Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT202-450-M2024/it202-api-project-milestone-3-2024-m24/grade/mm2849

IT202-450-M2024 - [IT202] API Project Milestone 3 2024 m24

Submissions:

Submission Selection

1 Submission [active] 8/4/2024 1:58:22 AM

Instructions

COLLAPSE

Overview Video: https://youtu.be/-4hlb9MXrQE

1. Implement the Milestone 3 features from the project's proposal document: https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88EWVwfo/viev
2. Make sure you add your ucid/date as code comments where code changes are done
3. All code changes should reach the Milestone3 branch
4. Create a pull request from Milestone3 to dev and keep it open until you get the output PDF from this assignment.
5. Gather the evidence of feature completion based on the below tasks.
6. Once finished, get the output PDF and copy/move it to your repository folder on your local machine.
7. Run the necessary git add, commit, and push steps to move it to GitHub
8. Complete the pull request that was opened earlier
9. Create and merge a pull request from dev to prod
10. Upload the same output PDF to Canvas

Branch name: Milestone3

Tasks: 21 Points: 10.00

API (1 pt.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Data Related to Users

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	What's the concept/association?
#2	1	What sort of relationship is it (one to many, many to one, many to many, etc)
#3	1	Note any other considerations

Response:

The concept/association of this online application will allow users to purchase many countries across the world. The relationship employed for this is many-to-many, with unique pairs. Admins will be able to fetch all countries in the world and edit them. Then, once obtained, users will be able to purchase them.

COLLAPSE

Task #2 - Points: 1

Text: Updating Entities

Checklist *The checkboxes are for your own tracking

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

Response:

When an update occurs, whether manually or by API, it is automatically applied to what the users see. When an administrator manually updates a country, it is automatically updated for all users. Users will always view the most recent and new info.

COLLAPSE

Handle Data Association (1 pt.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the code

Details:

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

Include ucid/date comments for each code screenshot

#1) Show the related code

Caption (required) ✓

Describe/highlight what's being shown

Showing the related code for Handle Data Association.

Explanation (required) ✓

Explain in concise steps how this logically works and mention which option your application handles regarding association

PREVIEW RESPONSE

This code works by establishing a database connection when a user is logged in. It then inserts data into the database using the users used_ids and the countries country_ids. When the user purchases the country, the success message "Congratulations on purchasing a country" appears. If there are any issues, the user will be prompted with "The country is not available." For my project, I selected the purchased option, which may also

be saved as a favorite or wish list.

COLLAPSE

Task #2 - Points: 1

Text: Screenshot of the association table(s)

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

Caption (required) ✓

Describe/highlight what's being shown

Showing the table made to handle the data association.

Explanation (required) ✓

Describe each column/association table

PREVIEW RESPONSE

To handle the data association, I created a new table called UserCountries. The data in this database will include the user_ids of those who purchased the country. This table will keep track of the user's purchased countries and country IDs. The table will also provide the user's purchasing time for the country.

COLLAPSE

Task #3 - Points: 1

Text: Add related links

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Include the heroku prod link for the page that creates the association
#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/countries.php

URL
https://it202-mm2849-prod-3453894141df.herok

URL #2
https://github.com/mm2849/mm2849-IT202-450/pull/51

URL
https://github.com/mm2849/mm2849-IT202-45(

ADD ANOTHER URL

Current User's Association Page (2 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of this page

Details:

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

Some screenshots may be repeated in subtasks, but ensure they highlight the specific subtask requirement.

#1) Show the summary of the results

Caption (required) ✓
Describe/highlight what's being shown

#2) Show the single view buttons/links

Caption (required) ✓
Describe/highlight what's being shown

#3) Show variations of the number

Caption (required) ✓
Describe/highlight what's being shown
Showing the variations

#4) Show variations of the

Showing the summary of the results with relevant information per entity.

Showing the delete all button.

of the number of shown items count and show the count of total number of associated items to the user

Caption (required) ✓
Describe/highlight what's being shown
Showing variations of the filter/sort including no results

^COLLAPSE ^

Task #2 - Points: 1

Text: Screenshot the code

Details:
Include ucid/date comments for each code screenshot

#1) Show the code related to fetching

Caption (required) ✓
Describe/highlight what's being shown
Showing the code related to fetching the user's associations.

