getMessage()); // mmt 4/17/2024 You, 19 hours ago * Uncommitted change
64     }
65 } <- #57-65 catch(PDOException $e)
66
67 } <- #45-67 if(!$error)
68 } <- #13-68 if ($_SERVER["REQUEST_METHOD"] === "POST")
69
70 ?>
71
72 <div class="container-fluid">
73     <h3>Add NBA Team</h3>
74     <form method = "POST" onsubmit = "return validate(this);">
75         <div class="form-group">
76             <label for="name">Name</label>
77             <input type="text" class="form-control" <?php if (isset($error['name'])) echo 'is-invalid'; ?>" id="name" name="name"
78             <?php if (isset($error['name'])) : ?>
79                 <div class="invalid-feedback"><?php echo $error['name']; ?></div>
80             <?php endif; ?>
81         </div>
82         <div class="form-group">
83             <label for="nickname">Nickname</label>
84             <input type="text" class="form-control" <?php if (isset($error['nickname'])) echo 'is-invalid'; ?>" id="nickname" name="nickname"
85             <?php if (isset($error['nickname'])) : ?>
86                 <div class="invalid-feedback"><?php echo $error['nickname']; ?></div>
```

screenshot of create page code

Checklist Items (0)

```
<form method = "POST" onsubmit = "return validate(this);">
    <div class="form-group">
        <?php endif; ?>
    </div>
    <div class="form-group">
        <label for="nickname">Nickname</label> <!-- mmt 4/17/2024 -->
        <input type="text" class="form-control" <?php if (isset($error['nickname'])) echo 'is-invalid'; ?>" id="nickname" name="nickname"
        <?php if (isset($error['nickname'])) : ?>
            <div class="invalid-feedback"><?php echo $error['nickname']; ?></div>
        <?php endif; ?>
    </div>
    <div class="form-group">
        <label for="city">City</label>
        <input type="text" class="form-control" <?php if (isset($error['city'])) echo 'is-invalid'; ?>" id="city" name="city"
        <?php if (isset($error['city'])) : ?>
            <div class="invalid-feedback"><?php echo $error['city']; ?></div>
        <?php endif; ?>
    </div>
    <div class="form-group">
        <label for="logo">Logo</label>
        <input type="text" class="form-control" <?php if (isset($error['logo'])) echo 'is-invalid'; ?>" id="logo" name="logo"
        <?php if (isset($error['logo'])) : ?>
            <div class="invalid-feedback"><?php echo $error['logo']; ?></div>
        <?php endif; ?>
    </div>
    <button type="submit" class="btn btn-primary">Add Team</button>
</form>
</div>
```

screenshot of create page code

Checklist Items (0)

```
public_html > Project > admin > create_team.php > script > validate
106 <script>
107     function validate(form) { // mmt 4/17/2024
108         const teamInput = form.name.value;
109         const nicknameInput = form.nickname.value;
110         const cityInput = form.city.value;
111         const logoInput = form.logo.value;
112
113     let isValid = true;
114     Manan Thakore, 3 days ago + new validation
115     if (!teamInput) {
116         flash("[js] Team name is required", "warning");
117         isValid = false;
118     }
119     if (!nicknameInput) {
120         flash("[js] Team nickname is required", "warning");
121         isValid = false;
122     }
123
124     if (!cityInput) {
125         flash("[js] Team city is required", "warning");
126         isValid = false;
127     }
128
129     if (!logoInput) {
130         flash("[js] Team logo URL is required", "warning");
131         isValid = false;
132     }
133     return isValid;
134 } <- #109-136 function validate(form)
```

screenshot of edit page code

Checklist Items (0)

