

# PHP Form Validation

Course Code: CSC 3222

Course Title: WEB TECHNOLOGIES



**Dept. of Computer Science  
Faculty of Science and Technology**

<b>Lab No:</b>	3	<b>Week No:</b>	3	<b>Semester:</b>	
<b>Lecturer:</b>	<i>Supta Richard Philip &amp; richard@aiub.edu</i>				



# Lecture Outline

- 1. Learning Objectives**
- 2. PHP Form Handling**
  - I. HTTP POST
  - II. HTTP GET
  - III. HTTP GET vs HTTP POST
- 3. PHP Form Validation**
- 4. Books and References**



# Learning Objectives

- In this Lab, we will learn more details about HTML form elements i.e. different type of form, designing different type of HTML form and form action.
- 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:
  - ```
<html>
<body>
<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
</body>
</html>
```



A screenshot of a simple HTML form. The form consists of two text input fields labeled "Name:" and "E-mail:", and a single "Submit" button. The "Name:" field is currently empty, and the "E-mail:" field also appears empty. The "Submit" button is located at the bottom right of the form area.



# PHP Form Handling

## HTTP POST

- When the user fills out the form and clicks the submit button, the form data is sent for processing to a PHP file named "welcome.php". The form data is sent with the HTTP POST method.
- welcome.php file
- ```
<html>
<body>
Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>
</body>
</html>
```



# PHP Form Handling

## HTTP GET

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



# PHP Form Handling

## HTTP GET vs HTTP POST

- Both GET and POST create an array (e.g. `array( key1 => value1, key2 => value2, key3 => value3, ... )`). This array holds key/value pairs, where keys are the names of the form controls and values are the input data from the user.
- `$_GET` is an array of variables passed to the current script via the URL parameters.
- Information sent from a form with the GET method is **visible to everyone** (all variable names and values are displayed in the URL).
- `$_POST` is an array of variables passed to the current script via the HTTP POST method.
- Information sent from a form with the POST method is invisible to others



# 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 Example

\* required field

Name:  \*

E-mail:  \*

Website:

Comment:

Gender:  Female  Male  Other \*



# PHP Form Validation

Field	Validation Rules
Name	Required. + Must only contain letters and whitespace
E-mail	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

# PHP Functions

The **empty()** function checks whether a variable is empty or not. This function returns false if the variable exists and is not empty, otherwise it returns true.

The following values evaluates to empty:

- 0
- 0.0
- "0"
- ""
- NULL
- FALSE
- array()

# PHP Functions

↗ <?php  
\$a = 0;

```
// True because $a is empty
if (empty($a)) {
    echo "Variable 'a' is empty.<br>";
}

?>
```

[https://www.w3schools.com/php/func\\_var\\_empty.asp](https://www.w3schools.com/php/func_var_empty.asp)

# PHP Functions

- ↗ The **isset()** function checks whether a variable is set, which means that it has to be declared and is not NULL.
- ↗ This function returns true if the variable exists and is not NULL, otherwise it returns false.

# PHP Functions

- ↗ 

```
<?php
$a = 0;
// True because $a is set
if (isset($a)) {
    echo "Variable 'a' is set.<br>";
}

$b = null;
// False because $b is NULL
if (isset($b)) {
    echo "Variable 'b' is set.";
}
?>
```
- ↗ [https://www.w3schools.com/php/func\\_var\\_isset.asp](https://www.w3schools.com/php/func_var_isset.asp)

# PHP Functions

The **preg\_match()** function returns whether a match was found in a string.

```
<?php  
$str = "Visit W3Schools";  
$pattern = "/w3schools/i";  
echo preg_match($pattern, $str);  
?>
```

[https://www.w3schools.com/php/func\\_regex\\_preg\\_match.asp](https://www.w3schools.com/php/func_regex_preg_match.asp)



# 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 Stoltz 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](https://www.w3schools.com/php/php_forms.asp)