Submission Worksheet

IT202-450-M2024 - [Generic] Module 5 Multi-Dimension PHP Problems

Submissions:

Submission Selection

1 Submission [active] 6/23/2024 5:39:37 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) 👁

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


```php
// Start edits

foreach ($birds as $bird){
    $subset[] = [
        'name' => $bird['name'],
        'color' => $bird['color'],
        'region' => $bird['region']
    ];
}

// End edits
echo "<pre>" . var_export($subset, true) . "</pre>";
You, 13 hours ago • adding template files ...
```

Caption (required) ✓   **Caption (required)** ✓

Describe/highlight what's being shown
showing output from heroku dev for problem 1

URL (required) ✓
URL must be Heroku prod
https://it202-jed56-
prod-553fb5e41cd0.herokuapp.com/m5/problem1.php

Describe/highlight what's being shown
showing the code solution for problem 1

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

📄 **PREVIEW RESPONSE**

The foreach loop iterates through each item in the given arrays. For each item in the original array given, the subset array is passed in the values for name, color, and region, and are assigned the birds name color and region. I simply am taking these attributes from the original array and passing them into the subset array without including the other attributes.

● **Problem 2 (3 pts.)**
^COLLAPSE ^

●
^COLLAPSE ^

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

ℹ️ 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) 👁️

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


```php
// Start edits

foreach ($cars as $car){
    $age = $currentYear - $car['year'];
    $isClassic = $age >= $classic_age;
    $processedCars[] = array_merge($car, ['age' => $age, 'isClassic' => $isClassic]);
}

// End edits
echo "<pre>" . var_export($processedCars, true) . "</pre>";
```

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

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

showing the output from heroku dev for problem 2

URL (required) ✓

URL must be Heroku prod

https://it202-jed56-
prod-553fb5e41cd0.herokuapp.com/problem2.php

showing the code solution for problem 2

Explanation (required) ✓

Explain in concise steps how this logically works

📄 PREVIEW RESPONSE

The foreach loop iterates over each car in the given arrays. For each car, an age variable is created and the assigned value is the current year minus the cars year. Then an isClassic boolean variable is created and assigned the value True or False depending if the age is greater than or equal to the classic_age value. Finally, I take the processed Cars array and merge it with the orignal array to add the two new attributes.

● **Problem 3 (3 pts.)**

⌃COLLAPSE ⌃

● **Task #1 - Points: 1**

⌃COLLAPSE ⌃

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) 👁

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


```php
// Start edits

foreach($users as $user){
    foreach ($activities as $activity){
        if ($user['userId'] == $activity['userId']){
            $joined[] = array_merge($user, $activity);
            break;
        }       You, 13 hours ago · adding completed problems …
    }
}

// End edits
echo "<pre>" . var_export($joined, true) . "</pre>";
```

Caption (required) ✓

Describe/highlight what's being shown

showing output from heroku dev for problem 3

Caption (required) ✓

Describe/highlight what's being shown

showing code solution for problem 3

URL (required) ✓
URL must be Heroku prod
https://it202-jed56-
prod-553fb5e41cd0.herokuapp.com/problem3.php

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

📄 PREVIEW RESPONSE

The foreach loops iterates through each user in the given array of users, and for each of the users it iterates through the activities in the given arrays. It then checks if the userId's match in both user and activity, and if so then they are merged and joined into the joined array. It then breaks out of the inner loop and goes on to the next user.

● **Reflection (1 pt.)**
∧COLLAPSE∧

●
∧COLLAPSE∧ **Task #1 - Points: 1**

Text: **Reflect on your experience**

ℹ **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:

This assignment was pretty fun. I didn't really have any problems writing the solutions besides having to look up some simple syntax for merging arrays in PHP. I ran into an issue with heroku where I could not login, but that was due to my internet connection. I was able to get it connected after a while and then I was able to see the app on my browser.

●
∧COLLAPSE∧ **Task #2 - Points: 1**

Text: **Include the pull request link for this branch**

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

URL #1
https://github.com/jordand2003/jed56-it202-450/pull/28

● Task #3 Points: 1

^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: Large View

Small   Medium   Large

Files

23 mins   M5/problem3.php

18 mins   M5/problem2.php

18 mins   M5/problem1.php

Showing WakaTime duration for each problem.

End of Assignment