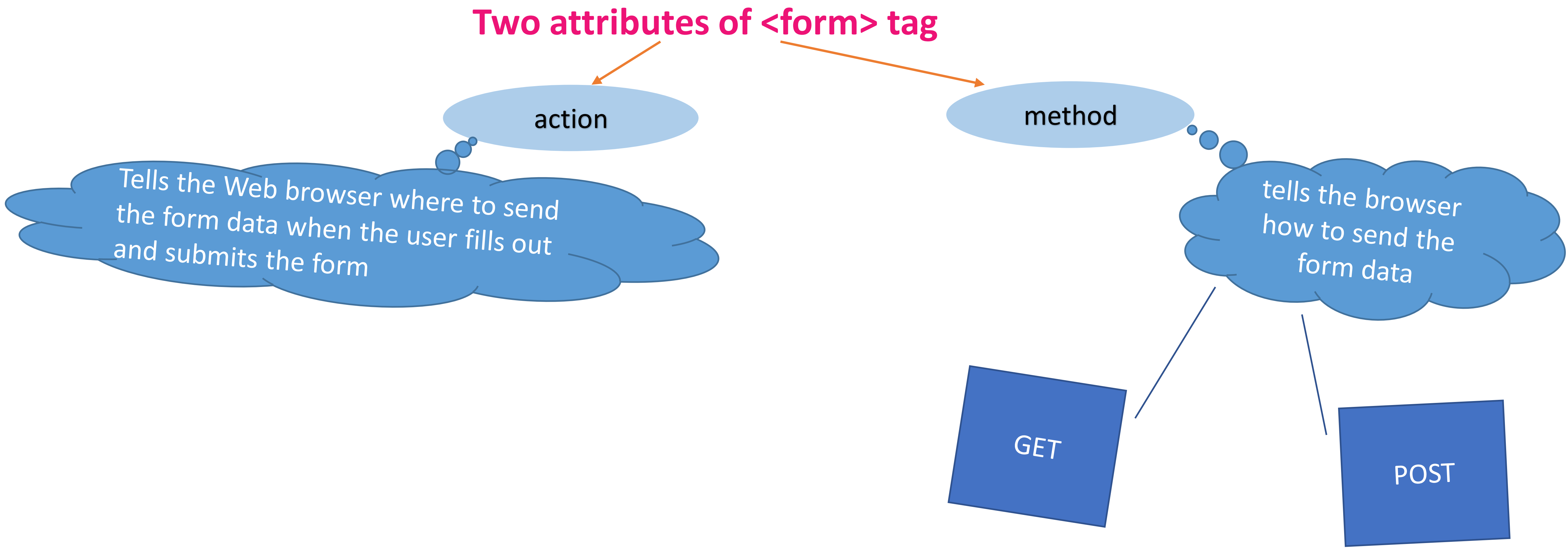
How HTML Form works?

All Web forms start with an opening **< form >** tag, and end with a closing **< /form >** tag


Two attributes of <form> tag
action
Tells the Web browser where to send the form data when the user fills out and submits the form
method
tells the browser how to send the form data
GET
POST

The GET Method

```
http://www.test.com/index.htm?name1=value1&name2=value2
```

- The GET method produces a long string that appears in your server logs, in the browser's Location: box.
- The GET method is restricted to send upto 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.
- The PHP provides **$_GET** associative array to access all the sent information using GET method.

The POST Method

The POST method transfers information via HTTP headers. The information is encoded put into a header.

- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol.
- By using Secure HTTP you can make sure that your information is secure.
- The PHP provides **$_POST** associative array to access all the sent information using POST method.

