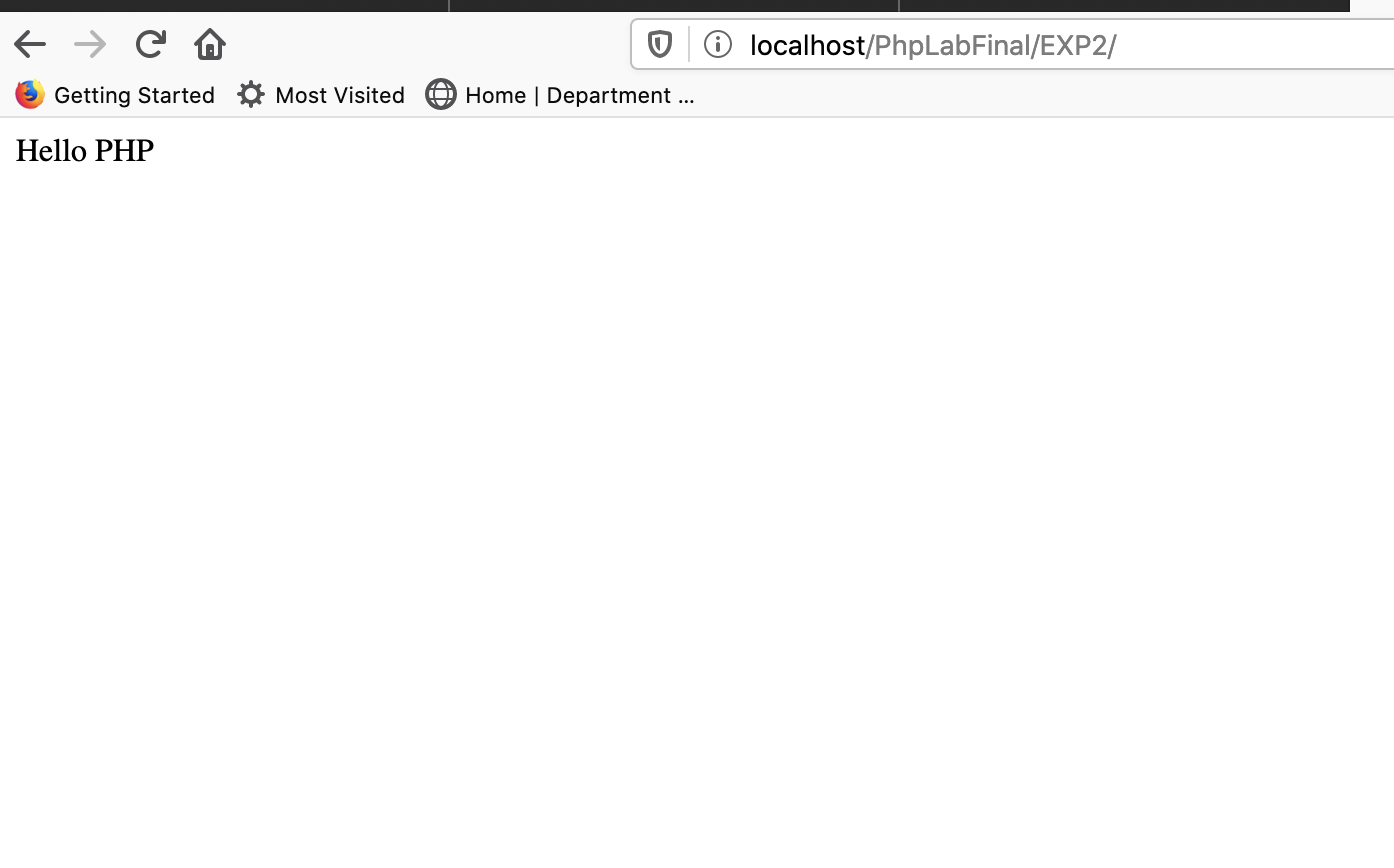
PHP LAB EXPERIMENTS

1. XAMP installation and configuration: Apache Server & MySQL
2. Write a PHP script to display “Hello PHP” using echo or Print.

<?php

echo "Hello PHP";

?>

OUTPUT:

1. Write a PHP script to print a textbox value using PHP. Use the HTML form tag and submit button to perform the experiment.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

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

<input type="text" name="txt" required>

<button type="submit" name="sub">Submit</button>

</form>

<?php

if (isset($\_POST['sub'])) {

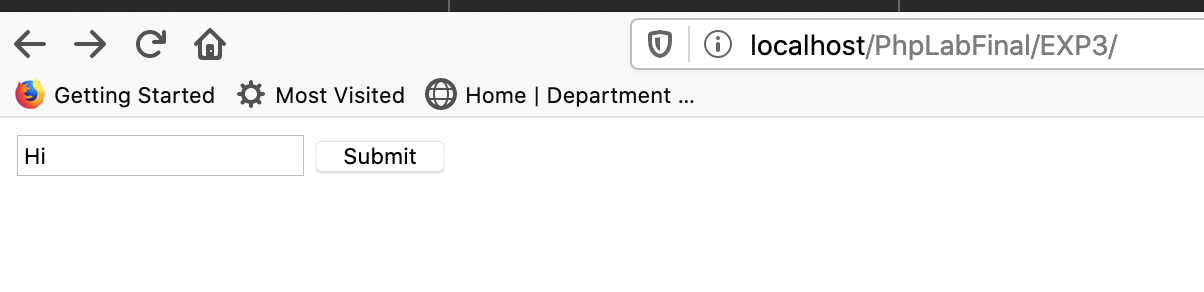
echo $\_POST['txt'];

