Lab03: PHP Form Handling

Mahesh Kumar
Assistant Professor (Adhoc)

Department of Computer Science
Acharya Narendra Dev College
University of Delhi

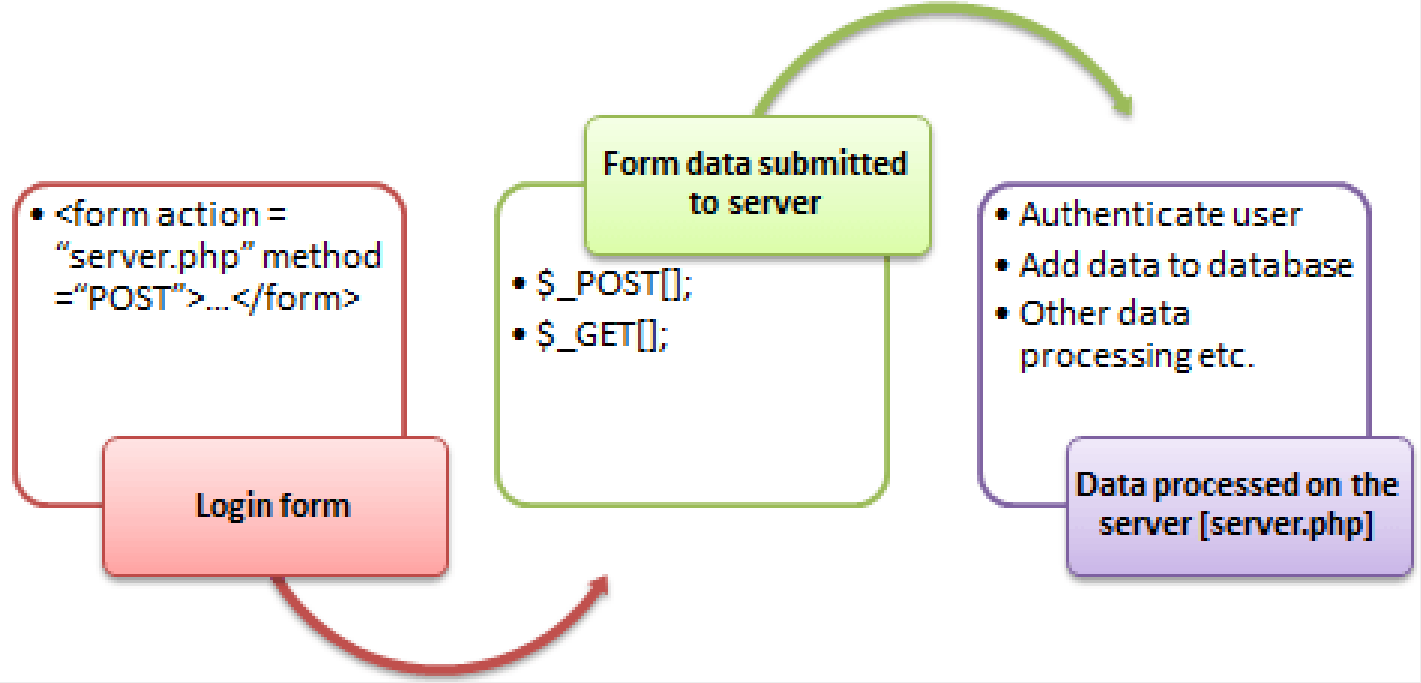
Course webpage
[http://www.mkbhandari.com/mkwiki]

What is Form?

- Forms are used to get input from the user and submit it to the web server for processing.
- A form is an HTML tag that contains graphical user interface items (elements) such as textbox, check boxes, radio buttons etc.
- The form is defined using the <form>...</form> tags and GUI items are defined using form elements such as input.
- 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.
- The PHP superglobals $_GET and $_POST are used to collect form-data.
- 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.

What is Form?

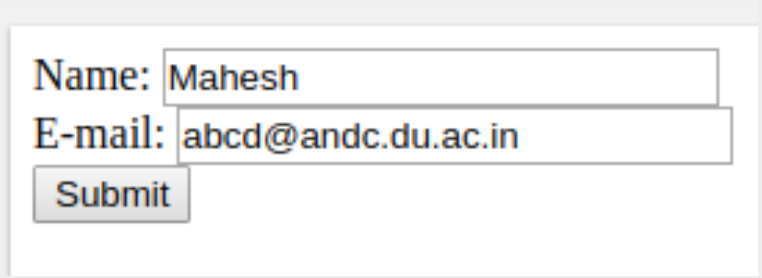
- The diagram below illustrates the form handling process.



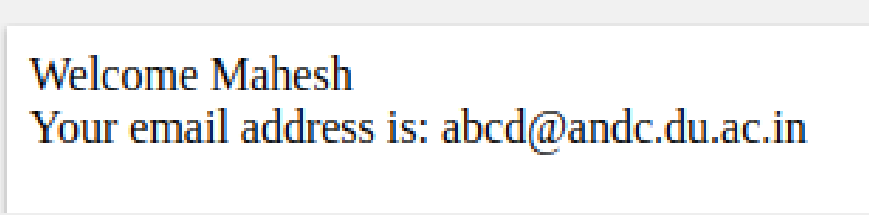
- Both GET and POST are treated as $_GET and $_POST.

A Sample Form – POST Method

```
<!DOCTYPE HTML>
<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>
```




```
<html>
<body>
Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>
</body>
</html>
```



GET VS POST Methods

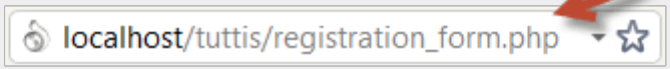
POST	GET
Values not visible in the URL	Values visible in the URL
Has not limitation of the length of the values since they are submitted via the body of HTTP	Has limitation on the length of the values usually 255 characters. This is because the values are displayed in the URL. Note the upper limit of the characters is dependent on the browser.
Has lower performance compared to Php_GET method due to time spent encapsulation the Php_POST values in the HTTP body	Has high performance compared to POST method dues to the simple nature of appending the values in the URL.
Supports many different data types such as string, numeric, binary etc.	Supports only string data types because the values are displayed in the URL
Results cannot be book marked	Results can be book marked due to the visibility of the values in the URL

GET VS POST Methods

FORM SUBMISSION POST METHOD

```
<form action="registration_form.php" method="POST">
    First name: <input type="text" name="firstname"><br>
    Last name: <input type="text" name="lastname">
    <br>
    <input type="hidden" name="form_submitted" value="1"/>
    <input type="submit" value="Submit">
</form>
```

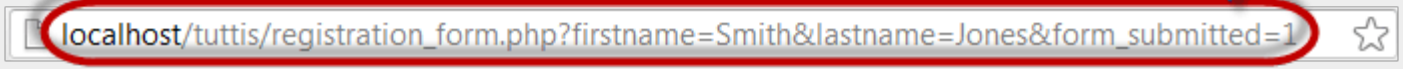
Submission URL does not show form values



FORM SUBMISSION GET METHOD

```
<form action="registration_form.php" method="GET">
    First name: <input type="text" name="firstname"><br>
    Last name: <input type="text" name="lastname">
    <br>
    <input type="hidden" name="form_submitted" value="1"/>
    <input type="submit" value="Submit">
</form>
```

SUBMISSION URL SHOWS FORM VALUES



Lab Exercise No. 1

```
//Input a String
<!Doctype html>
   <head>
      <title>PHP program1 </title>
   </head>
   <body>
      <form method="POST" action="revstr.php">
         Enter a String:
         <input type="text" name="str" required>
         <input type="submit" value="SUBMIT">
      </form>
   </body>
   </html>
```

```
//Reverse of String
<?php
      $str1=$_POST['str'];
      echo "Original String = ".$str1."<br/>";
      $rstr1=strrev($str1);
      echo "Reverse String = ".$rstr1."<br/>";
?>
```

Practice Set No. 16

```
// PHP code to get the Fibonacci series using recursion
<?php
// Recursive function for fibonacci series.
function Fibonacci($number) {
        // if and else if to generate first two numbers
        if ($number = = 0)
                return 0;
        else if ($number == 1)
                return 1;
        // Recursive Call to get the upcoming numbers
        else
                return (Fibonacci($number-1) +
                                Fibonacci($number-2));
}
// Driver Code
$number = 10;
for ($counter = 0; $counter < $number; $counter++){
        echo Fibonacci($counter),' ';
}
?>
```

References

Reference for this topic

- [Introduction to PHP]
  https://www.w3schools.com/php/default.asp

- [Install and Run PHP]
  https://www.techomoro.com/how-to-run-a-php-application-on-ubuntu-18-04-2-lts/

- [GeeksforGeeks]
  https://www.geeksforgeeks.org/php/