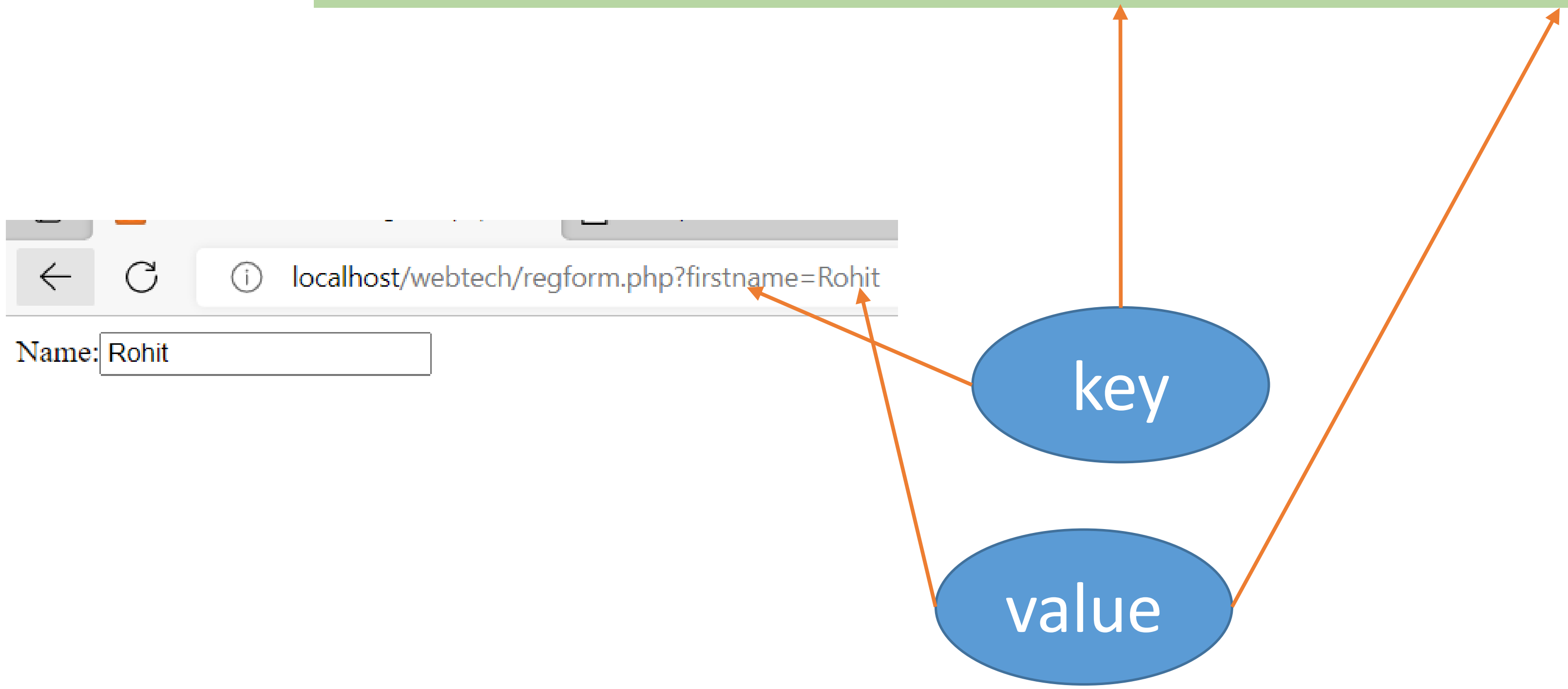
Capturing form data with PHP

Superglobal array	Description
$_GET	Contains a list of all the field names and values sent by a form using the get method
$_POST	Contains a list of all the field names and values sent by a form using the post method
$_REQUEST	Contains the values of both the **$_GET** and **$_POST** arrays combined, along with the values of the $_COOKIE superglobal array

Capturing form data with PHP

Each of these three superglobal arrays contains the **field names** from the sent form as **array keys**, with the **field values** themselves as **array values**.

```
Name: < input type=”text ” name=”firstname”  value=“Rohit”/ >
```



