

|  |  |
| --- | --- |
| LAB 4  PHP FORM | |
| COURSE CODE | **DAT 21303** |
| COURSE NAME | **WEB DEVELOPMENT** |
| FACULTY | **CENTRE OF DIPLOMA STUDIES (CeDS)**  **DEPARTMENT OF INFORMATION TECHNOLOGY** |
| STUDENT NAME | **QAIREL QAYYUM BIN MUHAMAD RIDHUAN**  **(212363)**  **WAN MUHAMMAD ALIF FIRDAUS BIN WAN HANAPI**  **(AA210292)** |
| SECTION | **6** |
| LECTURER NAME | **MISS NADIA** |
| SUBMIT DATE |  |

**Question 1:**  Based on Figure 1.1 is about HTML form woven with PHP. Just write it back in your editor and run it from your localhost. Provide the output with full and logical explanation line by line.

//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"]);

$gender = test\_input($\_POST["gender"]);

$comment = test\_input($\_POST["comment"]);

$website = test\_input($\_POST["website"]);

Declaring Variables for the form to key in value. Test input function (test\_input) is called to validate every variable input.

function test\_input($data){

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

Form Input Validatation

* trim($data) will strip unnecessary characters (extra space, tab, newline) from the user input data (with the PHP trim() function).
* stripslashes($data) will remove backslashes () from the user input data (with the PHP stripslashes() function).
* htmlspecialchars($data) converts special characters to HTML entities.

?>

<h2>Tutorial point absolute classes registration</h2>

<form method= "post"action = "form.php">

<table>

<tr>

<td>Name:</td>

<td><input type= "text"name="name"></td>

</tr>

<tr>

<td>E-mail:</td>

<td><input type = "text" name= "email"value="<?php echo $email?>"></td>

</tr>

<tr>

<td>Specific Time:</td>

<td><input type="text"name="website"></td>

</tr>

<tr>

<td>Class details:</td>

<td><textarea name = "comment" rows = "5" cols = "40"></textarea></td>

</tr>

<tr>

<td>Gender:</td>

<td>

<input type = "radio" name = "gender" value = "female">Female

<input type = "radio" name = "gender" value = "male">Male

</td>

</tr>

<tr>

<td>

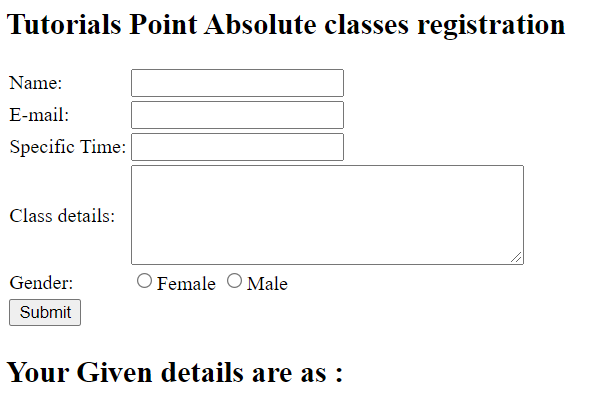
<input type = "submit" name = "submit" value = "Submit">

</td>

</tr>

</table>

</form>



Form Interface;

* <td> stands for table data. Everything between <td> and </td> are the content of the table cell.
* <tr> stands for table row.
* Input text = “text” to collect text input
* value="<?php echo $email?>" is to transfer data to the variable.
* textarea name="comment" rows="5" cols="40" is to specifies the space user can key text data.
* <input type="radio" name="gender" value="female">Female

<input type="radio" name="gender" value="male">Male is option for user to choose any one.

<?php

echo "<h2>Your Given details are as :</h2>";

echo $name;

echo "<br>";

echo $email;

echo "<br>";

echo $website;

echo "<br>";

echo $comment;

echo "<br>";

echo $gender;

?>

</body>

</html>

</body>

</html>

Code to display the data output that user has key in.

