Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT202-452-M2024/generic-module-5-multi-dimension-php-problems/grade/sha38

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

Submissions:

Submission Selection

1 Submission [active] 6/22/2024 3:16:41 PM

Instructions

^ COLLAPSE ^

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

Guide:

- 1. Make sure you're in the dev branch locally and $_{
 m git\ pull\ origin\ dev}$ any pending changes.
- 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 <code>public_html</code> folder in a subfolder of your choice.
- 5. Move the unedited template files to GitHub.
 - 1. git add .
 - git commit -m "adding template files"
 - 3. git push origin branch name (see below)
 - Create and open a pull request from the homework branch to main (leave it open until later steps).
- Note: As you work, it's recommended to add/commit at least after each solution is done (i.e., 3+ times in this case).
 - Make sure the files are saved before doing this.
- Fill in the items in the worksheet below (save as often as necessary).
- Once finished, export the worksheet.
- 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.
- If everything is good, continue to submit.
 - 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).
- Create a pull request from the homework related branch to main (i.e., dev <- "homework branch").
- 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 ^

Task #1 - Points: 1

Text: Problem 1 Evidence

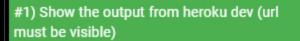
Details:

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





Processor of the control of the cont

The control of the co

Caption (required) 🗸

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



 \odot

Caption (required) 🗸

Shows output from heroku dev

URL (required) ✓ URL must be Heroku prod

https://sha38-prod-

ae798a1dca98.herokuapp.com/M5/problem1.php

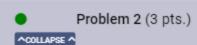
Describe/highlight what's being shown shows code solution

Explanation (required) ~

Explain in concise steps how this logically works



Firstly, we iterate through each element, '\$bird' in the \$birds array, which is what the foreach function does. We know what the birds array is an associative array. We append to the \$subset array using \$subset[] = and then we assign the values in the array. Since we want the name, color, and region, we make a new associative array with the keys name, color, and region, ex:"name" => \$bird["name"],... After all the elements in the birds array have been irerated through, the subset array will contain the extracted data for each bird.





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)



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



Caption (required) <

Describe/highlight what's being shown shows the output from heroku dev

URL (required) <

URL must be Heroku prod

https://sha38-prod-

ae798a1dca98.herokuapp.com/M5/problem2.php

Caption (required) <

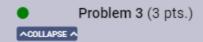
Describe/highlight what's being shown shows code solution

Explanation (required) <

Explain in concise steps how this logically works



Firstly, we retireve the current year using PHP 'data()' function which sets \$currentYear to the current year. We iterate over each car in the cars array using the foreach function and then calulcate the age by subtracting the cars manufacturing year from the current (year (\$age = \$currentYear - \$car['year'];). Then we check to see if the \$age of the car exceeds classic age(25), if it does, isClassic becomes true, otherwise it is false. We then build a new associative array named Processed Cars, and construct the array to contain "id", "make", "model", and "year", "age", and "isClassic". This function when echoed out, displays the processedCars which contains original car data along with "age" and "isClassic".





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)



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









Caption (required) 🗸

Describe/highlight what's being shown shows output from heroku dev

URL (required) <

URL must be Heroku prod

https://sha38-prod-

ae798a1dca98.herokuapp.com/M5/problem3.php



Caption (required) <

Describe/highlight what's being shown shows the code solution

Explanation (required) ~

Explain in concise steps how this logically works



We have a empty array \$joined where the joined user info will go. First, we iterate trhough each users in the \$user array, additionally we have another foreach loop inside which iterates over each activity in the \$activities array. Then we check to see if userId of the current user matches the userId of the current activity in line 57. When we find a match, we have a new associative array "\$joined[]" which contain the keys "userId", "name", "age", and "activity," these properties are copied from current user and current activity. We also have a break statement which exits the inner foreach loop once a match is found. We then return the results after all users have been checked.