Explanation (required)
✓
Explain in concise steps how this logically works and mention how you determine the result list (include the association logic and filters)
PREVIEW RESPONSE
The code begins by establishing a connection with the database. The code then

#2) Show the code related to the

Caption (required) ✓
Describe/highlight what's being shown
Showing the code related to the display of the results

Explanation (required)
✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The code proceeds to create the rows for the country cards to be displayed. When the count is 0, the user will receive the message "No results found", since there is a if state saying that if count===0 then flash the message.

#3) Each record should have

Caption (required)
Describe/highlight what's being shown
Missing caption

Explanation (required)
Explain in concise steps how this logically works
PREVIEW RESPONSE
Missing text

#4) Each record should have

Caption (required) ✓
Describe/highlight what's being shown
Showing button link for delete.

Explanation (required)
✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The code is started by adding the remove button. When this button is clicked, the user will be prompted whether or not to remove the country. If the user clicks cancel then they get redirect to the my_countries page.
Note: In my project, I

includes a variable, which allows the user to delete all of their countries, and when the button is hit, the user is displayed with a message saying if the action was successful or error. The code then creates the search form, which includes the country's name, local name, and continent, as well as a limit that the user may choose to display the number of countries they want to see. The total record count is then returned, displaying the total number of countries purchased by that particular user.

included a delete link for all countries to which the user is associated. Apologies for missing the section where each entity needed a delete button.

#5) Show the logic for deleting all

Caption (required) ✓
Describe/highlight what's being shown
Showing the logic for deleting all associations for the user.

Explanation (required) ✓
Explain in concise steps how this logically works

PREVIEW RESPONSE

The $GET is set for remove to work. The following statement will remove the userid

#6) Show the logic related to the count

Caption (required) ✓
Describe/highlight what's being shown
Showing the logic related to the count of all associated items to the user.

Explanation (required) ✓
Explain in concise steps how this logically works

PREVIEW RESPONSE

The get_total_count is a helpers function which

#7) Show the logic related to the count

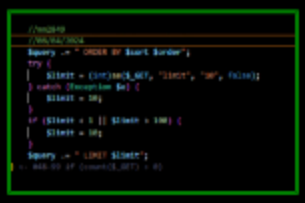
Caption (required) ✓
Describe/highlight what's being shown
Showing the logic related to count of the items on the page.

Explanation (required) ✓
Explain in concise steps how this logically works

PREVIEW RESPONSE

The idea behind render_result_count is that it takes ($results)

#8) Show the logic related to filter/sort



Caption (required) ✓
Describe/highlight what's being shown
Showing the logic related to filter/sort

Explanation (required) ✓
Explain in concise steps how this logically works

PREVIEW RESPONSE

The code above defines the sort and limit. The code then searches for a limit with a valid number

connected with the country. The data will also be removed from the database where the userid gets saved which for my project was UserCountries. Flash messages are generated to notify the user whether their action was successful or unsuccessful. Then at the bottom of the code there is a button created for the delete which when clicked will ask the user if they are sure about deleting the country associated with them.

is being called from render_functions file. The db helpers file will create the function and gets the total count of numbers of rows. This code will get the total count from the DB Countries and perform an inner join between the UserCountries.

and ($total_records), the first being the number of results on the page and the second being the total number of records.

of at least 1 and a maximum of 100. If no number is entered, the code will export a maximum of 10, because 10 is defined as the limit.

COLLAPSE

Task #3 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Include the heroku prod link for the page that creates the association
#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/my_countries.php

url
https://it202-mm2849-prod-3453894141df.herok

URL #2
https://github.com/mm2849/mm2849-IT202-450/pull/51

url
https://github.com/mm2849/mm2849-IT202-45(

+ ADD ANOTHER URL

COLLAPSE

All Users Association Page (likely an admin page) (2 pts.)

Task #1 - Points: 1

^COLLAPSE ^

Text: Screenshots of this page

Details:
Make sure the heroku dev url is visible in the address bar

Some screenshots may be repeated in subtasks, but ensure they highlight the specific subtask requirement.

