

# Submission Worksheet

**CLICK TO GRADE**

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-2-2024/grade/lm457>

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

### Submissions:

Submission Selection

1 Submission [active] 4/15/2024 6:10:44 PM

### 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
<input checked="" type="checkbox"/> #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
<input checked="" type="checkbox"/> #2	1	Columns should be logical and thought out (not valid to have a single field of JSON data or similar)
<input checked="" type="checkbox"/> #3	1	Clearly caption screenshots

Task Screenshots:

#### Gallery Style: Large View

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

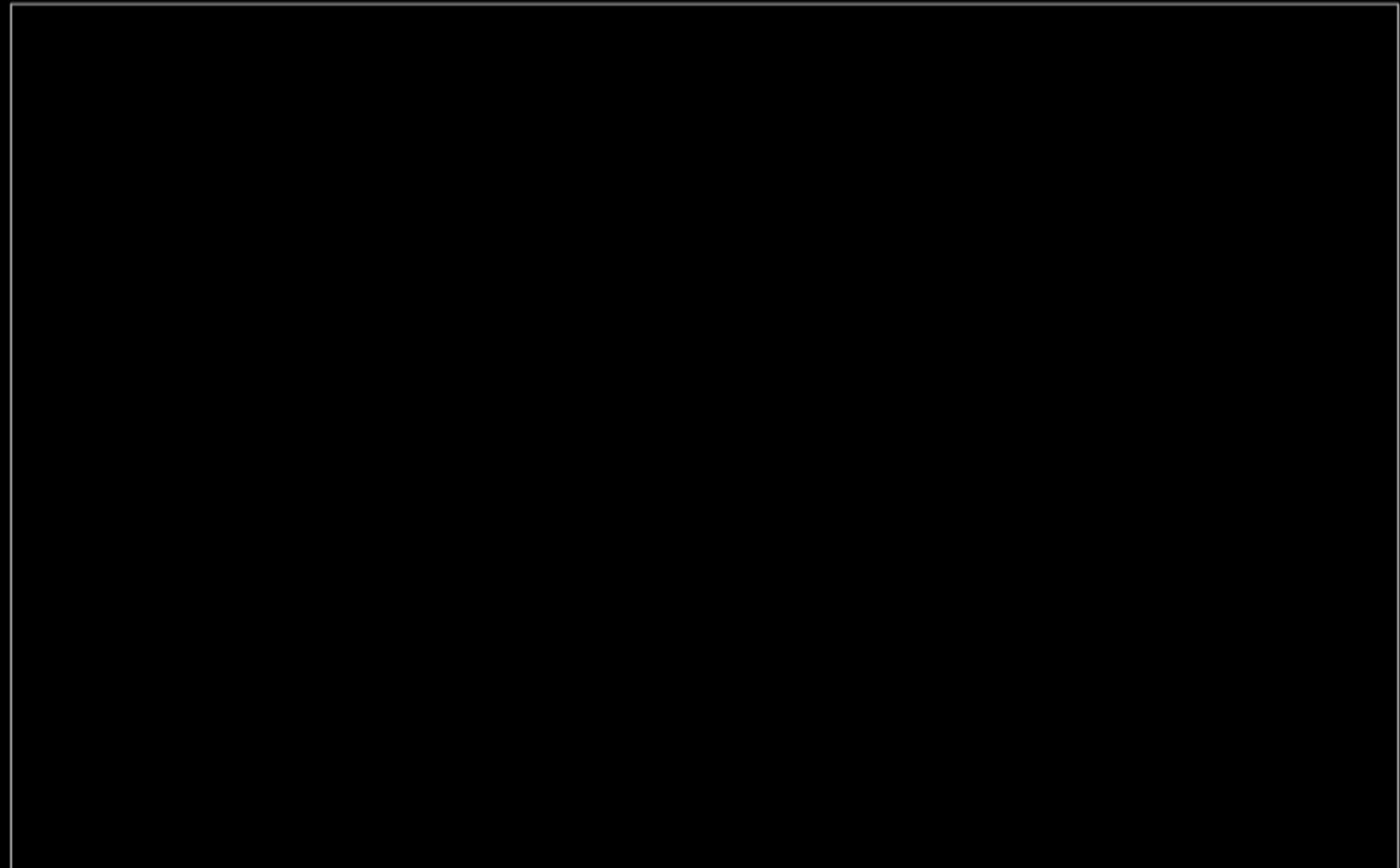


Table with initial ID column plus additional API data columns.

#### 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

## Table with more API Data columns.

### 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

SELECT * FROM tourist_info LIMIT 100														
friday_open	friday_close	saturday_open	saturday_close	sunday_open	sunday_close	popular_tour_title	primary_category	price	partner	tour_url	product_code			
720	1440	720	1440	720	1440	The High Roller at The LINC	Attraction Tickets	\$28.89	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	5084LASHIG			
(NULL)	(NULL)	600	840	600	840	Lion Habitat Ranch Admissi	Attraction Tickets	\$25.65	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	6072P1			
900	1440	900	1440	720	1440	Las Vegas Hop-On Hop-Off	Hop-on Hop-off Tours	\$55.00	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	5096LASHOP			
540	1260	540	1260	540	1260	Las Vegas The Mob Museu	Museum Tickets & Passes	\$34.95	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	5932MOB			
720	1440	720	1440	720	1440	Eiffel Tower Viewing Deck A	Attraction Tickets	\$28.75	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	5084LASEIF			
720	1560	600	1560	600	1500	STRAT SkyPod Observation	Attraction Tickets	\$24.00	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	9412P6			
0	1439	0	1439	0	1439	Where the Spots Are: Las V	Walking Tours	\$49.00	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	138742P1			
0	1439	0	1439	0	1439	Hoover Dam Tour from Las	4WD, ATV & Off-Road Tou	\$74.00	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	3951HDT			
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	Las Vegas Hop-On Hop-Off	Hop-on Hop-off Tours	\$55.00	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	5096LASHOP			
360	1140	360	1140	360	1140	Scooter Tours of Red Rock C	Vespa, Scooter & Moped Tou	\$135.00	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	14272P1			
660	1080	660	1080	660	1080	Titanic: The Artifact Exhibit	Attraction Tickets	\$38.99	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	5156TITANIC			
600	1380	600	1440	600	1440	High Roller World's Largest	Attraction Tickets	\$28.89	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	5084LASHIG			
0	1439	0	1439	0	1439	Las Vegas Downtown and F	Historical & Heritage Tou	\$36.37	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	6756FCCWT			
570	1020	570	1020	480	1020	Count Kustom's Las Vegas C	Half-day Tours	\$79.99	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	7496P2			
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	Dolce Springs 1-Hour Priv	4WD, ATV & Off-Road Tou	\$265.65	Viator	<a href="https://www.tripadvisor.com">https://www.tripadvisor.com</a>	107180P7			
700	1400	700	1400	700	1400	Seven Day Seventy Acre Blk	Bike Tour	\$99.99	Trivago	<a href="http://www.trivago.com/777acres">www.trivago.com/777acres</a>	3743HD3842			

Table with even more API Data columns.

### 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

* is_api tinyint(1)	* created timestamp	* modified timestamp
0	2024-04-14 02:39:39	2024-04-15 21:04:38
0	2024-04-14 02:56:23	2024-04-14 02:56:23
0	2024-04-14 02:56:57	2024-04-14 02:56:57
0	2024-04-14 02:57:50	2024-04-14 02:57:50
0	2024-04-14 02:59:21	2024-04-14 02:59:21
0	2024-04-14 03:00:51	2024-04-14 03:00:51
0	2024-04-14 03:13:19	2024-04-14 03:13:19
0	2024-04-14 03:16:58	2024-04-14 03:16:58
0	2024-04-14 03:34:23	2024-04-14 03:34:23
0	2024-04-14 15:28:56	2024-04-14 15:28:56
0	2024-04-14 15:29:45	2024-04-15 00:58:05
1	2024-04-15 01:07:07	2024-04-15 01:07:07
1	2024-04-15 01:18:05	2024-04-15 01:18:05
1	2024-04-15 01:18:35	2024-04-15 01:18:35
1	2024-04-15 01:19:19	2024-04-15 01:19:19
0	2024-04-15 15:28:38	2024-04-15 15:28:38

Table with the necessary created and modified columns as well as an is\_api column for differentiating between API data and manually created data.

### 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

Points		
#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:

I picked a tourist location API that fetches an incredible amount of data about popular destinations across the world based on a location\_id a user passes that is associated with a specific location. The raw data output of my API was very extensive so I made the deliberate choice to only pull in what I perceived to be the most relevant data. For starters, each location contains necessary parameters in the form of location ID, language, currency, and national ID. Apart from those, the data fields I pulled in were the locations' names, descriptions, ranking, reviews, ratings, contact info, timings, popular tours, and their associated info. Having the name and description form fields are pretty self-explanatory in that they're the most essential details about a specific tourist location. Then, the addition of ranking, reviews, and ratings allows users to gain a more thorough understanding of the locations and what they have to offer and to just see what other people have thought about them. Contact info is for allowing users to do their own research about the place and timings allowing users to make appropriate travel decisions. Popular tour info gives users an idea of possible excursions and activities available at a specific location. Although there is still an abundance of data I could retrieve from my API, I found these to be the most crucial for building the foundation of my travel guide app, with possible inclusions later down the line.

Regarding normalization, I only needed a single table to store my data and didn't have to normalize it to fit into several tables. However, I will note that the Popular tour info actually contained several popular tours available at a location rather than just one for a large number of locations, but for the sake of the complexity of my Web App, I decided I'd only choose the top most tour. But if I wanted to implement that feature, it would require the use of data normalization into separate tables, with the new popular tours table holding the id of the tourist\_info table as its foreign key to match all of the tours available for a specific location.

#### 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/Kith07/spring2024-IT202/pull/51>

#### Data Creation Page (2 pts.)

[^COLLAPSE ^](#)

#### Task #1 - Points: 1

Text: Screenshots of the creation page

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

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

National LocationID  
88888

Location Name  
Eighty Eight Land

Location Ranking  
#1 in EightLand

Location Description  
Land filled with 8s

Location Rating  
5.0

Location Number of Reviews  
8888

Location Website  
www.888.com

Location Address  
888 Eighty Lane

Location Phone  
+1 888-888-8888

**Part 1 of my creation page with valid data filled in the form fields.****Checklist Items (1)**

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

Write a Review Link  
www.review888.com

Location Monday Close Time  
1600

Location Tuesday Open Time  
800

Location Tuesday Close Time  
1600

Location Wednesday Open Time  
800

Location Wednesday Close Time  
1600

Location Thursday Open Time  
800

Location Thursday Close Time  
1600

Location Friday Open Time  
800

Part 2 of the creation page with valid data filled in for the form fields.

### Checklist Items (1)

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

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "it202-Im457" and contains a form for creating a tourist attraction. The form fields include:

- Location Friday Close Time: 1600
- Location Saturday Open Time: 800
- Location Saturday Close Time: 1600
- Location Sunday Open Time: 800
- Location Sunday Close Time: 1600
- Location Popular Tours Title: EightyLand Sky Diving Tour
- Location Popular Tour Category: Sky Diving
- Location Popular Tour Price: \$88.88
- Location Popular Tour Partner: Trivago
- Location Popular Tour Link: www.trivago.com/B88
- Location Popular Tour Product Code: BSGDGF888

At the bottom of the form is a blue button labeled "Add Tourist Place".

Part 3 of the creation page with valid data for the form fields.

### Checklist Items (1)

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

Stuff IT202 [Milestone] IT202-Gen Learn r022-Im4 r022-Im4 Untitled d Tourist Attr (500) Disc Projects

https://it202-im457-dev-7b44497d46f0.herokuapp.com/project/admin/create\_touristAPI.php

Tourist Attractions Home Profile Admin Tourist Locations Logout

### Fetch Tourist Location

National LocationID

**Fetch Details**

Fetch API page with valid data in form field.

#### Checklist Items (1)

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

Stuff IT202 [Milestone] IT202-Gen Learn r022-Im4 r022-Im4 Untitled d Tourist Attr (500) Disc Projects

https://it202-im457-dev-7b44497d46f0.herokuapp.com/project/admin/create\_touristAPI.php

Tourist Attractions Home Profile Admin Tourist Locations Logout

Inserted Successfully 54

### Fetch Tourist Location

National LocationID

**Fetch Details**

**Checklist Items (1)****#2 Show how the API data is fetched for API data (must be server-side)**

The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is titled "it202-lm457" and displays a URL: [https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/create\\_touristManual.php](https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/create_touristManual.php). The page content is a form for adding a tourist location. At the top, there is a green success message box containing the text "Inserted Successfully 48". Below the message, the form fields are listed: National LocationID, Location Name, Location Ranking, Location Description, Location Rating, Location Number of Reviews, Location Website, Location Address, Location Phone, Write a Review Link, and Location Monday Open Time. All fields are empty or have placeholder text. The bottom of the page has a dark footer bar.

Creation page successfully inserts manual data into table.

**Checklist Items (1)****#3 Show examples of validation messages**

The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is titled "it202-lm457" and displays a URL: [https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/create\\_touristAPI.php](https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/create_touristAPI.php). The page content is a form for fetching a tourist location. A yellow validation message box at the top states "Tourist Place Not Found". Below the message, the form fields are listed: National LocationID, Enter ID, and Fetch Details. The "Enter ID" field is empty. The "Fetch Details" button is blue with white text.

## Fetch API Page validation for invalid location ID

### Checklist Items (1)

#3 Show examples of validation messages

Part 1 of Creation Page JS Validation for required form fields when passed with empty values.