```
123
124     if (!cityInput) {
125         flash("[js] Team city is required", "warning");
126         isValid = false;
127     }
128
129     if (!logoInput) {
130         flash("[js] Team logo URL is required", "warning");
131         isValid = false;
132     }
133     return isValid; // mmt 4/17/2024
134 } <- #109-136 function validate(form)
135 </script>
136 <?php require(__DIR__ . "/../../../../partials/flash.php"); ?>
```

screenshot of create page code

Checklist Items (0)



[COLLAPSE](#)

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

Record Details							
ID	Name	Nickname	City	Logo	Created	Modified	is_api
1	parsippany giants	giants	parsippany	asbdfbaksdjbkasbd	2024-04-13 21:39:36	2024-04-16 19:47:42	0

record via the creation page (is_api shows 0 because its not from the api)

Checklist Items (0)



24	New York Knicks	Knicks	New York	https://upload.wikimedia.or	2024-04-14 17:19:50	2024-04-14 17:19:50	1
30	Sacramento Kings	Kings	Sacramento	https://upload.wikimedia.or	2024-04-14 18:27:34	2024-04-14 18:27:34	1

record fetched from the api (is_api shows 1 because its from the api)

Checklist Items (0)

Task #4 - Points: 1

Text: Explain the process

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#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)
#2	1	Provide a high-level step-by-step of how creating custom data works and gets added to your DB (include how duplicates are handled)
#3	1	Briefly describe the validations for the applicable fields
#4	1	Describe how duplicate data is handled

Response:

Fetching API data involves sending a request to the API endpoint to retrieve desired information, parsing the JSON response received, checking for duplicates in the database based on unique criteria, and inserting the data into the database if it's not a duplicate. Duplicates are managed by identifying existing records before insertion and taking appropriate actions such as skipping insertion or updating existing records. Creating custom data entails receiving form submissions, validating the data for required fields and formats, checking for duplicates in the database, and inserting the data if validation passes and no duplicates are found. Field validations include ensuring that team name, city, and state are present and符合相应的格式。对于重复的数据，如果插入失败，通常会忽略该条目或更新现有记录。

nickname, city, and logo URL are provided. Duplicate data is handled by catching database errors during insertion attempts, and if a duplicate entry is detected, a warning message is displayed to the user.



[^COLLAPSE ^](#)

Task #5 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://mmt-prod-6beda717bbfa.herokuapp.com/Project/admin/create_team.php

URL #2

<https://github.com/mananthakore/mmt-it202-008/pull/58>

URL #3

<https://github.com/mananthakore/mmt-it202-008/pull/60>



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

[^COLLAPSE ^](#)



[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the list page

Checklist

*The checkboxes are for your own tracking

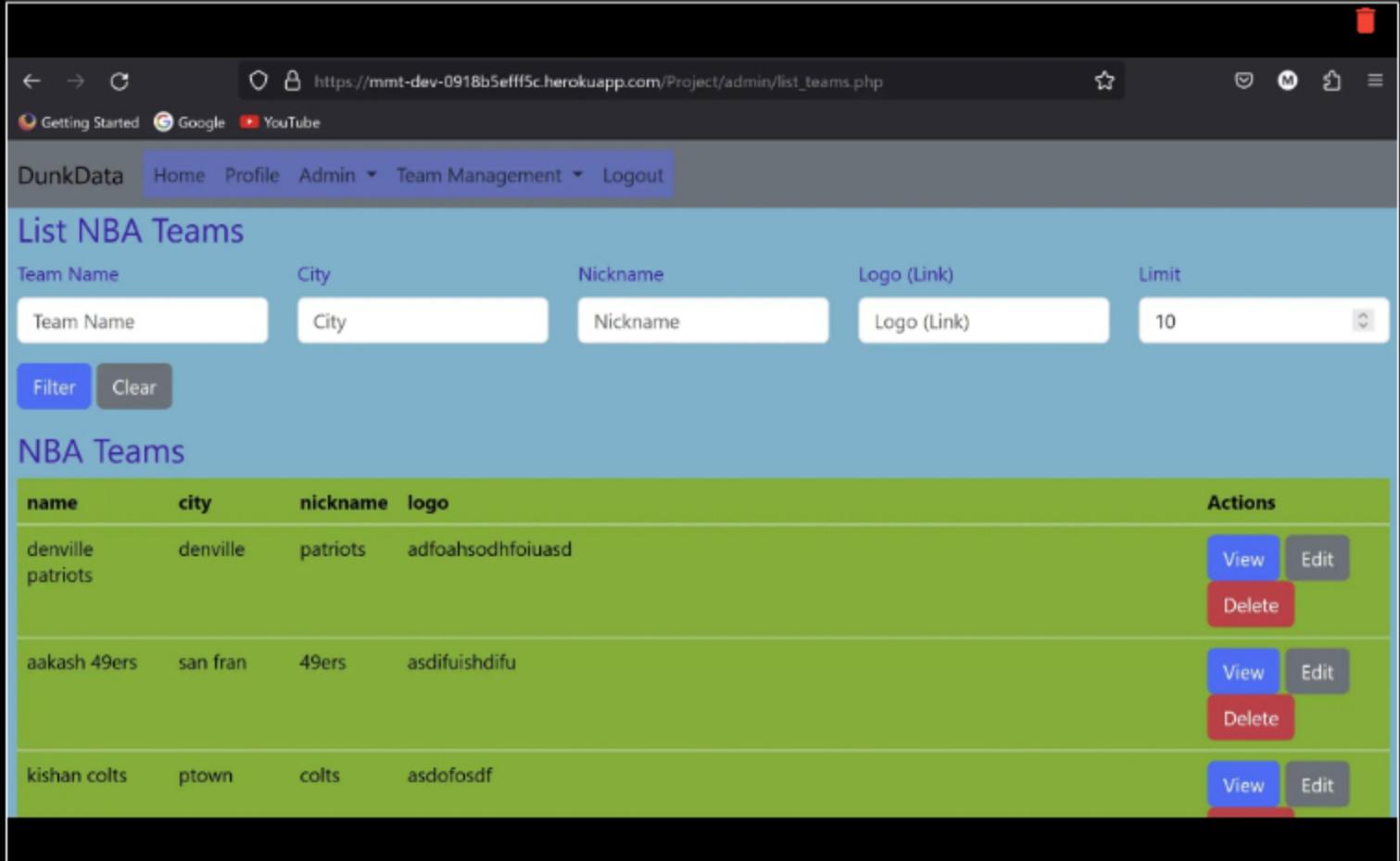
#	Points	Details
#1	1	Show the page of your entities listed (have a reasonable number shown)
#2	1	Show the filter/sort form based on your data and the required limit field
#3	1	Demonstrate a few varied filters/sorts
#4	1	Demonstrate a filter that doesn't have any records (should show an appropriate message)
#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)
#6	1	Each list item should have a summary of the entity (likely won't be the entire entity data)

#7	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
#8	1	Make sure the heroku dev url is visible in the address bar
#9	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

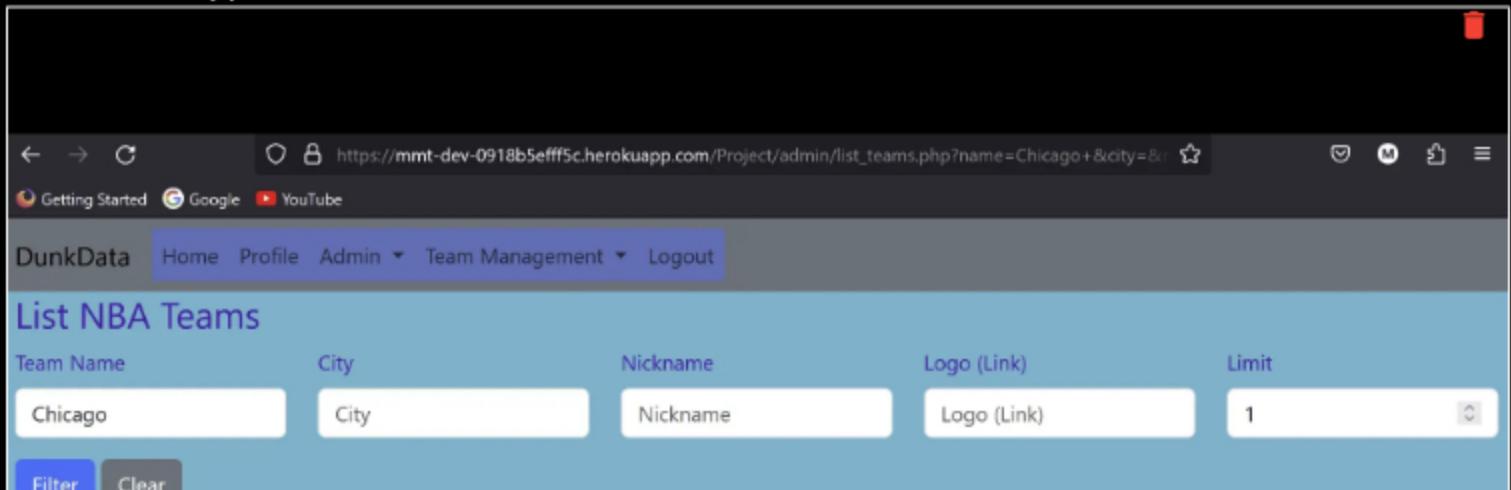
Small Medium Large



The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/list_teams.php. The page title is "DunkData". The main content is titled "List NBA Teams". Below it is a search/filter section with fields for "Team Name", "City", "Nickname", "Logo (Link)", and a "Limit" dropdown set to 10. There are "Filter" and "Clear" buttons. The main area displays a table of NBA teams with columns: name, city, nickname, logo, and Actions. The first row has actions for View, Edit, and Delete. The second row has actions for View, Edit, and Delete. The third row has actions for View and Edit.

screenshot of list teams page

Checklist Items (0)



The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/list_teams.php?name=Chicago+&city=&. The page title is "DunkData". The main content is titled "List NBA Teams". Below it is a search/filter section with fields for "Team Name" containing "Chicago", "City", "Nickname", "Logo (Link)", and a "Limit" dropdown set to 1. There are "Filter" and "Clear" buttons. The main area displays a table of NBA teams with columns: name, city, nickname, logo, and Actions. The first row has actions for View, Edit, and Delete. The second row has actions for View, Edit, and Delete. The third row has actions for View and Edit.

name	city	nickname	logo	Actions
Chicago Bulls	Chicago	Bulls	https://upload.wikimedia.org/wikipedia/fr/thumb/d/d1/Bulls_de_Chicago_logo.svg/1200px-Bulls_de_Chicago_logo.svg.png	<button>View</button> <button>Edit</button> <button>Delete</button>

screenshot of filtered nba teams

Checklist Items (0)

DunkData Home Profile Admin ▾ Team Management ▾ Logout

List NBA Teams

Team Name	City	Nickname	Logo (Link)	Limit
Team Name	San Fran	Bulls	Logo (Link)	1

Filter Clear

NBA Teams

No records to show

screenshot of no records to show message

Checklist Items (0)

DunkData Home Profile Admin ▾ Team Management ▾ Logout

List NBA Teams

Team Name	City	Nickname	Logo (Link)	Limit
Chicago	City	Nickname	Logo (Link)	2

Filter Clear

NBA Teams

NBA Teams					Actions	
name	city	nickname	logo			
Chicago Bulls	Chicago	Bulls	https://upload.wikimedia.org/wikipedia/fr/thumb/d/d1/Bulls_de_Chicago_logo.svg/1200px-Bulls_de_Chicago_logo.svg.png		View	Edit
Chicago	Chicago	Vikings	aksdkfjksadfkjsak		View	Edit

screenshot of limit and filter feature

Checklist Items (0)

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
require(__DIR__ . "/../../../../partials/nav.php"); // mmt 4/17/2024

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

// Build search form
$form = [
    ["type" => "text", "name" => "name", "placeholder" => "Team Name", "label" => "Team Name", "include_margin" => false],
    ["type" => "text", "name" => "city", "placeholder" => "City", "label" => "City", "include_margin" => false],
]
```

```

["type" => "text", "name" => "nickname", "placeholder" => "Nickname", "label" => "Nickname", "include_margin" => false],
["type" => "text", "name" => "logo", "placeholder" => "Logo (Link)", "label" => "Logo (Link)", "include_margin" => false],
["type" => "number", "name" => "limit", "label" => "Limit", "value" => "10", "include_margin" => false]
];
} <- #10-16 $form =
```