#1) Show the summary of the results

Caption (required) ✓
Describe/highlight what's being shown
Showing the summary of the results with relevant information per entity

#2) Show the single view buttons/links

Caption (required) ✓
Describe/highlight what's being shown
Showing the delete button. Single view button is on the available countries page where the users can find and purchase.

#3) Show the username related to the

Caption (required) ✓
Describe/highlight what's being shown
Showing the username related to a specific entity.

#4) Show variations of the number

Caption (required) ✓
Describe/highlight what's being shown
Showing variations of the number of shown items count and showing the count of total number of associated items

#5) Show variations of the

Caption (required) ✓
Describe/highlight what's being shown
Showing variations of the filter/sort including no results

COLLAPSE

Task #2 - Points: 1

Text: Screenshot the code

Details:
Include ucid/date comments for each code screenshot

#1) Show the code related to fetching

Caption (required) ✓
Describe/highlight what's being shown
Showing the code related to fetching all associations

Explanation (required)
✓
Explain in concise steps how this logically works and mention how you determine the result list (include the association logic and filters)
PREVIEW RESPONSE
The query goes on to select the Users, the User Countries and the Countries it then goes into Joining the countries table to Usercountries table and that gets joined to users to fetch details. Further on the query will fetch the results and stores

#2) Show the code related to the

Caption (required) ✓
Describe/highlight what's being shown
Showing the code related to the display of the results

Explanation (required)
✓
Explain in concise steps how this logically works and mention the logic for handling the username requirements
PREVIEW RESPONSE
First, the code retrieves the username using $GET. When a username is provided, the code searches for a user who matches the query. The code then renders the input fields and provides a render button for the user to click after entering values into the filter columns. After that, each country's card is displayed using render_country_card, which is retrieved from

#3) Each record should have

Caption (required)
Describe/highlight what's being shown
Missing caption

Explanation (required)
Explain in concise steps how this logically works
PREVIEW RESPONSE
Missing text

#4) Each record should have

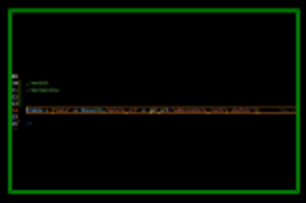
Caption (required)
Describe/highlight what's being shown
Missing caption

Explanation (required)
Explain in concise steps how this logically works
PREVIEW RESPONSE
Missing text

them into $r which is results.

the render functions.

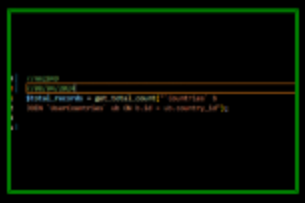
#5) Each record should have



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

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
Note: The delete Button has been added to the code. I was unable to set the delete option in such a way that it removed the relationship. The reason it isn't working is because I couldn't create a new file that would remove the association between the country and user. The plan was to create an association file that would remove the relationship between the user and the country from their tables.

#6) Show the logic related to the count



Caption (required) ✓
Describe/highlight what's being shown
Showing the logic related to the count of all associated items

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The get_total_count is a helpers function which is being called from render_functions file. The db helpers file will create the function and gets the total count of numbers of rows. This code will get the total count from the DB Countries and perform a join between the UserCountries, Where the Usercountries will be connected to the country_id.

#7) Show the logic related to the count

Caption (required) ✓
Describe/highlight what's being shown
Showing the logic related to the count of the items on the page

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The idea behind render_result_count is that it takes ($results) and ($total_records), the first being the number of results on the page and the second being the total number of records.

#8) Show the logic related to filter/sort

Caption (required) ✓
Describe/highlight what's being shown
Showing the logic related to filter/sort

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The code above defines the sort and limit. The code then searches for a limit with a valid number of at least 1 and a maximum of 100. If no number is entered, the code will export a maximum of 10, because 10 is defined as the limit.

COLLAPSE

Task #3 - Points: 1
Text: Add related links

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Include the heroku prod link for the page that creates the association
#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/country_associations.php

URL
https://it202-mm2849-prod-3453894141df.herok

URL #2
https://github.com/mm2849/mm2849-IT202-450/pull/52

URL
https://github.com/mm2849/mm2849-IT202-45(

+ ADD ANOTHER URL

Unassociated Page (2 pts.)

