

	CSE 482L: Internet and Web Technology Lab	Lab Manual
	Faculty: naqib.hussain@northsouth.edu Instructor: shabbir.ahmed01@northsouth.edu TA: afsana01@northsouth.edu Updated: 2023-Oct-25	6

Goal: Explore PHP

- Its usage, OOP
- Form handling
- Cookie handling
- Session handling

PHP, its usages, OOP concepts

Understand the fundamentals of PHP. Discuss in lab, ask the lab instructor when you have a question. Read from online learning materials such as W3schools ^a and Mozilla Developers Network (MDN) ^b

Make the following 2 PHP files

File name: index.php
<pre> <form action="process.php" method="POST"> <label for="name">Enter your name:</label><input type="text" id="name" />
 <label for="age">your age:</label><input type="text" id="age" />
 <input type="submit" /> <input type="reset" /> </form> </pre>

File name: process.php
<pre> Welcome <?php echo \$_POST["name"].".."; ?>
 You are <?php echo \$_POST["age"]; ?> years old! </pre>

Task:

1. Take an array of names using form. Copy the elements in reverse order into another array. Print both arrays.
2. Take a string message using form and split it into words. Print the words in both original and reverse order.

3. Take multiple integers separated by commas using form field and print frequency for each number. i.e.
1, 1, 2, 3, 1, 4, 2, 6
Output:
1 occurred 3 times
2 occurred 2 times and so on.
4. Write a PHP function `power($number, $degree)` that raises \$number to the power of \$degree using `for` loop.
Input: `power(2, 4)` Output: 16
Now take two integers using form and test the function.
5. Write a PHP function `validate password ($pass)` that tests for following criteria:
- Length must be within 8-32 characters
 - must contain at least one uppercase, one lowercase and one numeric character
 - cannot contain any special character

Test the function taking input from user.

PHP form handling and validation

Create a form similar to the one shown below. Apply what you learned on the earlier lab session.

PHP Form Validation Example

Name:

E-mail:

Mobile:

Website:

Comment:

Gender: ☐ Man ☐ Woman

When submitted, the form should be validated on server side PHP. Use PHP regular expression handler function. The validation rules are:

1. User name cannot not contain any empty space or special character
2. Password and Repeat Password matches
3. Email is valid
4. Mobile number contains numbers and '+' only
5. Zip code is a number

You may start from the following form

File name: **myform.php**

```
<!DOCTYPE HTML>
<html>
<head></head>
<body>
<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = $mobile = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $mobile = test_input($_POST["mobile"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
<h2>PHP Form Validation Example</h2>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
    Name: <input type="text" name="name"><br><br>
    E-mail: <input type="text" name="email"><br><br>
    Mobile: <input type="tel" name="mobile"><br><br>
    Website: <input type="text" name="website"><br><br>
    Comment: <textarea name="comment" rows="5" cols="40"></textarea>
    <br><br>
```

```

Gender:
<input type="radio" name="gender" id="man" value="man">
<label for="man">Man</label>
<input type="radio" name="gender" id="woman" value="woman">
<label for="woman">Woman</label>
<br><br>
<input type="submit" name="submit" value="Submit">
</form>

<?php
    echo "<h2>Your Inputs:</h2>";
    echo "Name: ".$name."<br>". "Email: ".$email."<br>". "Mobile: ".$mobile."<br>";
    echo "Website: ".$website."<br>". "Comment: ".$comment."<br>". "Gender: ".$gender;
?>
</body>
</html>

```

Check any required fields and if error then show it.

```

if (empty($_POST["name"])) {
    $nameErr = "Name is required";
} else {
    $name = test_input($_POST["name"]);
}

```

Validate inputs before using.

```

$email = test_input($_POST["email"]);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $emailErr = "Invalid email format";
}

```

PHP Cookies

Steps:

1. Request this page from a browser while it's running from your server.
2. Go to your browser and check that there is a cookie on the browser or not. If there none then refresh this page and check again.
3. Then delete this cookie and reload the page so that you can see the cookie again. If you set any value into your cookie, you can see that value of the cookie.

File name: **cookie.php**

```
<!DOCTYPE html>
<?php
    $cookie_name = "user";
    $cookie_value = "Alex Porter";
    setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
?>
<html>
<body>
<?php
    if(!isset($_COOKIE[$cookie_name])) {
        echo "Cookie named '" . $cookie_name . "' is not set!";
    } else {
        echo "Cookie '" . $cookie_name . "' is set!<br>";
        echo "Value is: " . $_COOKIE[$cookie_name];
    }
?>
    <p><strong>Note:</strong> reload page to see the new value of the cookie.</p>
</body>
</html>
```

Tasks:

1. Now set the cookie at different path on line number 5 and then do the steps 1-3.
 - a. Do you see cookie set?
 - b. Will this cookie get forwarded?
2. Play with expiration time of cookie. Set a short time. Let's say, now its 1:14 pm and set a time for 30 second (1:14:30 pm) and see what happen?

PHP Session Handling

File name: **my_session.php**

```
<?php session_start();?>
<!DOCTYPE html>
<html>
<body>
<?php
    // remove all session variables
    session_unset();
    // destroy the session
    session_destroy();
    echo "All session variables now removed, and the session is destroyed."
?>
</body>
</html>
```

Question: What do you see if you comment out `session_destroy()` and then check content of `.tmp` folder under XAMPP (ie `c:\xampp\tmp\`). Then uncomment it and check it again. What does this method do to sessions stored on the filesystem?

End Notes

^a PHP from W3Schools: <https://www.w3schools.com/php/>

^b PHP from MDN: <https://developer.mozilla.org/en-US/docs/Glossary/PHP>