### Checklist Items (1)

#3 Show examples of validation messages

## Part 2 of the Creation Page JS Validation with required form fields passed with empty values.

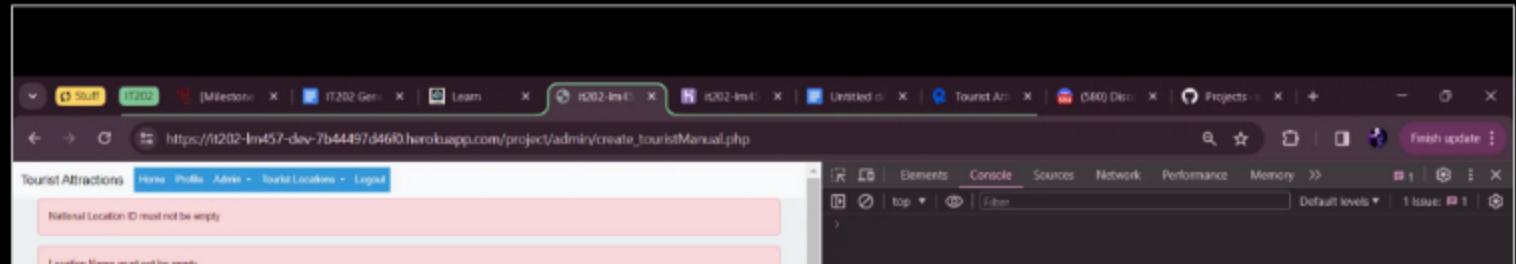
### Checklist Items (1)

#3 Show examples of validation messages

## Part 3 of Creation Page JS Validation for required form fields when passed with empty values.

### Checklist Items (1)

#3 Show examples of validation messages



Location Name must not be empty

Location Ranking must not be empty

Location Description must not be empty

Location Rating must not be empty

Location Number of Reviews must not be empty

Location Website must not be empty

Location Address must not be empty

Location Phone must not be empty

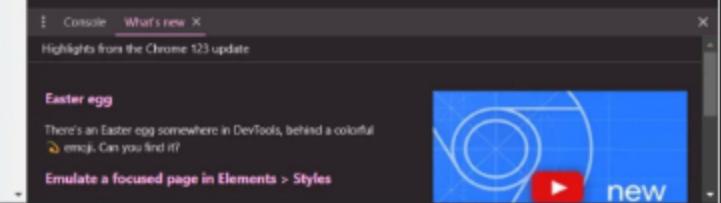
Write a Review Link must not be empty

Location Monday Open Time must not be empty

Location Monday Close Time must not be empty

Location Tuesday Open Time must not be empty

Location Tuesday Close Time must not be empty



## Part 1 of Creation Page PHP Validation with required form fields passed with empty values.

### Checklist Items (1)

#3 Show examples of validation messages

## Part 2 of Creation Page PHP Validation with required form fields passed with empty values.

### Checklist Items (1)

#3 Show examples of validation messages

Location Popular Tour Product Code must not be empty

An error occurred

### Add Tourist Location

National LocationID  
Enter ID

Location Name  
Enter Location Name

Location Ranking  
Enter Location Ranking

Location Description  
Enter Location Description

Location Rating  
Enter Location Rating

Location Number of Reviews  
Enter Location Number of Reviews

Location Website  
Enter Location Website

Location Address  
Enter Location Address

Location Phrase  
Enter Location Phrase

Write a Review Link  
Enter Write a Review Link

Console What's new X

Highlights from the Chrome 123 update

Easter egg

There's an Easter egg somewhere in DevTools, behind a colorful 🌈 emoji. Can you find it?

Emulate a focused page in Elements > Styles

new

### Part 3 of the Creation Page PHP Validation with required form fields passed with empty values.

#### Checklist Items (1)

##### #3 Show examples of validation messages

Duplicate Entry

### Fetch Tourist Location

National LocationID  
Enter ID

Patch Details

### Checklist Items (1)

#3 Show examples of validation messages

Creation Page Validation to ensure another record with the same location ID isn't created in our database due to the column's unique constraint.

### Checklist Items (1)

#3 Show examples of validation messages

Task #2 - Points: 1

Text: Screenshots of creation page code

### Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Form should have correct data types for each property being requested
<input checked="" type="checkbox"/> #2	1	Form should have correct validation for each field (HTML, JS, and PHP)
<input checked="" type="checkbox"/> #3	1	Successful creation should have a user-friendly message
<input checked="" type="checkbox"/> #4	1	Any errors should have user-friendly messages
<input checked="" type="checkbox"/> #5	1	Include the form/process for fetching API data
<input checked="" type="checkbox"/> #6	1	Include some indicator between custom data and API data

		INCLUDES SOME INSTRUCTIONS ON HOW TO REVIEW THIS CHECKLIST
#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



```
<div class="container-fluid">
  <h3>Fetch Tourist Location</h3>
  <form method="POST">
    <?php render_input([{"type" => "text", "name" => "nationallocation_id", "placeholder" => "Enter ID", "label" => "National LocationID", "rules" => ["required" => "required"]}); ?>
    <?php render_button(["text" => "Fetch Details", "type" => "submit"]); ?>
  </form>
</div>
```

Fetch API Page's form code with appropriately defined data types for each required property. Also has HTML validation through the required input type.

### Checklist Items (1)

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

Creation Data Page with appropriately defined data types for each requested property. Also has HTML validation through the use of the input type required attribute.

#### Checklist Items (2)

#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)

```
<?php
if (isset($_POST["nationallocation_id"])) {
    $nationalLocID = se($_POST, "nationallocation_id", "", false);
    if ($nationalLocID) {
        $selectedPlace = fetch_touristPlace($nationalLocID);
        if (!$selectedPlace){
            flash("Tourist Place Not Found", "warning");
        }
        error_log("API Data" . var_export($selectedPlace, true));
    } else {
        flash("Invalid National ID", "warning");
    }
}
```

Fetch API Page with PHP validation for the required form field. Also has user friendly error message in case of an invalid id that is passed.

#### Checklist Items (1)

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

**Creation Page JS Validation** for each form field ensuring that no empty field is passed as an acceptable value with user-friendly error messages.

**Checklist Items (1)**

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

**Creation Page PHP Validation** that ensures another round of verification on the server side to disallow empty values for

the form fields with user-friendly error messages.

#### Checklist Items (1)

#3 Successful creation should have a user-friendly message

"Inserted Successfully" message upon correct insertion of data from an API call on the Fetch API Page from an admin user into the database. Also has user friendly error messages for duplicate entry and improper API calls.

#### Checklist Items (1)

#3 Successful creation should have a user-friendly message

**"Inserted Successfully"** message upon correct insertion of data from a manual insertion on the Create Page by an admin user into the database. Also has user-friendly error messages for duplicate entries and improper passage of queries.

#### Checklist Items (1)

#3 Successful creation should have a user-friendly message

The fetch function used in the Fetch API page to make an API call and retrieve the appropriate information based on the location Id that is passed.

#### Checklist Items (1)

#5 Include the form/process for fetching API data

**Data Insertion on the Fetch API page that showcases the insertion of the value "1" to the is\_api column that indicates whether the data in the database is from the API or manually created by an admin user.**

#### Checklist Items (1)

#6 Include some indicator between custom data and API data

```
<?php
//note we need to go up 1 more directory
require(__DIR__ . "/../../../../partials/nav.php");

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

Role check to ensure that the Fetch API page can only be accessed by an Admin user.

#### Checklist Items (2)

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

#8 Include ucid/date comments for each code screenshot

```
<?php  
//note we need to go up 1 more directory  
require(__DIR__ . "/../../../../partials/nav.php");  
  