\$query = "SELECT id, name, city, nickname, logo FROM `NBA_Teams` WHERE 1=1";
\$params = [];
\$session_key = \$_SERVER["SCRIPT_NAME"];
\$is_clear = isset(\$_GET["clear"]);
if (\$is_clear) {
 session_delete(\$session_key);
 unset(\$_GET["clear"]);
 die(header("Location: " . \$session_key));
} else {
 \$session_data = session_load(\$session_key);
}

screenshot of list page code

Checklist Items (0)

```

30 if (count($_GET) == 0 && isset($session_data) && count($session_data) > 0) { // mmt 4/17/2024
31     if ($session_data) {
32         $_GET = $session_data;
33     }
34 } <- #30-34 if (count($_GET) == 0 && isset($session_data) && count($sessi...
35
36 if (count($_GET) > 0) {
37     session_save($session_key, $_GET);
38     $keys = array_keys($_GET);
39
40     foreach ($form as $k => $v) {
41         if (in_array($v["name"], $keys)) {
42             $form[$k]["value"] = $_GET[$v["name"]];
43         }
44     } <- #40-44 foreach ($form as $k => $v)
45
46     // Filter by team name
47     $name = se($_GET, "name", "", false);
48     if (!empty($name)) {
49         $query .= " AND name LIKE :name";
50         $params[":name"] = "%$name%";
51     }
52
53     // Filter by city
54     $city = se($_GET, "city", "", false);
55     if (!empty($city)) {
56         $query .= " AND city LIKE :city";
57         $params[":city"] = "%$city%";
58     }

```

screenshot of list page code

Checklist Items (0)

```

// Filter by nickname
$nickname = se($_GET, "nickname", "", false);
if (!empty($nickname)) { // mmt 4/17/2024
    $query .= " AND nickname LIKE :nickname";
    $params[":nickname"] = "%$nickname%";
}

// Filter by logo (link)
$logo = se($_GET, "logo", "", false);
if (!empty($logo)) {
    $query .= " AND logo LIKE :logo";
}

```

```
$params[":logo"] = "%$logo%";  
}  
try {  
    $limit = (int)se($_GET, "limit", "10", false);  
} catch (Exception $e) {  
    $limit = 10;  
}  
if ($limit < 1 || $limit > 100) [  
    $limit = 10;] Manan Thakore, 20 hours ago • php validation, styling, limit feature  
}  
} <- #36-81 if (count($_GET) > 0)  
  
$db = getDB();  
$stmt = $db->prepare($query);  
$results = [];  
try {  
    $stmt->execute($params);
```

screenshot of list page code

Checklist Items (0)

```
88     $r = $stmt->fetchAll();  
89     if ($r) {  
90         $results = $r;  
91     }  
92 } catch (PDOException $e) {  
93     error_log("Error fetching NBA teams: " . var_export($e, true));  
94     flash("Unhandled error occurred", "danger");  
95 }  
96  
97 $table = [  
98     "data" => $results,  
99     "title" => "NBA Teams",  
100    "ignored_columns" => ["id"], // mmt 4/17/2024  
101    // Add edit and delete URLs if needed  
102    "edit_url" => get_url("admin/edit_teams.php"),  
103    "delete_url" => get_url("admin/delete_teams.php"),  
104    "view_url" => get_url("admin/view_team.php")  
105]; <- #97-105 $table =  
106 ?>  
107 <div class="container-fluid">  
108     <h3>List NBA Teams</h3>  
109     <form method="GET">  
110         <div class="row mb-3" style="align-items: flex-end;">  
111             <?php foreach ($form as $k => $v) : ?>  
112                 <div class="col">  
113                     <?php render_input($v); ?>  
114                 </div>  
115             <?php endforeach; ?>  
116         </div>
```

screenshot of list page code

Checklist Items (0)

```
115             <?php endforeach; ?>  
116         </div>  
117     <?php if ($error) : ?>
```

```
117     <?php render_button(["text" => "Search", "type" => "submit", "text" => "Filter"]); ?>
118     <a href="?clear" class="btn btn-secondary">Clear</a>
119 </form> <!-- mmt 4/17/2024 -->
120     <?php render_table($table); ?>
121 </div>
122
123     <?php require_once(__DIR__ . "/../../../../partials/flash.php"); ?>
124
```

screenshot of list page code

Checklist Items (0)

