

Experiment No. 3

Title: Form Handling and Validation using PHP

Batch: B2 Roll No: 16010420117 Experiment No.:3

Aim: Design a Form and show handling of form Inputs and validation using PHP

Resources needed: Windows OS, Web Browser, Editor, XAMPP Server

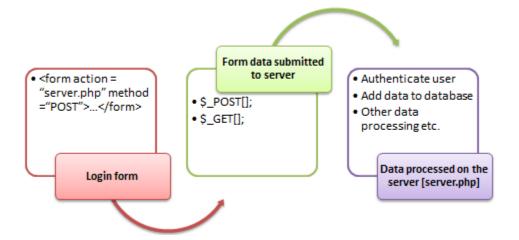
Pre Lab/ Prior Concepts:

Students should have prior knowledge of HTML/CSS/Basic Programming.

Theory:

When you login into a website or into your mail box, you are interacting with a form. Forms are used to get input from the user and submit it to the web server for processing.

The diagram below illustrates the form handling process.



A form is an HTML tag that contains graphical user interface items such as input box, check boxes radio buttons etc.

The form is defined using the <form>...</form> tags and GUI items are defined using form elements such as input.

- Forms come in handy when developing flexible and dynamic applications that accept user input.
- Forms can be used to edit already existing data from the database

Submitting the form data to the server

The action attribute of the form specifies the submission URL that processes the data. The method attribute specifies the submission type.

PHP POST method

- This is the built in PHP super global array variable that is used to get values submitted via HTTP POST method.
- The array variable can be accessed from any script in the program; it has a global scope.
- This method is ideal when you do not want to display the form post values in the URL.
- A good example of using post method is when submitting login details to the server.

It has the following syntax.

```
<?php

$_POST['variable_name'];
?>
```

HERE,

- "\$ POST[...]" is the PHP array
- "'variable_name'" is the URL variable name.

PHP GET method

- This is the built in PHP super global array variable that is used to get values submitted via HTTP GET method.
- The array variable can be accessed from any script in the program; it has a global scope.
- This method displays the form values in the URL.

• It's ideal for search engine forms as it allows the users to book mark theresults.

It has the following syntax.

```
<?php
$_GET['variable_name'];
?>
```

HERE,

- "\$_GET[...]" is the PHP array
- "'variable_name'" is the URL variable name.

The below diagram shows the difference between get and post

FORM SUBMISSION POST METHOD

```
<form action="registration form.php"
    First name: <input type="text" name="firstname
    Last name: <input type="text" name="lastname">
    <br>
    <input type="hidden" name="form submitted" value="1"/>
    <input type="submit" value="Submit">
</form>
           Submission URL does not show form values
                   localhost/tuttis/registration_form.php
              FORM SUBMISSION GET METHOD
       <form action="registration form.php" method="GB
           First name: <input type="text" name
           Last name: <input type="text" name="lastname">
           <input type="hidden" name="form submitted" value="1"/>
           <input type="submit" value="Submit">
       </form>
             SUBMISSION URL SHOWS FORM VALUES
```

localhost/tuttis/registration_form.php?firstname=Smith&lastname=Jones&form_submitted=;

Processing the registration form data

The registration form submits data to itself as specified in the action attribute of the form. When a form has been submitted, the values are populated in the \$_POST super global array.

We will use the PHP isset function to check if the form values have been filled in the \$_POST array and process the data. We will modify the registration form to include the PHP code that processes the data. Below is the modified code

```
<html>
<head>
        <title>Registration Form</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
   <?php if (isset($_POST['form_submitted'])): ?>
    //this code is executed when the form is submitted
        <h2>Thank You <?php echo $_POST['firstname']; ?> </h2>
       You have been registered as
           <?php echo $_POST['firstname'] . ' ' . $_POST['lastname']; ?>
        Go <a href="/registration form.php">back</a> to the form
        <?php else: ?>
            <h2>Registration Form</h2>
           <form action="registration_form.php" method="POST">
```

HERE,

<?php if (isset(\$_POST['form_submitted'])): ?> checks if the form_submitted hidden field has been filled in the \$_POST[] array and display a thank you and first name message.

If the form_fobmitted field hasn't been filled in the \$_POST[] array, the form is displayed.

Working with check boxes, radio buttons

If the user does not select a check box or radio button, no value is submitted, if the user selects a check box or radio button, the value one (1) or true is submitted.

We will modify the registration form code and include a check button that allows the user to agree to the terms of service.

