

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-450-M2024/it202-api-project-milestone-2-2024-m24/grade/mm2849>

IT202-450-M2024 - [IT202] API Project Milestone 2 2024 M24

Submissions:

Submission Selection

1 Submission [active] 7/27/2024 1:57:26 AM ▾

Instructions

[^ COLLAPSE ^](#)

Implement the Milestone 2 features from the project's proposal document:

<https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88EWVwfo/view>

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: 23 Points: 10.00

 Define Appropriate Tables for Data (1 pt.)

[^ COLLAPSE ^](#)

 Task #1 - Points: 1

Text: Screenshots of Table SQL

Checklist

*The checkboxes are for your own tracking

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

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

#1)
Screenshot
of the table



Caption (required) ✓

Describe/highlight what's being shown
Showing Screenshot of the table.

Explanation (required)



Explain the columns and what data they represent (briefly), also note any normalization that may have been necessary

PREVIEW RESPONSE

The data for my project include code, code2, name, localname, continent, region, independence year, surface area, and government form. The majority of the data is varchar, but for the independence year, I had to convert it to smallint so that the year could be determined correctly using the API.

Task #2 - Points: 1

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

COLLAPSE

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

i Details:

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

URL #1

<https://github.com/mm2849/mm2849-IT202-450/pull/43>

URL #2

<https://github.com/mm2849/mm2849-IT202-451>

+ ADD ANOTHER URL



Data Creation Page (2 pts.)

[COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the creation page

i Details:

Heroku dev url must be visible in all relevant screenshots

#1) Show potentially valid data



Caption (required) ✓

Describe/highlight what's being shown
Showing valid data filled in for the custom creation page.

#2) Show how the API data is



Caption (required) ✓

Describe/highlight what's being shown
Showing how the API data is fetched for API data

#3) Show examples of validation



Caption (required) ✓

Describe/highlight what's being shown
Showing examples of validation messages

#4) Show an example of successful



Caption (required) ✓

Describe/highlight what's being shown
Showing an example of successful creation message

Explanation (required)



Briefly explain the code

PREVIEW RESPONSE

In the code the keys are loaded in the create_trip from load_api keys. The API key is being stored in .env file and gets pulled from that into the load_api_keys.

#5)
Design/Style
should be



Caption (required) ✓

Describe/highlight what's being shown
Showing Design/Style considered.

Explanation (required)



Briefly explain your design choices

PREVIEW RESPONSE

I choose this type of design because I believe it is really easy to read. A large field allows the user to see what they're typing clearly. These colors also make it easier to comprehend any error or success messages that may appear. Overall, I felt that this design was the greatest option for my project.

[COLLAPSE](#)

Task #2 - Points: 1

Text: Screenshots of creation page code

Details:

Include ucid/date comments for each code screenshot

#1) Form
should have
correct data



Caption (required) ✓

Describe/highlight
what's being shown
Showing forms creation
page.

Explanation (required)



Briefly talk about each
field type and why it was
chosen

[PREVIEW RESPONSE](#)

The first line of code I
have is for who has
authorization to search
or create a page, and I
assigned that position to
the administrator. The
code then searches for a
certain name, which is a
country, and uses the
API to provide the user
with all of the
information about that
country and add it to the
database. The code will
then display on screen
that this was received
from the API, then the

#2) Form
should have
correct



Caption (required)

Describe/highlight
what's being shown
Missing caption

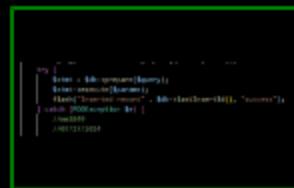
Explanation (required)

Briefly explain the
validations

[PREVIEW RESPONSE](#)

Missing text

#3)
Successful
creation



Caption (required) ✓

Describe/highlight
what's being shown
Showing Successful
creation should have a
user-friendly message

Explanation (required)



Explain how
duplicate/existing data
is handled

[PREVIEW RESPONSE](#)

When the data is getting
entered in the database
1062 handles the
duplicate/existing data.
It detects whether the
SQL error code in the
exception \$e is 1062,
indicating a
duplicate/existing entry
error. If this is the case,
the user will see a
warning message "A
country with this name
already exists."