Task #3 - Points: 1

Text: Explain how the page works

Checklist

*The checkboxes are for your own tracking

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

Response:

The filter and sorting functionality on the screen allows users to search for NBA teams based on various criteria such as team name, city, nickname, and logo link. Users can enter search terms in the provided input fields and set a limit on the number of results displayed. The design choice includes a simple form layout with text input fields for each search parameter and a submit button to apply the filters. Additionally, a "Clear" button is provided to reset the filters. The view is accessible to users with administrative roles, allowing them to interact with the view, edit, and delete links associated with each displayed team. The displayed data is presented in a tabular format, showing the team name, city, nickname, and logo link for each NBA team. Users can navigate through the results and perform actions such as editing or deleting team information as needed.

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

URL #1https://mmt-prod-6beda717bbfa.herokuapp.com/Project/admin/list_teams.php**URL #2**<https://github.com/mananthakore/mmt-it202-008/pull/59>**URL #3**<https://github.com/mananthakore/mmt-it202-008/pull/68>**URL #4**<https://github.com/mananthakore/mmt-it202-008/pull/69>

 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:

Small

Medium

Large

Invalid id passed

List NBA Teams

Team Name	City	Nickname	Logo (Link)	Limit
<input type="text" value="Team Name"/>	<input type="text" value="City"/>	<input type="text" value="Nickname"/>	<input type="text" value="Logo (Link)"/>	<input type="text" value="10"/>
<input type="button" value="Filter"/> <input type="button" value="Clear"/>				

NBA Teams

name	city	nickname	logo	Actions
denville	denville	patriots	adfoahsodhfoliuasd	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
aakash	49ers	san fran	49ers	<input type="button" value="View"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

missing id redirects back to list page

Checklist Items (0)

https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/view_team.php?id=40

Utah Jazz



Utah Jazz
Nickname: Jazz
City: Utah

screenshot of single view page for Utah Jazz team

Checklist Items (0)

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

```
public_html > Project > admin > view_team.php > ...
...
1  <?php
2
3  require(__DIR__ . "/../../../../partials/nav.php"); // mmt 4/17/2024
4
5  if (!has_role("Admin")) {
6      flash("You do not have permission to view this page", "warning");
7      die(header("location: $BASE_PATH" . "/home.php"));
8  }
9
10 $id = se($_GET, "id", -1, false);
11
12 $teamData=[];
13 if($id>-1){
14     $db=getDB();
15     $query="SELECT name, nickname, city, logo FROM `NBA_Teams` WHERE id=:id";
16     try{
17         $stmt=$db->prepare($query);
18         $stmt->execute([":id"=>$id]);
19         $teamData=$stmt->fetch(PDO::FETCH_ASSOC);
20     } <- #16-20 try
21     catch(PDOException $e){
22         error_log("Error fetching team data: " . var_export($e, true));
23         flash("Error fetching team data", "danger");
24     }
25 } <- #13-25 if($id>-1)
26 else{
```

```
27     flash("Invalid id passed", "danger");
28     die(header("Location: " . get_url("admin/list_teams.php")));
```

screenshot of view page code

Checklist Items (0)

```
26     else{
27         flash("Invalid id passed", "danger");
28         die(header("Location: " . get_url("admin/list_teams.php"))); // mmt 4/17/2024
29     }
30 ?>
31
32 <div class="container mt-4 d-flex justify-content-center">
33     <div class="card" style="width: 18rem;">
34         <?php if(!empty($teamData["logo"])):>
35             " class="card-img-top" alt="Team Logo">
36         <?php endif;?>
37
38         <div class="card-body">
39             <h5 class="card-title"><?php echo($teamData["name"]);?></h5>
40             <p class="card-text">Nickname: <?php safer_echo($teamData["nickname"]);?></p>
41             <p class="card-text">City: <?php safer_echo($teamData["city"]);?></p>
42         </div>
43     </div>
44 </div>
45
46 <?php require(__DIR__ . "/../../../../partials/flash.php");?>
```

screenshot of view page code

Checklist Items (0)

Task #3 - Points: 1

Text: Explain how the page works

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Provide the high-level steps for handling the DB lookup and presenting the data

Response:

Initially, it checks the user's permissions to ensure they have the necessary access rights. Upon validation, the page proceeds to query the database, executing a SELECT statement on the NBA_Teams table to retrieve team names, nicknames, cities, and logo URLs. The retrieved data is then fetched and stored for further processing. At the same time, the page formats the data for presentation, adding it into a table or list format, where each row represents an NBA team and each column represents a specific attribute. Once formatted, the data is displayed on the web page for user viewing. Throughout the process, robust error handling is implemented to catch and manage any encountered errors, including database connection issues or unexpected errors.



▲COLLAPSE▲

Task #4 - Points: 1**Text: Add related links****Checklist*****The checkboxes are for your own tracking**

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

URL #1https://mmt-prod-6beda717bbfa.herokuapp.com/Project/admin/view_team.php**URL #2**<https://github.com/mananthakore/mmt-it202-008/pull/63>**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**

Edit Team

[Back](#)**Team Name**

denville eagles

Team Nickname

eagles

City

denville

Logo URL

askhdfkhsdkf

[Update](#)**before denville eagles gets updated to denville patriots**

Checklist Items (0)

[←](#) [→](#) [C](#)https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/edit_teams.php?id=2

Updated record

Edit Team

[Back](#)**Team Name**

denville patriots

Team Nickname

patriots

City

denville

Logo URL

askhdfkhsdkf

after denville eagles gets updated to denville patriots

Checklist Items (0)

A screenshot of a web browser window. The address bar shows the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/edit_teams.php?id=2. The page title is "Edit Team". The navigation bar includes links for "Home", "Profile", "Admin", "Team Management", and "Logout". A yellow validation message box contains the text "[js] Team name is required". Below the message, the "Team Name" field is empty. The "Team Nickname" field contains the value "patriots". At the bottom of the page is a dark footer bar with the text "js validation for empty team name".

Checklist Items (0)

A screenshot of a web browser window, identical to the one above but with a different validation message. The address bar shows the same URL. The page title is "Edit Team". The navigation bar includes links for "Home", "Profile", "Admin", "Team Management", and "Logout". A yellow validation message box contains the text "[server] Team is required". Below the message, the "Team Name" field is empty. The "Team Nickname" field contains the value "patriots". At the bottom of the page is a dark footer bar with the text "php validation for empty team name".

