

Lab 6 – Introduction to PHP

Setting Up Your Environment

1. Ensure you have a web server environment (like XAMPP or WampServer) set up and running.
2. Create a new folder (i.e. **lab_6**) in your server's root directory for this exercise.

Exercise 1

1. Inside your existing folder (**lab_6**), create a new PHP file (i.e. **contact_info.php**) and type the following PHP code block at the beginning of the file:

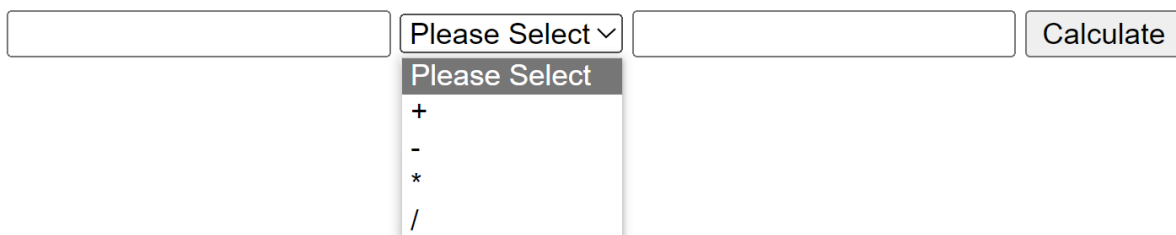
```
$title = "Contact Information";  
$name = "John Doe";  
$email = "john@apu.edu.my";  
$address[0] = "Lot 6 Technology Park Malaysia";  
$address[1] = "Bukit Jalil";  
$address[2] = "Kuala Lumpur";  
$postcode = "57000";  
$telephone = "0389961000";
```

2. Add HTML to your document and create a full contact page using PHP to output the value of the variables created above. Make use of the **heading**, **list** and **any other HTML element** you feel is appropriate.

Exercise 2

Create a basic calculator using PHP. Users should be able to enter two numbers and select an operation to perform. The web page should then display the result of the calculation.

Calculator



The image shows a web form for a calculator. It consists of two text input fields for numbers, a dropdown menu for selecting an operation, and a 'Calculate' button. The dropdown menu is currently open, displaying the following options: '+', '-', '*', and '/'. The text 'Please Select' is visible both above the dropdown and inside the menu header.

The step-by-step instructions for this exercise:

1. Inside your existing folder (**lab_6**), create a new PHP file named **calculator.php**.
2. In **calculator.php**, create an HTML form that includes:
 - Two input fields for entering numbers (use **<input type="number">**).
 - A dropdown select box to choose the operation (addition, subtraction, multiplication, division).
 - A submit button to perform the calculation.

3. In your PHP code, check if the form has been submitted (use `if ($_SERVER["REQUEST_METHOD"] == "POST") { ... }`).
4. Retrieve the values of the two numbers and the selected operation from the form submission.
5. Implement the calculations for addition, subtraction, multiplication, and division based on the selected operation.
6. Ensure to handle division by zero errors and display an error message if necessary.
7. Test your calculator application by entering different numbers and selecting different operations.
8. Ensure that the application calculates and displays the result accurately.

Exercise 3

This exercise involves working with PHP, HTML, and CSS.

Similar to the previous JavaScript exercise, create a webpage that automatically changes its background image based on the time of day. This time, use PHP to automate the background changes.

1. Inside your exercise folder, create a new PHP file named `background.php`.
2. In `background.php`, use PHP to get the current time from the user's computer using the `date()` function. Use `date_default_timezone_set()` function to specify the timezone.
3. Based on the current time, determine whether it's morning, afternoon, evening or night.
4. In your HTML code within `background.php`, set the background image of the webpage based on the time of day determined in step 3.
5. You can use inline CSS or add a CSS class to the body element with a background image property.
6. Test your webpage at different times of the day to ensure that it displays the appropriate background image.
7. You may want to add some additional text or content to the page to make it more interesting.

Note:

- To change the default time zone, use `date_default_timezone_set()`
<https://www.php.net/manual/en/function.date-default-timezone-set.php>
<https://www.php.net/manual/en/timezones.php>
- To retrieve date / time based on the time zone, use `date()`
<https://www.php.net/manual/en/function.date.php>