#4) Any
errors should
have user-



Caption (required) ✓

Describe/highlight
what's being shown
Showing errors should
have user-friendly
messages

Explanation (required)



Briefly describe the
scenarios

[PREVIEW RESPONSE](#)

One of the issues that a
user may encounter is
that the country already
exists in the database. A
further potential issue is
if the country's given
name is invalid, in which
case an error message
will be displayed. If the
user manually creates a
country and does not fill
out the required fields,
an error message will
appear.

from the API rather than created manually with the `is_api`. The user also has the option of creating their own custom country. Once the user has manually established a country and entered the necessary data, the information will be added to the database, and `is_api` will be zero because the user has not utilized the API for this. I thought it was a good idea to include all of the fields, including `code`, `code2`, `name`, `localname`, `continent`, `region`, `independence year`, `surface area`, and `government form`, so that the user could quickly define the nation.

#5) Include the form/process



```
const express = require('express');
const router = express.Router();
const { Country } = require('../models');
const { Op } = require('sequelize');

router.get('/api/countries', async (req, res) => {
  const { name, continent, region, independenceYear, surfaceArea, governmentForm } = req.query;
  const filters = {};
  if (name) filters.name = name;
  if (continent) filters.continent = continent;
  if (region) filters.region = region;
  if (independenceYear) filters.independenceYear = independenceYear;
  if (surfaceArea) filters.surfaceArea = surfaceArea;
  if (governmentForm) filters.governmentForm = governmentForm;
  const countries = await Country.findAll({
    where: filters,
    limit: 100
  });
  res.json(countries);
});

module.exports = router;
```

Caption (required) ✓
Describe/highlight what's being shown
Showing the form/process for fetching API data

Explanation (required)



Briefly explain the steps (include how duplicates are handled)

#6) Include some indicator



```
const express = require('express');
const router = express.Router();
const { Country } = require('../models');
const { Op } = require('sequelize');

router.get('/api/countries', async (req, res) => {
  const { name, continent, region, independenceYear, surfaceArea, governmentForm } = req.query;
  const filters = {};
  if (name) filters.name = name;
  if (continent) filters.continent = continent;
  if (region) filters.region = region;
  if (independenceYear) filters.independenceYear = independenceYear;
  if (surfaceArea) filters.surfaceArea = surfaceArea;
  if (governmentForm) filters.governmentForm = governmentForm;
  const countries = await Country.findAll({
    where: filters,
    limit: 100
  });
  res.json(countries);
});

module.exports = router;
```

Caption (required) ✓
Describe/highlight what's being shown
Showing some indicator between custom data and API data

Explanation (required)



Briefly mention what the indicator is (i.e., `api_id` if the API has ids or `is_api` if the API is a custom endpoint)

#7) Include any other rules like



```
const express = require('express');
const router = express.Router();
const { Country } = require('../models');
const { Op } = require('sequelize');

router.get('/api/countries', async (req, res) => {
  const { name, continent, region, independenceYear, surfaceArea, governmentForm } = req.query;
  const filters = {};
  if (name) filters.name = name;
  if (continent) filters.continent = continent;
  if (region) filters.region = region;
  if (independenceYear) filters.independenceYear = independenceYear;
  if (surfaceArea) filters.surfaceArea = surfaceArea;
  if (governmentForm) filters.governmentForm = governmentForm;
  const countries = await Country.findAll({
    where: filters,
    limit: 100
  });
  res.json(countries);
});

module.exports = router;
```

Caption (required) ✓
Describe/highlight what's being shown
Showing other rules like role guards and login checks

Explanation (required)



Briefly explain the logic/reasoning

PREVIEW RESPONSE

 PREVIEW RESPONSE

The code will search for a specific name given by the user, which will be the country's name, and then utilize the API to provide the user with all of the information about that country and add it to the database. The code will then display on screen that this was received from the API rather than created manually with the `is_api`. Duplicates are handled before the data is entered into the database, and if the existing data already exists, a friendly error message is displayed.

 as a boolean-like column, etc)

 PREVIEW RESPONSE

The indication "is_api" determines if the user has searched for an country or entered a country manually. The \$quote is_api is set to 1 for all search results and manually set to 0 so that the user may later determine whether or not the API was used.

For the project I was working on, I felt it would be a good idea to give the administrator the ability to search for or create a country so that the API would not be fetched on a continuous basis. I assumed that if this went into production, roles would be assigned to other specific users.