if (!has_role("Admin")) { //UCID: LM457  
    flash("You don't have permission to view this page", "warning"); //DATE: 4/16/2024  
    die(header("Location: $BASE_PATH" . "/home.php"));  
}  
?>
```

Role check on the Creation Page to ensure that only an Admin user has access to manually create entries for our table.

#### Checklist Items (2)

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

#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
<input checked="" type="checkbox"/> #1	1	Show at least one record fetched from the API
<input checked="" type="checkbox"/> #2	1	Show at least one record created via the creation form
<input checked="" type="checkbox"/> #3	1	Note what differs
<input checked="" type="checkbox"/> #4	1	Clearly caption screenshots

#### Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Records fetched from my API as denoted by the 1 value in the is\_api column.

### **Checklist Items (1)**

## #1 Show at least one record fetched from the API

## Checklist Items (1)

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

## 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	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)
<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:

When a user is on the fetch API page, they're presented with a form field for the required parameter of my tourist location API, "national ID". The form field is ensured to be required and not empty, to prevent false API calls. Once the user enters a value for the National ID field and submits the page, the data value in the field is retrieved using a POST request. Once PHP receives the form data, it uses a function that makes an API call from a remote online API (tourist locations). If the entered value for National ID is valid and exists in the API data, which otherwise would print a user-friendly invalid ID error or Tourist location not found error respectively, it sends the data to my PHP script which now uses a SQL query to insert the retrieved data into my database table with pre-defined columns corresponding to the data I'm pulling in. While inserting, if PHP encounters a 1062 error, denoting duplicate value, it delivers a user-friendly message stating that a location with such a national ID already exists within the database, due to the unique constraint implemented on the national ID column. Otherwise, the query is successfully passed and the API data is appropriately filled into the columns of my database and the user receives a success message.

As for the creation page, once a user makes their way here, they are prompted with several form fields containing all the required data necessary for the successful creation of a tourist location followed by its successful insertion into my database. Once again, after the user enters a value and submits the form, the PHP script ensures that the field value is set and retrieves the form data from a POST request. Next up, an empty array to contain the passed-in custom data is declared and the form data retrieved from the POST request is parsed through in key-value pairs with keys denoting the name of each field and the value denoting the actual value stored to those keys. The PHP script then implements server-side validation using an array containing the names and labels of all required fields and parsing through that array in key-value pairs again to ensure all of them are not empty. Once all fields are appropriately filled in and our previously empty array now contains all of the user data, an SQL insert query is used to inject all of the data into the matching database table columns, with successful insertion leading to a user-friendly success message, and any errors to prompt user-friendly error messages. Particularly noteworthy among those error messages, once again is the duplicate entry warning which ensures that no customer tourist location is created with a national ID that is already associated with an existing location within my database since the national ID column is made to be unique in my table.

Apart from the PHP server-side validations, which are usually a last resort, JS validations on the client side also take place using code embedded within script tags. Using an event listener function, JS ensures that once a form is submitted, and HTML validation is disabled temporarily, all required form fields do not accept empty values and database insertion doesn't take place. The first initial layer of validation is done by HTML, where the form code contains the input type attribute 'required' for all necessary form fields to ensure an empty form isn't submitted in the first place.

### 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://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/create\\_touristAPI.php](https://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/create_touristAPI.php)

#### URL #2

[https://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/create\\_touristManual.php](https://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/create_touristManual.php)

#### URL #3

<https://github.com/Kith07/spring2024-IT202/pull/52>

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

## Task Screenshots:

## Gallery Style: Large View

Small      Medium      Large



## List of Tourist Locations Data

LocationID	Name	Ranking	Rating	Min_Reviews	Website	Address	Phone	Created	Actions
6637961	The LINQ Promenade	#21 of 746 things to do in Las Vegas	4.5	3347	https://www.caesars.com/linq	3545 Las Vegas Blvd S between The LINQ Hotel & Casino and Flamingo Las Vegas, Las Vegas, NV 89169-0921	+1 702-322-1543	2024-04-15 01:17:07	<button>View</button> <button>Edit</button> <button>Delete</button>
902614	Fremont Street Experience	#11 of 746 things to do in Las Vegas	4.0	46217	http://vegasexperience.com/viva-vision-light-show	425 Fremont Street, Las Vegas, NV 89101-6626	+1 702-678-5689	2024-04-15 01:18:05	<button>View</button> <button>Edit</button> <button>Delete</button>
2409034	Shelly American, Inc.	#15 of 746 things to do in Las Vegas	4.5	1647	http://shellyamerican.com/	5405 E Bonanza Rd, Las Vegas, NV 89119-3211	+1 702-942-7325	2024-04-15 01:18:26	<button>View</button> <button>Edit</button> <button>Delete</button>
135665	Mount Charleston	#13 of 746 things to do in Las Vegas	4.5	1456	http://www.goincharleston.com	Robbers Roost Trail, Las Vegas, NV 89109	+1 702-672-5488	2024-04-15 01:19:15	<button>View</button> <button>Edit</button> <button>Delete</button>
7777777	Serenty Seven Acres	#1 in SevenLand	4.9	7777	www.777acres.com	777 Serenty Lane, Sacramento, CA	+1 777-777-7777	2024-04-15 22:37:02	<button>View</button> <button>Edit</button> <button>Delete</button>
888888	Eighty Eight Land	#1 in EightLand	5.0	8888	www.888.com	888 Eighty Lane	+1 888-888-8888	2024-04-15 22:42:25	<button>View</button> <button>Edit</button> <button>Delete</button>
501163	KA by Cirque du Soleil	#56 of 746 things to do in Las Vegas	4.5	8615	http://www.mgmgrand.com/entertainment/ka-cirque-du-soleil-show.aspx	3799 Las Vegas Blvd S MGM Grand, Las Vegas, NV 89109-4319	+1 702-531-1825	2024-04-15 23:38:01	<button>View</button> <button>Edit</button> <button>Delete</button>
1224568	Mirage Volcano	#37 of 746 things to do in Las Vegas	4.5	6631	http://www.mirage.com/attractions/volcano.aspx	3400 Las Vegas Blvd S Mirage Hotel & Casino, Las Vegas, NV 89109-0967	+1 702-791-7111	2024-04-15 23:43:28	<button>View</button> <button>Edit</button> <button>Delete</button>



List Tourist Locations Page that displays all of my records in the database including both those fetched from the API and those that were manually created.

## Checklist Items (1)

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



## Tourist Locations Filter

ID Search	Name Search	Minimum Rating	Minimum Reviews	Minimum Date	Maximum Date	Sort	Order	Limit
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button"/>	<input type="button"/>	<input type="button"/>
<input type="button" value="Filter"/>	<input type="button" value="Clear"/>							

List Filter Form based on the select columns I found applicable from my API as well as the required sort, order, and limit fields.

### Checklist Items (2)

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

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

The screenshot shows a web browser window with the URL [https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/list\\_touristLocs.php?NationalID=6637961&name=&min\\_rating=&min\\_num\\_reviews=&date\\_min=&date\\_max=&sort\\_by=Rating&order=desc&limit=10](https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/list_touristLocs.php?NationalID=6637961&name=&min_rating=&min_num_reviews=&date_min=&date_max=&sort_by=Rating&order=desc&limit=10). The browser has multiple tabs open, including 'Staff', 'IT202', 'Milestone', 'IT202 Gen...', 'Learn', 'it202-lm457', 'it202-lm457', 'it202-lm457', 'Untitled d...', 'Tourist Attr...', 'Pull requ...', and 'Finish update'. The main content area is titled 'Tourist Locations Filter' and contains a form with fields for ID Search (6637961), Name Search, Minimum Rating (Minimum Rating), Minimum Reviews (Minimum Reviews), Minimum Date (mm/dd/yyyy), Maximum Date (mm/dd/yyyy), Sort (Rating), Order (desc), and Limit (10). Below the form is a table titled 'List of Tourist Locations Data' with columns: NationalID, name, ranking, rating, num\_reviews, website, address, phone, created, and Actions. It shows one row for 'The LINQ Promenade' with NationalID 6637961, name 'The LINQ Promenade', ranking #21 of 745 things to do in Las Vegas, rating 4.5, reviews 3347, website <https://www.caesars.com/linq>, address '3645 Las Vegas Blvd S between The LINQ Hotel & Casino and Flamingo Las Vegas, Las Vegas, NV 89109-0521', phone '+1 702-322-0543', created '2024-04-15 01:07:07', and Actions buttons for View, Edit, and Delete.

Search Filter based on the Location ID of a tourist location.

### Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

The screenshot shows a web browser window with the URL [https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/list\\_touristLocs.php?NationalID=&name=F&min\\_rating=&min\\_num\\_reviews=&date\\_min=&date\\_max=&sort\\_by=Rating&order=desc&limit=10](https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/list_touristLocs.php?NationalID=&name=F&min_rating=&min_num_reviews=&date_min=&date_max=&sort_by=Rating&order=desc&limit=10). The browser has multiple tabs open, including 'Staff', 'IT202', 'Milestone', 'IT202 Gen...', 'Learn', 'it202-lm457', 'it202-lm457', 'it202-lm457', 'Untitled d...', 'Tourist Attr...', 'Pull requ...', and 'Finish update'. The main content area is titled 'Tourist Locations Filter' and contains a form with fields for ID Search (ID Search), Name Search (F), Minimum Rating (Minimum Rating), Minimum Reviews (Minimum Reviews), Minimum Date (mm/dd/yyyy), Maximum Date (mm/dd/yyyy), Sort (Rating), Order (desc), and Limit (10). Below the form is a table titled 'List of Tourist Locations Data' with columns: NationalID, name, ranking, rating, num\_reviews, website, address, phone, created, and Actions. It shows one row for 'Flamingo Las Vegas' with NationalID 6637961, name 'Flamingo Las Vegas', ranking #21 of 745 things to do in Las Vegas, rating 4.5, reviews 3347, website <https://www.caesars.com/flamingo>, address '3645 Las Vegas Blvd S between The LINQ Hotel & Casino and Flamingo Las Vegas, Las Vegas, NV 89109-0521', phone '+1 702-322-0543', created '2024-04-15 01:07:07', and Actions buttons for View, Edit, and Delete.

Filter Clear

### List of Tourist Locations Data

LocationID	name	rankMap	rating	num_reviews	website	address	phone	created	Actions
902614	Fremont Street Experience	#11 of 745 things to do in Las Vegas	4.0	45217	<a href="http://vegosexperience.com/las-vegas-light-show">http://vegosexperience.com/las-vegas-light-show</a>	425 Fremont Street, Las Vegas, NV 89101-5629	+1 702-478-5800	2024-04-15 01:18:05	<button>View</button> <button>Edit</button> <button>Delete</button>



Partial Name Search Filter based on whether the inputted letter is contained within the name of the tourist location.

### Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

Filter by Minimum Rating of 4.9 showcasing only two existing records of tourist location with at least a rating of 4.9 out of 5

### Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

All records based on a minimum reviews filter of 2000 showcasing only those tourist locations that have at least a minimum of 2000 reviews.

## Checklist Items (1)

### #3 Demonstrate a few varied filters/sorts

NationalID	name	ranking	rating	num_reviews	website	address	phone	created	Actions
102514	Fremont Street Experience	#11 of 746 things to do in Las Vegas	4.0	48217	http://vegasexperience.com/viva-vision-light-show/	425 Fremont Street, Las Vegas, NV 89101-5620	+1 702-578-5666	2024-04-15 01:18:05	<button>View</button> <button>Edit</button> <button>Delete</button>
9637941	The LINQ Promenade	#21 of 746 things to do in Las Vegas	4.5	3047	http://www.caesars.com/linq	3645 Las Vegas Blvd S between The LINQ Hotel & Casino and Flamingo Las Vegas, Las Vegas, NV 89109-0021	+1 702-322-0543	2024-04-15 01:07:07	<button>View</button> <button>Edit</button> <button>Delete</button>
2409634	Shelly American, Inc.	#15 of 746 things to do in Las Vegas	4.5	1547	http://shellyamerican.com/	6485 Danforth St, Las Vegas, NV 89119-3211	+1 702-942-7325	2024-04-15 01:18:35	<button>View</button> <button>Edit</button> <button>Delete</button>
135565	Mount Charleston	#13 of 746 things to do in Las Vegas	4.5	1450	http://www.gomtcharleston.com	Rubbers Ranch Trail, Las Vegas, NV 89103	+1 702-972-5466	2024-04-15 01:19:19	<button>View</button> <button>Edit</button> <button>Delete</button>
501163	KA by Cirque du Soleil	#50 of 746 things to do in Las Vegas	4.5	6615	http://www.ingmgrand.com/events/tourmentka-cirque-du-soleil-show.aspx	3799 Las Vegas Blvd S MGM Grand, Las Vegas, NV 89109-4219	+1 702-531-3826	2024-04-15 03:36:01	<button>View</button> <button>Edit</button> <button>Delete</button>
1224856	Mirage Volcano	#37 of 746 things to do in Las Vegas	4.5	6031	http://www.mirage.com/attractions/volcano.aspx	3499 Las Vegas Blvd S Mirage Hotel & Casino, Las Vegas, NV 89109-8907	+1 702-791-7111	2024-04-15 03:43:28	<button>View</button> <button>Edit</button> <button>Delete</button>
7777777	Seventy Seven Acres	#1 in SevenLand	4.9	7777	www.777acres.com	777 Seventy Lane, Sacramento, CA	+1 777-777-7777	2024-04-15 22:37:02	<button>View</button> <button>Edit</button> <button>Delete</button>
8888888	Eighty Eight Land	#1 in EightLand	5.0	8888	www.888.com	888 Eighty Lane	+1 888-888-8888	2024-04-15 22:42:25	<button>View</button> <button>Edit</button> <button>Delete</button>

All records based on the minimum data filter where only those tourist locations that were added before the 15th of April to the database were displayed.

Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

All records that were displayed based on the maximum date filter where only those that were entered into the database before the 16th of April are retrieved.

Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

**Sorting by the Rating option in descending order as denoted by the - sign in the order column. Displays all tourist locations from highest to lowest rated.**

**Checklist Items (1)**

#3 Demonstrate a few varied filters/sorts

**Sorting based on the number of reviews in ascending order. Displays all tourist locations from those with the lowest to highest number of reviews.**

**Checklist Items (1)**

#3 Demonstrate a few varied filters/sorts

Sorting based on number of reviews in descending order BUT with the use of the limit function that restricts the number of records retrieved to the value entered, which in this case is 4

Checklist Items (1)

#3 Demonstrate a few varied filters/sorts

A combination of sorting by number of reviews in descending order combined with a partial search on the tourist location name as well as a limit of 4 records at a time on the screen.

Checklist Items (1)

### #3 Demonstrate a few varied filters/sorts

Tourist Attractions Home Profile Admin - Tourist Locations - Logout

Tourist Locations Filter

ID Search	Name Search	Minimum Rating	Minimum Reviews	Minimum Date	Maximum Date	Sort	Order	Limit
ID Search:	Z	Minimum Rating	Minimum Reviews	mm/dd/yyyy	mm/dd/yyyy	Number of Reviews	desc	4

Filter Clear

List of Tourist Locations Data

No records to show

A Search Name filter for the letter Z, which isn't found in any of the tourist locations' names in my database, leading to the "No Records to show" message on the screen.

### Checklist Items (1)

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

Every single list item has three action buttons on the right, with the ability to navigate to a single view(details), edit the page, and delete the record. Also, each list item has a moderate summary of its entire associated data containing only the most important columns as displayed by the fewer number of columns on screen, with the bulk of the data being displayed in the more thorough single-view page.

### Checklist Items (2)

#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)

### Task #2 - Points: 1

**Text:** Screenshots of the list page code

#### Checklist

\*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small      Medium      Large

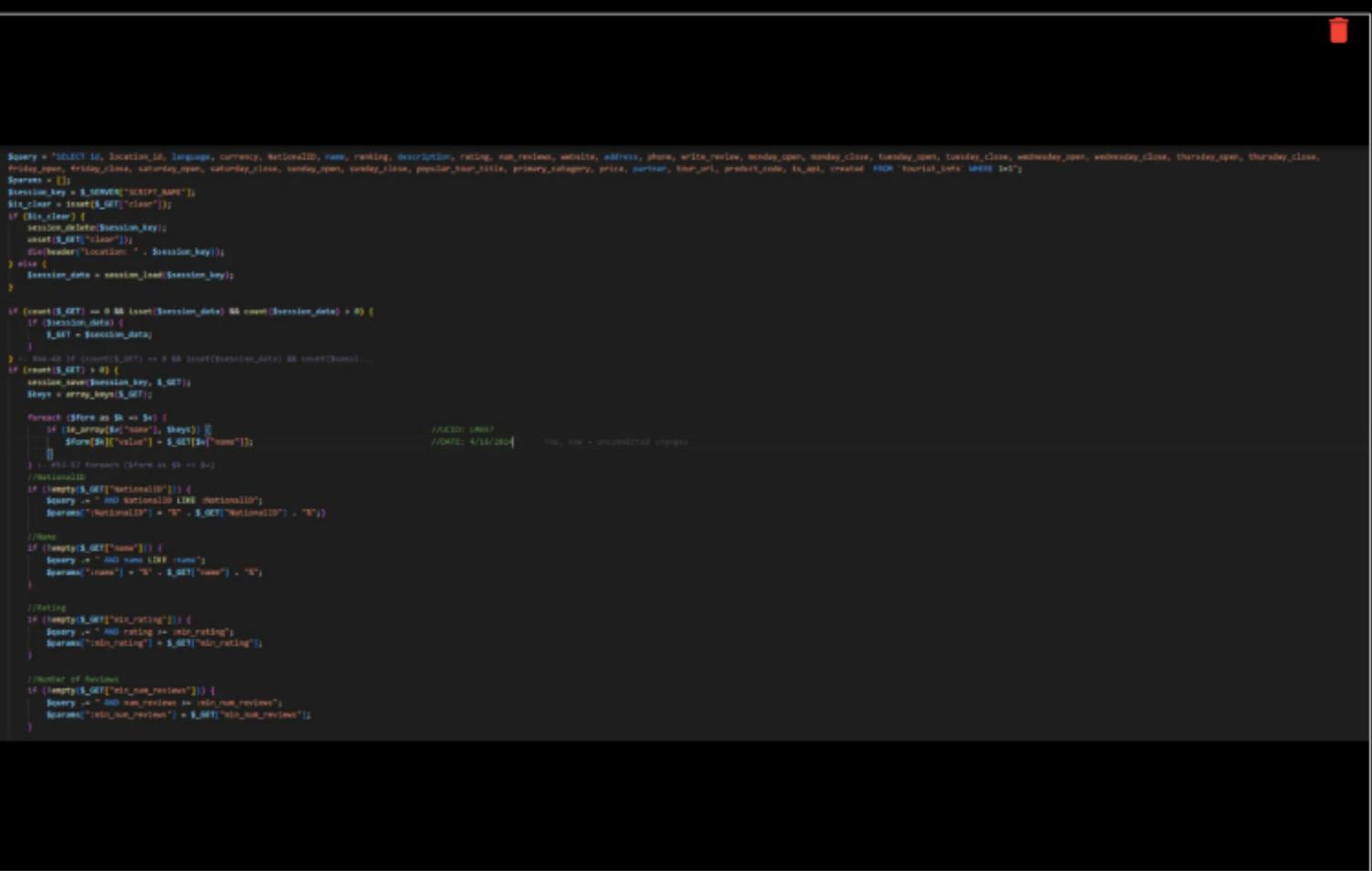
```
// Form Search
$form = [
    ["type" => "number", "name" => "NationalID", "placeholder" => "ID Search", "label" => "ID Search", "include_margin" => false],
    ["type" => "text", "name" => "name", "placeholder" => "Name Search", "label" => "Name Search", "include_margin" => false],
    ["type" => "decimal", "name" => "min_rating", "placeholder" => "Minimum Rating", "label" => "Minimum Rating", "pattern" => "\d*\.\d*", "include_margin" => false],
    ["type" => "number", "name" => "min_num_reviews", "placeholder" => "Minimum Reviews", "label" => "Minimum Reviews", "include_margin" => false],
    ["type" => "date", "name" => "date_min", "placeholder" => "Minimum Date", "label" => "Minimum Date", "include_margin" => false],
    ["type" => "date", "name" => "date_max", "placeholder" => "Maximum Date", "label" => "Maximum Date", "include_margin" => false],
    ["type" => "select", "name" => "sort", "label" => "Sort", "options" => [{"rating" => "Rating", "num_reviews" => "Number of Reviews", "date" => "Date"}, "include_margin" => false, "value" => "rating"],
    ["type" => "select", "name" => "order", "label" => "Order", "options" => [{"asc" => "+", "desc" => "-"}, "include_margin" => false],
```

```
["type" => "number", "name" => "limit", "label" => "Limit", "value" => "10", "include_margin" => false],  
]);
```

## The form structure of the list/sort feature on the List page.

### Checklist Items (1)

#### #1 Show the filter/sort form generation



A screenshot of a browser window displaying a complex PHP code snippet. The code is used to generate a filter/sort form for a database query. It includes logic for session management, URL parameters, and database queries. A red trash can icon is visible in the top right corner of the browser window.

