

Week 2 Exercise – PHP

Weighting: 2%

In this exercise we are going to create simple PHP applications.

Task 1 – Social Media Application – Array Implementation

Implement a web application in PHP which displays users's posts. A post has a user (or username), user's image, message and a date.

The implementation of this task involves:

1. Define an associative array which contains posts. For now, simply hard code at least 3 posts into the array. (Later on, in this course we'll retrieve the data from a database.)
2. Iterate through the array to place each post in an html document for display.

Note: Appropriate PHP variable names should be used.

Hint: Follow the lecture video on social media array implementation.

Task 2 – Social Media Application – Object-oriented Implementation.

Re-implement Task 1 using object-oriented PHP.

Instead of storing a post as an associative array, a post should now be stored as a post object.

Furthermore, in addition to storing and displaying posts, also store and display user comments. A post can have 0 to many user comments. A comment has a user and the comment/message. For this task, store comments as an associative array similar to what you did for posts in task 1.

Hint: refer to the Social Media example in Object-Oriented section of Lecture 2 video for partial implementation. Note: the implementation shown in the lecture video does not contain image and date. So you will need to add them.

Your implementation should also adhere to the following:

- A PHP file should contain only one class.
- Class files should be placed in a subdirector called *classes*.
- Classes need to place in a namespace called *wad*.
- Each class file should have the appropriate (code) comments. As a minimum:

- There should be a comment describing the class.
- There should be a comment for each function in the class.

Task 3 – Implement Comments by using Classes and Objects

This task tests the understanding you have gained from the previous task.

Re-implement Task 2 by converting Comments from associative array implementation to object implementation. You will need to:

1. Create a class called Comment in the file called comment.php. A comment contains a user (name) and a comment. Implement a constructor, and functions to retrieve comment data.
2. Update post.php to “use” the comment class, and include/require comment.php. Update the addComment() function in the Post class so that a comment object is created and added to the \$comments array.
3. Add some comments to some posts. This can be done in index.php after Posts have been seeded.
4. Update the display code to retrieve comment for display.

Note: the solution for this task is NOT demonstrated in the lecture video. You need to work this out! Apply what you have learnt from Task 2.

Task 4 – Type declarations

Reimplement your task 2 or task 3 by declaring types for function arguments, return values, and class properties. See lecture material on PHP Updates.

Explain to your peer reviewer the changes you have made and also the advantage of having type declaration.