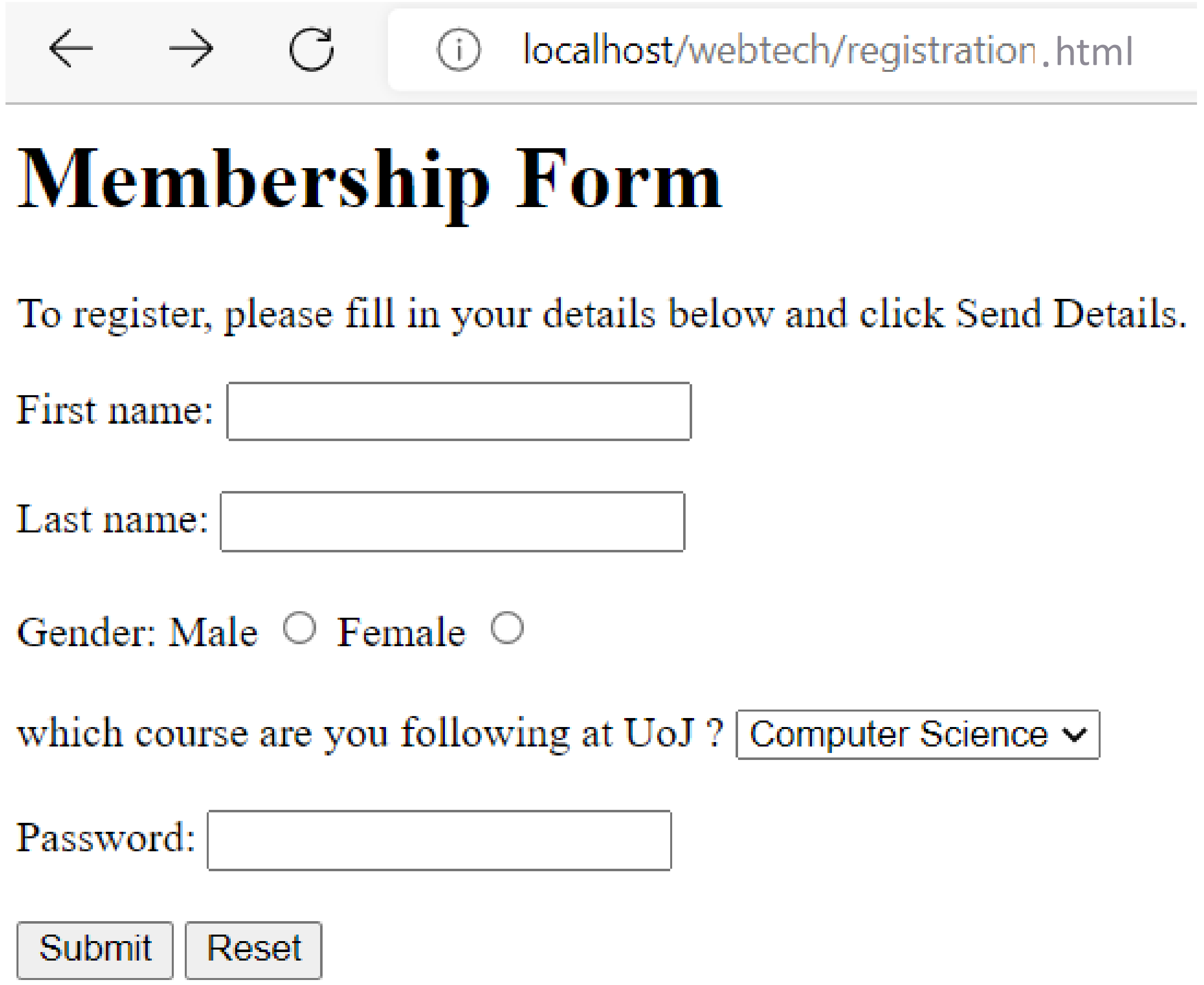
localhost/webtech/regform.php?firstname=Rohit

Name: Rohit

Capturing form data with PHP

Task 01:

- Create a file named “registration.html” with the following input fields.