Checklist Items (0)

The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/edit_teams.php?id=2. The page title is "Edit Team". A navigation bar at the top includes links for "Getting Started", "Google", "YouTube", "Home", "Profile", "Admin", "Team Management", and "Logout". A yellow message box displays the error "[js] Team nickname is required". The "Team Name" field contains "denville patriots". The "Team Nickname" field is empty. A footer message box at the bottom right says "js validation for empty nickname".

[js] Team nickname is required

Edit Team

Back

Team Name

denville patriots

Team Nickname

Team Nickname

js validation for empty nickname

Checklist Items (0)

The screenshot shows a web browser window with the URL https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/edit_teams.php?id=2. The page title is "Edit Team". A navigation bar at the top includes links for "Getting Started", "Google", "YouTube", "Home", "Profile", "Admin", "Team Management", and "Logout". A yellow message box displays the error "[server] Nickname is required". The "Team Name" field contains "denville patriots". The "Team Nickname" field is empty. A footer message box at the bottom right says "[server] validation for empty nickname".

[server] Nickname is required

Edit Team

Back

Team Name

denville patriots

Team Nickname

Team Nickname

[server] validation for empty nickname

Checklist Items (0)

← → C https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/edit_teams.php?id=2

Getting Started Google YouTube

[js] Team city is required

Edit Team

Back

Team Name

denville patriots

Team Nickname

patriots

City

City

js validation for empty city

Checklist Items (0)

← → C https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/edit_teams.php?id=2

Getting Started Google YouTube

[server] City is required

Edit Team

Back

Team Name

denville patriots

Team Nickname

patriots

City

City

php validation for empty city

Checklist Items (0)

The screenshot shows a web application titled "Edit Team". The URL is https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/edit_teams.php?id=2. A yellow validation message box at the top states "[js] Team logo URL is required". The form fields are as follows:

- Team Name: denville patriots
- Team Nickname: patriots
- City: denville
- Logo URL: Logo URL

A blue "Update" button is located at the bottom left.

js validation for empty logo url

Checklist Items (0)

The screenshot shows a web application titled "Edit Team". The URL is https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/edit_teams.php?id=2. A yellow validation message box at the top states "[server] Logo URL is required". The form fields are as follows:

- Team Name: denville patriots
- Team Nickname: patriots
- City: denville
- Logo URL: Logo URL

A blue "Update" button is located at the bottom left.

php validation for empty logo url

Checklist Items (0)

Task #2 - Points: 1

Text: Screenshots of edit 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 edit should have a user-friendly message
#4	1	Any errors should have user-friendly messages
#5	1	Include any other rules like role guards and login checks
#6	1	Include ucid/date comments for each code screenshot
#7	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```
public_html > Project > admin > edit_teams.php > ...
You, 19 hours ago | 2 authors (Manan Thakore and others)
1  <?php
2  require(__DIR__ . "/../../../../partials/nav.php");
3  // mmt 4/17/2024 You, 19 hours ago * Uncommitted changes
4  if(!has_role("Admin")){
5      flash("You do not have permission to view this page", "warning");
6      die(header("location: $BASE_PATH" . "/home.php"));
7  }
8
9  $id = se($_GET, "id", -1, false);
10
11 if ($_SERVER["REQUEST_METHOD"] === "POST") {
12     $name = se($_POST, "name", "", false);
13     $nickname = se($_POST, "nickname", "", false);
14     $city = se($_POST, "city", "", false);
15     $logo = se($_POST, "logo", "", false);
16
17     $errors = false;
18
19     // Server-side validation
20     if (empty($name)) {
21         flash("[server] Team is required", "warning");
22     }
23 }
```

```
22     $errors = true;
23 }
24
25 if (empty($nickname)) {
26     flash("[server] Nickname is required", "warning");
27     $errors = true;
28 }
29
```

screenshot of edit page code

Checklist Items (0)

```
24 Manan Thakore, 2 days ago * some validation fixes
25 if (empty($nickname)) {
26     flash("[server] Nickname is required", "warning"); // mmt 4/17/2024
27     $errors = true;
28 }
29
30 if (empty($city)) {
31     flash("[server] City is required", "warning");
32     $errors = true;
33 }
34
35 if (empty($logo)) {
36     flash("[server] Logo URL is required", "warning");
37     $errors = true;
38 }
39
40 if (!$errors) {
41
42     $db = getDB();
43     $query = "UPDATE `NBA_Teams` SET ";
44     $params = [];
45
46     foreach ($_POST as $k => $v) {
47         if ($params) {
48             $query .= ",";
49         }
50         $query .= "$k=:{$k}";
51         $params[":{$k}"] = $v;
52     } <- #46-52 foreach ($_POST as $k => $v)
53 }
```

screenshot of edit page code

Checklist Items (0)

```
54     $query .= " WHERE id=:id";
55     $params[":id"] = $id;
56
57     try {} // mmt 4/17/2024
58     $stmt = $db->prepare($query);
59     $stmt->execute($params); Manan Thakore, 2 days ago * some validation fixes
60     flash("Updated record", "success");
61 } <- #57-61 try
62 catch (PDOException $e) {
63     error_log("Something broke with the query" . var_export($e, true));
64 }
65 } <- #40-65 if (!$errors)
66 } <- #11-66 if ($_SERVER["REQUEST_METHOD"] === "POST")
67
68
69 $team = [];
70 if($id > -1){
71     // Fetch team data
72     $db = getDB();
73     $query = "SELECT name, nickname, city, logo FROM `NBA_Teams` WHERE id=:id";
74 }
```

```
74     try{
75         $stmt = $db->prepare($query);
76         $stmt->execute([":id" => $id]);
77         $team = $stmt->fetch();
78     } <- #75-79 try
79     catch(PDOException $e){
80         error_log("Error fetching record: " . var_export($e, true));
81         flash("Error fetching record", "danger");
82     }
```

screenshot of edit page code

Checklist Items (0)

