



CET-223

Web Technologies

Experiment # 22

Experiment Title

PHP Scripts – II

Assessment of CLO(s): 03

Performed on _____

Student Name:			
Roll No.		Group	
Semester		Session	

Total (Max)	Performance (03)	Viva (03)	File (04)	Total (10)
Marks Obtained				
Remarks (if any)				

Experiment evaluated by

Instructor's Name	Engr. Bilal Iqbal		
Date		Signature	

OBJECTIVE:

This lab focuses on handling form data in PHP using the GET and POST methods. Students will learn to create HTML forms and process the submitted data on the server.

What is Form?

Forms are used to get input from the user and submit it to the web server for processing.

When and why we are using forms?

- 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.
- Form submission type **POST or GET**.

There are two ways the browser client can send information to the web server.

- The GET Method
- The POST Method

The GET Method

The GET method sends the encoded user information appended to the page request.

- The GET method produces a long string that appears in your server logs, in the browser's Location: box.
- The GET method is restricted to send up to 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.
- The data sent by GET method can be accessed using QUERY_STRING environment variable.
- The PHP provides \$_GET associative array to access all the sent information using GET method.
- The GET method should not be used while sending any sensitive information.
- A limited amount of data can be sent using method = "get". This limit should not exceed 2048 characters.
- For security reasons, never use the GET method to send highly sensitive information like username and password, because it shows them in the URL.

```
<html>
  <body>

    <form action = "<?php $_PHP_SELF ?>" method = "GET">
      Name: <input type = "text" name = "name" />
      Age: <input type = "text" name = "age" />
      <input type = "submit" />
    </form>
  </body>
</html>
```

Post Method

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.

The POST method does not have any restriction on data size to be sent.

Lab Experiment 22: PHP Scripts - II

The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.

The POST method is useful for sending any sensitive information because the information sent using the POST method is not visible to anyone.

There is no limitation on size of data to be sent using the POST Method. You can send a large amount of information using this method.

Binary and ASCII data can also be sent using the POST method.

```
<html>
  <body>

    <form action = "one.php" method = "Post">
      Name: <input type = "text" name = "name" />
      Age: <input type = "text" name = "age" />
      <input type = "submit" />
    </form>
  </body>
</html>
```

One.php

```
<html>
  <body>

    Welcome <?php echo $_POST["name"]; ?> </br>
    Your age is: <?php echo $_POST["age"]; ?>

  </body>
</html>
```

Example 3

```
<html>
<head>
<title>form handling </title>
<style>
body {
  border: 1px solid #996633
}
h1 {
  margin-left:380px; color: #0F0;
}
input[type=text]{
  width: 150px; display: block; margin-left:420px; margin-bottom:
10px;background-color: #FFF;
}
input[type=submit] {
  width: 150px; margin-left:420px; display: block;
}
label {
  margin-left:420px;
}
</style>
```

```
</head>
<body>
<h1>Form Handling</h1>
<form action="get.php" method="get">
  <label for="firstName">First Name:</label><br>
  <input type="text" name="txt_fname" /><br>

  <label for="firstName">Last Name:</label><br>
  <input type="text" name="txt_lname" /><br>

  <label for="Salary">Salary:</label><br>
  <input type="text" name="txt_salary" /><br>
  <input type="submit" />
</form>
</body>
</html>
```

Get.php

```
<html>
<head>
<style>
h1 {
    margin-left:200px;
    color: #666;
}
table, td, th {
    margin-left:200px;
    border: 1px solid green;
}
th {

    width:150px;
    background-color: green;  color: white;
}
</style>
</head>
<body>
<h1>Values retrived by GET method</h1>
<?php
    echo "<table>";
    echo "<tr>";
        echo "<th>First Name</th>";
        echo "<th>Last Name</th>";
        echo "<th>Salary</th>";

    echo "</tr>";

    echo "<tr>";
        echo "<td>" . $_GET["txt_fname"] . "</td>";
        echo "<td>" . $_GET["txt_lname"] . "</td>";
        echo "<td>" . $_GET["txt_salary"] . "</td>";

    echo "</tr>";
```

```
        echo "</table>";

?>
</body>
</html>
```

Example 4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Untitled Document</title>
</head>

<body>
<form method="post">
<input type="text" name="one" placeholder="Enter First
number"/><br/>
<input type="text" name="two" placeholder="Enter Second
Number"/><br/>
<select name="option">
  <option value="plus">+</option>
  <option value="minus">_</option>
  <option value="multi">*</option>
  <option value="div">/</option>
</select><br/>
<input type="submit" name="btn" value="Claculate"/>

</form>

<?php

if(isset($_POST['btn'])) {

    $first=$_POST['one'];
    $second=$_POST['two'];
    $option=$_POST['option'];
    $ans=$first+$second;
    echo 'Ans is:' . $ans;

}
?>

<h1> <?php echo $ans ?></h1>
</body>
</html>
```

Validate Form Data with PHP

The first thing we will do is to pass all variables through PHP's htmlspecialchars() function.

- Strip unnecessary characters (extra space, tab, newline) from the user input data (with the PHP trim() function).
- Remove backslashes (\) from the user input data (with the PHP stripslashes() function)

```
<!DOCTYPE HTML>
<html>
<head>
</head>
<body>

<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";

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

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($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>

    Comment: <textarea name="comment" rows="5"
cols="40"></textarea>
    <br><br>
    Gender:
    <input type="radio" name="gender" value="female">Female
    <input type="radio" name="gender" value="male">Male
    <input type="radio" name="gender" value="other">Other
    <br><br>
    <input type="submit" name="submit" value="Submit">
</form>

<?php
echo "<h2>Your Input:</h2>";
echo $name;
```

```
echo "<br>";  
echo $email;  
echo "<br>";  
echo $website;  
echo "<br>";  
echo $comment;  
echo "<br>";  
echo $gender;  
?>  
  
</body>  
</html>
```

Lab Tasks

Apply PHP get and post method after designing below form.



Name:

Email:

Age:

phone number: