Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT202-007-F2024/generic-module-5-multi-dimension-php-problems/grade/jd755

Course: IT202-007-F2024
Assigment: [Generic] Module 5 Multi-Dimension PHP Problems
Student: Jeremy R. (jd755)

Submissions:

Submission Selection

1 Submission [submitted] 10/18/2024 9:23:29 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

Group

75%

Group: Problem 1
Tasks: 1
Points: 3

COLLAPSE

Task

75%

Group: Problem 1
Task #1: Problem 1 Evidence
Weight: ~100%
Points: ~3.00

COLLAPSE

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)

Columns: 1

Sub-Task

50%

Group: Problem 1
Task #1: Problem 1 Evidence
Sub Task #1: Show the output from heroku dev (url must be visible)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Output in heroku dev

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task URLs

URL #1

Missing URL

URL

Sub-Task

100%

Group: Problem 1

Task #1: Problem 1 Evidence

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

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Code in VSC

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

I first declare a variable that holds what sub array is currently being worked on. This is named subArrNum. I then create a for loop that creates a subarray in subset and assigns to it the specific data needed.

End of Task 1

End of Group: Problem 1

Task Status: 0/1

Group

75%

Group: Problem 2

Tasks: 1

Points: 3

COLLAPSE

Task

75%

Group: Problem 2
Task #1: Problem 2 Evidence
Weight: ~100%
Points: ~3.00

COLLAPSE

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)

Columns: 1

Sub-Task

50%

Group: Problem 2
Task #1: Problem 2 Evidence
Sub Task #1: Show the output from heroku dev (url must be visible)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Problem 2 in heroku dev

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task URLs

URL #1

Missing URL

URL

Sub-Task

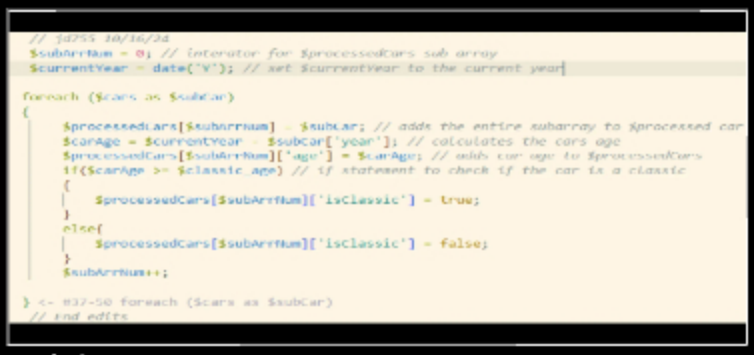
100%

Group: Problem 2
Task #1: Problem 2 Evidence
Sub Task #2: Show the code solution (ucid/date as comment must be present)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```
// jd755 10/16/24
$subArrNum = 0; // iterator for $processedCars sub array
$currentYear = date('Y'); // set $currentYear to the current year

foreach ($cars as $subCar)
{
    $processedCars[$subArrNum] = $subCar; // adds the entire subarray to $processed car
    $carAge = $currentYear - $subCar['year']; // calculates the cars age
    $processedCars[$subArrNum]['age'] = $carAge; // adds car age to $processedCars
    if($carAge >= $classic_age) // if statement to check if the car is a classic
    {
        $processedCars[$subArrNum]['isClassic'] = true;
    }
    else{
        $processedCars[$subArrNum]['isClassic'] = false;
    }
    $subArrNum++;