Task #3 - Points: 1

Text: Screenshot of records from DB

#1) Show at least one record



Caption (required) ✓
Describe/highlight what's being shown (note what differs from a custom record)
Showing at least one record fetched from the API

#2) Show at least one record



Caption (required) ✓
Describe/highlight what's being shown (note what differs from the API record)
Showing at least one record created via the creation form

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-mm2849->

https://it202-mm2849-prod-3453894141df.herokuapp.com/admin/list_country.php



URL #2

<https://github.com/mm2849/mm2849-IT202-450/pull/43>

<https://github.com/mm2849/mm2849-IT202-450>



+ ADD ANOTHER URL

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

▲ COLLAPSE ▲

Task #1 - Points: 1

Text: Screenshots of the list page

ⓘ Details:

Heroku dev url must be visible in all relevant screenshots

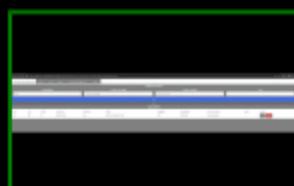
#1) Show the page of your entities



Caption (required) ✓

Describe/highlight what's being shown
Showing the page of my entities listed

#2) Show the filter/sort form based



Caption (required) ✓

Describe/highlight what's being shown
Showing the filter/sort form based on my data and the required limit field

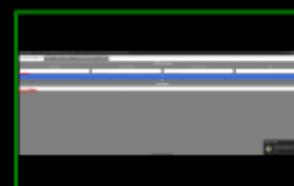
#3) Demonstrate a few varied



Caption (required) ✓

Describe/highlight what's being shown
Demonstrating a few varied filters/sorts

#4) Demonstrate a filter that



Caption (required) ✓

Describe/highlight what's being shown
Demonstrating a filter that doesn't have any records.

Explanation (required)

✓

Briefly describe the page

 PREVIEW RESPONSE

The page describes various aspects, including the country's name and code, code2, name, localname, continent, region, independence year, surface area, and government form. The page also has a filter option, which allows the user to locate a specific country in the list if they are unable to identify it. They could also search countries based on continent. The user can also limit the list to only see two of the three countries. There is also the ability to edit and delete the country from the page.

field

Explanation (required)

✓

Briefly mention what's available to the user

 PREVIEW RESPONSE

The user can filter the results with the country name, country local name, continent and the limit. There is also a clear option where the user can clear the field set and view the list again with all the countries.

#5) Each list item should have a link of



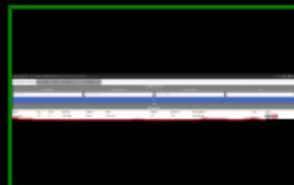
Caption (required) ✓
Describe/highlight what's being shown
Showing list items with actions.

Explanation (required)

✓

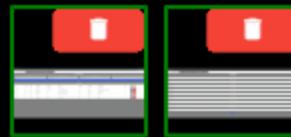
Mention which users can interact with the view, edit, and delete

#6) Each list item should have a



Caption (required) ✓
Describe/highlight what's being shown
Showing Each list item with a summary of the entity

#7) Design/Style should be



Caption (required) ✓

Describe/highlight what's being shown
Showing Design/Style Considered

Explanation (required)

✓

Briefly explain your design choices

 PREVIEW RESPONSE

I chose this design

links

 PREVIEW RESPONSE

The admins are the users who have access to the view, edit, and remove links in this project. Administrators can grant admin rights to a specific user to change and delete the country.

Links and design

because I feel it will be very easy for the administrator to understand and use. A big field helps the administrator easily see what they are typing. The different colors also make it easier to understand any error or success messages that may show.

Task #2 - Points: 1

Text: Screenshots of the list page code

Details:

Include ucid/date comments for each code screenshot

#1) Show the filter/sort form



Caption (required) ✓

Describe/highlight what's being shown
Showing the filter/sort form generation.

Explanation (required)



Briefly explain how the code works (i.e., some stuff may be dynamic)

 PREVIEW RESPONSE

The code modifies the form values and adds a condition to the query

#2) Show the DB query and how the



Caption (required) ✓

Describe/highlight what's being shown
Showing the DB query and how the filter/sort is handled.