**Question 2:**  Create a simple form such as in figure 1.2 below and attach a control structure (if else) to validate if the input is greater than 100 the message will say “true” and if the input value below than hundred the message is “false”. The message must be shown by using JavaScript.

<!DOCTYPE html>

<html><head></head>

<body>

<?php

$number = $\_GET["number"];

echo "number: ".$number;

?>

<p id="demo">..</p>

  <script>

      <?php echo $number; ?>;

      var x = "<?php echo "$number"?>";

      let greeting;

      if(x < 100)

      {

        greeting = "False";

      }

      else

      {

        greeting = "True";

      }

      document.getElementById("demo").innerHTML = greeting;

  </script>

</body>

</html>

<html>

    <body>

    <form method="get" action="Lab3Web.php">

    Input Number: <input type="text" name="number"><br>

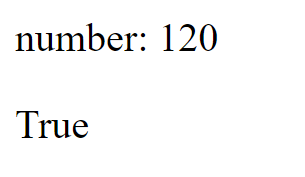
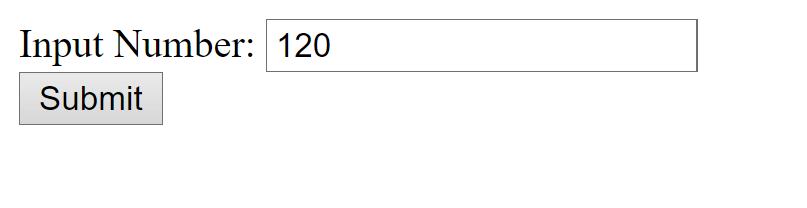
    <input type="submit" name="submit" value="Submit">

</form>

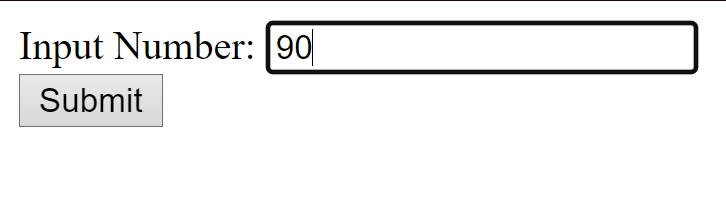
    </body>

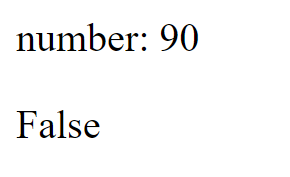
</html>

CODE QUESTION 2



**TRUE NUMBER**

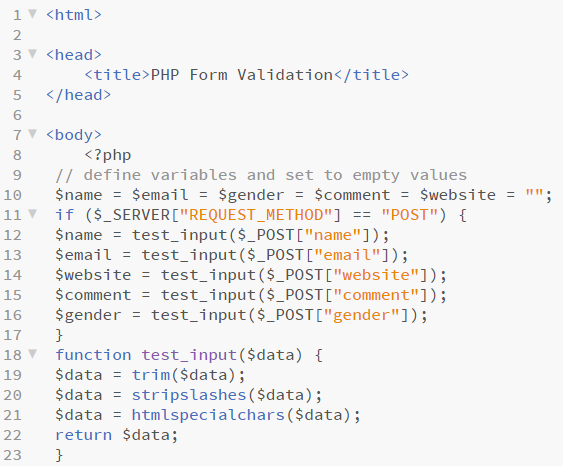
****



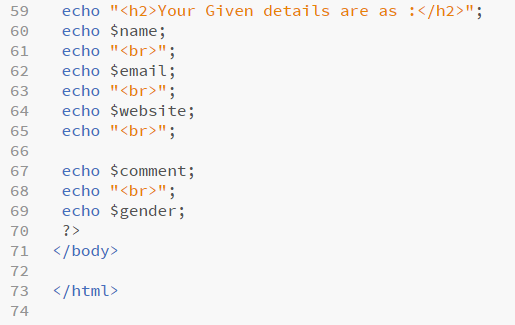
**FALSE NUMBER**

**Question 3:** Modify the given codes in question 1 in order to keep the values in the form and to show the values in the input fields after the user hits the submit button.