```
Query = "SELECT id, location_id, language, currency, NationalID, name, ranking, description, rating, num_reviews, website, address, phone, write_review, monday_open, monday_close, tuesday_open, tuesday_close, wednesday_open, wednesday_close, thursday_open, thursday_close, friday_open, friday_close, saturday_open, saturday_close, sunday_open, sunday_close, popular_hour_start, primary_category, price, partner, tour_url, product_code, is_apl, created FROM tourist_attr WHERE 1=1";  
Session = [];  
Session_key = $_SESSION['SCRIPT_NAME'];  
$_SESSION['clear'] = inset($_GET['clear']);  
if (!$_SESSION['clear']) {  
    session_delete($Session_key);  
    unset($_GET['clear']);  
    die(header("Location: " . $Session_key));  
} else {  
    Session_data = session_load($Session_key);  
}  
  
if (count($_GET) == 0 AND count(Session_data) == count(Session_data) > 0) {  
    if ($Session_data) {  
        $Session_key = $Session_data;  
    }  
}  
if (isset($_GET) && count($_GET) == 0 AND count(Session_data) == count(Session_data)) {  
    if ($Session_data) {  
        $Session_key = $Session_data;  
    }  
}  
$_GET = array();  
$_SESSION['clear'] = $Session_key;  
$_SESSION['Session'] = $Session_data;  
  
foreach ($key as $k => $v) {  
    if (is_array($v['value'])) {  
        $value = $v['value'][0];  
    } else {  
        $value = $v['value'];  
    }  
    $key = $k; // Set key to current key  
    $value = $value ? $value : null; // Set value to null if empty  
    $Session[$key] = $value; // Add key-value pair to session array  
}  
$_SESSION['Session'] = $Session;  
if (!empty($_GET['NationalID'])) {  
    $Query .= " AND NationalID LIKE '%". $_GET['NationalID'] . "%'";  
    $Param['NationalID'] = "%". $_GET['NationalID'] . "%";  
}  
  
if (!empty($_GET['name'])) {  
    $Query .= " AND name LIKE '%". $_GET['name'] . "%'";  
    $Param['name'] = "%". $_GET['name'] . "%";  
}  
  
if (!empty($_GET['min_rating'])) {  
    $Query .= " AND rating >= ". $_GET['min_rating'];  
    $Param['min_rating'] = $_GET['min_rating'];  
}  
  
if (!empty($_GET['min_num_reviews'])) {  
    $Query .= " AND num_reviews >= ". $_GET['min_num_reviews'];  
    $Param['min_num_reviews'] = $_GET['min_num_reviews'];  
}
```

Part 1 showcasing the DB query and how part of the filter/sort functionality is handled in the code.

### Checklist Items (1)

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

Part 2 showcasing the rest of the filter/sort functionality alongside the restriction on the limit field and how all of its queried from the database.

### Checklist Items (1)

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

```
Table = ["id", "name", "title" => "List of Tourist Locations Beta", "ignored_columns" => ["id", "location_id", "language", "currency", "description", "website_review", "monday_open", "monday_close", "tuesday_open", "tuesday_close", "wednesday_open", "wednesday_close", "thursday_open", "thursday_close", "friday_open", "friday_close", "saturday_open", "saturday_close", "sunday_open", "sunday_close", "popular_tour_title", "primary_category", "price", "partner", "tour_url", "product_code", "id_url", "edit_url" => get_url("admin/edit_touristlocation.php"), "delete_url" => get_url("admin/delete_touristlocation.php"), "view_url" => get_url("admin/view_touristlocationdetails.php")];
}