Explanation (required)



Briefly explain the related logic

 PREVIEW RESPONSE

The limit field is restricted to a maximum of 100. The filter option

#3) Show how the output is



Caption (required) ✓

Describe/highlight what's being shown
Showing how the output is generated and displayed.

Explanation (required)



Briefly explain the related logic

 PREVIEW RESPONSE

The above query includes all columns. To provide an output of the

#4) Show any restrictions



Caption (required) ✓

Describe/highlight what's being shown
Showing restrictions like role guard or login checks

Explanation (required)



Briefly explain the logic/reasoning

 PREVIEW RESPONSE

For the project I was working on, I felt it would be a good idea to

for name, local name, and continent based on all columns in the table countries. It also determines the column and order, ensures that they are valid, and limits the number of entries the user may enter.

This API endpoint allows the user to enter either a country name, local name, or continent. The user will be able to enter a single letter and be shown with a list of countries that contain with that letter. The user can also choose a specific limit that they want to see and will be presented with that number of countries.

For the API endpoint users entered, we use a foreach method with only name, local, name, and continent as inputs.

give the administrator the ability to search for or create a country so that the API would not be fetched on a continuous basis. I assumed that if this went into production, roles would be assigned to other specific users.

Task #3 - 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-mm2849-prod-3453894141df.herokuapp.com/project/admin/create_trip.php

UR

<https://it202-mm2849-prod-3453894141df.herokuapp.com/>



URL #2

<https://github.com/mm2849/mm2849-IT202-450/pull/43>

UR

<https://github.com/mm2849/mm2849-IT202-450/pull/43>



[+ ADD ANOTHER URL](#)

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

[^COLLAPSE ^](#)

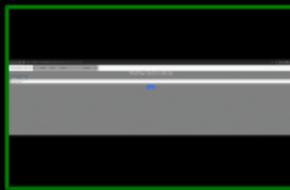
Task #1 - Points: 1

Text: Screenshots of the details page

Details:

Heroku dev url must be visible in all relevant screenshots

#1) Entity
should be
fetch by id



Caption (required) ✓

*Describe/highlight
what's being shown*
Showing Entity should
be fetch by id

#2) A
missing id
should



Caption (required)

*Describe/highlight
what's being shown*
Missing caption

#3)
Design/Style
should be



Caption (required) ✓

*Describe/highlight
what's being shown*
Showing Design/Style
should be considered

#4) Data
shown
should be



Caption (required) ✓

*Describe/highlight
what's being shown*
Showing Data shown
should be more
detailed/inclusive.

Explanation (required)



*Briefly explain your
design choices*

PREVIEW RESPONSE

I chose this design
because I feel it will be
very easy for the
administrator to
understand and use. A
big field helps the
administrator easily see
all the details about that
country.

Explanation (required)



*Briefly explain the
details shown and how it
differs from the list view*

PREVIEW RESPONSE

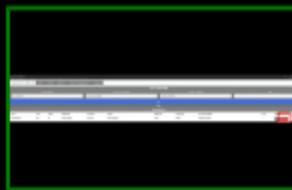
The data above includes
all information about the
country.

#5) There
should be a
link to edit



Caption (required) ✓
*Describe/highlight
what's being shown*
Showing the edit link

#6) There
should be a
link to delete



Caption (required) ✓
*Describe/highlight
what's being shown*
Showing the delete link

 COLLAPSE

Task #2 - Points: 1

Text: Screenshots of the details page code

Details:

Include uid/date comments for each code screenshot

#1) Show how id is fetched



Caption (required) ✓

Describe/highlight what's being shown
Showing Screenshot of how id is fethched.

Explanation (required)

✓
Briefly explain the logic and how it's handled when the property is missing or not "valid"

 PREVIEW RESPONSE

When the property is missing or invalid, an error code stating "error search countries" will appear.

#2) Show the DB query to get the



Caption (required) ✓

Describe/highlight what's being shown
Showing the DB query to get the record

Explanation (required)

✓
Briefly explain the logic

 PREVIEW RESPONSE

First, the code creates a connection with the database. Then it tries to retrieve the query's results. If a problem occurs, the user will be informed with the error message "Error Searching for Country."

