Chapter 3-part 1:Form Handling

HTML Form: POST Method
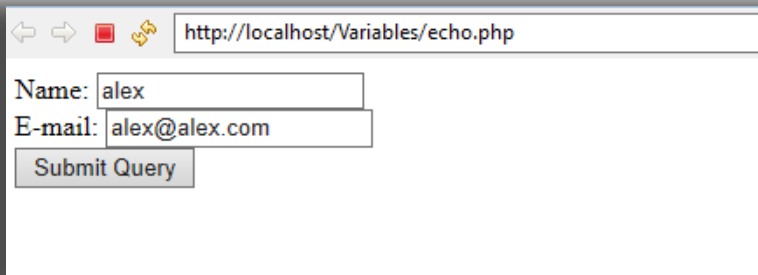
When the user fills out the form and clicks the submit button, the form data is sent for processing to a PHP file named "welcome.php".

```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>
```

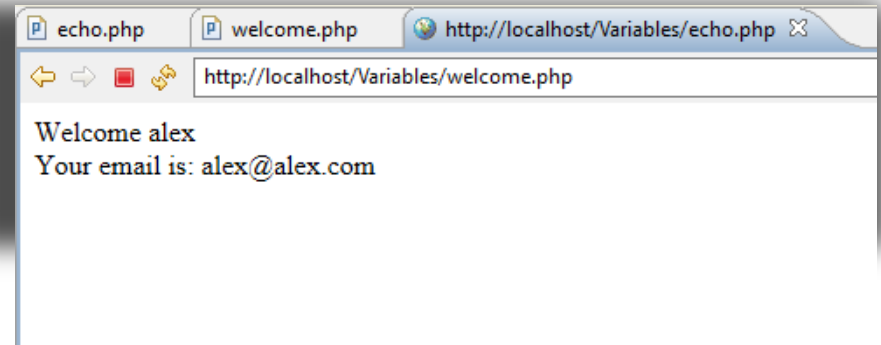# PHP File : $_POST [...]

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

The output could be something like this:

http://localhost/Variables/echo.php

Name: alex
E-mail: alex@alex.com
Submit Query

P echo.php   P welcome.php   http://localhost/Variables/echo.php

http://localhost/Variables/welcome.php

Welcome alex
Your email is: alex@alex.com

HTML Form: GET Method

Same example as the previous form using a get method
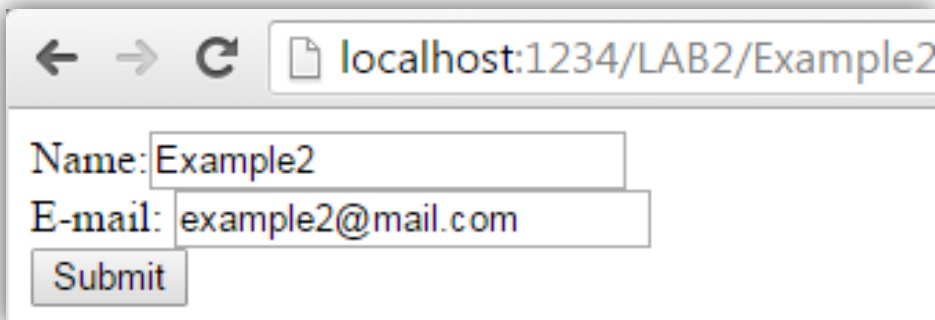
```
<html>
<body>
<form action="welcome.php" method="get">
Name:  <input type="text" name="name"><br/>
E-mail: <input type="text" name="email"><br/>
        <input type="submit">
</form>
</body>
</html>
```

PHP File : $_GET [...]

```html
<html>
<body>
        Welcome <?php echo $_GET["name"]; ?><br/>
        Your email is: <?php echo $_GET["email"];  ?>
</body>
</html>
```
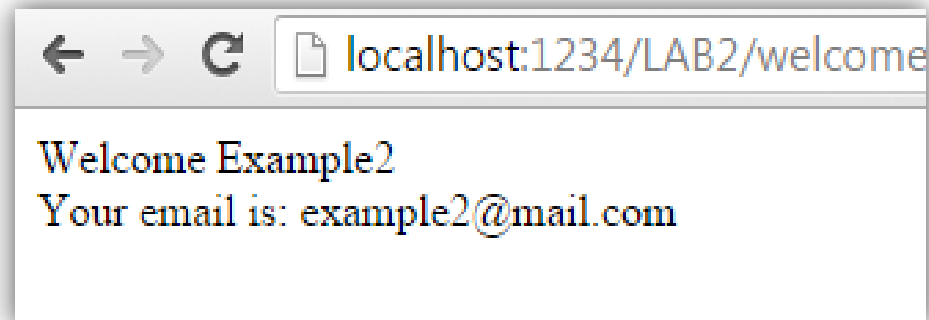
The output could be something like this:

localhost:1234/LAB2/Example2

Name: Example2
E-mail: example2@mail.com
Submit

localhost:1234/LAB2/welcome

Welcome Example2
Your email is: example2@mail.com

GET vs. POST

Both **GET** and **POST** create an **array that holds key/value pairs**, where keys are the names of the form controls and values are the input data from the user.

