PHP Form Validation

Course Code: **CSC 3222** Course Title: WEB TECHNOLOGIES

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	**3**	**Week No:**	**3**	**Semester:**	
Lecturer:	*Supta Richard Philip & richard@aiub.edu*				

Lecture Outline

1. **Learning Objectives**
2. **PHP Form Handling**
3. **PHP Form Validation**
4. **Books and References**

Learning Objectives

- In this lecture , we will learn more details about HTML form elements i.e. different type of form, designing different type of HTML form.
- We will also learn HTTP GET and POST.
- Handling form data using $_GET or $_POST methods and validations form data using PHP.

PHP Form Handling

- The PHP superglobals **$_GET** and **$_POST** are used to collect form-data.
- The example displays a simple HTML form with two input fields and a submit button using “get” and “post”
- And See the difference.
- ```
  <html>
  <body>
  <form action="welcome.php" method="post">
  Name: <input type="text" name="name"><br>
  E-mail: <input type="text" name="ema
  <input type="submit">
  </form>
  </body>
  </html>
  ```


  ```

PHP Form Handling

HTTP GET

- The same result could also be achieved using the HTTP GET/POST method:
- Welcome_get.php file
- ```
  <html>
  <body>
  Welcome <?php echo $_POST["name"]; ?><br>
  Your email address is: <?php echo $_POST["email"]; ?>
  </body>
  </html>
  ```
  ```

METHOD: GET vs. POST

	GET	**POST**
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data

METHOD: GET vs. POST

History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

PHP Superglobals

Some predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

The PHP superglobal variables are:

- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET

PHP $GLOBALS

- $GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).
- PHP stores all global variables in an array called $GLOBALS[*index*]. The *index* holds the name of the variable.

```php
<?php
$x = 75;
$y = 25;

function addition() {
  $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

addition();
echo $z;
?>
```

PHP $_SERVER

- $_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

```php
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

PHP $_REQUEST

- PHP $_REQUEST is a PHP super global variable which is used to collect data after submitting an HTML form.

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_REQUEST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>

</body>
```

PHP $_POST

PHP $_POST is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_POST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>
```

PHP $_GET

- PHP $_GET is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".

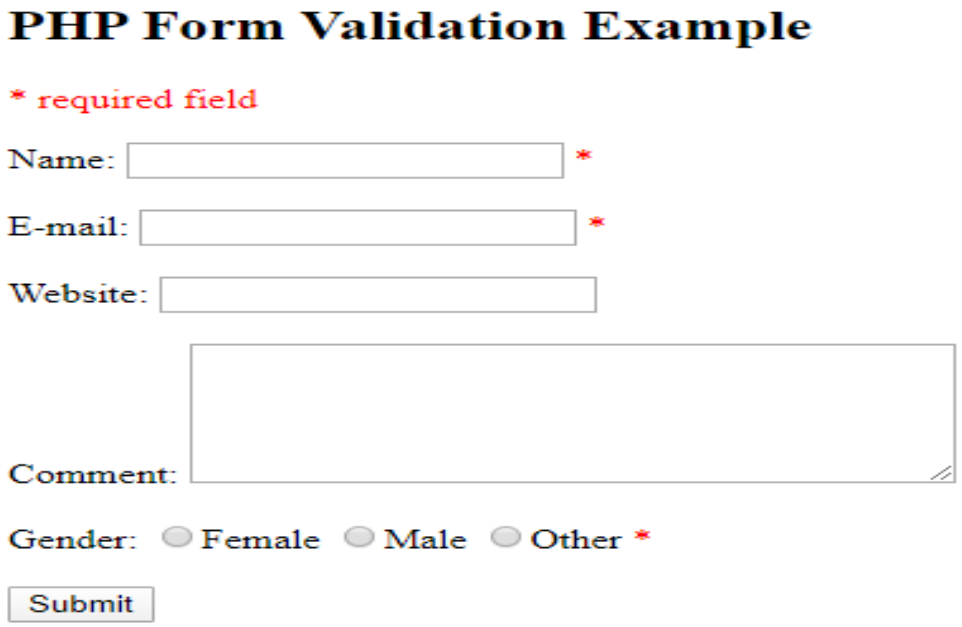
```
<html>
<body>

<?php
echo "Study " . $_GET['subject'] . " at " . $_GET['web'];
?>

</body>
</html>
```

PHP Form Validation

- Proper validation of form data is important to protect your form from hackers and spammers!
- Lets consider the example.



PHP Form Validation

- When your validation is passed, the hold the data in object.
- Now write a class for student using name, email, etc. attributes(See the form).
- Use this student object to hold the student data.

Books

- W3Schools Online Web Tutorials; URL: http://www.w3schools.com
- PHP Documentation; URL: http://www.php.net/docs.php
- Sams Teach Yourself Ajax JavaScript and PHP All in One; Phil Ballard and Michael Moncur; Sams Publishing; 2010
- JavaScript Phrasebook; Christian Wenz; Sams Publishing; 2007
- PHP and MySQL Web Development, 4/E; Luke Welling and Laura Thomson; Addison-Wesley Professional; 2009
- JavaScript for Programmers Paul J. Deitel and Harvey M. Deitel; Prentice Hall; 2009
- Beginning PHP5, Apache, and MySQL Web Development; Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz and Michael K. Glass; Wiley Publishing; 2005
- XML in a Nutshell, 3/E; Elliotte Rusty Harold and W. Scott Means; O'Reilly Media; 2004

References

1. https://www.w3schools.com/php/php_forms.asp