### Tourist Locations Filter



Show methods: 0/17



| ID     | Name                                | Title                               | Language | Currency | Description                                                              | Website Review | Monday Open | Monday Close | Tuesday Open | Tuesday Close | Wednesday Open | Wednesday Close | Thursday Open | Thursday Close | Friday Open | Friday Close | Saturday Open | Saturday Close | Sunday Open | Sunday Close | Popular Tour Title | Primary Category | Price | Partner | Tour URL | Product Code | Edit URL | Delete URL | View URL |       |       |       |       |       |
|--------|-------------------------------------|-------------------------------------|----------|----------|--------------------------------------------------------------------------|----------------|-------------|--------------|--------------|---------------|----------------|-----------------|---------------|----------------|-------------|--------------|---------------|----------------|-------------|--------------|--------------------|------------------|-------|---------|----------|--------------|----------|------------|----------|-------|-------|-------|-------|-------|
| 1      | Paris Eiffel Tower                  | Paris Eiffel Tower                  | French   | EUR      | Paris Eiffel Tower is a famous landmark in Paris, France.                | 100            | 08:00       | 18:00        | 09:00        | 19:00         | 10:00          | 20:00           | 11:00         | 21:00          | 12:00       | 22:00        | 13:00         | 23:00          | 14:00       | 24:00        | 15:00              | 25:00            | 16:00 | 26:00   | 17:00    | 27:00        | 18:00    | 28:00      | 19:00    | 29:00 | 20:00 | 30:00 | 21:00 | 31:00 |
| 2      | London Big Ben                      | London Big Ben                      | English  | GBP      | London Big Ben is a famous landmark in London, UK.                       | 150            | 09:00       | 19:00        | 10:00        | 20:00         | 11:00          | 21:00           | 12:00         | 22:00          | 13:00       | 23:00        | 14:00         | 24:00          | 15:00       | 25:00        | 16:00              | 26:00            | 17:00 | 27:00   | 18:00    | 28:00        | 19:00    | 29:00      | 20:00    | 30:00 | 21:00 | 31:00 |       |       |
| 3      | New York City                       | New York City                       | English  | USD      | New York City is a major city in the United States.                      | 200            | 10:00       | 20:00        | 11:00        | 21:00         | 12:00          | 22:00           | 13:00         | 23:00          | 14:00       | 24:00        | 15:00         | 25:00          | 16:00       | 26:00        | 17:00              | 27:00            | 18:00 | 28:00   | 19:00    | 29:00        | 20:00    | 30:00      | 21:00    | 31:00 |       |       |       |       |
| 4      | Rome Colosseum                      | Rome Colosseum                      | Italian  | EUR      | Rome Colosseum is a famous landmark in Rome, Italy.                      | 130            | 08:00       | 18:00        | 09:00        | 19:00         | 10:00          | 20:00           | 11:00         | 21:00          | 12:00       | 22:00        | 13:00         | 23:00          | 14:00       | 24:00        | 15:00              | 25:00            | 16:00 | 26:00   | 17:00    | 27:00        | 18:00    | 28:00      | 19:00    | 29:00 | 20:00 | 30:00 |       |       |
| 5      | Bangkok Wat Phra Kaew               | Bangkok Wat Phra Kaew               | Thai     | THB      | Bangkok Wat Phra Kaew is a famous landmark in Bangkok, Thailand.         | 170            | 07:00       | 19:00        | 08:00        | 20:00         | 09:00          | 21:00           | 10:00         | 22:00          | 11:00       | 23:00        | 12:00         | 24:00          | 13:00       | 25:00        | 14:00              | 26:00            | 15:00 | 27:00   | 16:00    | 28:00        | 17:00    | 29:00      | 18:00    | 30:00 | 19:00 | 31:00 |       |       |
| 6      | Singapore Merlion                   | Singapore Merlion                   | English  | SGD      | Singapore Merlion is a famous landmark in Singapore.                     | 190            | 08:00       | 20:00        | 09:00        | 21:00         | 10:00          | 22:00           | 11:00         | 23:00          | 12:00       | 24:00        | 13:00         | 25:00          | 14:00       | 26:00        | 15:00              | 27:00            | 16:00 | 28:00   | 17:00    | 29:00        | 18:00    | 30:00      | 19:00    | 31:00 |       |       |       |       |
| 7      | Paris Louvre Museum                 | Paris Louvre Museum                 | French   | EUR      | Paris Louvre Museum is a famous museum in Paris, France.                 | 160            | 09:00       | 18:00        | 10:00        | 19:00         | 11:00          | 20:00           | 12:00         | 21:00          | 13:00       | 22:00        | 14:00         | 23:00          | 15:00       | 24:00        | 16:00              | 25:00            | 17:00 | 26:00   | 18:00    | 27:00        | 19:00    | 28:00      | 20:00    | 29:00 | 21:00 | 30:00 |       |       |
| 8      | London British Museum               | London British Museum               | English  | GBP      | London British Museum is a famous museum in London, UK.                  | 180            | 10:00       | 19:00        | 11:00        | 20:00         | 12:00          | 21:00           | 13:00         | 22:00          | 14:00       | 23:00        | 15:00         | 24:00          | 16:00       | 25:00        | 17:00              | 26:00            | 18:00 | 27:00   | 19:00    | 28:00        | 20:00    | 29:00      | 21:00    | 30:00 |       |       |       |       |
| 9      | New York Metropolitan Museum of Art | New York Metropolitan Museum of Art | English  | USD      | New York Metropolitan Museum of Art is a famous museum in New York City. | 210            | 11:00       | 20:00        | 12:00        | 21:00         | 13:00          | 22:00           | 14:00         | 23:00          | 15:00       | 24:00        | 16:00         | 25:00          | 17:00       | 26:00        | 18:00              | 27:00            | 19:00 | 28:00   | 20:00    | 29:00        | 21:00    | 30:00      |          |       |       |       |       |       |
| 10     | Rome Colosseum                      | Rome Colosseum                      | Italian  | EUR      | Rome Colosseum is a famous landmark in Rome, Italy.                      | 140            | 09:00       | 18:00        | 10:00        | 19:00         | 11:00          | 20:00           | 12:00         | 21:00          | 13:00       | 22:00        | 14:00         | 23:00          | 15:00       | 24:00        | 16:00              | 25:00            | 17:00 | 26:00   | 18:00    | 27:00        | 19:00    | 28:00      | 20:00    | 29:00 | 21:00 | 30:00 |       |       |
| 11     | Bangkok Wat Phra Kaew               | Bangkok Wat Phra Kaew               | Thai     | THB      | Bangkok Wat Phra Kaew is a famous landmark in Bangkok, Thailand.         | 150            | 08:00       | 19:00        | 09:00        | 20:00         | 10:00          | 21:00           | 11:00         | 22:00          | 12:00       | 23:00        | 13:00         | 24:00          | 14:00       | 25:00        | 15:00              | 26:00            | 16:00 | 27:00   | 17:00    | 28:00        | 18:00    | 29:00      | 19:00    | 30:00 |       |       |       |       |
| 12     | Singapore Merlion                   | Singapore Merlion                   | English  | SGD      | Singapore Merlion is a famous landmark in Singapore.                     | 170            | 09:00       | 20:00        | 10:00        | 21:00         | 11:00          | 22:00           | 12:00         | 23:00          | 13:00       | 24:00        | 14:00         | 25:00          | 15:00       | 26:00        | 16:00              | 27:00            | 17:00 | 28:00   | 18:00    | 29:00        | 19:00    | 30:00      |          |       |       |       |       |       |
| 13     | Paris Eiffel Tower                  | Paris Eiffel Tower                  | French   | EUR      | Paris Eiffel Tower is a famous landmark in Paris, France.                | 160            | 10:00       | 19:00        | 11:00        | 20:00         | 12:00          | 21:00           | 13:00         | 22:00          | 14:00       | 23:00        | 15:00         | 24:00          | 16:00       | 25:00        | 17:00              | 26:00            | 18:00 | 27:00   | 19:00    | 28:00        |          |            |          |       |       |       |       |       |
| 14     | London Big Ben                      | London Big Ben                      | English  | GBP      | London Big Ben is a famous landmark in London, UK.                       | 180            | 11:00       | 20:00        | 12:00        | 21:00         | 13:00          | 22:00           | 14:00         | 23:00          | 15:00       | 24:00        | 16:00         | 25:00          | 17:00       | 26:00        | 18:00              | 27:00            | 19:00 | 28:00   |          |              |          |            |          |       |       |       |       |       |
| 15     | New York City                       | New York City                       | English  | USD      | New York City is a major city in the United States.                      | 200            | 12:00       | 21:00        | 13:00        | 22:00         | 14:00          | 23:00           | 15:00         | 24:00          | 16:00       | 25:00        | 17:00         | 26:00          | 18:00       | 27:00        | 19:00              | 28:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 16     | Rome Colosseum                      | Rome Colosseum                      | Italian  | EUR      | Rome Colosseum is a famous landmark in Rome, Italy.                      | 130            | 10:00       | 19:00        | 11:00        | 20:00         | 12:00          | 21:00           | 13:00         | 22:00          | 14:00       | 23:00        | 15:00         | 24:00          | 16:00       | 25:00        | 17:00              | 26:00            | 18:00 | 27:00   |          |              |          |            |          |       |       |       |       |       |
| 17     | Bangkok Wat Phra Kaew               | Bangkok Wat Phra Kaew               | Thai     | THB      | Bangkok Wat Phra Kaew is a famous landmark in Bangkok, Thailand.         | 150            | 09:00       | 19:00        | 10:00        | 20:00         | 11:00          | 21:00           | 12:00         | 22:00          | 13:00       | 23:00        | 14:00         | 24:00          | 15:00       | 25:00        | 16:00              | 26:00            | 17:00 | 27:00   |          |              |          |            |          |       |       |       |       |       |
| 18     | Singapore Merlion                   | Singapore Merlion                   | English  | SGD      | Singapore Merlion is a famous landmark in Singapore.                     | 170            | 10:00       | 20:00        | 11:00        | 21:00         | 12:00          | 22:00           | 13:00         | 23:00          | 14:00       | 24:00        | 15:00         | 25:00          | 16:00       | 26:00        | 17:00              | 27:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 19     | Paris Louvre Museum                 | Paris Louvre Museum                 | French   | EUR      | Paris Louvre Museum is a famous museum in Paris, France.                 | 160            | 11:00       | 20:00        | 12:00        | 21:00         | 13:00          | 22:00           | 14:00         | 23:00          | 15:00       | 24:00        | 16:00         | 25:00          | 17:00       | 26:00        | 18:00              | 27:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 20     | London British Museum               | London British Museum               | English  | GBP      | London British Museum is a famous museum in London, UK.                  | 180            | 12:00       | 21:00        | 13:00        | 22:00         | 14:00          | 23:00           | 15:00         | 24:00          | 16:00       | 25:00        | 17:00         | 26:00          | 18:00       | 27:00        |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 21     | New York Metropolitan Museum of Art | New York Metropolitan Museum of Art | English  | USD      | New York Metropolitan Museum of Art is a famous museum in New York City. | 210            | 13:00       | 22:00        | 14:00        | 23:00         | 15:00          | 24:00           | 16:00         | 25:00          | 17:00       | 26:00        | 18:00         | 27:00          |             |              |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 22     | Rome Colosseum                      | Rome Colosseum                      | Italian  | EUR      | Rome Colosseum is a famous landmark in Rome, Italy.                      | 140            | 10:00       | 19:00        | 11:00        | 20:00         | 12:00          | 21:00           | 13:00         | 22:00          | 14:00       | 23:00        | 15:00         | 24:00          | 16:00       | 25:00        | 17:00              | 26:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 23     | Bangkok Wat Phra Kaew               | Bangkok Wat Phra Kaew               | Thai     | THB      | Bangkok Wat Phra Kaew is a famous landmark in Bangkok, Thailand.         | 150            | 09:00       | 19:00        | 10:00        | 20:00         | 11:00          | 21:00           | 12:00         | 22:00          | 13:00       | 23:00        | 14:00         | 24:00          | 15:00       | 25:00        | 16:00              | 26:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 24     | Singapore Merlion                   | Singapore Merlion                   | English  | SGD      | Singapore Merlion is a famous landmark in Singapore.                     | 170            | 10:00       | 20:00        | 11:00        | 21:00         | 12:00          | 22:00           | 13:00         | 23:00          | 14:00       | 24:00        | 15:00         | 25:00          | 16:00       | 26:00        |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 25     | Paris Eiffel Tower                  | Paris Eiffel Tower                  | French   | EUR      | Paris Eiffel Tower is a famous landmark in Paris, France.                | 160            | 11:00       | 20:00        | 12:00        | 21:00         | 13:00          | 22:00           | 14:00         | 23:00          | 15:00       | 24:00        | 16:00         | 25:00          | 17:00       | 26:00        |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 26     | London Big Ben                      | London Big Ben                      | English  | GBP      | London Big Ben is a famous landmark in London, UK.                       | 180            | 12:00       | 21:00        | 13:00        | 22:00         | 14:00          | 23:00           | 15:00         | 24:00          | 16:00       | 25:00        | 17:00         | 26:00          |             |              |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 27     | New York City                       | New York City                       | English  | USD      | New York City is a major city in the United States.                      | 200            | 13:00       | 22:00        | 14:00        | 23:00         | 15:00          | 24:00           | 16:00         | 25:00          | 17:00       | 26:00        |               |                |             |              |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 28     | Rome Colosseum                      | Rome Colosseum                      | Italian  | EUR      | Rome Colosseum is a famous landmark in Rome, Italy.                      | 130            | 10:00       | 19:00        | 11:00        | 20:00         | 12:00          | 21:00           | 13:00         | 22:00          | 14:00       | 23:00        | 15:00         | 24:00          | 16:00       | 25:00        | 17:00              | 26:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 29     | Bangkok Wat Phra Kaew               | Bangkok Wat Phra Kaew               | Thai     | THB      | Bangkok Wat Phra Kaew is a famous landmark in Bangkok, Thailand.         | 150            | 09:00       | 19:00        | 10:00        | 20:00         | 11:00          | 21:00           | 12:00         | 22:00          | 13:00       | 23:00        | 14:00         | 24:00          | 15:00       | 25:00        | 16:00              | 26:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 30     | Singapore Merlion                   | Singapore Merlion                   | English  | SGD      | Singapore Merlion is a famous landmark in Singapore.                     | 170            | 10:00       | 20:00        | 11:00        | 21:00         | 12:00          | 22:00           | 13:00         | 23:00          | 14:00       | 24:00        | 15:00         | 25:00          | 16:00       | 26:00        |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 31     | Paris Louvre Museum                 | Paris Louvre Museum                 | French   | EUR      | Paris Louvre Museum is a famous museum in Paris, France.                 | 160            | 11:00       | 20:00        | 12:00        | 21:00         | 13:00          | 22:00           | 14:00         | 23:00          | 15:00       | 24:00        | 16:00         | 25:00          | 17:00       | 26:00        |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 32     | London British Museum               | London British Museum               | English  | GBP      | London British Museum is a famous museum in London, UK.                  | 180            | 12:00       | 21:00        | 13:00        | 22:00         | 14:00          | 23:00           | 15:00         | 24:00          | 16:00       | 25:00        | 17:00         | 26:00          |             |              |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 33     | New York Metropolitan Museum of Art | New York Metropolitan Museum of Art | English  | USD      | New York Metropolitan Museum of Art is a famous museum in New York City. | 210            | 13:00       | 22:00        | 14:00        | 23:00         | 15:00          | 24:00           | 16:00         | 25:00          | 17:00       | 26:00        |               |                |             |              |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 34     | Rome Colosseum                      | Rome Colosseum                      | Italian  | EUR      | Rome Colosseum is a famous landmark in Rome, Italy.                      | 140            | 10:00       | 19:00        | 11:00        | 20:00         | 12:00          | 21:00           | 13:00         | 22:00          | 14:00       | 23:00        | 15:00         | 24:00          | 16:00       | 25:00        | 17:00              | 26:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 35     | Bangkok Wat Phra Kaew               | Bangkok Wat Phra Kaew               | Thai     | THB      | Bangkok Wat Phra Kaew is a famous landmark in Bangkok, Thailand.         | 150            | 09:00       | 19:00        | 10:00        | 20:00         | 11:00          | 21:00           | 12:00         | 22:00          | 13:00       | 23:00        | 14:00         | 24:00          | 15:00       | 25:00        | 16:00              | 26:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 36     | Singapore Merlion                   | Singapore Merlion                   | English  | SGD      | Singapore Merlion is a famous landmark in Singapore.                     | 170            | 10:00       | 20:00        | 11:00        | 21:00         | 12:00          | 22:00           | 13:00         | 23:00          | 14:00       | 24:00        | 15:00         | 25:00          | 16:00       | 26:00        |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 37     | Paris Eiffel Tower                  | Paris Eiffel Tower                  | French   | EUR      | Paris Eiffel Tower is a famous landmark in Paris, France.                | 160            | 11:00       | 20:00        | 12:00        | 21:00         | 13:00          | 22:00           | 14:00         | 23:00          | 15:00       | 24:00        | 16:00         | 25:00          | 17:00       | 26:00        |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 38     | London Big Ben                      | London Big Ben                      | English  | GBP      | London Big Ben is a famous landmark in London, UK.                       | 180            | 12:00       | 21:00        | 13:00        | 22:00         | 14:00          | 23:00           | 15:00         | 24:00          | 16:00       | 25:00        | 17:00         | 26:00          |             |              |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 39     | New York City                       | New York City                       | English  | USD      | New York City is a major city in the United States.                      | 200            | 13:00       | 22:00        | 14:00        | 23:00         | 15:00          | 24:00           | 16:00         | 25:00          | 17:00       | 26:00        |               |                |             |              |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 40     | Rome Colosseum                      | Rome Colosseum                      | Italian  | EUR      | Rome Colosseum is a famous landmark in Rome, Italy.                      | 130            | 10:00       | 19:00        | 11:00        | 20:00         | 12:00          | 21:00           | 13:00         | 22:00          | 14:00       | 23:00        | 15:00         | 24:00          | 16:00       | 25:00        | 17:00              | 26:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 41     | Bangkok Wat Phra Kaew               | Bangkok Wat Phra Kaew               | Thai     | THB      | Bangkok Wat Phra Kaew is a famous landmark in Bangkok, Thailand.         | 150            | 09:00       | 19:00        | 10:00        | 20:00         | 11:00          | 21:00           | 12:00         | 22:00          | 13:00       | 23:00        | 14:00         | 24:00          | 15:00       | 25:00        | 16:00              | 26:00            |       |         |          |              |          |            |          |       |       |       |       |       |
| 42     | Singapore Merlion                   | Singapore Merlion                   | English  | SGD      | Singapore Merlion is a famous landmark in Singapore.                     | 170            | 10:00       | 20:00        | 11:00        | 21:00         | 12:00          | 22:00           | 13:00         | 23:00          | 14:00       | 24:00        | 15:00         | 25:00          | 16:00       | 26:00        |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 43     | Paris Louvre Museum                 | Paris Louvre Museum                 | French   | EUR      | Paris Louvre Museum is a famous museum in Paris, France.                 | 160            | 11:00       | 20:00        | 12:00        | 21:00         | 13:00          | 22:00           | 14:00         | 23:00          | 15:00       | 24:00        | 16:00         | 25:00          | 17:00       | 26:00        |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |
| 44</td |                                     |                                     |          |          |                                                                          |                |             |              |              |               |                |                 |               |                |             |              |               |                |             |              |                    |                  |       |         |          |              |          |            |          |       |       |       |       |       |


```

```
<?php
//note we need to go up 1 more directory
require(__DIR__ . "/../../../../partials/nav.php");

if (!has_role("Admin")) {
    flash("You don't have permission to view this page", "warning"); //UCID: LM457
    die(header("Location: " . get_url("home.php"))); //DATE: 4/16/2024
}
?>
<?php
```

Role check to ensure only those users with Admin authority are able to access this listing page of all tourist locations.

#### Checklist Items (1)

#4 Show any restrictions like role guard or login checks

#### 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:

First, the PHP script utilizes an array to store the several filter and sort options that will interact with my database table columns. When a user submits specific or even all filter fields on the list page, the PHP script retrieves the filter form data using a GET request for those parameters. If no new filters are passed, the \$\_GET stores the current session data and ensures that existing filters still apply once returned to the list page. If new filters are found, an SQL query is constructed that includes all of the required table columns/fields, and the new session data is stored containing new filter data. The PHP script then adds these filter conditions to the SQL query using the WHERE clause to ensure any corresponding data is fetched. For instance, my filter form has the following fields: national ID, name, rating, reviews,

corresponding data is returned. For instance, my filter form has the following fields, location, rating, reviews, min date, max date, and the required sort, order, and list fields. Assuming inputs are entered into these fields, the SQL query is modified accordingly and retrieves any matching records. As for the sort, order, and list fields, those also work similarly in that users can choose the sorting field, with options of rating, reviews, or date in tandem with the two order options that denote whether to sort in a descending or ascending order. Here, an ORDER BY clause is used to sort specific fields in one of two ways. The limit functionality is also implemented similarly where the LIMIT clause is used to limit the number of matching queries fetched from the database for any filters.

Once the appropriate query is prepared and executed, the output is stored in a pre-defined results array which then itself is later nested within a larger 'table' array that contains only a select few columns from my table and also additional interactive buttons for viewing a details page, and edit page or the ability to delete that tourist location record. HTML table functionality is used to output this array onto the screen in a tabular format with the specified column names. Additionally, the filter/sort functionality is also displayed using HTML where the array that was initially defined containing all of the filter and sort options is parsed through in key-value pairs and a form field is created for each key name alongside button functionality for submitting the filter request and clearing the filters, which essentially just reset the list page to its initial unfiltered state. If the user navigates to another page while filters are applied, the current session data continues to be stored in the GET array, which allows users to return to the page in the state they left it in.

My CSS styling for the list page and all pages for that matter has remained consistent with regards to color palette and overall layout. Particularly for the filter/sort form, I ensured that the key names were aligned in a proper line using the HTML style "space-around" which worked the best for me in terms of ensuring each field was appropriately distant and spaced out from one another. The key names were enclosed within a first div tag containing the row class where they were all aligned horizontally followed by the displaying of the key values in another nested div tag containing the column class. The entirety of both forms is embedded within Bootstrap formatting, particularly the container-fluid class, again to ensure consistency across forms.

As previously mentioned, the tabular format of the list page results also contains three action buttons leading to a details page, an edit page, or the ability to delete that record. However, as of right now, only admin users have the authority to interact and access any of these pages/functionality.

#### 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 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://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/list\\_touristLocs.php](https://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/list_touristLocs.php)

#### URL #2

<https://github.com/Kith07/spring2024-IT202/pull/53>

#### URL #3

<https://github.com/Kith07/spring2024-IT202/pull/55>

● 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

The screenshot shows a web browser window with the URL [https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/view\\_locationDetails.php?id=33](https://it202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/view_locationDetails.php?id=33). The page title is "Tourist Attractions". The main content displays a tourist location named "The LINQ Promenade". Below the name are three buttons: "Back", "Edit", and "Delete". A large circular area is overlaid on the page content, obscuring the description and reviews sections.

National ID: 6637961

Name: The LINQ Promenade

Description: The LINQ Promenade is Caesars Entertainment's shopping, dining and entertainment district located at the center of the Las Vegas Strip which is home to 35+ restaurant and retail stores such as Brooklyn Bowl attractions at The LINQ Promenade such as The High Roller, North America's Tallest Observation wheel, and Fly LINQ Zipline, the strip's only zipline. There's something for everyone at The LINQ Promenade!

Rating: 4.5

Number of Reviews: 3347

The URL of the details page showcases that the entity is fetched by the id via the URL.

#### Checklist Items (1)

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

Attempt at accessing the details page with a missing id value, which simply redirects me back to the list page with an error message stating that an invalid id value was passed/obtained from the url.

#### Checklist Items (1)

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

A lot more thorough and detailed single view of each list item/tourist Location on the details page. It contains information from otherwise ignored columns on the list page. The page also contains links to either edit the single entity or delete the entity, both of which are exclusive to the admin role.

#### Checklist Items (1)

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

#### Task #2 - Points: 1

Text: Screenshots of the details page code

#### Checklist

\*The checkboxes are for your own tracking

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

#### Task Screenshots:

Gallery Style: Large View

Small      Medium      Large

