Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT202-452-M2024/generic-module-5-multi-dimension-php-problems/grade/vs53

IT202-452-M2024 - [Generic] Module 5 Multi-Dimension PHP Problems

Submissions:

Submission Selection

1 Submission [active] 6/24/2024 10:14:05 PM

Instructions

COLLAPSE

Overview video: https://youtu.be/lp568G93Noo

Guide:

1. Make sure you're in the dev branch locally and `git pull origin dev` any pending changes.
2. Make a new branch per the recommended branch name below (`git checkout -b ...`).
3. Grab the template code from https://gist.github.com/MattToegel/f7b0489fb0d8cee615d6626056ac5de2
4. Create individual PHP files for each problem and save the files inside your `public_html` folder in a subfolder of your choice.
5. Move the unedited template files to GitHub.
    1. `git add .`
    2. `git commit -m "adding template files"`
    3. `git push origin branch_name` (see below)
    4. Create and open a pull request from the homework branch to main (leave it open until later steps).
6. Note: As you work, it's recommended to add/commit at least after each solution is done (i.e., 3+ times in this case).
    1. Make sure the files are saved before doing this.
7. Fill in the items in the worksheet below (save as often as necessary).
8. Once finished, export the worksheet.
9. Add the output file to any location of your choice in your repository folder (i.e., a `Module5` folder).
10. Check that git sees it via `git status` .
11. If everything is good, continue to submit.
    1. Track the file(s) via `git add` .

2. Commit the changes via `git commit` (don't forget the commit message).
3. Push the changes to GitHub via `git push` (don't forget to refer to the proper branch).
4. Create a pull request from the homework related branch to main (i.e., dev <- "homework branch").
5. Open and complete the merge of the pull request (it should turn purple).
6. Locally checkout dev and pull the latest changes (to prepare for future work).

12. Take the same output file and upload it to Canvas.

Branch name: M5-MD-PHP-Problems

Tasks: 6 Points: 10.00

Problem 1 (3 pts.)

^COLLAPSE ^

^COLLAPSE ^

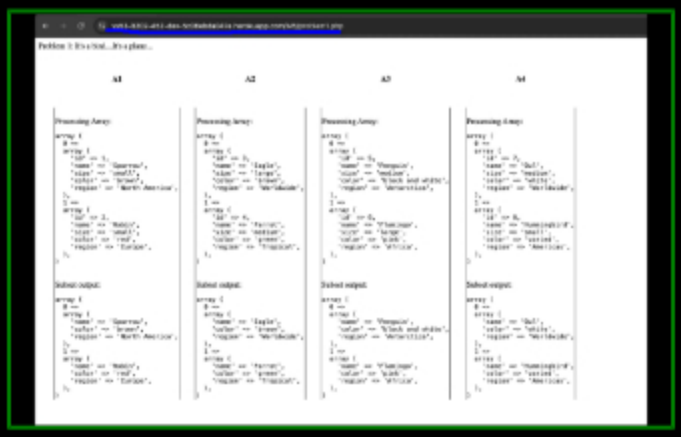
Task #1 - Points: 1

Text: Problem 1 Evidence

Details:
Only make edits where the template code mentions.

Solution should add logic to create a new array with only name, color, and region (subset of the original data)
Requires at least 2 screenshots (code + output from heroku dev)
Live URL must be Herokue Prod

#1) Show the output from heroku dev (url must be visible)



Caption (required) ✓

#2) Show the code solution (ucid/date as comment must be present)

```
// Note: use the $birds variable to iterate over, don't directly touch $a1-$a4
// TODO add logic here to create a new array with only name, color, and region
$subset = []; // result array
// Start edits
// UCID: ksss
// DATE: 05/24/2024
foreach ($birds as $bird) {
    $subset[] = [
        "name" => $bird["name"],
        "color" => $bird["color"],
        "region" => $bird["region"]
    ];
}
// End edits
echo "<pre>" . var_export($subset, true) . "</pre>";
```

Caption (required) ✓

Describe/highlight what's being shown
Showing the output from heroku dev, and I have the heroku dev url underlined in blue

URL (required) ✓
URL must be Heroku prod
https://vs53-it202-452-[redacted]-prod-1c77a1c304b2.herokuapp.com/[redacted]/problem1.php

Describe/highlight what's being shown
Showing my code solution for problem1, and my UCID and Date being underlined in blue

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
I looped through each "bird" in the "$birds" array. And then using the new array called "$subset", I included only the "name", "color", and "region" properties. Then this new array is printed to show only those details.