Both **GET** and **POST** are treated as **$_GET** and **$_POST**, which are superglobals.

$_GET is an array of variables passed to the current script via the URL parameters.

$_POST is an array of variables passed to the current script via the HTTP POST method.

When to use GET?

Information **sent from a form** with the **GET method** is **visible to everyone** (all variable names and values are displayed in the URL).

GET also has a limitation of 2000 characters to send.

GET may be used for sending non-sensitive data.

Note: GET should NEVER be used for sending passwords or other sensitive information!

When to use POST?

Information sent from a form with the POST method is **invisible to others** (all names/values are embedded within the body of the HTTP request)

POST has **no limits** on the amount of information to send.

POST supports **advanced functionality** such as support for multi-part binary input while uploading files to server.

Developers prefer POST for sending form data.

HTML Form Elements
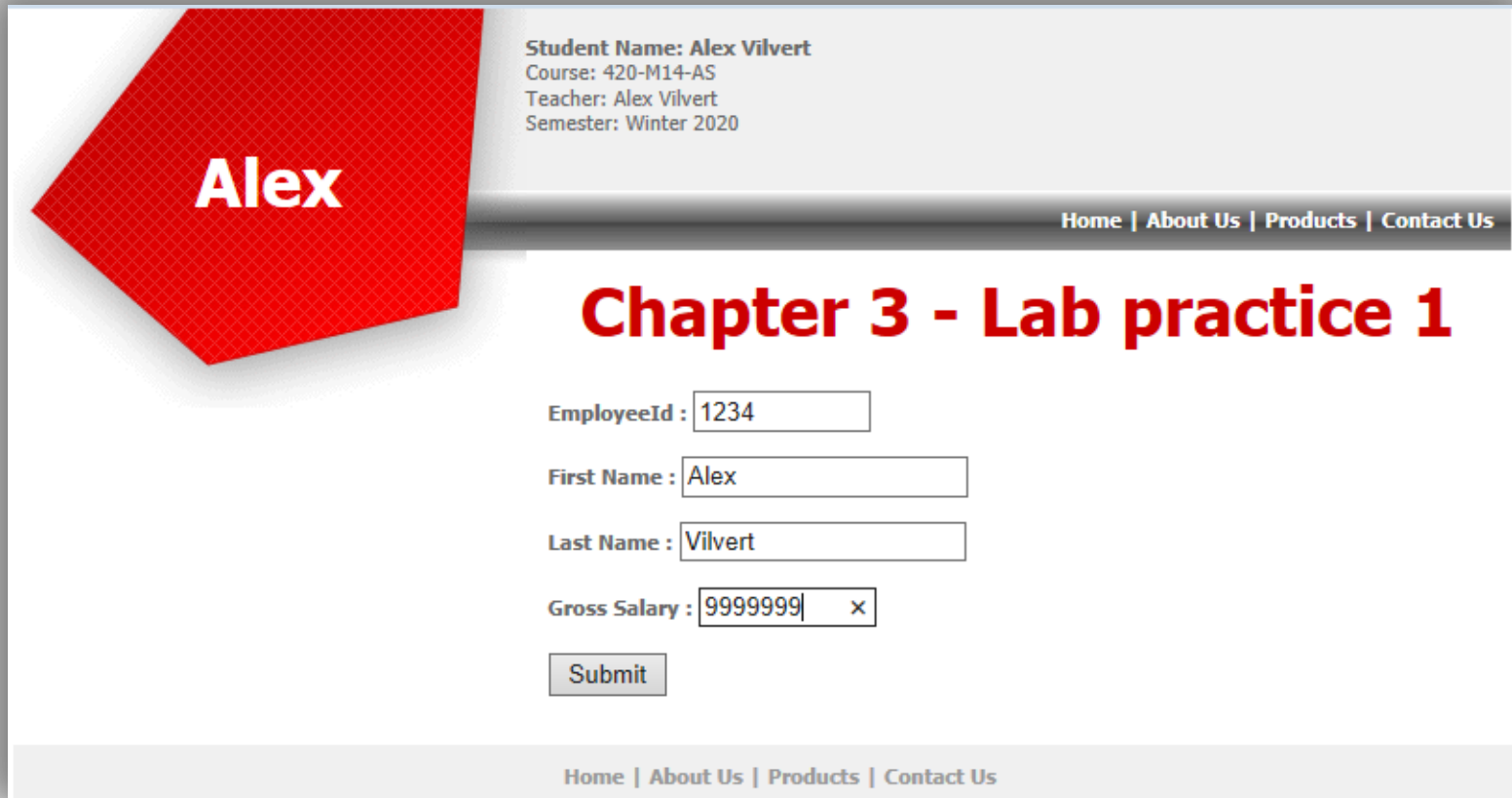
HTML Tag	Definition
<input>	specifies an input field, allows user to enter data.
<textarea>	defines a multi-line text input control.
<button>	defines a clickable button.
<select>	is used to create a drop-down list.
<option>	defines an option in a select list.
<fieldset>	used to group related elements in a form.
<label>	defines a label for an <input> element.