```
$id = se($_GET, "id", -1, false); //UCID: LM457
$output = [];
if ($id > -1) {
    $output[] = "UCID: " . $id;
    $output[] = "Date: " . date("m/d/Y");
}
echo implode("\n", $output); //DATE: 4/16/2024
```

The id is obtained from the URL corresponding to the details page of that list entity using the \$\_GET variable. It is also checked to make sure it has a value larger than -1 to ensure only valid IDs are passed.

### Checklist Items (1)

#### #1 Show how id is fetched

```
$id = $_GET['id']; // (LID) LID57
$output = '';
if ($id > -1) { // (DATE) 4/16/2024
    $db = getDB();
    $query = "SELECT id, location_id, language, currency, NationalID, name, ranking, description, rating, num_reviews, website, address, phone, write_review, monday_open, monday_close, tuesday_open, tuesday_close, wednesday_open, wednesday_close, thursday_open, thursday_close, friday_open, friday_close, saturday_open, saturday_close, sunday_open, sunday_close, popular_tour_title, primary_category, price, partner, tour_url, product_code FROM `tourist_info` WHERE id = ?"; // SQL
    try {
        $stmt = $db->prepare($query);
        $stmt->execute(['id' => $id]);
        $result = $stmt->fetch();
        if ($result) {
            $output = $result[0];
        }
    } catch (PDOException $e) {
        error_log("Error fetching data: " . var_export($e, true));
        flash("An error occurred", "danger");
    }
}
else {
    flash("Invalid ID passed", "warning");
    header("Location: " . get_url("admin/list_touristinfo.php"));
}
foreach ($output as $key => $value) {
    if ($is_null[$key]) {
        $output[$key] = 'N/A';
    }
}
// 80-90 foreach ($output as $key => $value)
```

DB query that accesses the specific record associated with the location ID obtained from the URL.

### Checklist Items (1)

#### #2 Show the DB query to get the record

```
<div class="container-fluid">
    <h1>Tourist Location: <span style="font-size: small;">$output["name"]</span></h1>
    <div>
        <a href="#">php echo get_url('admin/list_touristinfo.php'); </a> <span class="btn btn-secondary">Back</span>
        <a href="#">php echo get_url('admin/edit_touristinfo.php?id=' . $id); </a> <span class="btn btn-primary">Edit</span>
        <form method="POST" style="display: inline-block;">
            <input type="hidden" name="id" value="<?php echo $id; ?>">
            <button type="submit" class="btn btn-danger" onclick="return confirm('Are you sure you want to delete this record?')>Delete</button>
        </form>
    </div>
    <ul class="list-group">
        <li class="list-group-item">strong National ID: <strong>?php echo $output["NationalID"]; ?</strong></li>
        <li class="list-group-item">strong Name: <strong>?php echo $output["name"]; ?</strong></li>
        <li class="list-group-item">strong Description: <strong>?php echo $output["description"]; ?</strong></li>
        <li class="list-group-item">strong Rating: <strong>?php echo $output["rating"]; ?</strong></li>
        <li class="list-group-item">strong Number of Reviews: <strong>?php echo $output["num_reviews"]; ?</strong></li>
        <li class="list-group-item">strong Website: <strong>?php echo $output["website"]; ?</strong></li>
        <li class="list-group-item">strong Address: <strong>?php echo $output["address"]; ?</strong></li>
        <li class="list-group-item">strong Phone: <strong>?php echo $output["phone"]; ?</strong></li>
        <li class="list-group-item">strong Write Reviews: <strong>?php echo $output["write_review"]; ?</strong></li>
        <li class="list-group-item">strong Timings: <strong>?php echo $output["timings"]; ?</strong></li>
    </ul>
</div>
```