} <- 032-50 foreach ($cars as $subCar)
// end edits
```

Code in VSC

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

I first start by declaring the variable subArrNum with a similar purpose as problem 1. I also initalize currentYear to the current year by using the date() function with 'Y' passed in. This returns the current year. I then created a for loop that creates subarrays in the processedCars array. First, the entire sub array in cars is added to processedCars. I then calculate the cars age by subtracting the currentYear variable from the 'year' element in the sub array. I then add this age to the sub array. Lastly, I determine the car's status as a classic car by determining if the age is greater than or equal to 25.

End of Task 1

End of Group: Problem 2
Task Status: 0/1

Group

75%

Group: Problem 3
Tasks: 1
Points: 3

COLLAPSE

Task

75%

Group: Problem 3
Task #1: Problem 3 Evidence
Weight: ~100%
Points: ~3.00

COLLAPSE

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

Columns: 1

Sub-Task

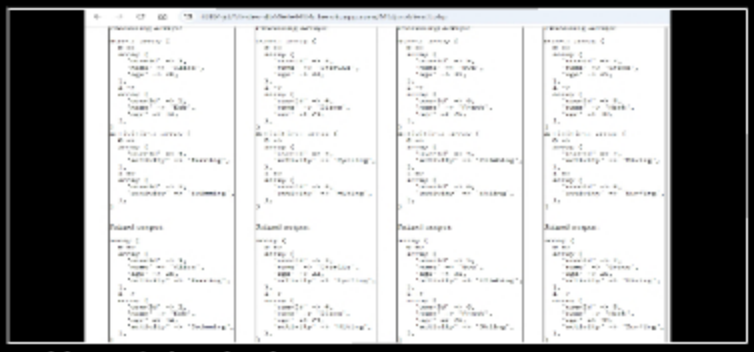
50%

Group: Problem 3
Task #1: Problem 3 Evidence
Sub Task #1: Show the output from heroku dev (url must be visible)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



problem 3 in heroku dev

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task URLs

URL #1

Missing URL

URL

Sub-Task

100%

Group: Problem 3
Task #1: Problem 3 Evidence
Sub Task #2: Show the code solution (ucid/date as comment must be present)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Code in VSC

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

I start by declaring the subArrNum variable that shares the same purpose as the previous problems. This time, I use a double for each loop: one for the user array and the other for the activity array. Inside the activity array, I check if their ID element is the same. If it is, I use two seperate for each loops to combine the contents of the users and activities arrays into the joined array variable. After that is complete, I break out of the loop as there is no more need to check the other activity array for a matching ID.

End of Task 1

End of Group: Problem 3
Task Status: 0/1

Group

100%

Group: Reflection
Tasks: 3
Points: 1

COLLAPSE

Task

100%

Group: Reflection
Task #1: Reflect on your experience
Weight: ~33%
Points: ~0.33

COLLAPSE

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.

Task Response Prompt

Response:

I found problem 3 to be the hardest as I had never had to join two arrays before. It took me a bit to come up with my solution but I get a feeling that there is a better way to solve it. I also originally hard coded the currentYear variable in problem 2. However, this would mean that the solution would turn incorrect after the new year, so I had to research if there was a function that returned the current year.

End of Task 1

Task

100%

Group: Reflection
Task #2: Include the pull request link for this branch
Weight: ~33%
Points: ~0.33

COLLAPSE

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

Task URLs

URL #1
https://github.com/legitdaylight/jd755-IT202-007/pull/23

URL
https://github.com/legitdaylight/jd755-IT202-007/

End of Task 2

Task

100%

Group: Reflection
Task #3: Add Screenshot of Wakatime
Weight: ~33%
Points: ~0.33

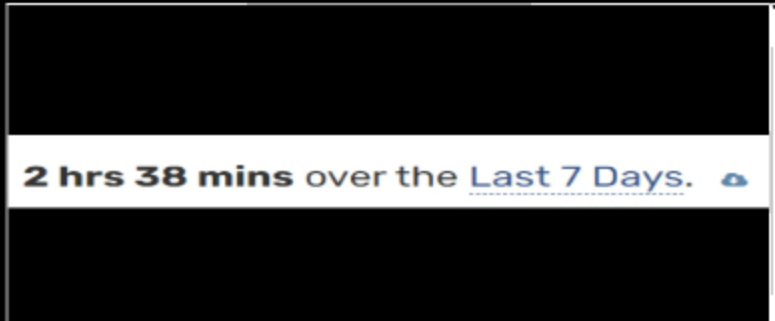
COLLAPSE

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: 2 Columns

4 2 1



Wakatime screenshot 1

Wakatime screenshot 2

End of Task 3

End of Group: Reflection
Task Status: 3/3

End of Assignment