Submission Worksheet

Submission Data

Course: IT202-450-M2025
Assignment: IT202 PHP Multi-Dimension Problems
Student: Mark Y. (may23)
Status: Submitted | **Worksheet Progress:** 100%
Potential Grade: 1000.00/1000.00 (100.00%)
Received Grade: 0.00/1000.00 (0.00%)
Started: 7/12/2025 11:57:21 AM
Updated: 7/12/2025 12:24:10 PM
Grading Link: https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-php-multi-dimension-problems/grading/may23
View Link: https://learn.ethereallab.app/assignment/v3/IT202-450-M2025/it202-php-multi-dimension-problems/view/may23

Instructions

- Overview Link: https://youtu.be/jd5giyXReaI

1. Ensure you read all instructions and objectives before starting.
2. Create a new branch from `dev` called `M6-Homework`
    1. `git checkout dev` (ensure proper starting branch)
    2. `git pull origin dev` (ensure history is up to date)
    3. `git checkout -b M6-Homework` (create and switch to branch)
3. Copy the template code from here: GitHub Repository - M6 Homework
    - It includes Problems 1-3 and `base.php`. Put all into an `M6` folder or similar inside your `public_html`
    - Immediately record to history
        - `git add public_html`
        - `git commit -m "adding M6 HW baseline files"`
        - `git push origin M6-Homework`
        - Create a Pull Request from `M6-Homework` to `dev` and keep it open
4. Fill out the below worksheet
    - Each Problem requires the following as you work
        - Ensure there's a comment with your UCID, date, and brief summary of how the problem was solved
        - Initial outline/plan of how you'll solve it via comments (add/commit after this stage)
        - Code solution (add/commit periodically as needed)
5. Once finished, click "Submit and Export"
6. Locally add the generated PDF to a folder of your choosing inside your repository folder and move it to Github
    1. `git add .`
    2. `git commit -m "adding PDF"`
    3. `git push origin M6-Homework`
    4. On Github merge the pull request from `M6-Homework` to `dev`
    5. On Github create a pull request from `dev` to `prod` and immediately merge. (This will trigger the `prod` deploy to make the heroku prod links work)

7. Upload the same PDF to Canvas
8. Sync Local
    1. `git checkout dev`
    2. `git pull origin dev`

Section #1: (250 pts.) Problem 1 - Subset

Progress: 100%

≔ Task #1 (250 pts.) - Edit the `processBirds` function to extract properties

Progress: 100%

Details:

- Only make edits where noted via provided comments
- Challenge: Extract the name, color, region of each bird into the $subset array
- Step 1: sketch out plan using comments (include ucid and date)
- Step 2: Add/commit your outline of comments (required for full credit)
- Step 3: Add code to solve the problem (add/commit as needed)

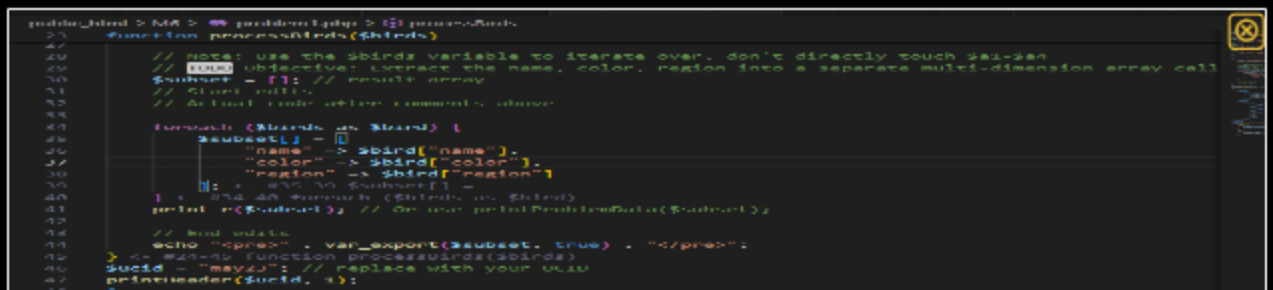
🖻 Part 1:

Progress: 100%

Details:
Two screenshots are expected

1. Snippet of relevant code showing solution (with ucid/date comment)
2. Full output of executing the program (visit the proper file on Heroku dev after a manual deploy)

sample output



relevant code

Saved: 7/12/2025 12:17:09 PM

Part 2:

Progress: 100%

Details:

- Direct link to the file in the homework related branch from Github (should end in `.php`)
- Direct link to the file on Heroku Prod (Just grab the base prod url and manually enter the path to the file)

URL #1
https://github.com/Mark-5555/may23-IT202-450/M6-Homework/public_html/M6/problem1.php

URL
https://github.com/Mark-5555/ma

URL #2
https://may23-it202-450-prod-70e1264ec6c5.herokuapp.com/M6/problem1.php

URL
https://may23-it202-450-prod-70e

Saved: 7/12/2025 12:17:09 PM

Part 3:

Progress: 100%

Details:
Briefly explain `how` the code solves the challenge (note: this isn't the same as `what` the code does)

Your Response:

The code solves the challenge by focusing only on the specific pieces of information we care about — the bird's name, color, and region — and ignoring the rest. It creates a new array called $subset and fills it with smaller arrays that each hold just those three details for every bird in the list. This makes the data cleaner and easier to work with for any future use, such as displaying it or sending it elsewhere. Instead of changing the original data, the code builds a new simplified version, which is exactly what the task asked for.

Saved: 7/12/2025 12:17:09 PM

Section #2: (250 pts.) Problem 2 - Adding Properties

Progress: 100%

☰ Task #1 (250 pts.) - Edit the `processCars` function to add properties

Progress: 100%

Details:

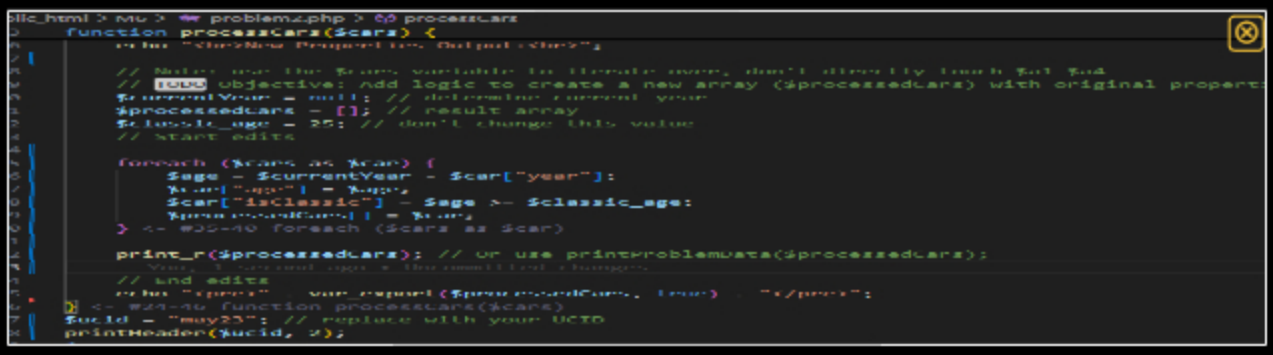
- Only make edits where noted via provided comments
- Challenge 1: Add a new property called age that's set from today's year and the car's year
- Challenge 2: Add a new property called isClassic that's true/false based on $classic_age
- Step 1: sketch out plan using comments (include ucid and date)
- Step 2: Add/commit your outline of comments (required for full credit)
- Step 3: Add code to solve the problem (add/commit as needed)

🖼 Part 1:

Progress: 100%

Details:
Two screenshots are expected

1. Snippet of relevant code showing solution (with ucid/date comment)
2. Full output of executing the program (visit the proper file on Heroku dev after a manual deploy)



P2 code

sample output

Saved: 7/12/2025 12:17:02 PM

Part 2:

Progress: 100%

Details:

- Direct link to the file in the homework related branch from Github (should end in `.php`)
- Direct link to the file on Heroku Prod (Just grab the base prod url and manually enter the path to the file)

URL #1
https://github.com/Mark-5555/may23-IT202-450/M6-Homework/public_html/M6/problem2.php

URL
https://github.com/Mark-5555/ma

URL #2
https://may23-it202-450-prod-70e1264ec6c5.herokuapp.com/M6/problem2.php

URL
https://may23-it202-450-prod-70e

Saved: 7/12/2025 12:17:02 PM

Part 3:

Progress: 100%

Details:
Briefly explain `how` the code solves the challenges (note: this isn't the same as `what` the code does)

Your Response:

The code solves the challenge by enhancing each car's information with how old the car is and whether it qualifies as a classic. It figures out the age by subtracting the car's model year from the current year. Then, it checks if that age is greater than or equal to a set number (like 25) to decide if it's a classic. Instead of changing the original data, it builds a new list of cars with the added properties, making the output cleaner and more useful for later tasks or display.

Saved: 7/12/2025 12:17:02 PM

Section #3: (250 pts.) Problem 3 - Join

Progress: 100%

Task #1 (250 pts.) - Edit the `joinArrays` function to combine two arrays based on a common key

Progress: 100%

Details:

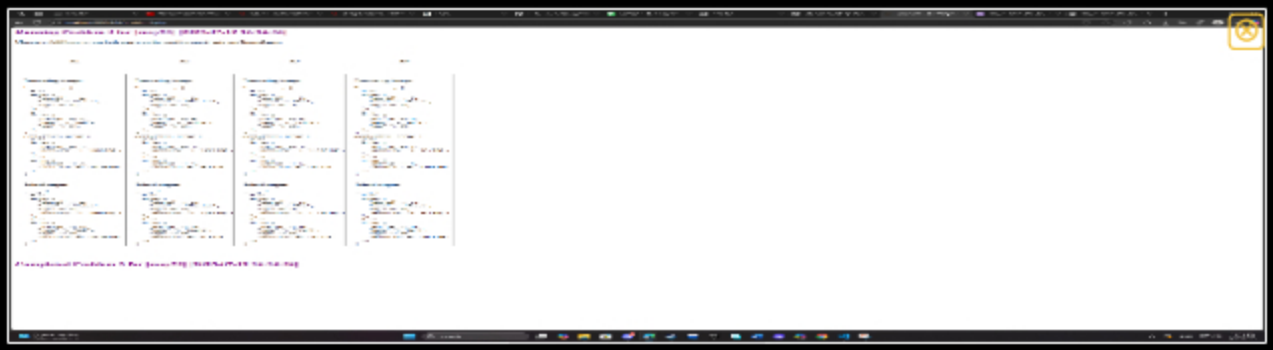
- Only make edits where noted via provided comments
- Challenge: Combine the data in both arrays by the userId property
- Step 1: sketch out plan using comments (include ucid and date)
- Step 2: Add/commit your outline of comments (required for full credit)
- Step 3: Add code to solve the problem (add/commit as needed)

Part 1:

Progress: 100%

Details:
Two screenshots are expected

1. Snippet of relevant code showing solution (with ucid/date comment)
2. Full output of executing the program (visit the proper file on Heroku dev after a manual deploy)



sample output

```
// Note: use the $users and $activities variables to iterate over, don't directly touch $a
// TODO Objective: Add logic to join both arrays on the userId property into one $joined a
$joined = []; // result array
// start edits
foreach ($users as $user) {
    foreach ($activities as $activity) {
        if ($user["userId"] === $activity["userId"]) {
            $joined[] = array_merge($user, $activity);
            break; // stop once a match is found
        }
    } <- #55 AM foreach ($activities as $activity)
} <- #54-61 foreach ($users as $user)        You, 1 second ago • Uncommitted changes

// End edits
echo "<pre>" . var_export($joined, true) . "</pre>";
} <- #48-66 function joinArrays($users, $activities)

$ucid = "may23"; // replace with your UCID
printHeader($ucid, 4);
?>
<table>
```

P3 code

Saved: 7/12/2025 12:16:59 PM

Part 2:

Progress: 100%

Details:

- Direct link to the file in the homework related branch from Github (should end in `.php`)
- Direct link to the file on Heroku Prod (Just grab the base prod url and manually enter the path to the file)

URL #1
https://github.com/Mark-5555/may23-IT202-450-M6-Homework/public_html/M6/problem3.php

URL
https://github.com/Mark-5555/ma

URL #2
https://may23-it202-450-prod-70e1264ec6c5.herokuapp.com/M6/problem3.php

URL
https://may23-it202-450-prod-70e

Saved: 7/12/2025 12:16:59 PM

Part 3:

Progress: 100%

Details:
Briefly explain `how` the code solves the challenges (note: this isn't the same as `what` the code does)

Your Response:

The code solves the challenge by combining two separate sets of information — one about users and one about their activities — into a single list. It looks at each user and finds their matching activity by comparing their userId. When it finds a match, it merges the two pieces of data together. This creates a new, organized list where each entry shows the user's details along with what they do, making the information easier to understand and use.