```

        <th>Day</th>
        <th>Open Hours</th>
    </tr>
    <tr>
        <td>Monday</td>
        <td><?php $output["monday_open"] = "Not Available"; ?> <?php $output["monday_close"] = "Not Available"; ?></td>
    </tr>
    <tr>
        <td>Tuesday</td>
        <td><?php $output["tuesday_open"] = "Not Available"; ?> <?php $output["tuesday_close"] = "Not Available"; ?></td>
    </tr>
    <tr>
        <td>Wednesday</td>
        <td><?php $output["wednesday_open"] = "Not Available"; ?> <?php $output["wednesday_close"] = "Not Available"; ?></td>
    </tr>
    <tr>
        <td>Thursday</td>
        <td><?php $output["thursday_open"] = "Not Available"; ?> <?php $output["thursday_close"] = "Not Available"; ?></td>
    </tr>
    <tr>
        <td>Friday</td>
        <td><?php $output["friday_open"] = "Not Available"; ?> <?php $output["friday_close"] = "Not Available"; ?></td>
    </tr>
    <tr>
        <td>Saturday</td>
        <td><?php $output["saturday_open"] = "Not Available"; ?> <?php $output["saturday_close"] = "Not Available"; ?></td>
    </tr>
    <tr>
        <td>Sunday</td>
        <td><?php $output["sunday_open"] = "Not Available"; ?> <?php $output["sunday_close"] = "Not Available"; ?></td>
    </tr>
</table></div>
</div>

```

Front-end of how the more detailed view of the single list entity is presented and structured on the webpage including the functionality for the Back, Edit, and Delete buttons.

### Checklist Items (1)

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

#### 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:

To navigate to the view/details page, the PHP script first uses a GET request to access the ID parameter from the URL. Assuming that the retrieved is a valid one, which otherwise would lead to a user-friendly Invalid ID message, a SQL SELECT query is prepared to access all of the available data columns associated with the retrieved ID using a WHERE clause to match IDs. Since the list page offers a summarized view of the entire data catalog for all tourist locations, the details page ensures every single available detail in the database is fetched and appropriately presented on the details page to provide a more thorough experience for the user. The output is stored within an array and a conditional check is made by initially parsing through it in key-value pairs to ensure any empty fields for columns are filled with an "N/A" text. The entire output array is then presented in a highly organized manner using the list-group class where each specific field is further embedded within list item <li> tags within the list-group-item class. Stylistic choices to bolden the names of fields were made to help differentiate between field names and values and particular care was taken for displaying the timings for each location, which were further formatted in a tabular structure with table headings and each day on a different table row. In addition to the details, the admin user also has access to navigation links leading back to either the list page or to the edit page for the specific location where the ID associated with the current page is added to the URL, or a delete option to remove the specific location record from my database. All of the forms were once again wrapped within Bootstrap's container-fluid class to ensure styling consistency across pages.

## 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://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/view\\_locationDetails.php?id=33](https://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/view_locationDetails.php?id=33)

### URL #2

<https://github.com/Kith07/spring2024-IT202/pull/57>

## Edit Data Page (2 pts.)

[^COLLAPSE ^](#)

## Task #1 - Points: 1

Text: Screenshots of the edit page

### Checklist

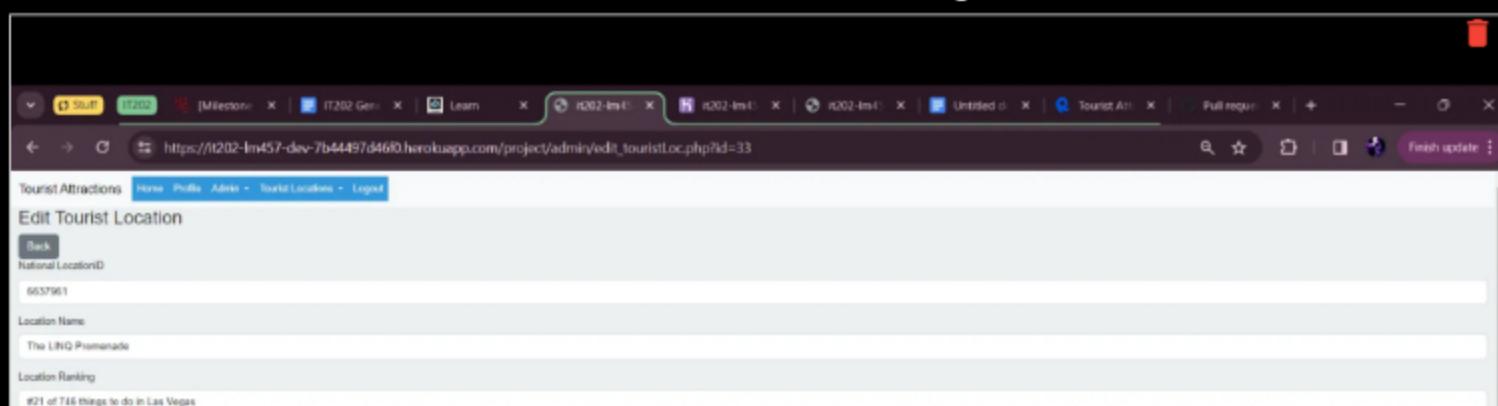
\*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show before and after screenshots of data you'll edit
<input checked="" type="checkbox"/> #2	1	Show examples of validation messages
<input checked="" type="checkbox"/> #3	1	Show an example of successful edit messages
<input checked="" type="checkbox"/> #4	1	Design/Style should be considered (i.e., bootstrap, custom css, etc)
<input checked="" type="checkbox"/> #5	1	Make sure the heroku dev url is visible in the address bar
<input checked="" type="checkbox"/> #6	1	Clearly caption screenshots

### Task Screenshots:

#### Gallery Style: Large View

Small      Medium      Large

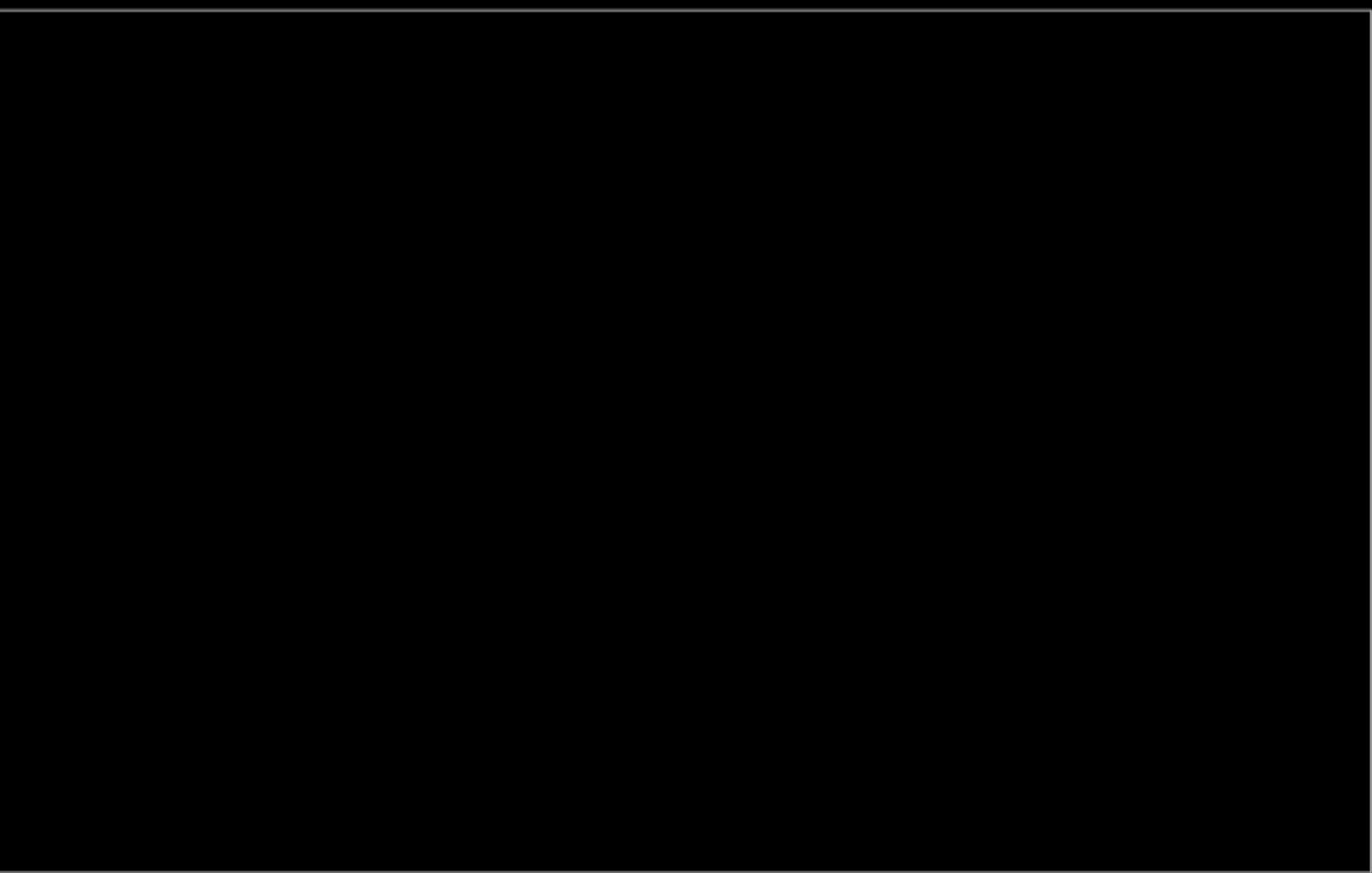


Location Description	The LINQ Promenade is Caesars Entertainment's shopping, dining and entertainment district located at the center of the Las Vegas Strip which is home to 35+ restaurant and retail stores such as Brooklyn Bowl, Gordon Ramsay Fish and Chips, IN-N-Out Burger, Razer, Yard House, and so much more! You can also catch a show at the LINQ Theater.
Location Rating	4.5
Location Number of Reviews	3347
Location Website	<a href="https://www.caesars.com/linq">https://www.caesars.com/linq</a>
Location Address	3545 Las Vegas Blvd S between The LINQ Hotel & Casino and Flamingo Las Vegas, Las Vegas, NV 89109-8921
Location Phone	+1 702 322-0543
Write a Review Link	<a href="https://www.tripadvisor.com/UserReview-g45963-d6037961-The_LINQ_Promenade-Las_Vegas_Nevada.html">https://www.tripadvisor.com/UserReview-g45963-d6037961-The_LINQ_Promenade-Las_Vegas_Nevada.html</a>
Location Monday Open Time	000
Location Monday Close Time	

**Before an edit takes place on the tourist location name on the edit page of a certain record.**

#### **Checklist Items (1)**

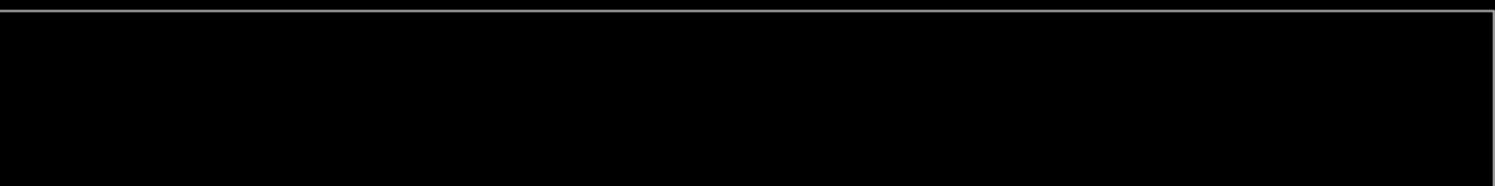
**#1 Show before and after screenshots of data you'll edit**



**After the edit of the tourist location name alongside a successful message denoting our update of this specific record.**

#### **Checklist Items (1)**

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



JS Validation messages when attempting to edit the record by passing empty values for required form fields.

Checklist Items (1)

#2 Show examples of validation messages

In a similar pattern, more JS validation errors when attempting to pass empty values for required form fields.

Checklist Items (1)

#2 Show examples of validation messages

## #2 Show examples of validation messages

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Edit Tourist Location" and has the URL [https://t202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/edit\\_touristLoc.php?id=33](https://t202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/edit_touristLoc.php?id=33). The page displays several validation error messages in red boxes:

- "NationalID" cannot be empty
- "name" cannot be empty
- "ranking" cannot be empty
- "description" cannot be empty
- "rating" cannot be empty
- "num\_reviews" cannot be empty
- "website" cannot be empty
- "address" cannot be empty
- "phone" cannot be empty
- "write\_review" cannot be empty
- "monday\_open" cannot be empty

The browser's developer tools are open, showing the "Elements" tab selected. A tooltip in the bottom right corner of the developer tools indicates an "Easter egg" feature.

PHP Validation when passing empty values for required form fields on the edit page after disabling HTML and JS validations.

### Checklist Items (1)

#### #2 Show examples of validation messages

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Edit Tourist Location" and has the URL [https://t202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/edit\\_touristLoc.php?id=33](https://t202-lm457-dev-7b44497d46f0.herokuapp.com/project/admin/edit_touristLoc.php?id=33). The page displays several validation error messages in red boxes:

- "NationalID" cannot be empty
- "name" cannot be empty
- "ranking" cannot be empty
- "description" cannot be empty
- "rating" cannot be empty
- "num\_reviews" cannot be empty
- "website" cannot be empty
- "address" cannot be empty
- "phone" cannot be empty
- "write\_review" cannot be empty
- "monday\_open" cannot be empty

The browser's developer tools are open, showing the "Elements" tab selected. A tooltip in the bottom right corner of the developer tools indicates an "Easter egg" feature.

Even more error messages showcasing further PHP validation for all fields when attempting to pass empty values in required form fields.

#### Checklist Items (1)

#2 Show examples of validation messages

Another instance of the successful edit message after properly updating a value for a form.

#### Checklist Items (1)

#3 Show an example of successful edit messages

Task #2 - Points: 1

Text: Screenshots of edit page code

#### Checklist

\*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Form should have correct data types for each property being requested
<input checked="" type="checkbox"/> #2	1	Form should have correct validation for each field (HTML, JS, and PHP)
<input checked="" type="checkbox"/> #3	1	Successful edit should have a user-friendly message
<input checked="" type="checkbox"/> #4	1	A successful edit message

<input checked="" type="checkbox"/> #4	1	Any errors should have user-friendly messages
<input checked="" type="checkbox"/> #5	1	Include any other rules like role guards and login checks
<input checked="" type="checkbox"/> #6	1	Include ucid/date comments for each code screenshot
<input checked="" type="checkbox"/> #7	1	Clearly caption screenshots

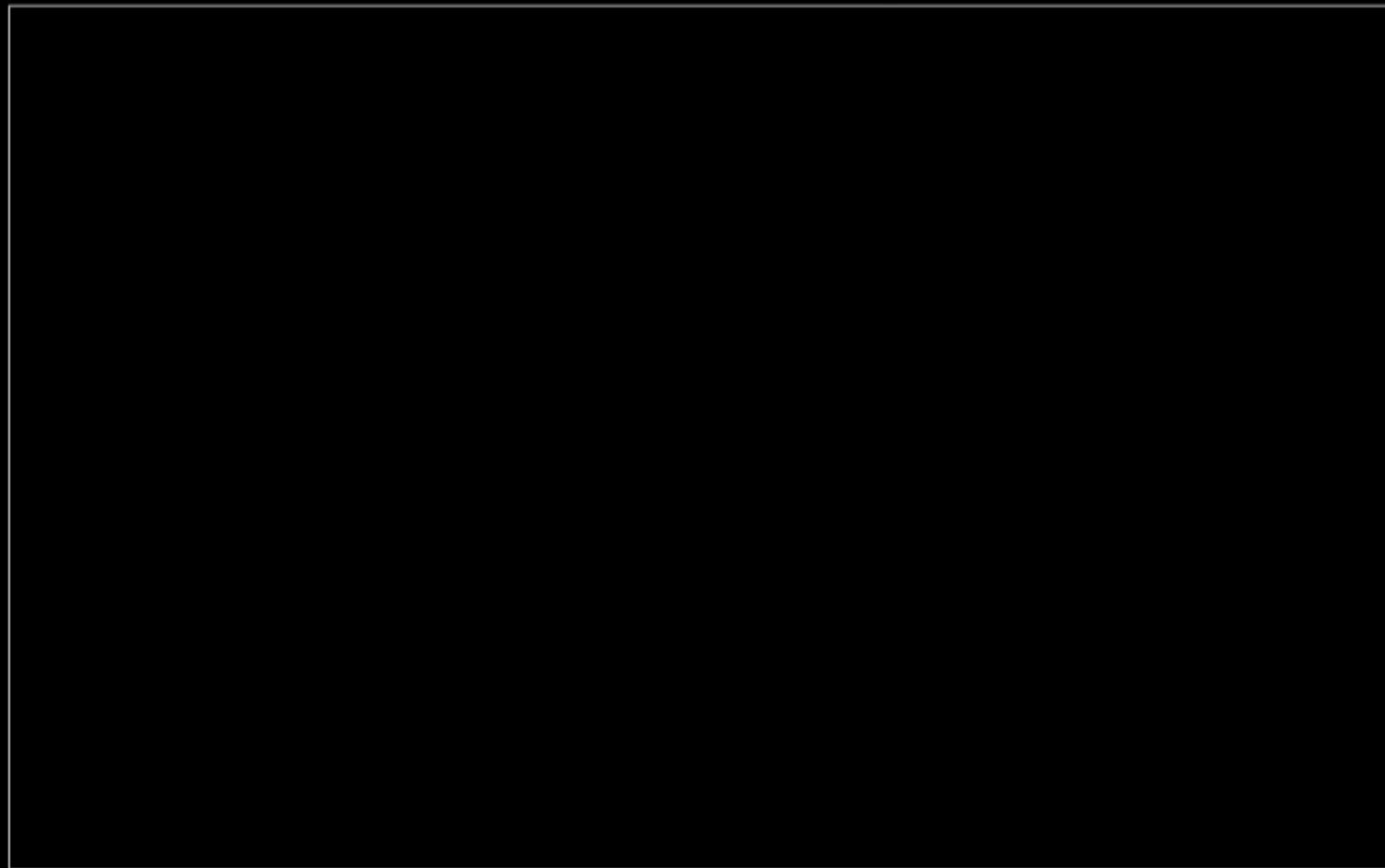
Task Screenshots:

Gallery Style: Large View

Small

Medium

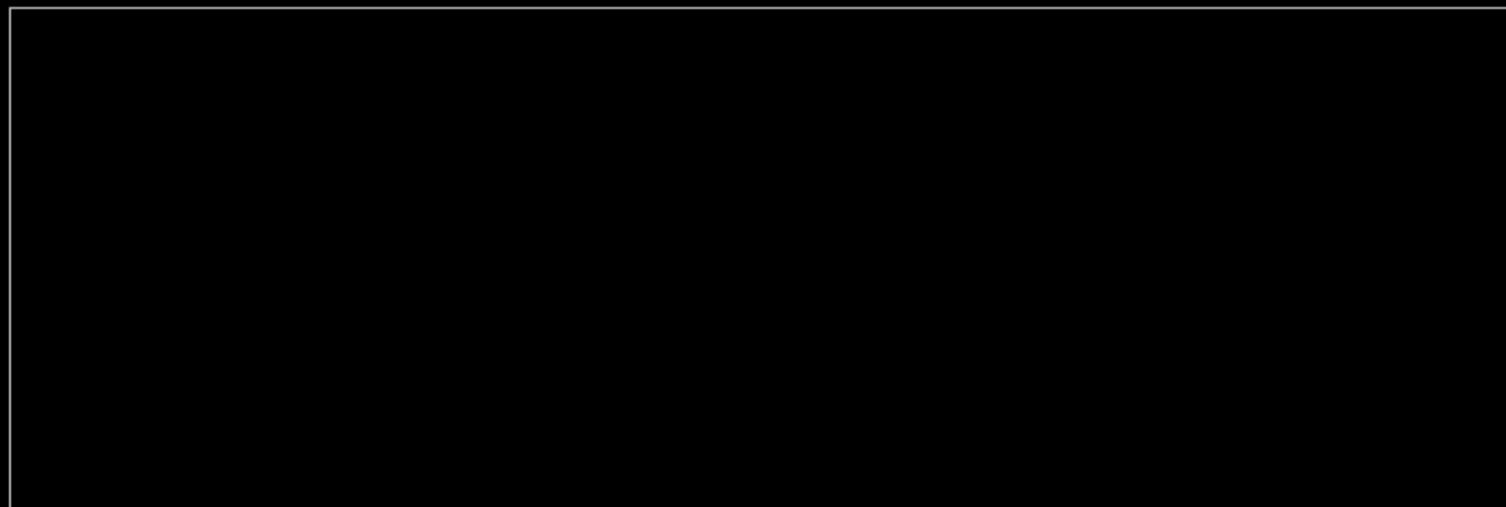
Large



Part 1 of the form with accurate data types for each requested/needed property alongside HTML validation in the form of the input type required attribute.

Checklist Items (1)

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



**Part 2 displaying how the form is actually structured and how it parses through the array of all the requested properties to output them onto the screen.**

**Checklist Items (1)**

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

**JS Validation code that ensures each requested/required property doesn't pass with an empty value on the client side.**

**Checklist Items (1)**

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

```

if (isset($_POST["NationalID"])) {
    $selectedPlace = [];
    foreach ($_POST as $key => $value) {
        if (in_array($key, ["NationalID", "name", "francing", "description", "rating", "num_reviews", "website", "address", "phone", "write_review", "monday_open", "monday_close", "tuesday_open", "tuesday_close", "wednesday_open", "wednesday_close", "thursday_open", "thursday_close", "friday_open", "friday_close", "saturday_open", "saturday_close", "sunday_open", "sunday_close", "popular_tour_title", "primary_category", "price", "partner", "tour_url", "product_code"])) {
            unset($_POST[$key]);
        }
    }
    $selectedPlace = $_POST;
}
// #10-23: Foreach ($_POST as $key => $value)
// PHP Server Side Validation for Empty Fields
$hasError = false;
foreach ($selectedPlace as $k => $v) {
    //UCID: UH457
    if (strlen($v) == 0) {
        //MESSAGE: "The field '$k' cannot be empty." -danger-
        $hasError = true;
    }
}
// #26-33: Foreach ($selectedPlace as $k => $v)

```

**PHP validation for the requested/required form fields that ensure no empty values are passed on the server side as an additional layer of security.**

#### Checklist Items (1)

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

The user is prompted with a user-friendly "Updated Record" success message upon appropriately editing specific or all form fields. In the case improper values are passed, the user is also prompted with user-friendly error messages highlighting that something went wrong with the query due to their inputs.

## Checklist Items (2)

#3 Successful edit should have a user-friendly message

#4 Any errors should have user-friendly messages

