CTIS 256 Web Technologies II

Note # 4

Serkan GENÇ

Web Form Processing

- Web Form is a way to gather user inputs for Web applications.
- Web Form is an html form that involves standard html user interface elements, namely, "*textbox*", "*password box*", "*checkbox*", "*radio button*", "*text area*", "*select/list box*", and "*submit button*".
- Web Form elements must be surrounded by **<form>** tag. Form tag has two important attributes; **method** and **action**.

memberForm.php

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Web Form</title>
    </head>
    <body>
        <!-- WEB FORM -->
        <form action="saveMember.php" method="POST">
            <table>
                <tr>
                    <td>Name</td>
                    <td><input type="text" name="mName" /></td>
                </tr>
                <tr>
                    <td colspan="2">
                        <input type="submit" name="saveSubmit" value="Save" />
                    </td>
                </tr>
            </table>
        </form>

    </body>
</html>
```

- *action = "php filename"*, data will be sent to this php file after the user clicks the submit button.
- *method = "POST"*, HTTP packet format when sending the data to the specified php file.

Form Processing Architecture

1 Request ***memberForm.php*** ***with GET method***

Send html form back

memberForm.php

saveMember.php

2 User fills the form and clicks submit button. Submit button triggers a new POST request with the data filled in.

3 Send data (mName=Ali) to ***saveMember.php*** with POST method

4 **PHP Engine** gets POST data from Web server, and creates $_POST, $_GET, $_SERVER super global arrays, and executes ***saveMember.php*** with those arrays.

5 Browser gets the result of saveMember.php and displays on the screen.

saveMember.php ← $_POST, $_GET,…

In the first request, the browser gets the html form, and after the user fills the form and clicks the submit button. In the second request, the browser sends the input data to the specified php program in the html form.

Access data from PHP

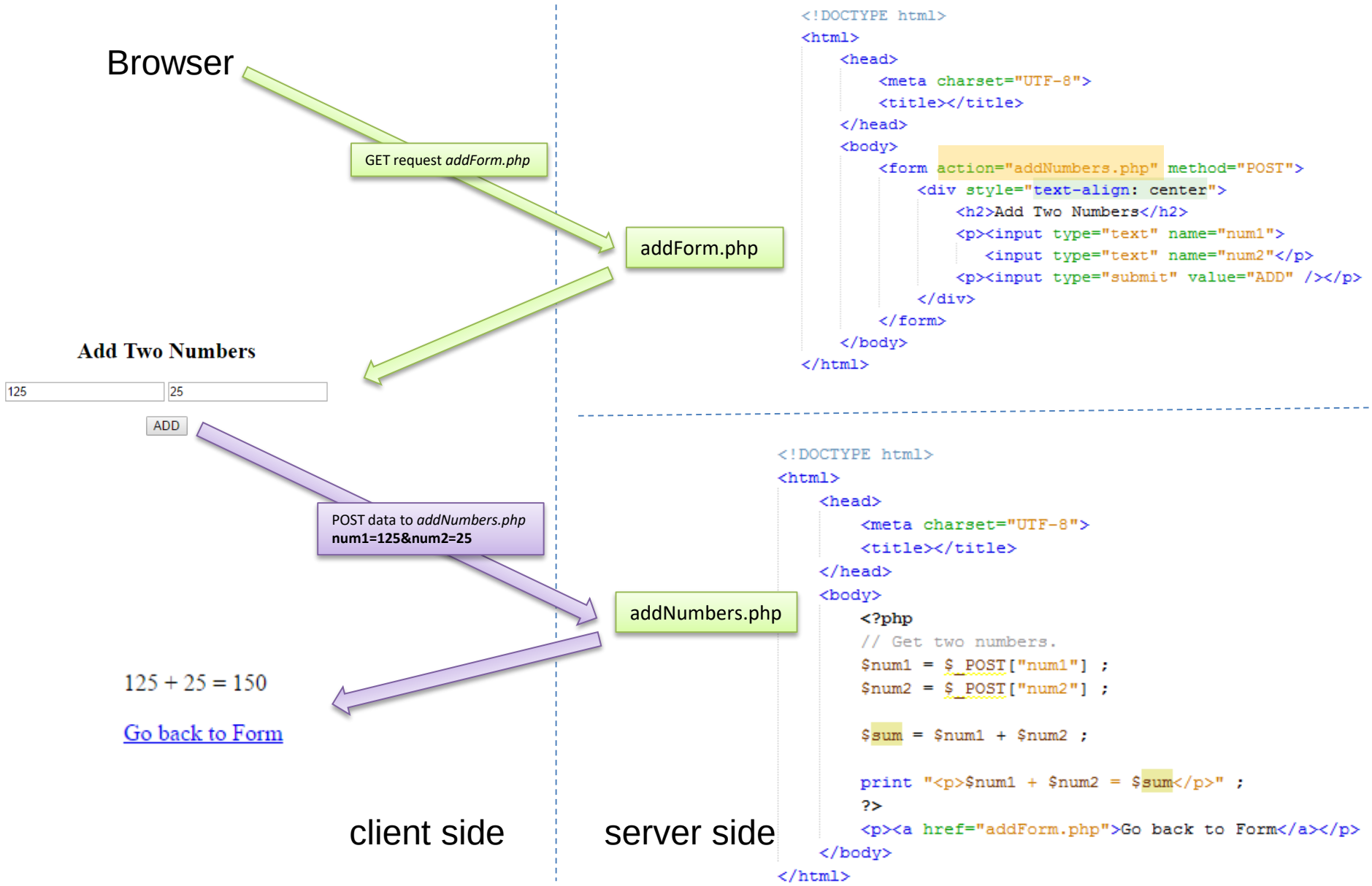
- PHP interpreter prepares $_POST/$_GET array with the data in the html form.
- Interpreter provides $_POST/$_GET array with the specified php file.
- PHP program can access the data through $_POST/$_GET array.
- When accessing a certain data, the name that we used in the form is used as the index to the array.
- For a checkbox, *isset()* function is used to test if it is checked or not.
- $_POST/$_GET data are always in **string** format.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <?php
        // Now, $_POST involves the data in the html form.

        // key is the name used in the html form.
        $userName = $_POST["mName"] ;

        // Here, we simply display what we get from the form.
        // Normally, we save this data into file or mostly in database.
        print "<p>You entered your name as $userName</p>" ;
        ?>
    </body>
</html>
```