#3) Show the code related to presenting



Caption (required)

Describe/highlight what's being shown
Missing caption

Explanation (required)

Briefly explain the logic

 PREVIEW RESPONSE

Missing text

Checklist

*The checkboxes are for your own tracking

Task #3 - Points: 1

Text: Add related links

#

Points

Details

Include the heroku prod link for this page

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

URL #1

https://it202-mm2849-prod-3453894141df.herokuapp.com/admin/list_country.php



URL

https://it202-mm2849-prod-3453894141df.herokuapp.com/admin/list_country.php

URL #2

<https://github.com/mm2849/mm2849-IT202-450/pull/43>



+ ADD ANOTHER URL



Edit Data Page (2 pts.)

[COLLAPSE](#)



Task #1 - Points: 1

Text: Screenshots of the edit page

[COLLAPSE](#)

i Details:

Heroku dev url must be visible in all relevant screenshots

#1) Show before and after



Caption (required) ✓

Describe/highlight what's being shown
Showing Show before and after screenshots of data edited

Explanation (required)

✓
Briefly explain what you changed

[PREVIEW RESPONSE](#)

In the screenshot above, I modified the continent of the United States, the

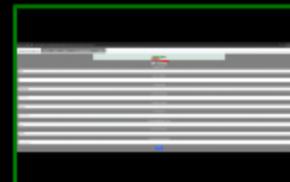
#2) Show examples of validation



Caption (required) ✓

Describe/highlight what's being shown
Showing examples of validation messages

#3) Show an example of successful



Caption (required) ✓

Describe/highlight what's being shown
Showing an example of successful edit messages

#4) Design/Style should be



Caption (required) ✓

Describe/highlight what's being shown
Showing Design/Style.

Explanation (required)

✓
Briefly explain your design choices

[PREVIEW RESPONSE](#)

I chose this design because I feel it will be very easy for the

or the United States, the year of independence, and the surface area.

administrator to understand and use. A big field helps the administrator easily see all the details that they are editing on that country.

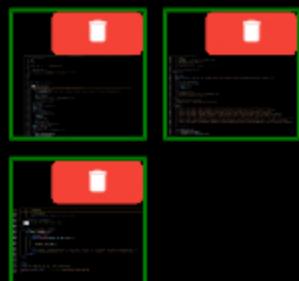
Task #2 - Points: 1

Text: Screenshots of edit page code

Details:

Include ucid/date comments for each code screenshot

#1) Form should have correct data



Caption (required) ✓

Describe/highlight what's being shown
Showing Form should have correct data types for each property being requested.

Explanation (required)

✓
Briefly explain the data type choices

PREVIEW RESPONSE

For my project, I decided that it would be beneficial to have all of the data types to modify the country as well. I

#2) Form should have correct



Caption (required)

Describe/highlight what's being shown
Missing caption

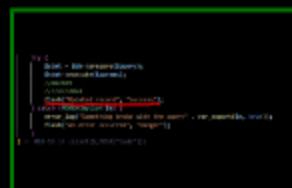
Explanation (required)

Briefly explain the validations

PREVIEW RESPONSE

Missing text

#3) Successful edit should



Caption (required) ✓

Describe/highlight what's being shown
Showing Successful edit should have a user-friendly message

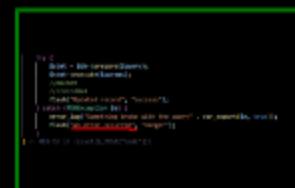
Explanation (required)

✓
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

PREVIEW RESPONSE

First, the code checks for all records in the database. Then it connects to the database and updates the record.

#4) Any errors should have user-



Caption (required) ✓

Describe/highlight what's being shown
Showing Any errors should have user-friendly messages

Explanation (required)

✓
Briefly explain the possible errors
 PREVIEW RESPONSE
A possible problem that the user may encounter is entering an invalid country code. A country's codes are limited to three, and anything more will result in an error. The same is true for the name.

included everything, including the country's name and code, code 2, name, localname, continent, region, independence year, surface area, and government form. When editing the country, the user will be prompted to specify a new surface area.

database, allowing it to edit a specific country.

true for the independence year. Independence year is set to four, and anything greater than four will result in an error.