```
<?php
//note we need to go up 1 more directory
require(__DIR__ . "/../../../../partials/nav.php");

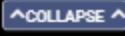
if (!has_role("Admin")) { //UCID: LM457
    flash("You don't have permission to view this page", "warning"); //DATE: 4/16/2024
    die(header("Location: $BASE_PATH" . "/home.php"));
}
?>
```

Role check to ensure only users with Admin authority are able to access and edit the tourist location records.

## Checklist Items (1)

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

 Task #3 - Points: 1

 Text: Screenshot of records from DB

### Checklist

\*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small      Medium      Large

Small

Medium

Large

Before any changes are made to the name of the first tourist location through the edit page.

**Checklist Items (1)**

#1 Show a before and after screenshot of the record

After the tourist location name of the first row is edited. The name was changed from the LINQ Promenade to just AFTER CHANGE showcasing how it has been updated with user editing

### Checklist Items (1)

#1 Show a before and after screenshot of the record

#### 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:

A user can access the edit page of a specific record in one of two ways: either go through the list page or the details page. Either way, the first step is done by retrieving the ID associated with that tourist location using a GET request to check for any ID parameters that were passed from the URLs of the list/details page. Assuming a valid ID is retrieved, which otherwise would deny access to the page and return the user back to where they initially were with a user-friendly error message, a SQL SELECT query containing all of the required form fields is prepared using the WHERE clause to match the ID retrieved from the GET request to any of those in the database. Once found, all of the data associated with each necessary form field is stored within an output array in key-value pairs. Next, a separate PHP form array is declared containing the names, labels, and HTML validations for each necessary form field and another array to store the key names of the values from our output array. Then, the form array is traversed in key-value pairs and a conditional check is made to see if the name of the current form field exists as a key in the newly defined keys array to ensure only those fields that have corresponding values in the retrieved data are populated. Assuming the check is made, the value in the output array is assigned to the corresponding "value" attribute of the matching form field. An HTML form to edit these fields is then displayed to the user who is now loaded into the edit page with pre-populated data in all of the required form fields. Once the required form fields are entered and the page is submitted, the PHP script retrieves the new form data from a POST request for each form field and assigns it to an array. Assuming the form data is properly stored in the array, which would otherwise lead to user-friendly error messages, a SQL UPDATE statement is prepared where each modified form field is changed using the SET clause as the form data is parsed through in key-value pairs and a check is made to see WHERE the ids are matching to ensure the successful update of the correct record. Once updated, the changes are reflected in the database and the user receives a user-friendly success message.

Regarding the validations for the form fields, several layers of checking were implemented using HTML, JS, and PHP. For starters, as previously mentioned, the PHP form array contained the names and labels of each required form field as well as the input type required attribute to ensure no empty fields were submitted in the first place. Disabling HTML validation, a second layer of validation on the client side takes place using JS, where the event listener function awaits form submission, upon which it also defines an array of each required field's name and label and parses through that array in a key-value format ensuring no empty values are passed for necessary form fields, and any exceptions are met with user-friendly error messages. A last round of server-side validations is made in PHP, where the array containing the validation strings will be used to validate the data from the client side and the database.

the output data after the initial data retrieval is traversed in a key-value format as well and ensured that no empty values (0 is allowed) were passed and any attempt to do so would result in user-friendly error messages.

### 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://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/edit\\_touristLoc.php?id=33](https://it202-lm457-prod-cf97d0c34365.herokuapp.com/project/admin/edit_touristLoc.php?id=33)

#### URL #2

<https://github.com/Kith07/spring2024-IT202/pull/54>

#### URL #3

<https://github.com/Kith07/spring2024-IT202/pull/58>

### Delete Handling (1 pt.)

[^COLLAPSE ^](#)

### 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

Success message after the deletion of a manual tourist location on the list page.

#### Checklist Items (1)

- #1 Show the success message of a delete

Error Message when an invalid id is passed or manually entered on the url.

#### Checklist Items (1)

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

## Delete handling code for hard deletes on both view and list pages

### Checklist Items (1)

#3 Show the code related to the delete processing

#### 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

**Before tourist location MOUNT CHARLESTON is deleted from the DB data.**

**Checklist Items (1)**

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

**After tourist location MOUNT CHARLESTON is deleted from the DB data table. Since I'm doing a hard delete, the entire record associated with the tourist location MOUNT CHARLESTON was removed.**

**Checklist Items (1)**

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



### 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:

Due to the overall complexity of my API and the necessary additional changes to my database table, I've primarily chosen to only implement hard deletes for the existing API records (Although I do hope I will be able to implement it during MS3). The only restriction currently in place with regards to delete authority is having Admin rights, since no other user has access to any of that functionality, and the Admin can delete any and every record.

To delete a tourist location record, the PHP script first accesses the ID associated with the record using a GET request to check for any passed ID parameters in the URL. If an invalid ID is obtained through the URL, the user is met with a user-friendly error message denoting the issue at hand. Assuming a valid ID is passed and retrieved, a SQL DELETE is prepared using the WHERE clause again to match the retrieved ID to its corresponding record within the database. If no record exists with that ID, another user-friendly message is prompted to the user denoting the failed attempt to delete the record alongside its ID. If an existing record is found, the SQL statement is executed and the record is deleted/removed from both the database and the list page and the user is given a success message. This delete functionality is implemented by adding its navigation link to the link/details pages.



### Task #4 - Points: 1

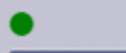
Text: Add the pull request link for the branch related to this feature

#### Details:

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

#### URL #1

<https://github.com/Kith07/spring2024-IT202/pull/56>



#### Misc (1 pt.)



### Task #1 - Points: 1

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



## Task Screenshots:

### Gallery Style: Large View

Small      Medium      Large

The screenshot shows a GitHub project board with three columns: Todo, In Progress, and Done. The Todo column has 0 items. The In Progress column has 0 items. The Done column has 16 items, which are listed as follows:

- spring2024-(T202 #43) Define the appropriate table or tables for your API [MS2]
- spring2024-(T202 #44) Data Creation Page [MS2]
- spring2024-(T202 #45) Data List Page (many items) [MS2]
- spring2024-(T202 #46) View Data Details Page (single item) [MS2]
- spring2024-(T202 #47) Edit Data Page [MS2]

Project Board after issues were moved to Done.

#### Task #2 - Points: 1

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

URL #1

<https://github.com/users/Kith07/projects/1/views/1>

#### Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

This was undoubtedly the most effort I've put into a single assignment/project throughout my academic career and the fact that the entire process was enjoyable came to me as a surprise. Yes, the work did get tedious, and I most definitely have myself to blame since I procrastinated until the very last minute, but it was ultimately an excellent learning and a very rewarding lesson. Since the bulk of MS1 code was given to us, I failed to realize just how much

effort was involved in building a web application. I thoroughly felt like I learned a lot more during this MS than I did during any of the actual theories I learned in lectures simply because I was forced to learn and experiment with things while trying to make them work. I've always preferred practical learning projects like these over theory-intensive ones, just because of their inherent ambiguous nature. Overall, although I'm still very much a novice, I feel like I've gained a concrete understanding of how to utilize and integrate PHP, JS, MySQL, HTML, CSS, etc. in order to build a well-functioning web application. Most importantly, though, I've gained a rudimentary understanding of what APIs are and how to work with them and fetch data from them, which is an essential skill for any IT-related student, especially since I only knew of APIs as a buzzword until last semester.

Regarding issues, I've faced so many during this assignment that I honestly don't even recall all of them. For starters, I was unaware of how to properly cache data until I had used up about 85% of all my API calls (I just made a new account later). Next up, inserting API data into my database was a problem of its own due to how my data was formatted (It had like 7 associative arrays nested within one another). During that process, I learned that my parameters for the API had only one possible combination with the entire data categorized by a separate National Location ID. So I had to go about hard-coding those parameter values and manually checking the national location ID from the user within the API data and making it a pseudo-parameter. Then I had several issues with labels for form fields, especially since I had too many of them so I kind of lost consistency across pages which caused a few errors. Lastly, although the PHP server-side validation itself wasn't too complex, the way my form was being passed made it difficult to disable JS Validation. So, that was another snippet of code I had to come up with to disable the submit event listener so as to effectively disable JS validation which allowed me to finally test my server-side validation. All of this is still just the tip of the iceberg when it comes to the errors I faced but I'm glad I came out on the bright side having fixed them all (Only took 35 hrs). Either way, I look forward to fleshing out the assignment in MS3.

#### 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

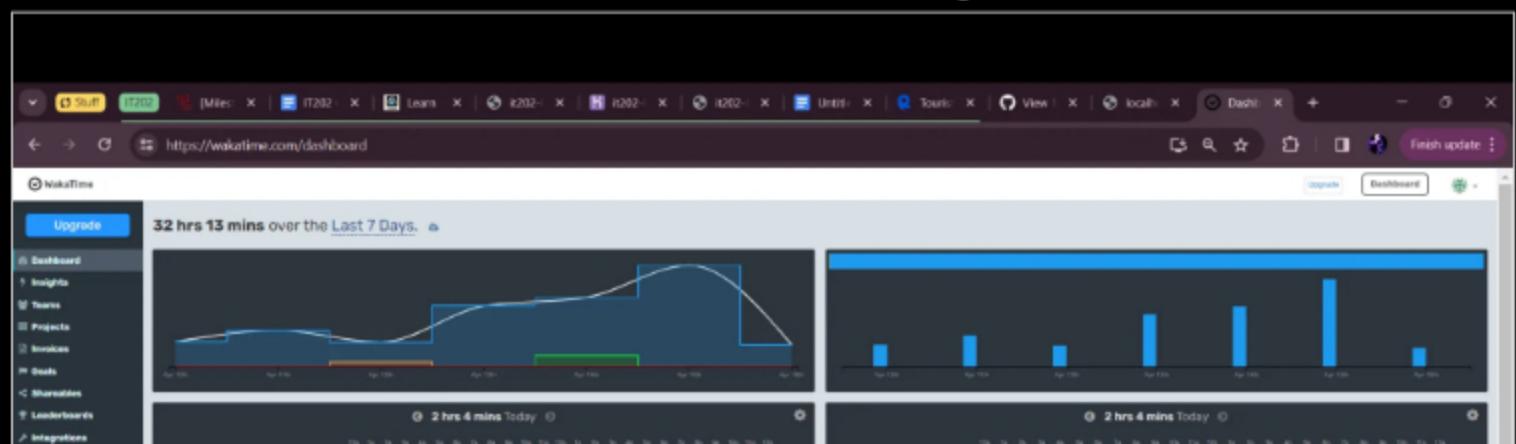
##### Task Screenshots:

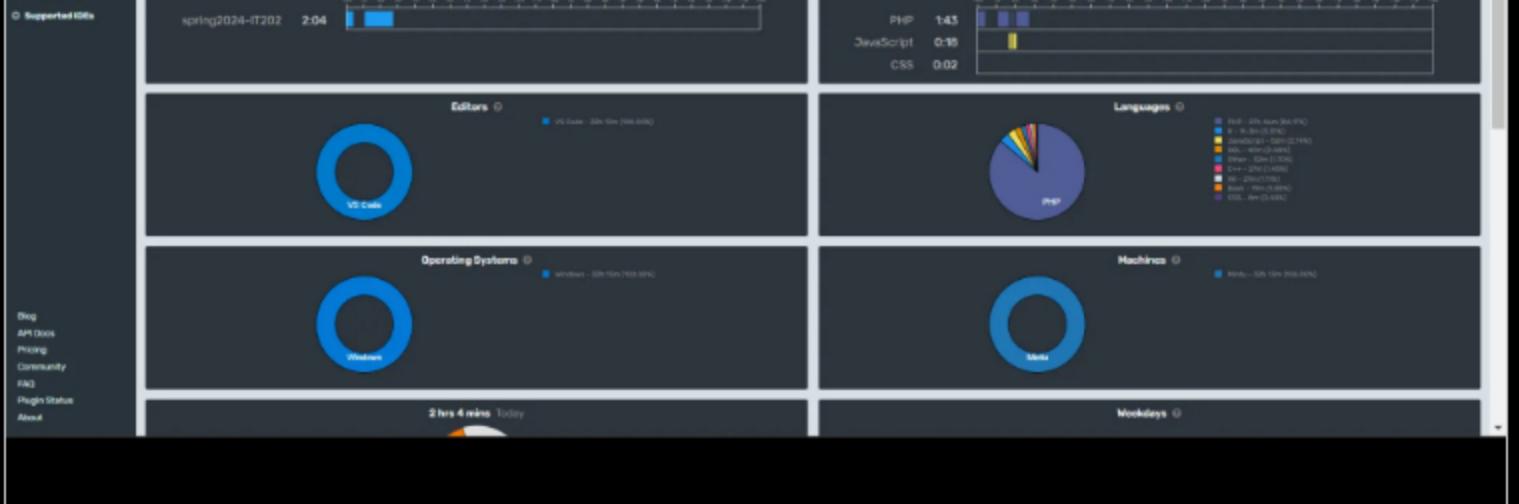
Gallery Style: Large View

Small

Medium

Large





WakaTime Screenshot covering the time spent across MS2.

End of Assignment