^COLLAPSE^

^COLLAPSE^

Task #1 - Points: 1

Text: Screenshots of this page

Details:
Make sure the heroku dev url is visible in the address bar

Some screenshots may be repeated in subtasks, but ensure they highlight the specific subtask requirement.

#1) Show the summary of the results

Caption (required) ✓
Describe/highlight what's being shown
Showing the screenshot of results with relevant

#2) Show the single view buttons/links

Caption (required) ✓
Describe/highlight what's being shown
Showing the screenshot with single view buttons.

#3) Show variations of the number

Caption (required) ✓
Describe/highlight what's being shown
Showing variations of the number of shown items count and show the count of total

#4) Show variations of the

Caption (required) ✓
Describe/highlight what's being shown
Showing variations of the filter/sort including no results.

information per entity.

number of associated items.

COLLAPSE

Task #2 - Points: 1

Text: Screenshot the code

Details:

Include ucid/date comments for each code screenshot

#1) Show the code related to fetching

Caption (required) ✓

Describe/highlight what's being shown

Showing the code related to fetching all unassociated entities

Explanation (required)

✓

Explain in concise steps how this logically works and mention how you determine the result list (include the unassociated logic and filters)

PREVIEW RESPONSE

In the above code, the query retrieves the id information from the UserCountries table when the user id is not present, so that all

#2) Show the code related to the

Caption (required) ✓

Describe/highlight what's being shown

Showing the code related to the display of the results.

Explanation (required)

✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The code renders the input fields and provides a render button for the user to click after entering values into the filter columns. After that, each country's card is displayed using render_country_card, which is retrieved from the render functions.

#3) Each record should have

Caption (required) ✓

Describe/highlight what's being shown

Showing the button for single view.

Explanation (required)

✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The view button is being built in a country card using the specified names, local name, and continent. It is then loaded into the render functions file. The render_country_card function then gets executed on the unassociated page.

#4) Show the logic related to the count

Caption (required) ✓

Describe/highlight what's being shown

Showing the logic related to the count of all unassociated items

Explanation (required)

✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The get_total_count is a helpers function which is being called from render_functions file. The db helpers file will create the function and gets the total count of numbers of rows. This code will get the total count from the DB Countries where the country will get the country_id from

countries without a given id are displayed on the accessible countries page.

UserCountries.

#5) Show the logic related to the count

Caption (required) ✓
Describe/highlight what's being shown
Showing the logic related to the count of the items on the page

Explanation (required)
✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The render_result_count is a function that is called by render functions. It will calculate the total_records received for all unassociated items and display the number of counts.

#6) Show the logic related to filter/sort

Caption (required) ✓
Describe/highlight what's being shown
Showing the logic related to filter/sort.

Explanation (required)
✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The code above defines the sort and limit. The code then searches for a limit with a valid number of at least 1 and a maximum of 100. If no number is entered, the code will export a maximum of 10, because 10 is defined as the limit.

COLLAPSE

Task #3 - Points: 1
Text: Add related links

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Include the heroku prod link for the page that creates the

#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/available_countries.php

URL
https://it202-mm2849-prod-3453894141df.herok

URL #2
https://github.com/mm2849/mm2849-IT202-450/pull/52

URL
https://github.com/mm2849/mm2849-IT202-45(

+ ADD ANOTHER URL

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

^COLLAPSE ^

^COLLAPSE ^

Task #1 - Points: 1

Text: Screenshots of the page

Details:
Make sure the heroku dev url is visible in the address bar

Some screenshots may be repeated in subtasks, but ensure they highlight the specific subtask requirement.

#1) Show the search form with valid

Caption (required) ✓
Describe/highlight what's being shown
Showing search form with valid data.

#2) Show the results of the search

Caption (required) ✓
Describe/highlight what's being shown
Showing the results of the search.

#3) Show the result of entities and

Caption (required) ✓
Describe/highlight what's being shown
Showing the result of entities and users being associated and unassociated.

