



# UNIVERSITY OF INFORMATION TECHNOLOGY AND SCIENCES (UITs)

LAB REPORT : 2

IT-422 : INTERNET PROGRAMMING LAB

---

## PHP Form Handling

---

*Submitted To:*

Zinia Anzum Tonni  
Lecturer,  
Department of IT, UITs

*Submitted By:*

Name: Nazmul Zaman  
Student ID:2014755055  
Department of IT UITs

24 MAY 2022

## Contents

|          |                                 |          |
|----------|---------------------------------|----------|
| <b>1</b> | <b>Abstrac</b>                  | <b>2</b> |
| <b>2</b> | <b>Source Code</b>              | <b>3</b> |
| 2.1      | (POST Method) . . . . .         | 3        |
| 2.2      | (index.php) . . . . .           | 3        |
| 2.3      | (home.php) . . . . .            | 3        |
| 2.4      | (GET Method) . . . . .          | 4        |
| 2.5      | (index2.php) . . . . .          | 4        |
| 2.6      | (home2.php) . . . . .           | 4        |
| 2.7      | (REQUEST Method) . . . . .      | 5        |
| 2.8      | (index3.php) . . . . .          | 5        |
| 2.9      | (home3.php) . . . . .           | 5        |
| 2.10     | (MATCHING USER INPUT) . . . . . | 6        |
| 2.11     | (index4.php) . . . . .          | 6        |
| 2.12     | (home4.php) . . . . .           | 6        |
| <b>3</b> | <b>Conclusion</b>               | <b>8</b> |

# 1 Abstrac

Web programming or internet programming refers to the writing, markup and coding involved in Web development, which includes Web content, Web client and server scripting and network security. The most common languages used for Web programming are XML, HTML, JavaScript, Perl 5 and PHP.

**Introduction** **POST METHOD:** In the PHP POST method, data from HTML FORM is submitted/collected using a super global variable `$_POST`. This method sends the encoded information embedded in the body of the HTTP request and hence the data is not visible in the page URL unlike the GET Method.

**GET METHOD:** PHP `$_GET` is a PHP super global variable which is used to collect form data after submitting an HTML form with `method="get"`. GET can also collect data sent in the URL. Assume we have an HTML page that contains a hyperlink with parameters: `<html>`, `<body>`

**REQUEST METHOD:** PHP `$_REQUEST` is a PHP super global variable which is used to collect data after submitting an HTML form.

## 2 Source Code

### 2.1 (POST Method)

### 2.2 (index.php)

```
1
2 <!DOCTYPE html>
3 <head>
4   <title>
5
6   </title>
7 </head>
8 <body>
9   <form action = "home.php" method = "POST" >
10
11   <input type = "text" name = "username" placeholder = "username" >
12   <input type = "text" name = "password" placeholder = "password" >
13   <input type = "submit" value = "Login" >
14
15
16 </form>
17 </body>
```

?

### 2.3 (home.php)

```
1 <?php
2
3   $user = $_POST ["username"];
4
5   $pass = $_POST ["password"];
6
7   echo $user ;
8   echo " <br > " ;
9   echo $pass ;
10
11 ?>
```

?

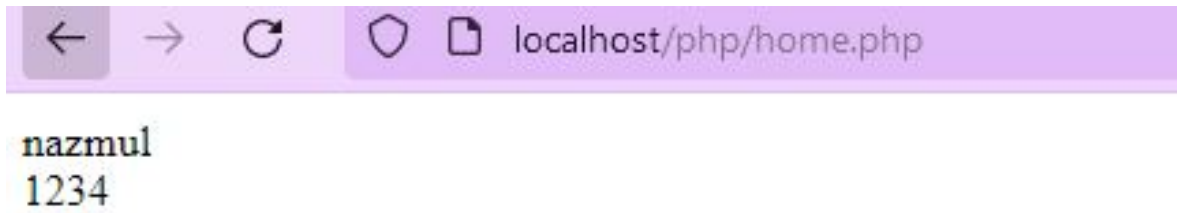


Figure 1: OUTPUT

## 2.4 (GET Method)

### 2.5 (index2.php)

```
1 <!DOCTYPE html>
2 <head>
3   <title>
4
5   </title>
6 </head>
7 <body>
8   <form action = "home.php" method = "GET" >
9
10  <input type = "text" name = "username" placeholder = "username" >
11  <input type = "text" name = "password" placeholder = "password" >
12  <input type = "submit" value = "Login" >
13
14
15
16 </form>
17 </body>
```

?

### 2.6 (home2.php)

```
1 <?php
2
3   $user = $_GET ["username"];
4
5   $pass = $_GET["password"];
6
7   echo $user ;
8   echo " <br > " ;
9   echo $pass ;
10
11 ?>
```

?