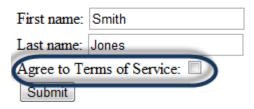
```
<?php if (!isset($_POST['agree'])): ?>
            You have not accepted our terms of service
            <?php else: ?>
                 <h2>Thank You <?php echo $_POST['firstname']; ?></h2>
                You have been registered as
                     <?php echo $_POST['firstname'] . ' ' . $_POST['lastname']; ?>
                 Go <a href="/registration_form2.php">back</a> to the form
            <?php endif; ?>
            <?php else: ?>
                         <h2>Registration Form</h2>
                         <form action="registration_form2.php" method="POST">
                             First name:
                             <input type="text" name="firstname">
                             <br/>br> Last name:
                              <input type="text" name="lastname">
                             <br/>
<br/>
dr> Agree to Terms of Service:
                              <input type="checkbox" name="agree">
                             <br
                              <input type="hidden" name="form_submitted" value="1" />
                              <input type="submit" value="Submit">
         >
                          <?php endif; ?>
</body
```

</html

-/1	۲.,		_ <
1</td <td>()</td> <td>m</td> <td>n></td>	()	m	n>

View the above form in a browser

Registration Form



Fill in the first and last names

Note the Agree to Terms of Service checkbox has not been selected.

Click on submit button

You will get the following results

Click on back to the form link and then select the checkbox

Registration Form

First name:	Smith	
Last name:	Jones	
Agree to Terms of Service: 🔽		
Submit		

Click on submit button

You will get the following results

Thank You Smith

You have been registered as Smith Jones

Go back to the form

Form Validation: Validation refers to checking the input submitted by the user. Using PHP,Server-Side validation of input is performed. In the Server-Side Validation, the input submitted by the user issent to the server and is validated on

the server. After validation is completed on server, server sends the feedback back to the client.

Few examples of performing validations such as required field, perform validation using regular expressions and validation of e-mail address are provided further

Example of required field validation for "name" field of form:

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

Displaying Validation Error message on form:

In this example, preg_match() function searches a string for pattern, returning true if the pattern exists, and false otherwise

Example of validating E-mail:

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

The easiest and safest way to check whether an email address is well-formed is to use PHP's filter_var() function.

Code:

```
<!DOCTYPE HTML>
<html>
<head>
    <style>
        .error {color: #FF0000;}
    </style>
</head>
<body>
    <?php
        $nameErr = $emailErr = $genderErr = $websiteErr = "";
        $mobNumberErr = "";
        $name = $email = $gender = $comment = $website = "";
        $mobnumber = "";
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
        if (empty($_POST["name"])) {
            $nameErr = "Please enter a valid name";
        } else {
            $name = test_input($_POST["name"]);
            if (!preg_match("/^[a-zA-Z-' ]*$/",$name)) {
            $nameErr = "Only letters and white space allowed";
        if (empty($_POST["email"])) {
```

```
$emailErr = "valid Email address";
        } else {
            $email = test_input($_POST["email"]);
            // check if e-mail address is well-formed
            if (!filter var($email, FILTER VALIDATE EMAIL)) {
            $emailErr = "The email address is incorrect";
        if (empty($ POST["mobnumber"])) {
            $mobNumberErr = "valid Mobile Number";
        } else {
            $email = test_input($_POST["mobnumber"]);
            if (preg_match("/^[a-zA-Z-' ]*$/",$name)) {
            $mobNumberErr = "Enter Only Neumeric values here";
        if (empty($_POST["website"])) {
            $website = "";
        } else {
            $website = test_input($_POST["website"]);
            if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-
9+\&@\#/\%?=\sim[!:,.;]*[-a-z0-9+\&@\#/\%=\sim[]/i",$website)) {
```

```
$websiteErr = "Enter a valid Webiste URL";
   if (empty($_POST["comment"])) {
       $comment = "";
   } else {
       $comment = test_input($_POST["comment"]);
   if (empty($_POST["gender"])) {
       $genderErr = "Please select a gender";
   } else {
       $gender = test_input($_POST["gender"]);
   function test_input($data) {
   $data = trim($data);
   $data = stripslashes($data);
   $data = htmlspecialchars($data);
   return $data;
<h2>PHP Form Validation Example</h2>
<span class="error">* required field</span>
```

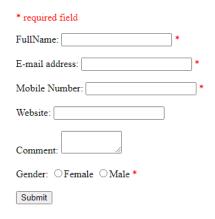
```
<form method="post" action="<?php echo</pre>
htmlspecialchars($ SERVER["PHP SELF"]);?>">
        FullName: <input type="text" name="name">
        <span class="error">* <?php echo $nameErr;?></span>
        <br><br><br>>
        E-mail address: <input type="text" name="email">
        <span class="error">* <?php echo $emailErr;?></span>
        <br><br><
        Mobile Number: <input type="text" name="mobnumber">
        <span class="error">* <?php echo $mobNumberErr;?></span>
        <br><br><br>>
        Website: <input type="text" name="website">
        <span class="error"><?php echo $websiteErr;?></span>
        <br><br><br><
        Comment: <textarea name="comment" rows="2" cols="10"></textarea>
        <br><br><br>>
        Gender:
        <input type="radio" name="gender" value="female">Female
        <input type="radio" name="gender" value="male">Male
            <span class="error">* <?php echo $genderErr;?></span>
        <br><br><br><
        <input type="submit" name="submit" value="Submit">
    </form>
    <?php
```

```
echo "<h2> Final Output:</h2>";
        echo $name;
        echo "<br>";
        echo $email;
        echo "<br>";
        echo $mobnumber;
        echo "<br>";
        echo $website;
        echo "<br>";
        echo $comment;
        echo "<br>";
        echo $gender;
</body>
</html>
```