Task #1 - Points: 1

Text: Reflect on your experience

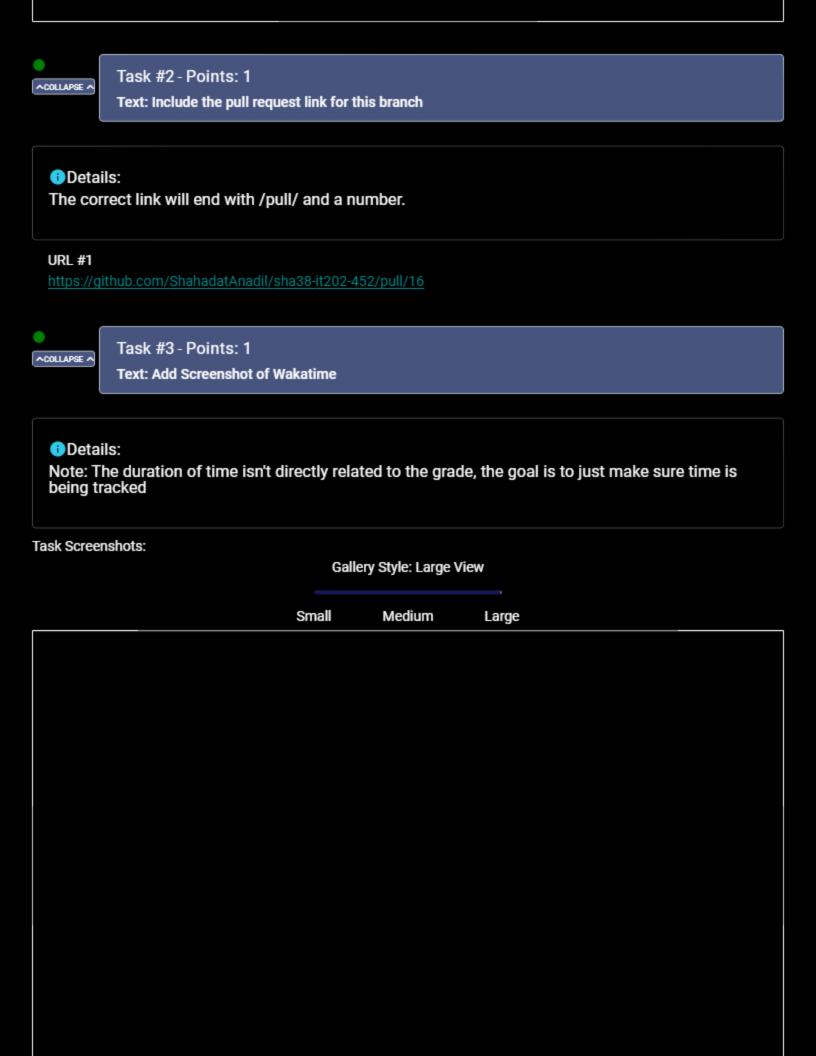
Details:

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:

One issue I had was with problem 2, specifically the currentYear. It would return all my cars as false for isClassic which was wrong. I realized that it was returning false because the currentYear was set to null. I then set the currentYear to the current year 2024 using \$currentYear= date("Y"); which helped resolve the problem. One thing I learned is how useful associative arrays are, they make data retrieval faster due to elements being accessed by their key, also really great for mapping relationships between data elements.



shows overall overview Files **Branches** 42 mins public_html/M5/problem2.php 1 hr 50 mins M5-MD-PHP-Problems 59 mins ProjectSetup 37 mins public_html/M5/problem3.php 22 mins Unknown 29 mins public_html/M5/problem1.php 10 mins M3-HTML-JS-CSS-Readings 21 mins .../001_create_table_users.sql 8 mins M4-SQL-Readings 20 mins ...tml/Project/sql/init_db.php 5 mins M3-Challenge-HW 18 mins ...06@sha38/alter-template.sql 5 mins main 10 mins .gitignore 11 secs dev 10 mins ...eadings_IT202-452-M2024.pdf 8 mins ...nd-quiz_IT202-452-M2024.pdf 6 mins ...ckpoint_IT202-452-M2024.pdf 5 mins public_html/M2/problem1.php 5 mins .git/MERGE_MSG 3 mins Untitled-1

shows specific for each files

End of Assignment