COLLAPSE

Task #2 - Points: 1

Text: Screenshots of the code

Details:
Include ucid/date comments for each code screenshot

#1) Search form field for finding a

Caption (required) ✓
Describe/highlight what's being shown
Showing search form field for finding a partial match of usernames.

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The users and usernames are initially created. Then it checks to see whether the username is not empty, and then searches the database for users. After searching for the user, it will indicate whether the user has any active countries or which countries have been assigned to that specific user.

#2) Search form field for finding a

Caption (required) ✓
Describe/highlight what's being shown
Showing Search form field for finding a partial match of entities.

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The given code will establish a connection with the database and list all the active countries Names, Local Names and continent.

#3) Code related to getting a

Caption (required) ✓
Describe/highlight what's being shown
Showing code related to getting a max of 25 results.

Explanation (required) ✓
Explain in concise steps how this logically works and describe the steps for the search and how it works for users and entities
PREVIEW RESPONSE
When a country's name is searched in the countries table, 25 countries will be shown. The users do not have that limit. **Note**: I missed out on the 25-user limit.

#4) Code that generates

Caption (required) ✓
Describe/highlight what's being shown
Showing code that generates the checkboxes next to each list.

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
When users are searched, a checkbox is added, allowing many users to be selected at once. The checkbox is then labeled with the user's specified country. When a country is searched, a checkbox is created, allowing numerous countries to be selected at once. The checkbox is then labeled with the countries that are assigned to the users.

#5) Code related to submitting

Caption (required) ✓

Describe/highlight what's being shown

Attaching code related to submitting the checkbox lists.

Explanation (required) ✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The render button function is being called from render functions file.The function render button is called, and the text is typed into Search, and the selected search is submitted.

#6) Code related to applying the

Caption (required) ✓

Describe/highlight what's being shown

Showing code related to applying the associations upon submission

Explanation (required) ✓

Explain in concise steps how this logically works and describe the steps for the associate/unassociate logic for the combination of users and entities

PREVIEW RESPONSE

First, in the given code, users and countries are issued individual ids. If the checkboxes for the user and the nation are not ticked, an error flash will appear. Once checked, a database connection will be established. This will insert the provided data into the UserCountries table, specifically in the columns of user_id and country_id. A success message will appear once the application has been successfully applied.

COLLAPSE

Task #3 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Include the heroku prod link for the page that creates the association
#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/assign_country_user.php

URL

https://it202-mm2849-prod-3453894141df.herok

URL #2

https://github.com/mm2849/mm2849-IT202-450/pull/53

URL

https://github.com/mm2849/mm2849-IT202-45(

+ ADD ANOTHER URL

COLLAPSE

Misc (1 pt.)