Output(Code with result Snapshot)

1. Initial form:

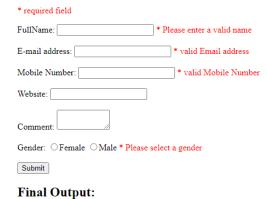
PHP Form Validation Example



Final Output:

2. Validation without any entries in the form:

PHP Form Validation Example



3. Entering wrong data in the input fields

PHP Form Validation Example

* required field
FullName: * Only letters and white space allowed
E-mail address: * The email address is incorrect
Mobile Number:
Website: Enter a valid Webiste URL
Comment:
Gender: ○Female ○Male *
Submit
Final Output:
1324234 asdf
asdf asdf female

4. Form with correct inputs

PHP Form Validation Example

* required field
FullName: *
E-mail address:
Mobile Number:
Website:
Comment:
Gender: OFemale OMale *
Submit
Final Output:
Soham Bhoir 12312312
https://soham-bhoir.vercel.app asdf male

Questions:-

What is the significance of \$_SERVER["PHP_SELF"] and htmlspecialchars() function?
 Ans:

• The \$_SERVER["PHP_SELF"] is a super global variable that returns the filename of the currently executing script.

So, the \$_SERVER["PHP_SELF"] sends the submitted form data to the page itself, instead of jumping to a different page. This way, the user will get error messages on the same page as the form.

htmlspecialchars replaces characters with special meaning in HTML with &-escaped entities. So, for example, 'becomes '. It doesn't turn %22 into ", however, because %22 has no special meaning in HTML, so it's safe to display it without modification.

If you want a form to be handled by the same URL that is used to display it, always use action="" rather than action=<?=\$ SERVER['PHP SELF']?> or action=<?=\$ SERVER['REQUEST URI']?>.

As you've already figured out, there are serious risks of cross-site scripting (XSS) if you use either of the \$_SERVER variables, because they contain user input and therefore cannot be trusted. So, unless you have a good reason that you need to tweak the URL somehow, just use action="".

2. Explain working of empty() and isset() with an example. Compare the working of both these functions

Ans:

The **empty()** function is an inbuilt function in PHP that is used to check whether a variable is empty or not.

These values are considered to be empty values:

- "" (an empty string)
- 0 (0 as an integer)
- 0.0 (0 as a float)
- "0" (0 as a string)
- NULL
- FALSE
- array() (an empty array)

<u>isset()</u> Function: The <u>isset()</u> function is an inbuilt function in PHP that is used to determine if the variable is declared and its value is not equal to NULL.

Parameters: This function accepts one or more parameters as mentioned above and described below.

- \$var: It contains the variable which needs to check.
- \$...: It contains the list of other variables.

Return Value: It returns TRUE if var exists and its value is not equal to NULL and FALSE otherwise.

Sr No.	isset() Function	empty() Function
1	The isset () function is an inbuilt function in PHP that is used to determine if the variable is declared and its value is not equal to NULL.	The empty () function is an inbuilt function in PHP that is used to check whether a variable is empty or not.
2	The isset () function will generate a warning or enotice when the variable does not exists.	The empty () function will not generate any warning or enotice when the variable does not exists.

3. Differentiatie between GET and POST method of Form input Submission to the server. Ans:

Difference between HTTP GET and HTTP POST

HTTP GET	HTTP POST	
In GET method we can not send large amount of data rather limited data is sent because the request parameter is appended into the URL.	In POST method large amount of data can be sent because the request parameter is appended into the body.	
GET request is comparatively better than Post so it is used more than the Post request.	POST request is comparatively less better than Get so it is used less than the Get request.	
GET request is comparatively less secure because the data is exposed in the URL bar.	POST request is comparatively more secure because the data is not exposed in the URL bar.	
Request made through GET method are stored in Browser history.	Request made through POST method is not stored in Browser history.	
GET method request can be saved as bookmark in browser.	POST method request can not be saved as bookmark in browser.	
Request made through GET method are stored in cache memory of Browser.	Request made through POST method are not stored in cache memory of Browser.	
Data passed through GET method can be easily stolen by attackers.	Data passed through POST method can not be easily stolen by attackers.	
In GET method only ASCII characters are allowed.	In POST method all types of data is allowed.	

Outcomes: CO2: Design forms and use session handling mechanism with web applications

Conclusion: We learnt and implemented the form validation in php successfully.

Grade: AA / AB / BB / BC / CC / CD/DD

Signature of faculty in-charge withdate

References:

Books:

- 1. Thomson PHP and MySQL Web Development Addison-Wesley Professional, 5th Edition 2016.
- 2. Peter MacIntyre, Kevin Tatroe Programming PHP O'Reilly Media, Inc, 4th Edition 2020
- **3.** Frank M. Kromann Beginning PHP and MySQL: From Novice to Professional, Apress 1st Edition, 2018
- **4.** https://www.w3schools.com/