#5) Include any other rules like



```
SELECT * FROM countries WHERE id = 1;
```

```
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | name | code | code_2 | name | localname | continent | region | independence_year | surface_area | government_form |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | USA   | US   | USA   | United States of America | United States of America | North America | North America | 1776-07-04 | 9.587e+12 | Federal Republic of America |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Caption (required) ✓

Describe/highlight what's being shown
Showing any other rules like role guards and login checks.

Explanation (required)



Briefly explain the logic/reasoning

PREVIEW RESPONSE

For the project I was working on, I thought it would be a good idea to allow the administrator to edit a country. The administrator will be the only person with the ability to edit a nation. The reasoning is that if users are permitted to edit the country, it will become more confusing for other users that visit the site. If an

administrator uses the correct information, it will benefit all users.

Task #3 - Points: 1

Text: Screenshot of records from DB

#1) Show a before and after



Caption (required) ✓

Describe/highlight what's being shown
Showing a before and after screenshot of the record

Explanation (required)



Explain what differs

PREVIEW RESPONSE

Showing that South Africa's Independence year and the surface area were edited.

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-mm2849-dev-69196d81a269.herokuapp.com/admin/edit_country.php?id=1](https://it202-mm2849-dev-69196d81a269.herokuapp.com/admin/edit_country.php?)

URL #2

<https://github.com/mm2849/mm2849-IT202-450/pull/44>

UR

[https://it202-mm2849-dev-69196d81a269.herokuapp.com/admin/edit_country.php?id=1](https://it202-mm2849-dev-69196d81a269.herokuapp.com/admin/edit_country.php?)



UR

<https://github.com/mm2849/mm2849-IT202-450/pull/44>



+ ADD ANOTHER URL

● Delete Handling (1 pt.)

[COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots related to delete

● Details:

Heroku dev url must be visible in all relevant screenshots

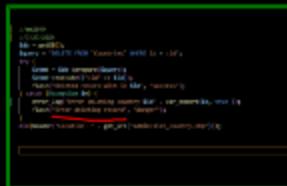
Include ucid/date comments for each code screenshot

#1) Show the success message of



Caption (required) ✓
Describe/highlight what's being shown
Showing the success message of a delete

#2) Show any error messages of



Caption (required) ✓
Describe/highlight what's being shown
Showing error messages of a failed delete

#3) Show the code related to the delete



Caption (required) ✓
Describe/highlight what's being shown
Showing the code related to the delete processing

#4) Explain the delete logic



Explanation (required)



Is it a soft or hard delete? Are there any necessary roles or restrictions? (can only delete their data, can only be done by admin, etc)?

[PREVIEW RESPONSE](#)

The delete option will be a hard delete, deleting the data completely from the database. The data can only be erased by the administrator.

Explanation (required)

✓
Explain in concise steps how this logically works

[PREVIEW RESPONSE](#)

First, the code above will establish a database connection. Then it

Explanation (required)

✓
Explain in concise steps how this logically works

[PREVIEW RESPONSE](#)

Initially, the code only allows the administrator to delete. Then it checks

connection. Then it builds a query to delete data from the database's countries table. If the country with the correct id is not provided, an error will occur during deletion.

to delete. Then it checks if the id is valid, and if not, it returns an invalid id error. The code will then establish a connection with the database. Then it creates a query to delete data from the database's nations table. If the correct country and id are not provided, an error will occur during deletion.

Task #2 - Points: 1

Text: Screenshots of the data

#1) Show a before and after



Caption (required) ✓

Describe/highlight what's being shown (note precisely what changed)

Showing that United Kingdom was in the database in the first screenshot and then deleted from the database.

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

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

URL #1

<https://github.com/mm2849/mm2849-IT202-450/pull/44>

th

<https://github.com/mm2849/mm2849-IT202-451>

+ ADD ANOTHER URL

● Misc (1 pt.)

[COLLAPSE ^](#)

● Task #1 - Points: 1

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Showing Screenshot of GitHub Project Board

● Task #2 - Points: 1

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

[COLLAPSE ^](#)

URL #1

<https://github.com/users/mm2849/projects/2>

UR

<https://github.com/users/mm2849/projects/2>

+ ADD ANOTHER URL

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

The main challenge I had when working on this assignment was connecting my API to VSCode and Heroku. After I got that to work, I had trouble fetching and creating both at the same time for the country. With the assistance of the professor, I was able to make both work at the same time.

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

Showing WakaTime Screenshot.

End of Assignment