COLLAPSE

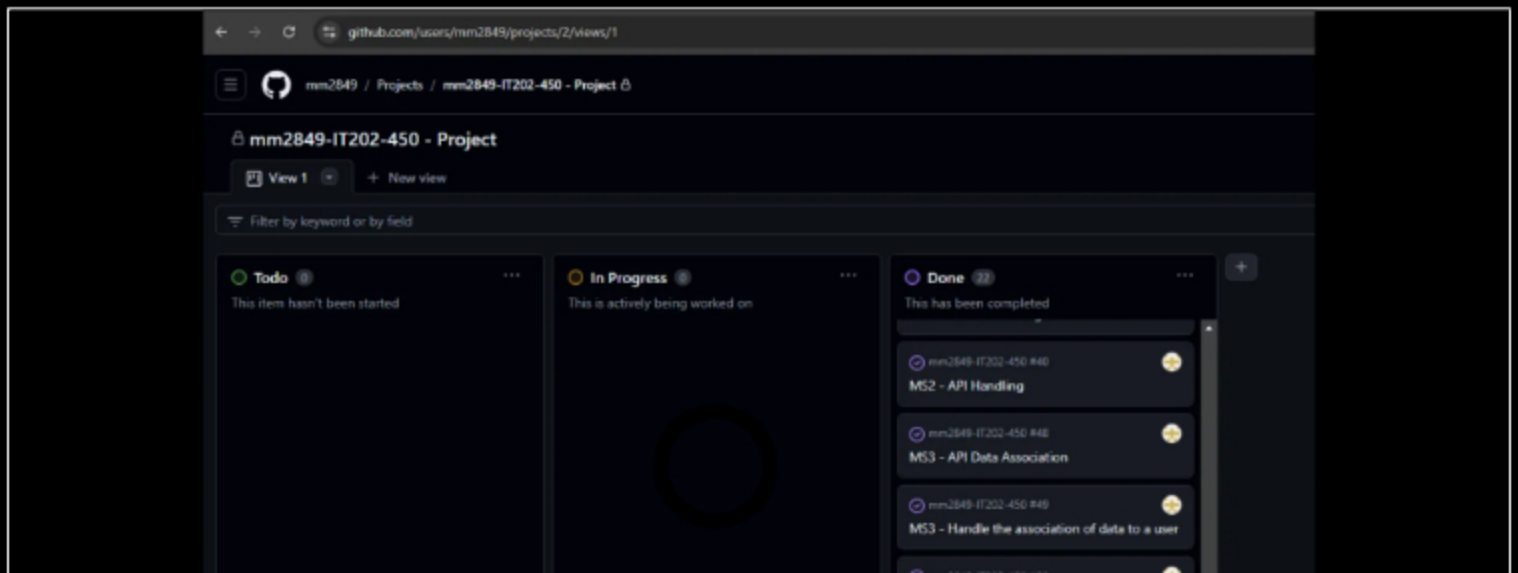
Task #1 - Points: 1

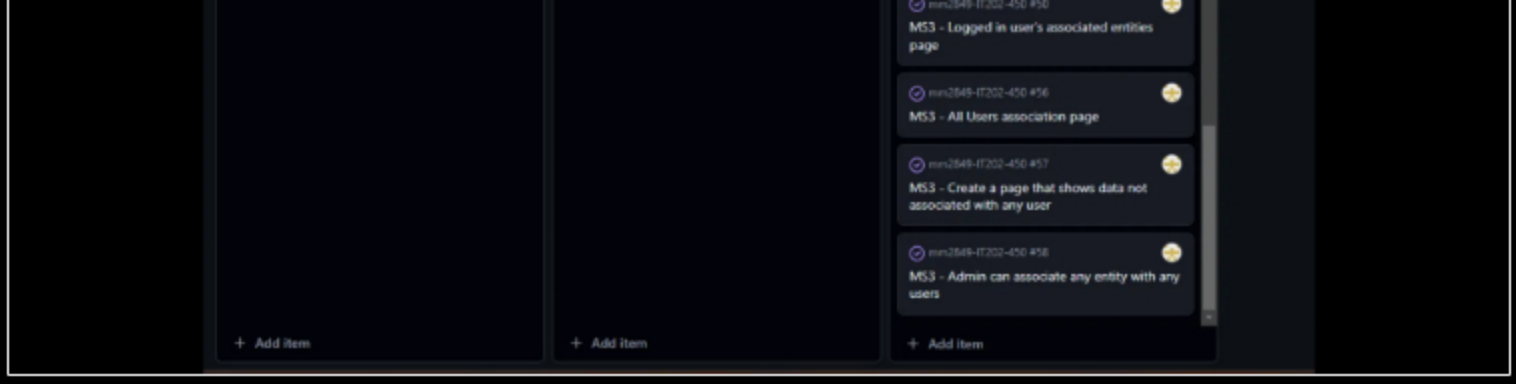
Text: Screenshot of your project board from GitHub (tasks should be in the proper column)

Task Screenshots:

Gallery Style: Large View

Small Medium Large





Showing screenshot of my project board from GitHub

COLLAPSE

Task #2 - Points: 1

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

URL #1

https://github.com/users/mm2849/projects/2/views/1

url

https://github.com/users/mm2849/projects/2/v

+ ADD ANOTHER URL

COLLAPSE

Task #3 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

I had a lot of trouble getting my country cards to function properly. When my country's cards were first presented, there was no info on them. The total_count was also a new experience because we had to write a new function in the db helpers file as well as a new file result_count. The last association country user was a little difficult, but I used the same approach from assign roles and adjusted some things based on the output I was wanting to get. Overall, Milestone 3 taught me a new concept of associating data with users, something I had never done before.