```
public_html / project / admin / edit_teams.php / ...
81         error_log("Error fetching record: " . var_export($e, true));
82         flash("Error fetching record", "danger"); // mmt 4/17/2024
83     }
84 } <- #70-84 if($id > -1)
85 else{
86     flash("Invalid ID passed", "danger");
87     die(header("Location:" . get_url("admin/list_teams.php")));
88 }
89
90 if($team){
91     $form = [
92         ["type" => "text", "name" => "name", "placeholder" => "Team Name", "label" => "Team Name", "value" => $team["name"]],
93         ["type" => "text", "name" => "nickname", "placeholder" => "Team Nickname", "label" => "Team Nickname", "value" => $team["nickname"]],
94         ["type" => "text", "name" => "city", "placeholder" => "City", "label" => "City", "value" => $team["city"], "rules" => "required"],
95         ["type" => "text", "name" => "logo", "placeholder" => "Logo URL", "label" => "Logo URL", "value" => $team["logo"]],
96     ]; <- #91-96 $form =
97 } <- #90-97 if($team)
98
99 ?>
100
101 <div class="container-fluid">
102     <h3>Edit Team</h3>
103     <div>
104         <a href="=php echo get_url("admin/list_teams.php"); ?&gt;" class="btn btn-secondary"&gt;Back&lt;/a&gt;
105     &lt;/div&gt;
106     &lt;form method = "POST" onsubmit = "return validate(this);"&gt;
107         &lt;?php foreach($form as $field) : ?&gt;
108             &lt;?php render_input($field); ?&gt;
109             &lt;br&gt;</pre
```

screenshot of edit page code

Checklist Items (0)

```
<?php endforeach; // mmt 4/17/2024?>
<?php render_button(["text" => "Update", "type" => "submit"]); ?>      You, 19 hours ago * Uncommitted changes
</form>
</div>

<script>
    function validate(form) {
        const teamInput = form.name.value;
        const nicknameInput = form.nickname.value;
        const cityInput = form.city.value;
        const logoInput = form.logo.value;

        let isValid = true;

        if (!teamInput) {
            flash("[js] Team name is required", "warning");
            isValid = false;
        }
    }
</script>
```

```

if (!nicknameInput) {
    flash("[js] Team nickname is required", "warning");
    isValid = false;
}

if (!cityInput) {
    flash("[js] Team city is required", "warning");
    isValid = false;
}

```

screenshot of edit page code

Checklist Items (0)

```

function validate(form) {
    if (!nicknameInput) {
        flash("[js] Team nickname is required", "warning");
        isValid = false;
    }

    Manan Thakore, 3 days ago * new validation
    if (!cityInput) { // mmt 4/17/2024
        flash("[js] Team city is required", "warning");
        isValid = false;
    }

    if (!logoInput) {
        flash("[js] Team logo URL is required", "warning");
        isValid = false;
    }
    return isValid;
} <- #115-142 function validate(form)
</script>

?php require_once(__DIR__ . "/../../../../partials/flash.php"); ?>

```

screenshot of edit page code

Checklist Items (0)

Task #3 - Points: 1

Text: Screenshot of records from DB

Checklist			*The checkboxes are for your own tracking
#	Points	Details	
#1	1	Show a before and after screenshot of the record	
#2	1	Note what differs	

#3

1

Clearly caption screenshots**Task Screenshots:****Gallery Style: Large View****Small****Medium****Large**

NBA_Teams

Properties DATA Monitor

SELECT * FROM `NBA_Teams` LIMIT 100

	Q	P	id	name	nickname	city	logo	created
			int	varchar(100)	varchar(100)	varchar(100)	varchar(255)	timestamp
	>	1	2	denville eagles	eagles	denville	askhdfkhsdkf	2024-04-13 21:54:32
	>	2	3	aakash 49ers	49ers	san fran	asdifuishdifu	2024-04-13 22:17:27
	>	3	4	kishan colts	colts	ptown	asdofosdf	2024-04-13 22:17:45
	>	4	5	jig seahawks	seahawks	seattle	sdhfuiuwfiweif	2024-04-13 22:19:46
	>	5	24	New York Knicks	Knicks	New York	https://upload.wikimedia.or	2024-04-14 17:19:50
	>	6	30	Sacramento Kings	Kings	Sacramento	https://upload.wikimedia.or	2024-04-14 18:27:34
	>	7	40	Utah Jazz	Jazz	Utah	https://upload.wikimedia.or	2024-04-14 16:59:01

before denville eagles gets updated**Checklist Items (0)**

NBA_Teams

Properties DATA Monitor

SELECT * FROM `NBA_Teams` LIMIT 100

	Q	P	id	name	nickname	city	logo	created
			int	varchar(100)	varchar(100)	varchar(100)	varchar(255)	timestamp
	>	1	2	denville patriots	patriots	denville	askhdfkhsdkf	2024-04-13 21:54:32
	>	2	3	aakash 49ers	49ers	san fran	asdifuishdifu	2024-04-13 22:17:27
	>	3	4	kishan colts	colts	ptown	asdofosdf	2024-04-13 22:17:45
	>	4	5	jig seahawks	seahawks	seattle	sdhfuiuwfiweif	2024-04-13 22:19:46
	>	5	24	New York Knicks	Knicks	New York	https://upload.wikimedia.or	2024-04-14 17:19:50
	>	6	30	Sacramento Kings	Kings	Sacramento	https://upload.wikimedia.or	2024-04-14 18:27:34
	>	7	40	Utah Jazz	Jazz	Utah	https://upload.wikimedia.or	2024-04-14 16:59:01

after denville eagles gets updated to denville patriots

Checklist Items (0)

Task #4 - Points: 1

Text: Explain the process

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#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
#2	1	Briefly describe the validations for the applicable fields

Response:

Before proceeding to the editing interface, it checks the user's administrative permissions. Upon confirmation, the script retrieves the team's existing data from the database based on the provided ID. It then renders an editable form populated with the retrieved data, allowing users to modify the team's name, nickname, city, and logo URL. Server-side validation ensures that all fields are filled out before updating the record in the database. If any validation errors occur, relevant messages are displayed to guide the user. After successfully updating the record, a confirmation message appears.

Task #5 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

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

URL #1

https://mmt-prod-6beda717bbfa.herokuapp.com/Project/admin/edit_teams.php

URL #2

<https://github.com/mananthakore/mmt-it202-008/pull/62>

URL #3

<https://github.com/mananthakore/mmt-it202-008/pull/68>

URL #4

<https://github.com/mananthakore/mmt-it202-008/pull/69>

● Delete Handling (1 pt.)

[COLLAPSE ^](#)

● Task #1 - Points: 1

Text: Screenshots related to delete

[COLLAPSE ^](#)

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show the success message of a delete
#2	1	Show any error messages of a failed delete (like id not being passed)
#3	1	Show the code related to the delete processing

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

The screenshot shows a web browser displaying a list of NBA teams. The URL in the address bar is https://mmt-dev-0918b5efff5c.herokuapp.com/Project/admin/list_teams.php. The page has a header with 'DunkData' and navigation links for Home, Profile, Admin, Team Management, and Logout. A success message 'Deleted record with id 41' is displayed in a box. Below it is a table titled 'List NBA Teams' with columns for Team Name, City, Nickname, Logo (Link), and Limit. There are input fields for Team Name, City, Nickname, and Logo (Link), and a dropdown for Limit set to 10. Buttons for Filter and Clear are present. The main content area shows a table of NBA teams with columns for name, city, nickname, logo, and actions (View, Edit, Delete). Two rows are visible: 'denville eagles' and 'akash 49ers'. The 'denville eagles' row has a logo placeholder 'askhdfkhskdfk'. The 'akash 49ers' row has a logo placeholder 'asdifuishdifu'. Action buttons for 'View', 'Edit', and 'Delete' are shown for each row.

name	city	nickname	logo	Actions
denville eagles	denville	eagles	askhdfkhskdfk	View Edit Delete
akash 49ers	san fran	49ers	asdifuishdifu	View Edit Delete

success message of deleting parsippany giants team

Checklist Items (0)

The screenshot shows a web application interface. At the top, there is a header with links for 'Getting Started', 'Google', 'YouTube', 'DunkData', 'Home', 'Profile', 'Admin', 'Team Management', and 'Logout'. Below the header, a pink box displays the error message 'Invalid id passed to delete'. The main section is titled 'List NBA Teams' and contains a table with columns: Team Name, City, Nickname, Logo (Link), and Limit. There are input fields for filtering these columns and buttons for 'Filter' and 'Clear'. The table lists two teams: 'denville eagles' and 'akash 49ers'. Each team row has 'View', 'Edit', and 'Delete' buttons. The entire page has a light blue background.

Team Name	City	Nickname	Logo (Link)	Limit
Team Name	City	Nickname	Logo (Link)	10

Filter **Clear**

NBA Teams

name	city	nickname	logo	Actions
denville eagles	denville	eagles	askhdfkhsdkdfk	View Edit Delete
aakash 49ers	san fran	49ers	asdifuishdifu	View Edit Delete

screenshot of error message of a delete

Checklist Items (0)

The screenshot shows a code editor with a dark theme. The file is named 'delete_teams.php'. The code is a PHP script that checks if the user has an 'Admin' role. If not, it redirects to 'home.php'. It then checks if a valid team ID is provided. If not, it shows an error message and redirects to 'list_teams.php'. If a valid ID is provided, it attempts to delete the record from the 'NBA_Teams' database table. If successful, it shows a success message. If there is an error, it logs the error and shows a danger message. The code uses sessions, requires functions.php, and includes error handling for PDO exceptions.