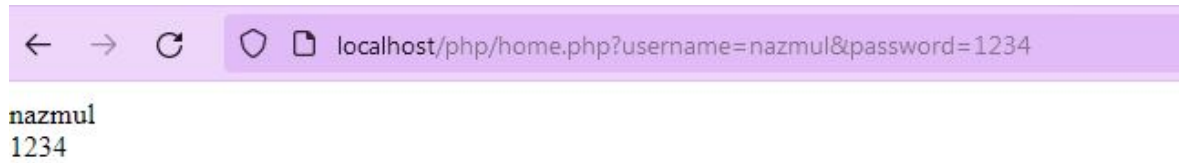


Figure 2: OUTPUT

## 2.7 (REQUEST Method)

## 2.8 (index3.php)

```
1 <!DOCTYPE html>
2 <head>
3   <title>
4
5   </title>
6 </head>
7 <body>
8   <form action = "home.php" method = "POST" >
9
10  <input type = "text" name = "username" placeholder = "username" >
11  <input type = "text" name = "password" placeholder = "password" >
12  <input type = "submit" value = "Login" >
13
14
15
16 </form>
17 </body>
```

?

## 2.9 (home3.php)

```
1 <?php
2
3   $user = $_REQUEST["username"];
4
5   $pass = $_REQUEST ["password"];
6
7   echo $user ;
8   echo " <br > " ;
9   echo $pass ;
10
11 ?>
```

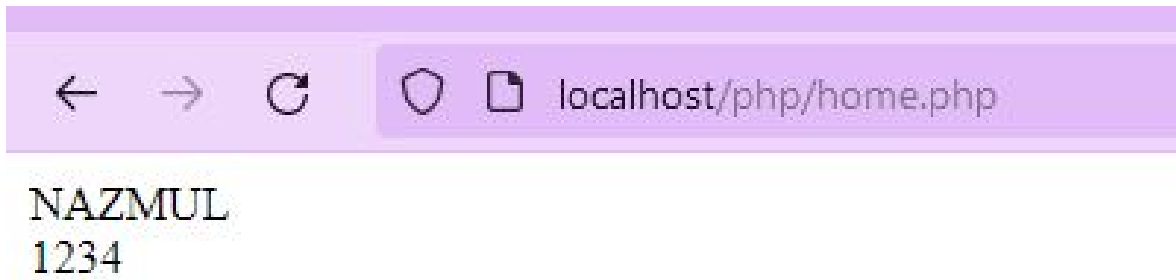


Figure 3: OUTPUT

?

## 2.10 (MATCHING USER INPUT)

## 2.11 (index4.php)

```
1 <!DOCTYPE html>
2 <head>
3 <title>
4
5 </title>
6 </head>
7 <body>
8 <form action = "home4.php" method = "POST" >
9
10 <input type = "text" name = "username" placeholder = "username" >
11 <input type = "text" name = "password" placeholder = "password" >
12 <input type = "submit" value = "Login" >
13
14
15
16 </form>
17 <body>
```

?

## 2.12 (home4.php)

```
1 <?php
2
3 $username = " NAZMUL " ;
4 $id = " 1234 " ;
5
6 $txt = $_POST [ " username " ];
```

```
7  $pass = $_POST [ " password " ];
8
9  if ( $username == $txt && $id == $pass )
10 {
11
12     echo " Login Successful " ;
13 }
14     else
15     echo " Incorrect password try again " ;
16 ? >
?
```

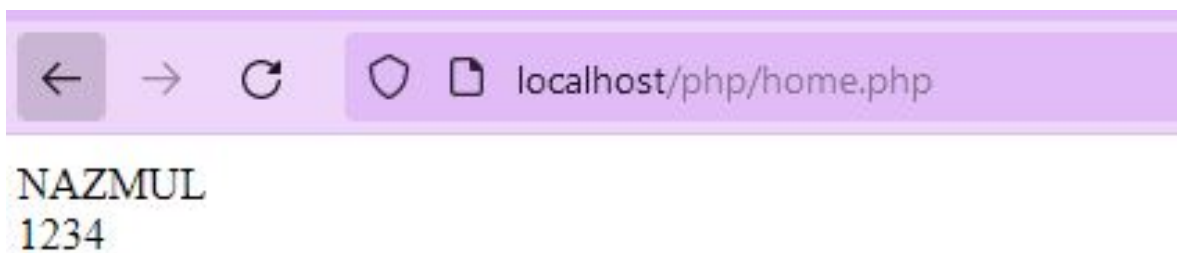


Figure 4: OUTPUT



### **3 Conclusion**

In this lab report we learn about different type of PHP form handling as like (GET,POST.REQUEST) and also learn why we use this different type method. which helped us to implement different place.