COLLAPSE

Task #4 - Points: 1

Text: WakaTime Screenshot

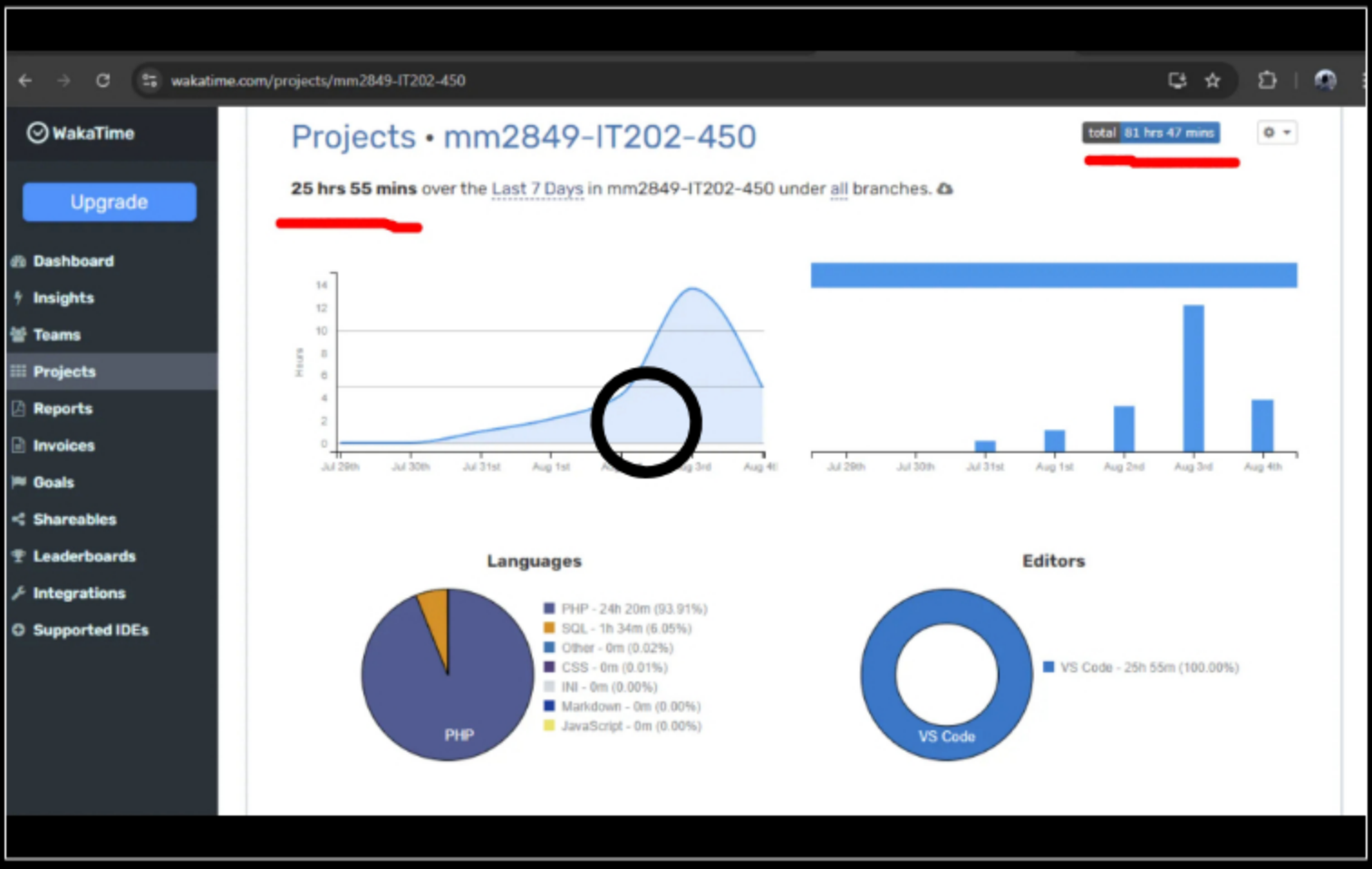
Details:

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