```
public_html > Project > admin > delete_teams.php > ...
1  <?php
2
3  session_start();
4  require_once(__DIR__ . "/../../../../lib/functions.php"); // mmt 4/17/2024
5
6  // Check if the user has admin role, redirect if not      Manan Thakore, 4 days ago * added delete team feature
7  if (!has_role("Admin")) {
8      flash("You do not have permission to view this page", "warning");
9      die(header("Location: $BASE_PATH" . "/home.php"));
10 }
11
12 $id = se($_GET, "id", -1, false);
13
14 // Validate if the id is valid
15 if ($id < 1) {
16     flash("Invalid id passed to delete", "danger");
17     die(header("Location: " . get_url("admin/list_teams.php")));
18 }
19
20 // Delete team record
21 $db = getDB();
22 $query = "DELETE FROM `NBA_Teams` WHERE id=:id";
23 try {
24     $stmt = $db->prepare($query);
25     $stmt->execute([":id" => $id]);
26     flash("Deleted record with id $id", "success");
27 } catch (PDOException $e) {
28     error_log("Error deleting team $id: " . var_export($e, true));
29     flash("Error deleting record", "danger");
}
```

Checklist Items (0)



```

21 $db = getDB();
22 $query = "DELETE FROM `NBA_Teams` WHERE id=:id";
23 try {
24     $stmt = $db->prepare($query);
25     $stmt->execute([":id" => $id]);
26     flash("Deleted record with id $id", "success");
27 } catch (PDOException $e) {
28     error_log("Error deleting team $id: " . var_export($e, true));
29     flash("Error deleting record", "danger");
30 }
31
32 // Redirect to the list of teams page
33 die(header("Location: " . get_url("admin/list_teams.php"))); // mmt 4/17/2024
34

```

Checklist Items (0)



Task #2 - Points: 1

Text: Screenshots of the data

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show a before and after screenshot of the DB data
#2	1	Note what changed (i.e., record removed or soft delete value changed)
#3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Properties DATA Monitor

```
SELECT * FROM `NBA_Teams` LIMIT 100
```

Search results Cost: 78ms Total 7

	Q	id	name	nickname	city	logo	created
		int	varchar(100)	varchar(100)	varchar(100)	varchar(255)	timestamp
>	1	2	denville eagles	eagles	denville	askhdfkhsksdfk	2024-04-13 21:54:32
>	2	3	aakash 49ers	49ers	san fran	asdifuishdifu	2024-04-13 22:17:27
>	3	4	kishan colts	colts	ptown	asdofoosdf	2024-04-13 22:17:45
>	4	5	jig seahawks	seahawks	seattle	sdhfuiuwfiweif	2024-04-13 22:19:46
>	5	24	New York Knicks	Knicks	New York	https://upload.wikimedia.or	2024-04-14 17:19:50
>	6	30	Sacramento Kings	Kings	Sacramento	https://upload.wikimedia.or	2024-04-14 18:27:34
>	7	40	Utah Jazz	Jazz	Utah	https://upload.wikimedia.or	2024-04-14 16:59:01
>	8	41	parsippany giants	giants	parsippany	assbkdfbksajkfsj	2024-04-17 18:08:40

before parsippany giants gets deleted

Checklist Items (0)

NBA_Teams x

Properties DATA Monitor

```
SELECT * FROM `NBA_Teams` LIMIT 100
```

Search results Cost: 24ms Total 6

	Q	id	name	nickname	city	logo
		int	varchar(100)	varchar(100)	varchar(100)	varchar(255)
>	1	2	denville eagles	eagles	denville	askhdfkhsksdfk
>	2	3	aakash 49ers	49ers	san fran	asdifuishdifu
>	3	4	kishan colts	colts	ptown	asdofoosdf
>	4	5	jig seahawks	seahawks	seattle	sdhfuiuwfiweif
>	5	24	New York Knicks	Knicks	New York	https://upload.wikimedia.or
>	6	30	Sacramento Kings	Kings	Sacramento	https://upload.wikimedia.or
>	7	40	Utah Jazz	Jazz	Utah	https://upload.wikimedia.or

after parsippany giants gets deleted

Checklist Items (0)

  Task #3 - Points: 1
Text: Explain the delete logic

Checklist

*The checkboxes are for your own tracking

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

Response:

The code executes a hard delete operation, permanently removing records from the database without retention. It restricts deletion functionality to users with the "Admin" role, ensuring only privileged users can access it. The script begins by checking the user's permissions and validating the ID parameter passed through the URL to ensure it's a valid integer greater than 0. Following this, it connects to the database and prepares a DELETE query to remove the specified record from the NBA_Teams table. Upon execution of the DELETE query, if successful, a success message is flashed to the user, confirming the deletion. In case of any errors during the deletion process, they are logged, and an error message is displayed to the user.

  Task #4 - Points: 1
Text: Add the pull request link for the branch related to this feature

 **Details:**

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

URL #1

<https://github.com/mananthakore/mmt-it202-008/pull/61>

 Misc (1 pt.)



  Task #1 - Points: 1
Text: Screenshot of your project board from GitHub (tasks should be in the proper column)

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

This screenshot shows a GitHub project board with three columns: Todo, In Progress, and Done.

- Todo:** 0 items. Description: "This item hasn't been started".
- In Progress:** 0 items. Description: "This is actively being worked on".
- Done:** 16 items. Description: "This has been completed".
 - mmt-it202-008 #51 MS2 - Data Creation Page
 - mmt-it202-008 #54 ... MS2 - API Handling
 - mmt-it202-008 #52 MS2 - Data List Page (many items)

At the bottom of each column are "+ Add item" buttons.

screenshot of project board

This screenshot shows a GitHub project board with three columns: Todo, In Progress, and Done.

- Todo:** 0 items. Description: "This item hasn't been started".
- In Progress:** 0 items. Description: "This is actively being worked on".
- Done:** 16 items. Description: "This has been completed".
 - mmt-it202-008 #56 MS2 - Delete Handling
 - mmt-it202-008 #55 ... MS2 - View Data Details Page (single item)
 - mmt-it202-008 #53 MS2 - Edit Data Page

At the bottom of each column are "+ Add item" buttons.

screenshot of project board

it202 2024 project

View 1 + New view

Filter by keyword or by field

Add status update Discard Save

Todo 0

This item hasn't been started

+ Add item

In Progress 0

This is actively being worked on

+ Add item

Done 16

This has been completed

- MS2 - View Data Details Page (single item)
- MS2 - Edit Data Page
- MS2 - Define the appropriate table or tables for your API

+ Add item

<https://github.com/mananthakore/mmt-it202-008/issues/50>

screenshot of project board

Task #2 - Points: 1

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

URL #1

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

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

The hardest issue I faced while working on the project was the php and JS validation. However, I realized that the issue I was facing was due to incorrect variable names and small minor errors. Another issue I was facing was my create page not adding a team when it should be. I was able to fix that error as well. Overall, Milestone 2 was very helpful as it helped me gain a better understanding of how APIs work and how to implement them in our own projects.

Task #4 - Points: 1

Text: WakaTime Screenshot

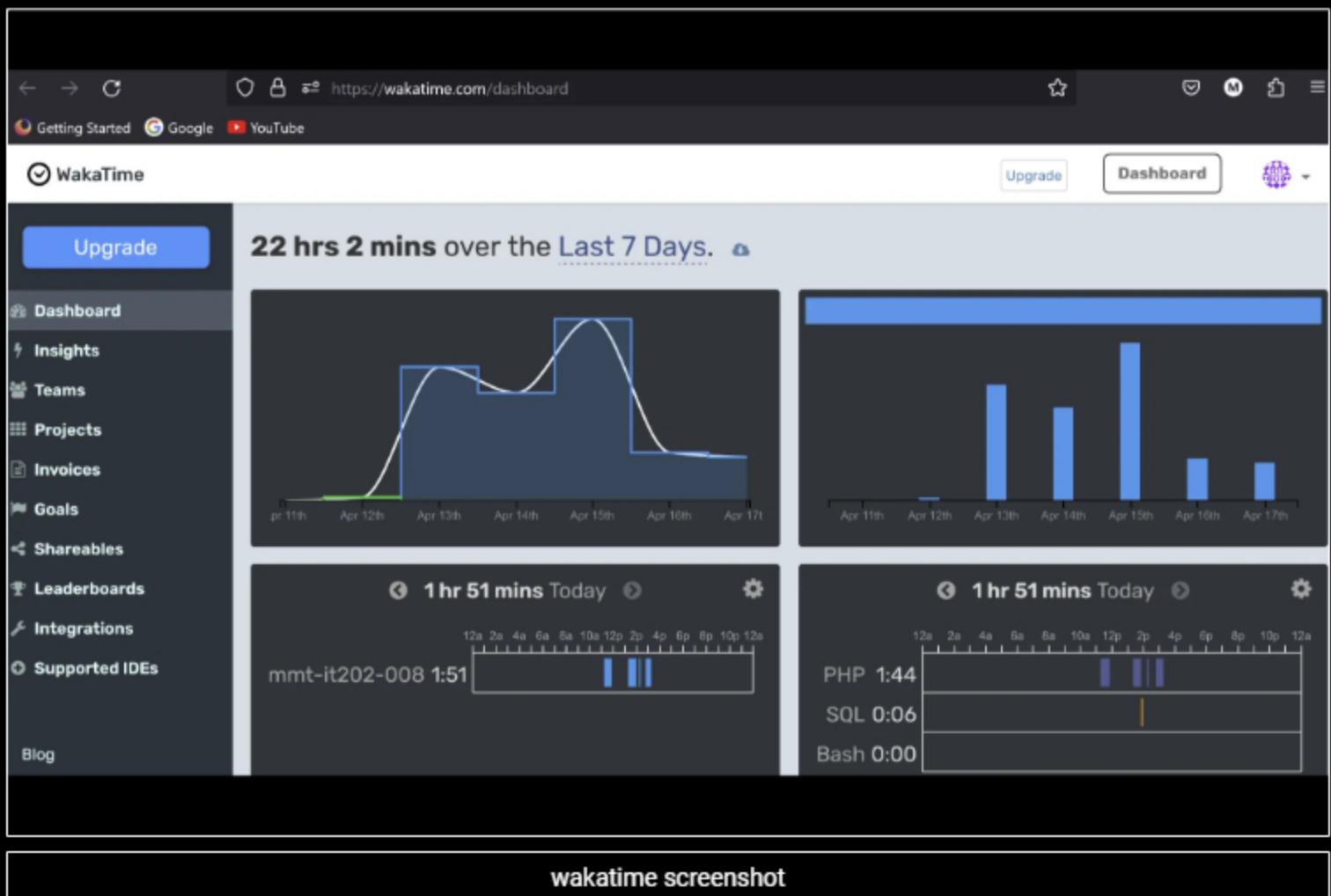
Details:

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