Problem 2 (3 pts.)

COLLAPSE

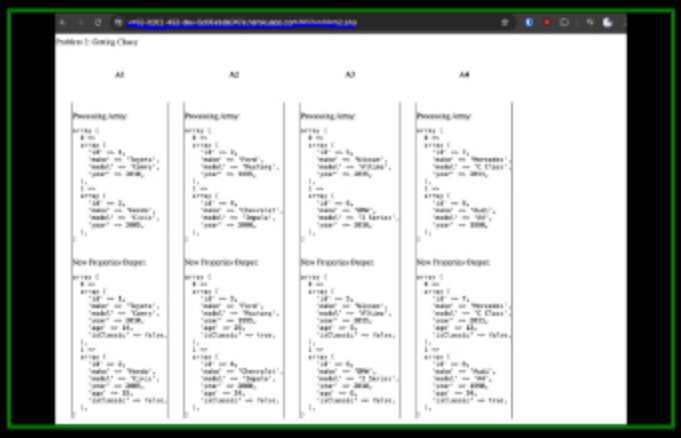
COLLAPSE

Task #1 - Points: 1
Text: Problem 2 Evidence

Details:
Only make edits where the template code mentions.

Solution should add logic to create a new array with original properties plus age and isClassic (extra data)
Requires at least 2 screenshots (code + output from heroku dev)
Live URL must be Herokue Prod

#1) Show the output from heroku dev (url must be visible)



Caption (required) ✓
Describe/highlight what's being shown
Showing the output from heroku dev for problem 2, and I have the heroku dev url underlined in blue

#2) Show the code solution (ucid/date as comment must be present)

Caption (required) ✓
Describe/highlight what's being shown
Showing my code solution for problem 2, and my UCID and Date being underlined in blue

URL (required) ✓
URL must be Heroku prod
https://vs53-it202-452-prod-1c77a1c304b2.herokuapp.com/M51/problem2.php

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
I looped through each car in the $cars array and created a new array called $processedCars. For each car I calculated its age and determined if it is a classic based on its age (if its greater than 25 its classic) Then I added these new properties (age and isClassic) to each car and included them in the new array.

Problem 3 (3 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Problem 3 Evidence

Details:
Only make edits where the template code mentions.
Solution should add logic to join the arrays on userId
Requires at least 2 screenshots (code + output from heroku dev)
Live URL must be Herokue Prod

#1) Show the output from heroku dev (url must be visible)

Caption (required) ✓
Describe/highlight what's being shown
Showing the output from heroku dev for problem 3, and I have the heroku dev url underlined in blue

URL (required) ✓

#2) Show the code solution (ucid/date as comment must be present)

Caption (required) ✓
Describe/highlight what's being shown
Showing my code solution for problem 3, and my UCID and Date being underlined in blue

Explanation (required) ✓

URL must be Heroku prod

https://vs53-it202-452-prod-1c77a1c304b2.herokuapp.com/problem3.php

Explain in concise steps how this logically works

PREVIEW RESPONSE

I looped through each user in the $users array and each activity in the $activities array. And for each user I checked if their userId matched with an activity's userId. And if they matched I combined the user and activity details into a new array called $joined.

Reflection (1 pt.)

COLLAPSE

Task #1 - Points: 1

Text: Reflect on your experience

Details:

Talk about any issues you had, how you resolved them, and anything you learned during this process.

Provide concrete details/examples. At least a few sentences.