Saved: 7/12/2025 12:16:59 PM

Section #4: (250 pts.) Misc

Progress: 100%

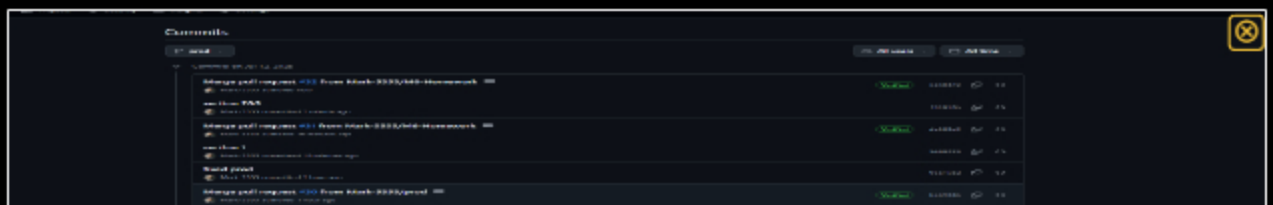
Task #1 (83.33 pts.) - Github Details

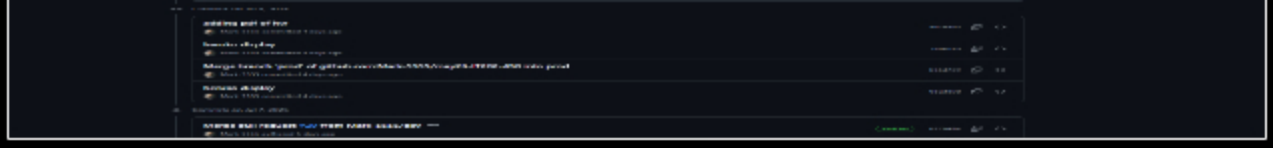
Progress: 100%

Part 1:

Progress: 100%

Details:
From the Commits tab of the Pull Request screenshot the commit history Following minimum should be present





commits history

Saved: 7/12/2025 12:21:07 PM

Part 2:

Progress: 100%

Details:
Include the link to the Pull Request (should end in `/pull/#`)

URL #1
https://github.com/Mark-5555/may23-IT202-450/pull/2

URL
https://github.com/Mark-5555/ma

Saved: 7/12/2025 12:21:07 PM

Task #2 (83.33 pts.) - WakaTime - Activity

Progress: 100%

Details:

- Visit the WakaTime.com Dashboard
- Click `Projects` and find your repository
- Capture the overall time at the top that includes the repository name
- Capture the individual time at the bottom that includes the file time
- Note: The duration isn't relevant for the grade and the visual graphs aren't necessary

wakatime-top

wakatime-bottom

Saved: 7/12/2025 12:22:22 PM

Task #3 (83.33 pts.) - Reflection

Progress: 100%

Task #1 (0.33 pts.) - What did you learn?

Progress: 100%

Details:
Briefly answer the question (at least a few decent sentences)

Your Response:

I learned how to handle and organize data in practical ways. I practiced extracting only the needed information from a larger set, adding new calculated properties to existing data, and combining two related data sets using a shared key. These tasks taught me how to manipulate arrays, apply logic to enhance data, and join information efficiently — all important skills for backend development and working with real-world data.

Saved: 7/12/2025 12:23:27 PM

Task #2 (0.33 pts.) - What was the easiest part of the assignment?

Progress: 100%

Details:
Briefly answer the question (at least a few decent sentences)

Your Response:

The easiest part was Section 1, where I just had to extract specific fields (name, color, and region) from each bird and organize them into a new array. It was straightforward and mostly involved simple looping and array access.

Saved: 7/12/2025 12:23:59 PM

Task #3 (0.33 pts.) - What was the hardest part of the assignment?

Progress: 100%

Details:
Briefly answer the question (at least a few decent sentences)

Your Response:

The hardest part was Section 3, because it required matching and merging data from two separate arrays based on a common key (userId). It involved nested loops and careful thinking to avoid mismatches or missing data, which made it more challenging and required more attention to detail.

Saved: 7/12/2025 12:24:10 PM