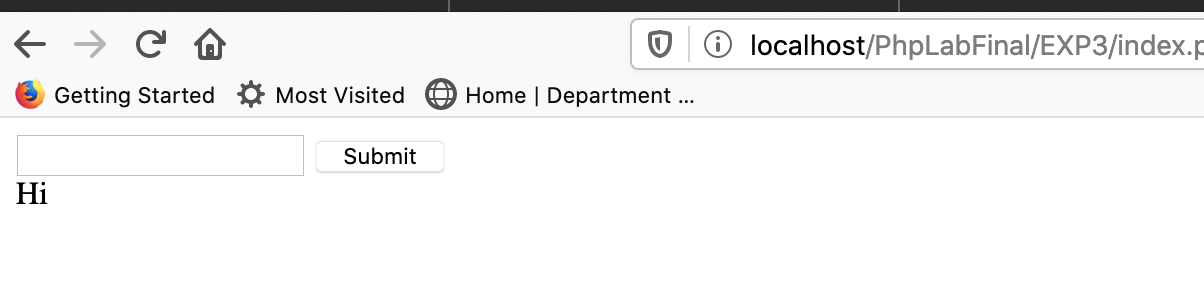
}

?>

</body>

</html>





1. Write a PHP script to perform the following string functions.
   1. String Length
   2. Reverse String
   3. Replace text within a string

<html>

<body>

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

<input type="text" name="strInp" required>

<button type="submit" name="sub">Submit</button>

</form>

<?php

if (isset($\_POST['sub'])) {

$x = $\_POST['strInp'];

echo "Length of the String = ".strlen($x)."</br>";

echo "Reverse String = ".strrev($x)."</br>";

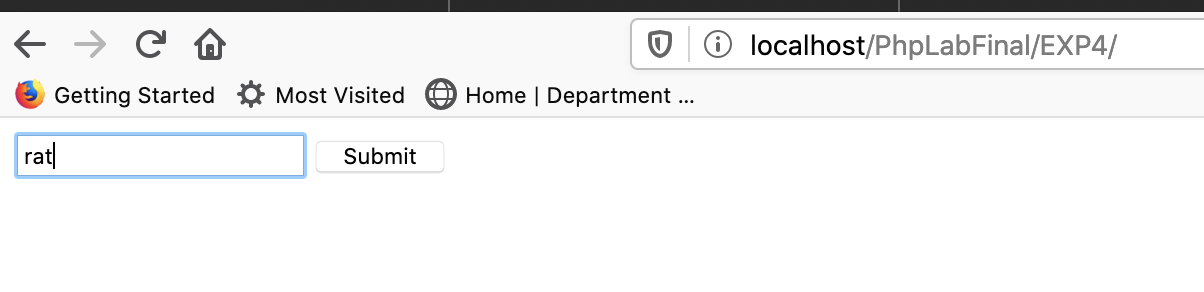
echo "Replacing \"r\" with \"S\" : ".str\_replace('r','S',$x);

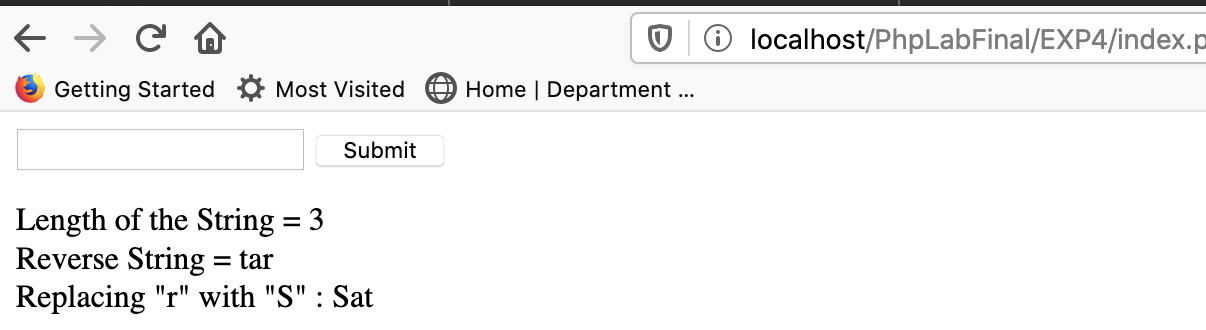
}

?>

</body>

</html>





1. Write a PHP script to perform basic calculator operations.

<html>

<body>

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

<input type="number" name="num1" id="num1" required>

<select name="sel" id="sel">

<option value="+" name="+">+</option>

<option value="-" name="-">-</option>

<option value="x" name="x">x</option>

<option value="÷" name="÷">÷</option>

</select>

<input type="number" name="num2" id="num2" required>

<button type="submit" name="sub">Calculate</button>

</form>

<?php

if (isset($\_POST['sub'])) {

$n1 = $\_POST['num1'];

$n2 = $\_POST['num2'];

$opr = $\_POST['sel'];

echo "$n1 $opr $n2 = ";

switch ($opr) {

case '+':

echo ($n1+$n2);

break;

case '-':

echo ($n1-$n2);

break;

case 'x':

echo ($n1\*$n2);

break;

case '÷':

echo ($n1/$n2);

break;

default:

# code...

break;

}

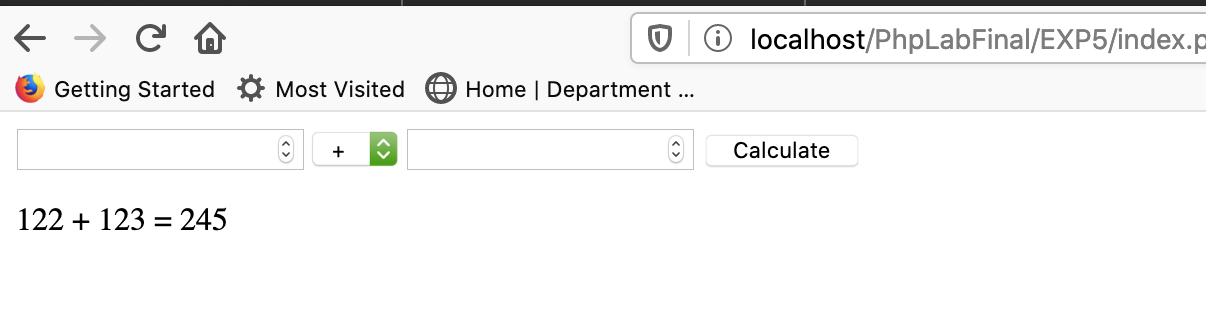
}

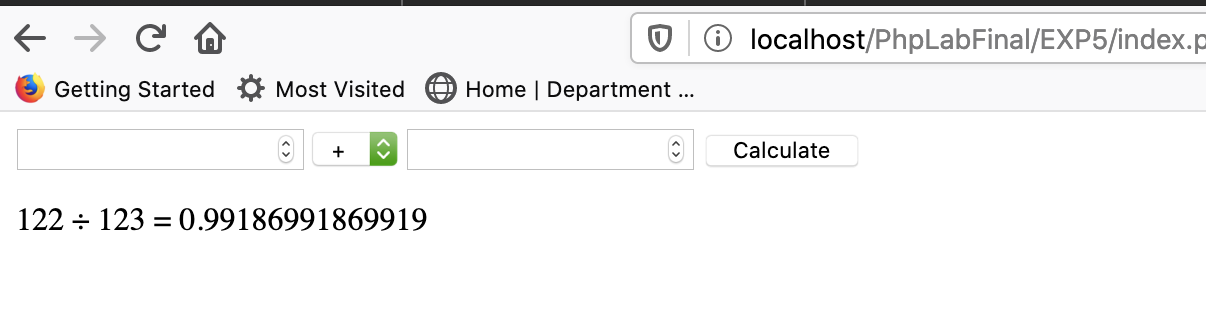
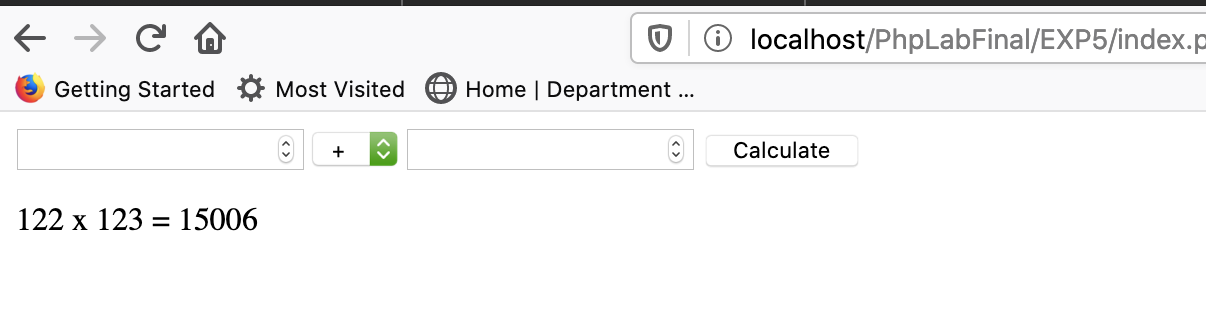
?>

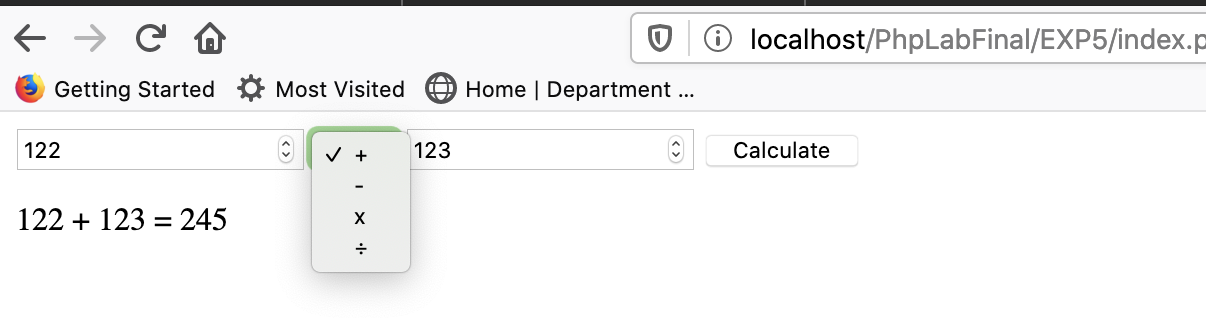
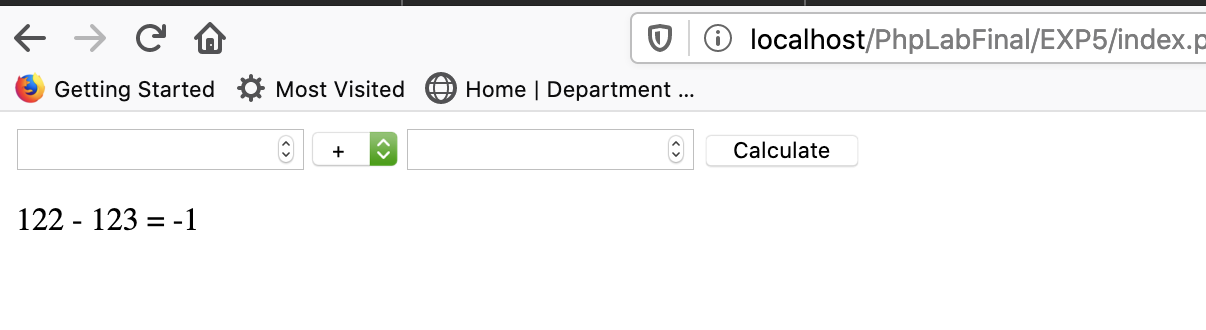
</body>

</html>

Output:







1. Write a PHP script to display the customer registration form submitted using the HTML page in tabular form.

index.php:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Form</title>

</head>

<body>

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

<div class="main-body">

<div class="row">

<div class="user-inst">

<label for="name">Name</label>

</div>

<div class="user-input">

<input type="text" name="name" id="name" placeholder="Enter name">

</div>

</div>

<!-- Written by Rishabh Panesar only -->

<div class="row">

<div class="user-inst">

<label for="email">Email</label>

</div>

<div class="user-input">

<input type="text" name="email" id="email" placeholder="Enter email" >

</div>

</div>

<div class="row">

<div class="user-inst">

<label for="Website">Website</label>

</div>

<div class="user-input">

<input type="text" name="Website" id="Website" placeholder="Enter Website" required>

</div>

</div>

<div class="row">

<div class="user-inst">

<label for="comment">Comment</label>

</div>

<div class="user-input">

<textarea name="comment" id="comment" cols="36" rows="5"></textarea>

</div>

</div>

<div class="row">

<div class="user-inst">

<label for="Phone">Phone</label>

</div>

<div class="user-input">

<input type="number" name="Phone" id="Phone" placeholder="Enter Phone" required>

</div>

</div>

<!-- Written by Rishabh Panesar only -->

<div class="row">

<div class="user-inst">

<label for="Password">Password</label>

</div>

<div class="user-input">

<input type="password" name="Password" id="Password" placeholder="Enter Password" required>

</div>

</div>

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

</div>

</form>

<!-- Written by Rishabh Panesar only -->

</body>

</html>

action.php:

<html>

<head>

</head>

<body>

<?php

if (isset($\_POST['sub'])) {

$name = $\_POST['name'];

$email = $\_POST['email'];

$website = $\_POST['Website'];

echo "<table border = '1'>

<tr>

<th>Name</th>

<th>Email</th>

<th>Website</th>

</tr>

<tr>

<td>$name</td>

<td>$email</td>

<td>$website</td>

</tr>

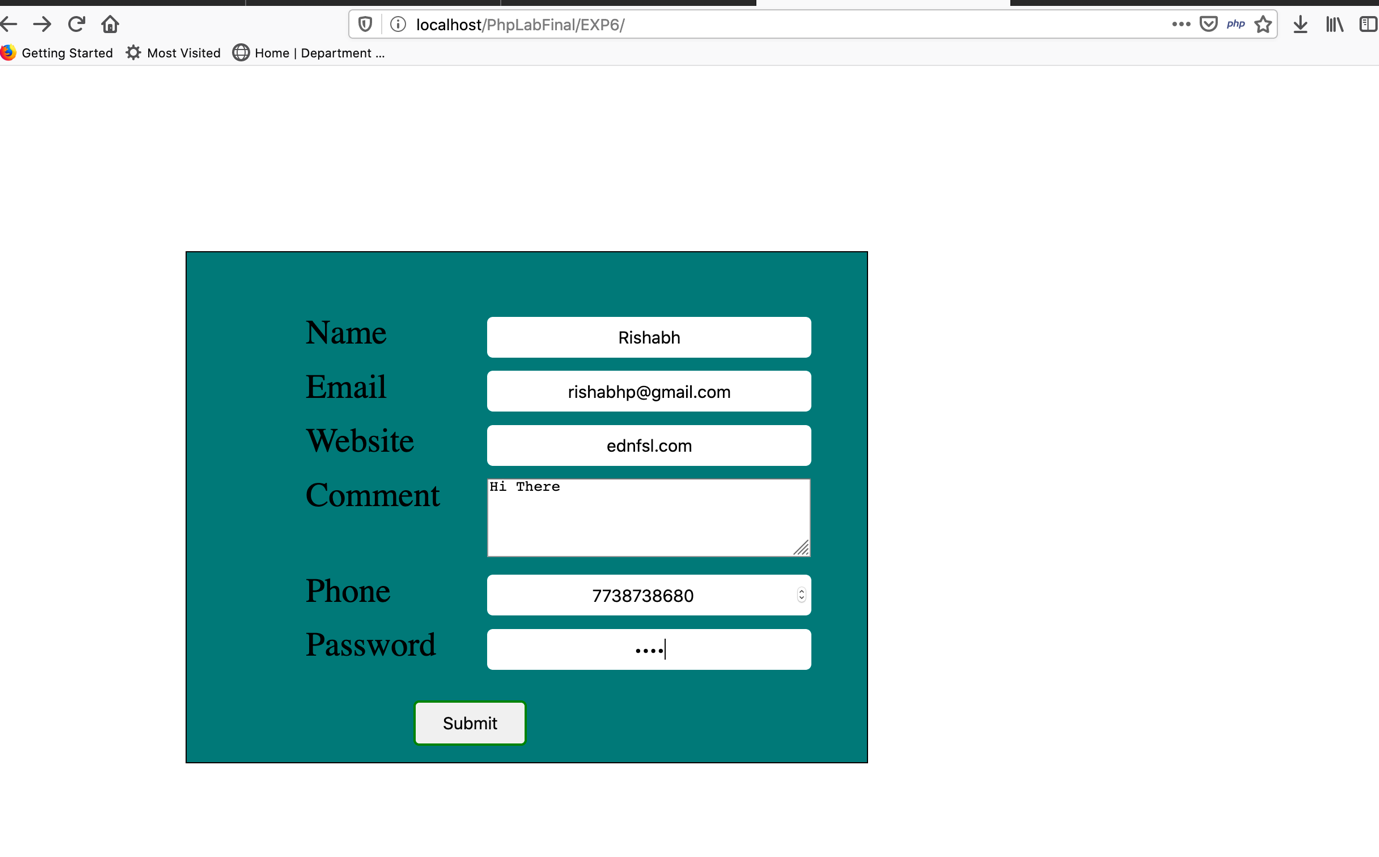
</table>";

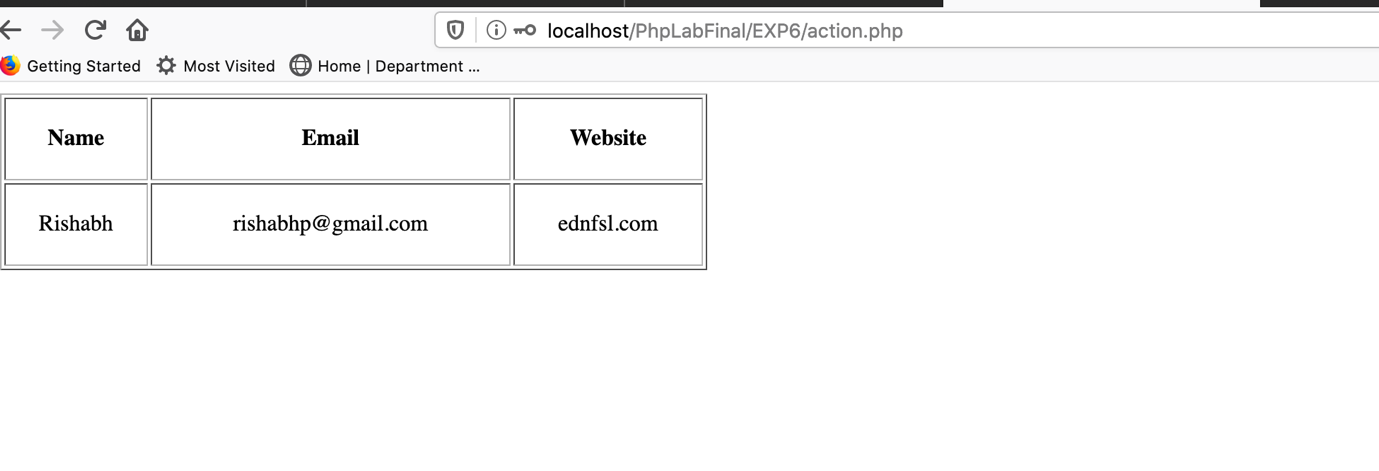
}

?>

</body>

</html>





1. Show XSS (Cross-Site Scripting) injection using an example.

index.php:

<html>

<head></head>

<body>

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

<h1>XSS Example</h1>

<label for="usrInp">Enter Username </label>

<input type="text" name="usrInp" id="usrInp">

<button type="submit" name="sub">Submit</button>

</form>

<?php

if (isset($\_POST['sub'])) {

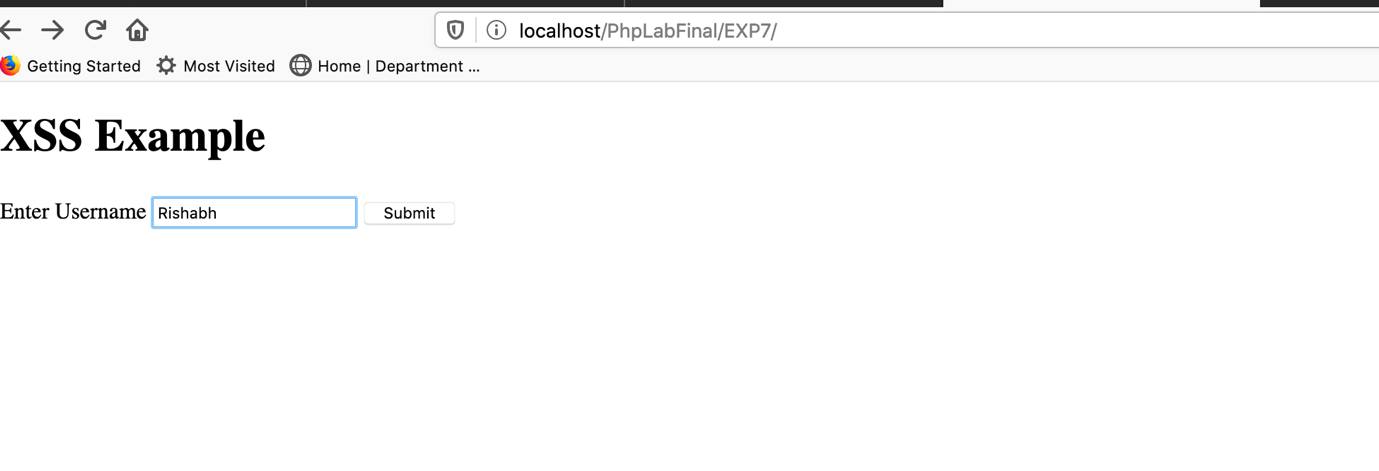
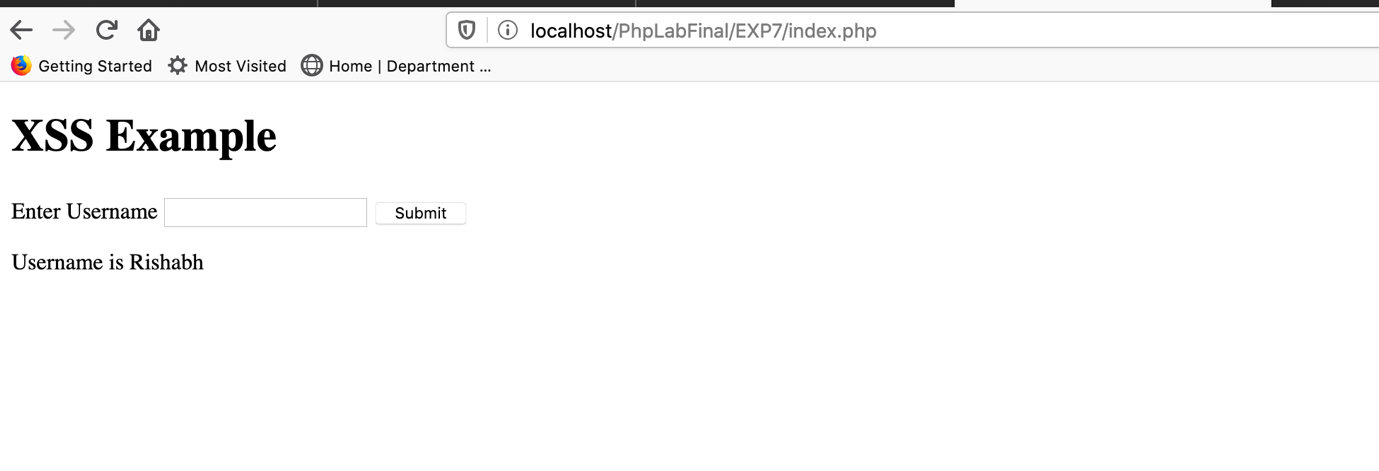
$x = $\_POST['usrInp'];

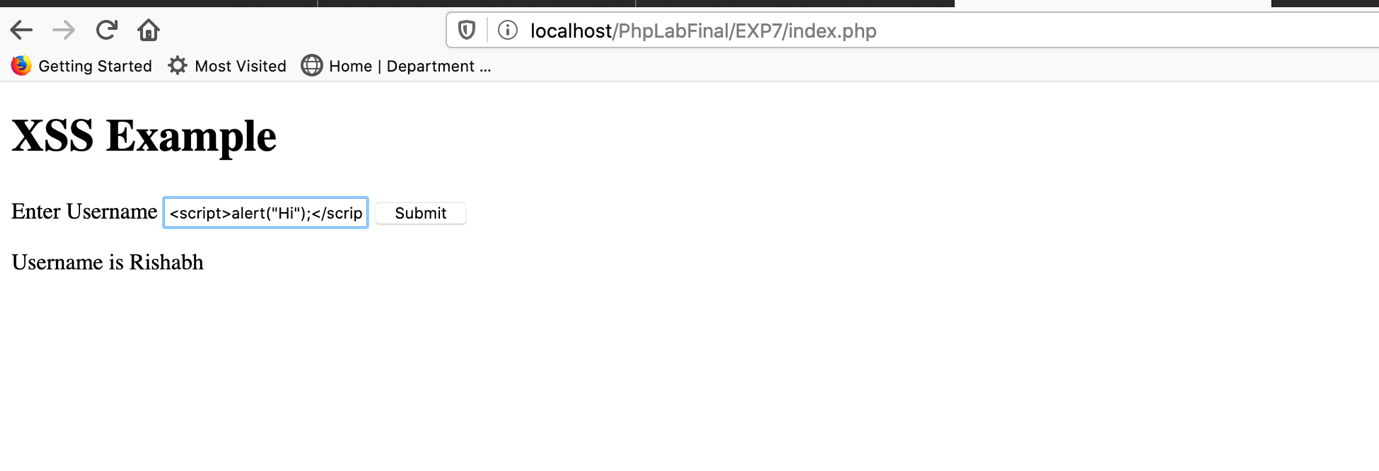
echo "Username is $x";

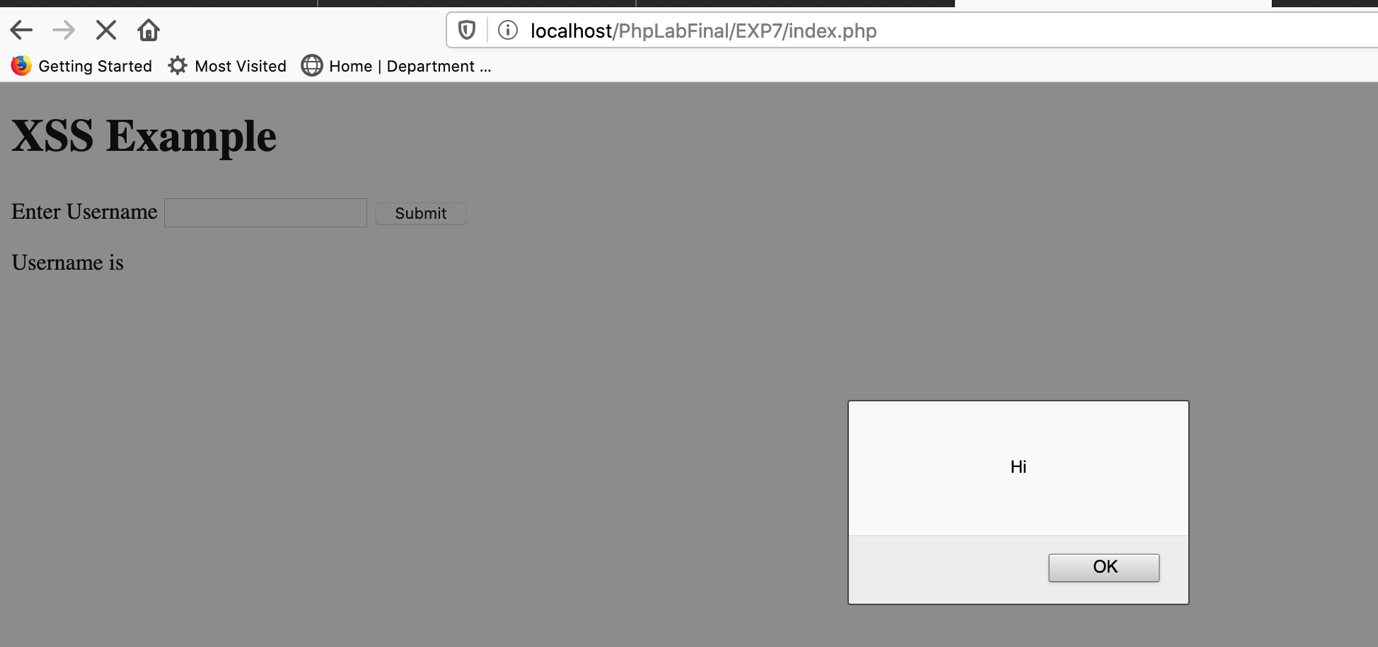
}

?>

</body>

</html>





1. Write a PHP script to secure your inputs using trim, stripslashes, and htmlspecialchars functions.

In order to prevent this attack, we need to use some special functions.

index.php:

<html>

<head></head>

<body>

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

<h1>XSS Example</h1>

<label for="usrInp">Enter Username </label>

<input type="text" name="usrInp" id="usrInp">

<button type="submit" name="sub">Submit</button>

</form>

<?php

if (isset($\_POST['sub'])) {

$x = $\_POST['usrInp'];

echo "Username is ".purifyInput($x);

}

function purifyInput($inp){

$inp = strip\_tags($inp);

$inp = stripslashes($inp);

$inp = htmlspecialchars($inp);

$inp = trim($inp,"\t \n\r");

return $inp;

}

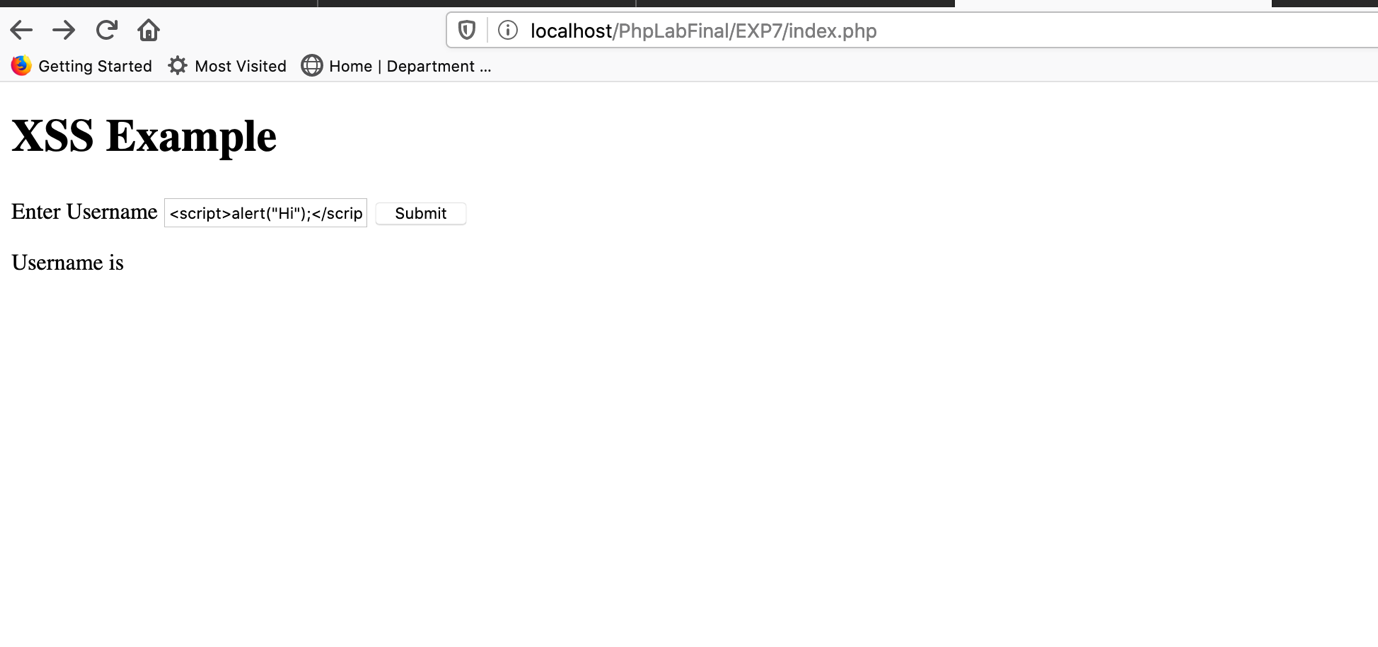
?>

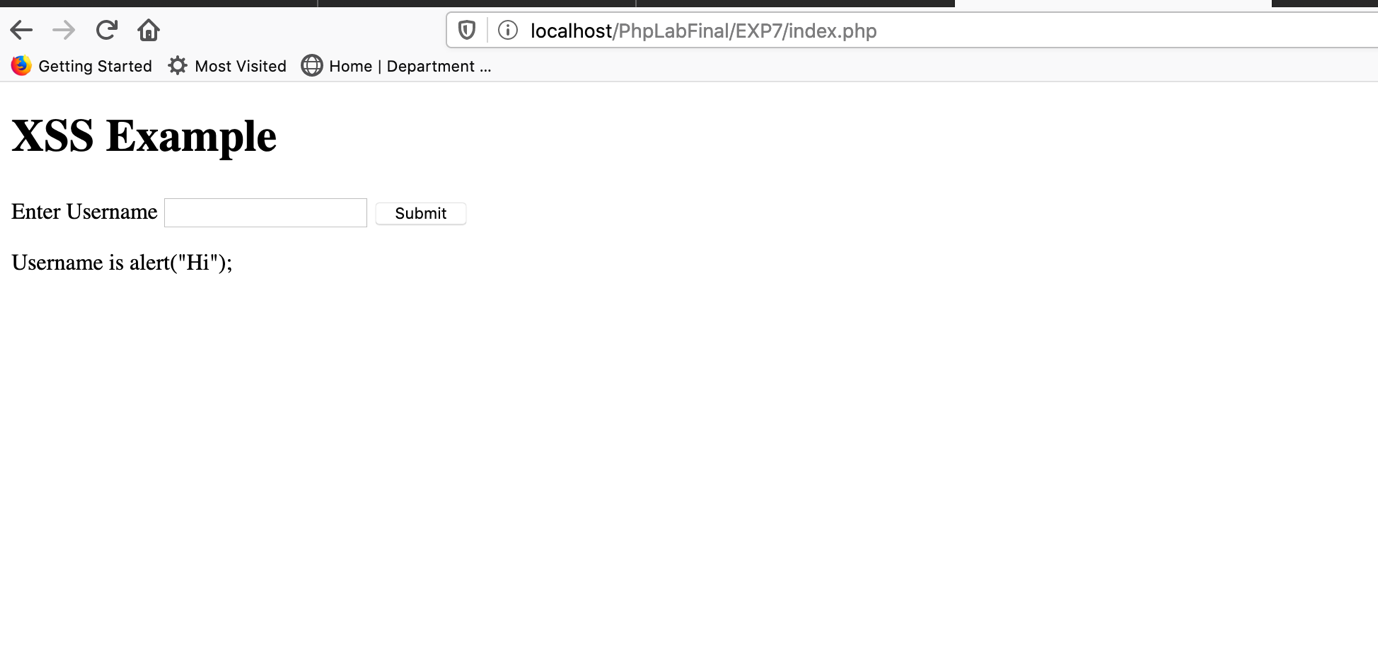
</body>

</html>

Now if we try to enter the script in the input, we will not see the same output.

Output:





1. Write a PHP script to validate URL, email and contact in a registration form using regular expressions.
2. Write a PHP script to show the functionality of “Include” and “require”.
3. Write a PHP script to understand the functionality of Cookies.
4. Write a PHP script to make a session and destroy a session using time.
5. Write a PHP script to show different functions of Date and time.
6. Write a PHP script to upload an image using checks like duplicate image, size of the image and format of the image.
7. Write PHP scripts to show the read, write and append operations using file handling.
8. Write PHP code to connect to the database and using prepared statement send form values to the database using MySQLi (object oriented)
9. Configure SMTP server, php.ini and sendmail.ini to send email using localhost.
10. Install Composer and create a project using Laravel framework and configure .env to connect to the database.