Response:

I had some problems with the last problem, which was problem 3. Because I had trouble matching the users and activies by userId and I forgot how to solve something like this. But then I remembered and figured out that I just needed to use nested loops to check for userId equality and then I merged the arrays. This defintely helped me brush up on some programming concepts and get better with using PHP.

Task #2 - Points: 1

Text: Include the pull request link for this branch

Details:

The correct link will end with /pull/ and a number.

URL #1

https://github.com/VikShah/vs53-it202-452/pull/12

Task #3 - Points: 1

Text: Add Screenshot of Wakatime

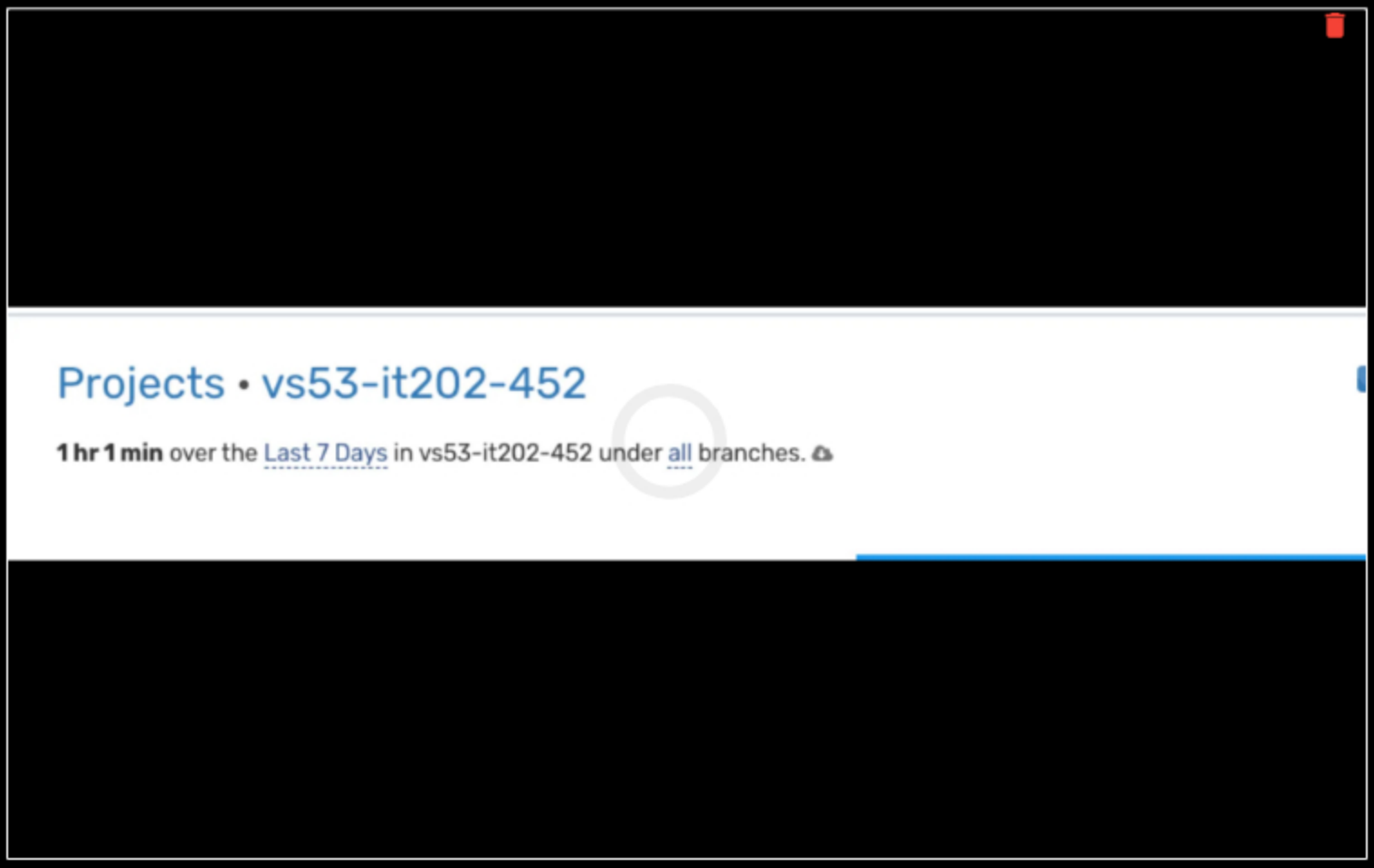
Details:

Note: The duration of time isn't directly related to the grade, the goal is to just make sure time is being tracked

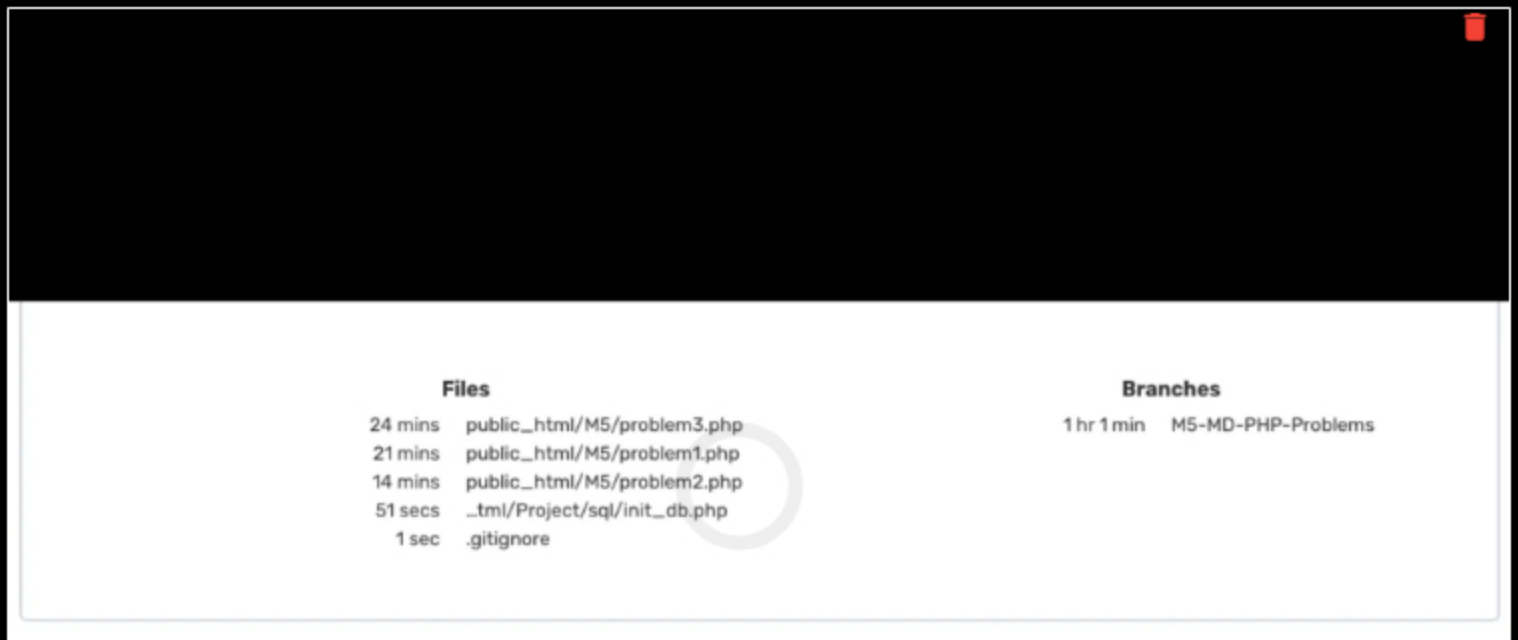
Task Screenshots:

Gallery Style: Large View

Small Medium Large



Showing the overall time that I spent in the project



Showing more specific detail for how long I spent in each problem

End of Assignment