Lab

Exercise 1: Form (Data Entry)

Modify the file **Template.php** to include a **Form** that allows the users to enter the following data:

Exercise 1: Form (Data Display)

After submitting the Form, the data is displayed on another page.
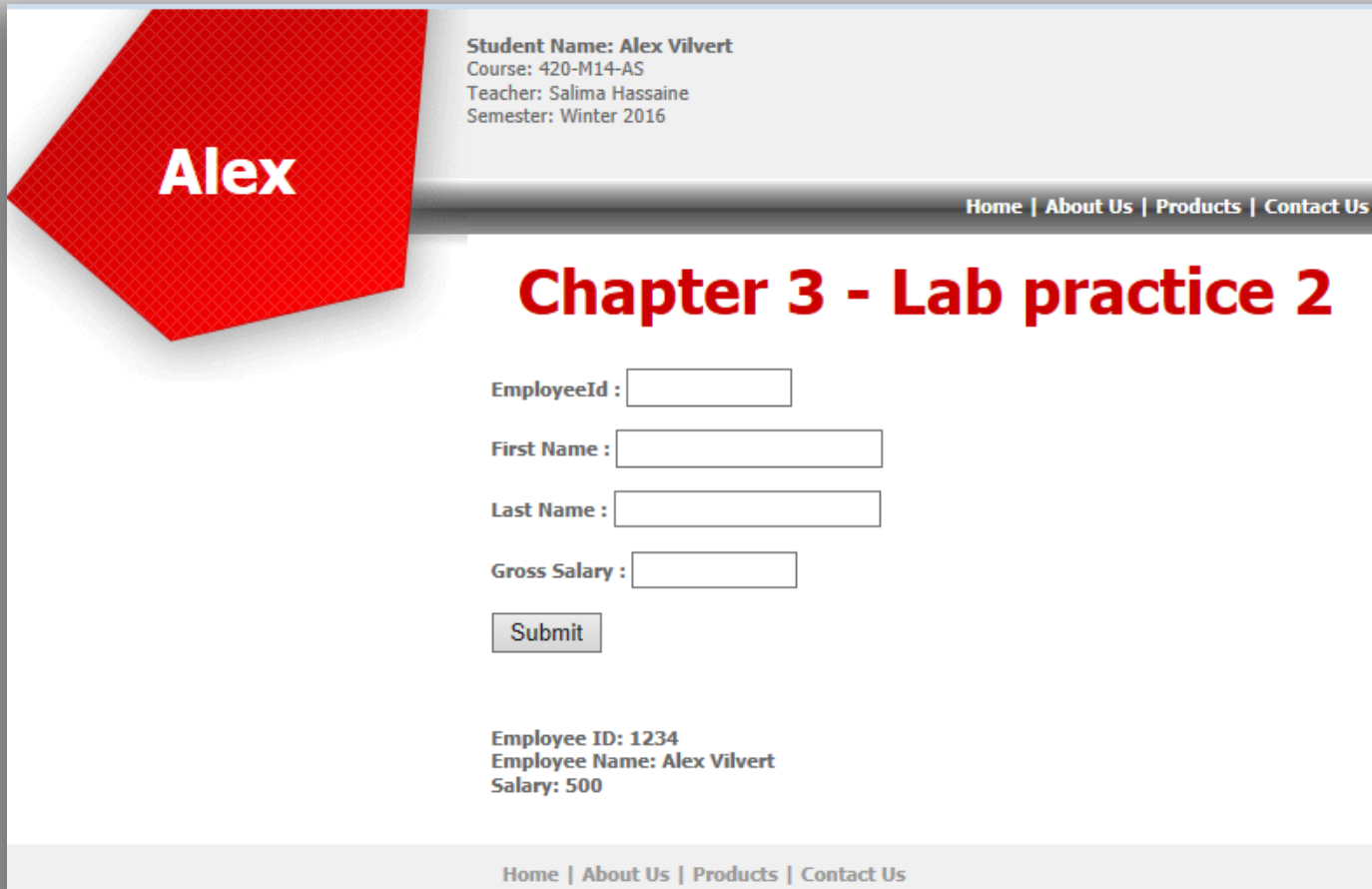
Exercise 2: Form (Data Display)

In this case the data is displayed on the same page.