SparkINN Novice Solution Pvt. Ltd

PHP Weekend Task: Building a Simple Blog Application

Objective: In this weekend task, you will apply the concepts learned in the PHP course to build a simple blog application. This task will help you consolidate your understanding of PHP syntax, data types, functions, and array handling. You will create a dynamic blog where users can add, view, edit, and delete posts. This task is designed to cover all the essential PHP topics learned so far.

Task Overview:

You will be required to:

1. **Set Up the Environment:**
   - Install PHP and set up a local web server (e.g., XAMPP, WAMP, or MAMP).
   - Ensure that PHP is correctly installed and configured.
2. **Create the Blog Application Structure:**
   - Create a new directory named `simple_blog` in your web server's root directory.
   - Inside the `simple_blog` directory, create the following folders:
     - `includes/`: For all reusable PHP files.
     - `posts/`: For storing post files.
     - `css/`: For stylesheets.
   - Create an `index.php` file as the entry point of your application.
3. **Implementing the Blog Posts:**
   - Create a PHP script to handle adding new blog posts.
   - Each blog post should include a title, content, and a publication date.
   - Use the PHP `DateTime` class to generate the publication date automatically.
   - Store the blog posts as associative arrays, where the title serves as the key and the content, date, and a unique ID as the values.
4. **Displaying Blog Posts:**
   - Use PHP to loop through the blog posts and display them on the `index.php` page.
   - Display each post’s title, content, and publication date.
   - Ensure that the posts are displayed in reverse chronological order (newest first).
5. **Editing and Deleting Posts:**
   - Implement functionality to edit existing blog posts.
   - Create a PHP script that allows users to update the content of a post.
   - Add a delete option that allows users to remove a blog post from the list.
6. **Working with PHP Data Types:**
   - Use different PHP data types for storing and manipulating data (e.g., strings for titles, arrays for storing posts).

    - Implement typecasting where necessary (e.g., converting input values to the correct data type).
    - Ensure that your application handles all data types correctly and performs type validation where appropriate.
7. **Using Functions in PHP:**
    - Create a set of reusable functions for tasks like adding a new post, deleting a post, and updating a post.
    - Store these functions in an `includes/functions.php` file.
    - Use functions to keep your code DRY (Don't Repeat Yourself).
8. **Handling Arrays in PHP:**
    - Use indexed and associative arrays to store blog posts.
    - Implement multi-dimensional arrays if needed, such as when storing posts with multiple attributes (title, content, date).
    - Use PHP array functions to manipulate the blog posts, such as sorting or filtering posts.
9. **File Inclusion:**
    - Use PHP's `include` and `require` statements to modularize your code.
    - Include the `functions.php` file in your `index.php` to access the functions you created.
    - Ensure that all included files are correctly linked and loaded.
10. **Styling the Blog:**
    - Create a simple CSS file in the `css/` directory.
    - Style your blog using CSS, ensuring that it is user-friendly and visually appealing.
    - Use basic CSS to structure the layout and format the text.

Submission:

- Zip your `simple_blog` folder and submit it.
- Ensure that all functionalities are working as expected.
- Include a README file with instructions on how to run your application.

This task will help you practice and consolidate your understanding of PHP. It combines the knowledge gained from the course into a single, practical project, allowing you to demonstrate your ability to build a functional web application using PHP.