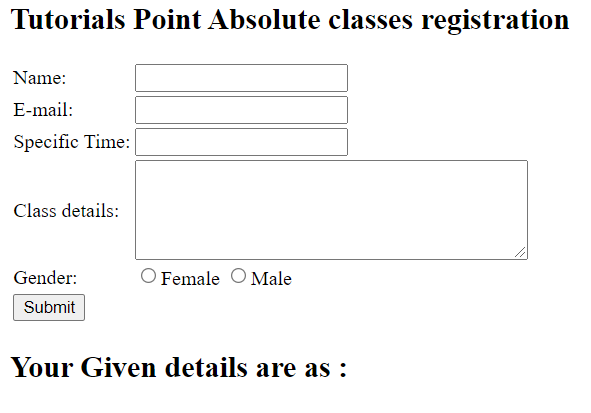
***CODE***

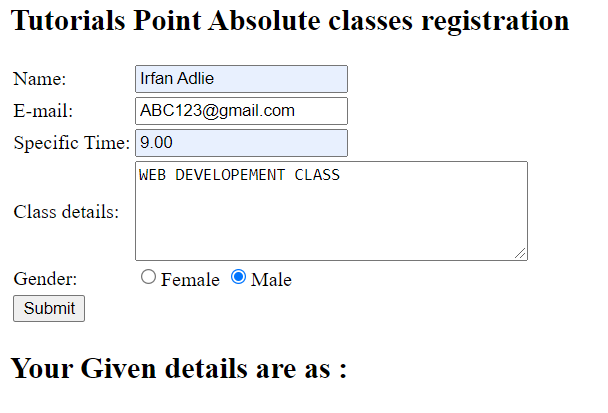


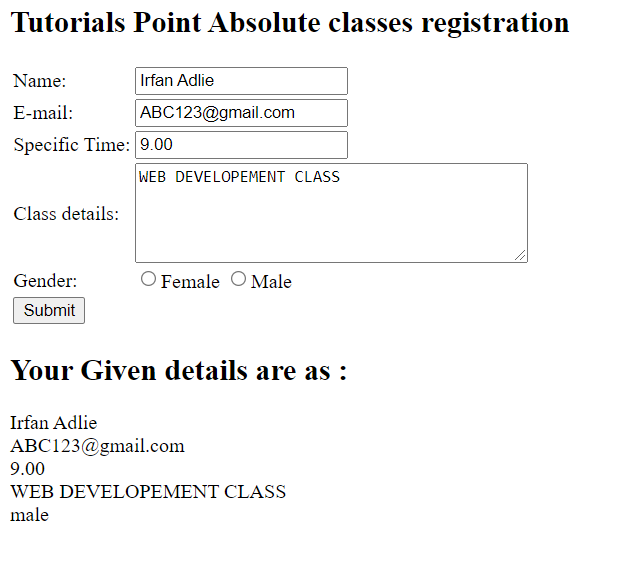




Interface/input/output



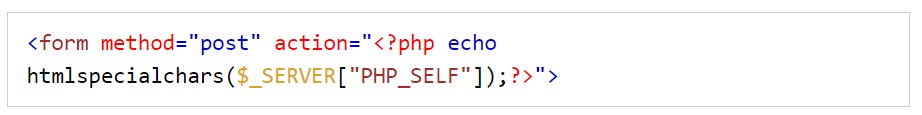




**Question 4:** How to Avoid $\_SERVER["PHP\_SELF"] Exploits?

$\_SERVER["PHP\_SELF"] exploits can be avoided by using the htmlspecialchars() function.

The form code should look like this:



The htmlspecialchars() function alters special characters to HTML entities. Then, if the user attempts to exploit the PHP\_SELF variable, it will result in the following output:



Therefore, the exploit attempt fails, and no harm is done.