bool isset($var) : it checks if the variable exists (assigned to a value) and it is not **null**.

bool empty($var): if the variable is not set, it returns true, and if the variable's value is empty string '' , '0', 0, 0.0, [], null, false, then it returns true.

Adding Two Numbers




```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <form action="addNumbers.php" method="POST">
            <div style="text-align: center">
                <h2>Add Two Numbers</h2>
                <p><input type="text" name="num1">
                   <input type="text" name="num2"></p>
                <p><input type="submit" value="ADD" /></p>
            </div>
        </form>
    </body>
</html>
```

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <?php
        // Get two numbers.
        $num1 = $_POST["num1"] ;
        $num2 = $_POST["num2"] ;

        $sum = $num1 + $num2 ;

        print "<p>$num1 + $num2 = $sum</p>" ;
        ?>
        <p><a href="addForm.php">Go back to Form</a></p>
    </body>
</html>
```

Access data from PHP

UI Type	HTML Code	Access in PHP
textbox	`<input type="text" name="userName" />`	`$name = $_POST["userName"] ;`
password box	`<input type="password" name="userPass" />`	`$pass = $_POST["userPass"] ;`
text area	`<textarea name="comment" cols="30" row="5"></textarea>`	`$comment = $_POST["comment"] ;`

radio button	`<input type="radio" name="company" value="samsung"> Samsung`
	`<input type="radio" name="company" value="sony"> Sony`
	`<input type="radio" name="company" value="lg"> LG`
		`$company = $_POST["company"] ;`

checkbox	`Smoking : <input type="checkbox" name="smoking" />`	`if ( isset($_POST["smoking"])) {`
		`  // checkbox checked.`
		`}`

multiple checkbox	`Languages: <input type="checkbox" name="lang[]" value="C"> C`
	`<input type="checkbox" name="lang[]" value="C++"> C++`
	`<input type="checkbox" name="lang[]" value="Java"> Java`
		`$lang = $_POST["lang"] ;`
		`// $language is an array filled`
		`// by selected languages.`

list box	`<select name="city">`
	`  <option value="0">Eskişehir</option>`
	`  <option value="1">Ankara</option>`
	`  <option value="2">Antalya</option>`
	`</select>`
		`$city = $_POST["city"] ;`

hidden	`<input type="hidden" name="formId" value="f344534567-123" />`	`$formID = $_POST["formId"] ;`

submit button	`<input type="submit" name="saveButton" value="Save" />`	`if ( isset($_POST["saveButton"])) {`
		`  // save submit button clicked.`
		`  // to test if there is more`
		`  // than one form`
		`}`

Form Example

Develop a form application consisting of two files; *q1_form.php* and *q1_process.php.* q1_form.php displays a form interface for a hotel reservation system. As you can see in Figure 1, a client selects the season for the holiday, and type of reservation **(Full Board**, and **Half Board**), number of person who will stay, and flight ticket. After clicking on "Reserve" submit button. Q1_process.php gets all data submitted by the client, and calculates the price**. Full Board costs 1000 TL, and Half Board costs 700** TL. If the client selects hot season, that is, June-August, he **pays 1.5 times more** than the price above. If the clients selects flight ticket, it adds 150 TL for each person. Q1.php displays the results as in Figure 2.

Figure 1

q1_form.php

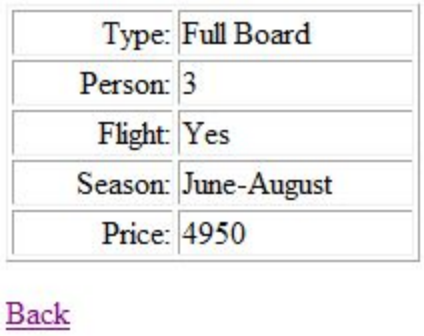
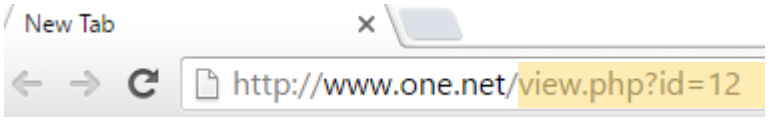


Figure 2

q1_process.php

Data with GET Method

- GET http packet is generated from location bar, links and forms with GET method.
- GET is used to request a resource (html, php, image, etc) from the server.
- If the resource is PHP, it means "Run the program and send me the output".
- With GET request, you can still send data/parameter through URL, called Query String.
- Values and variable names should be converted to urlencoded format using ***urlencode()*** function.
- When you click a link, browser generates a GET request. A link can involve data.
- Access data using $_GET array in php program.



Run *view.php* with id = 12, and send me the result.

Query String Format

?var1=value1&var2=value2&var3=value3

Links

```
<h2>MEMBERS</h2>
<p>
    <a href="viewMember.php?userID=72979"> Ali Gül </a>
</p>
```

MEMBERS

Ali Gül

click

```
<body>
    <?php
    $database = array(
        '72979' => array( 'name'=>'ali', 'surname' => 'gul',
                'city' => 'Ankara', 'mobil' => '+903124872977'
            )
    );
    $user = $_GET["userID"];
    var_dump( $database[$user]) ;
    ?>
</body>
```

urlencoded string

- In a URL, the query string should be converted to urlencoded format.
- It is also used in the preparation of data of the "**application/x-www-form-urlencoded**" media type, as is often used in the submission of HTML form data in HTTP POST/GET requests. PHP Engine decodes urlencoded string and put them into $_POST or $_GET super global arrays accordingly.
- string **urlencode**($input), string **urldecode**($encoded)
- In this conversion:
  - Space character is converted to + or %20.
  - Reserved ASCII characters ! * ' ( ) ; : @ & = + 4 , / ? # [ ] are converted to %hh where hh are two hexadecimal digits for its ASCII code.
  - Unreserved ASCII characters are used as they are.
  - Example: "ali kemal**&**ece" => "ali**+**kemal**%26**ece"
  - Other characters such Turkish characters (İ, Ç, ö, ğ, ü ,etc) are converted to UTF8 percentage coding.
  - Example: "Çağıl" => "%C3%87a%C4%9F%C4%B1l"
  - http://www.one.net/test/list.php?data=ali+Kemal%26ece&name= %C3%87a%C4%9F%C4%B1l