Membership Form

To register, please fill in your details below and click Send Details.

First name:

Last name:

Gender: Male ○ Female ○

which course are you following at UoJ ? Computer Science

Password:

Capturing form data with PHP

Task 02:

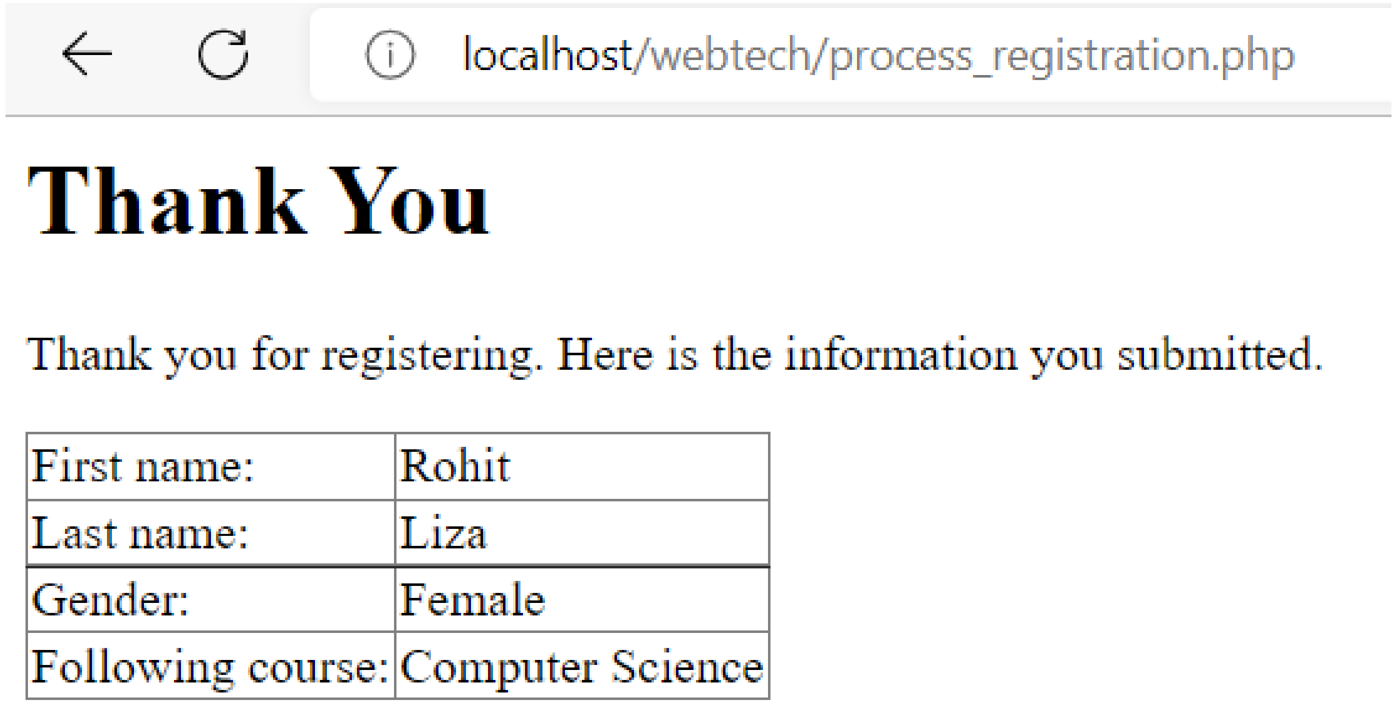
- Save the following script as process_registration.php in your document root (the folder where you placed registration.html) and get the details from registration.html and display

```
<html>
          <head>
                    <title>Thank You</title>
          </head>
          <body>
                    <h1>Thank You</h1>
                    <p>Thank you for registering. Here is the information you submitted.</p>
          <table border=1; style="border-collapse:collapse;">
                    <tr><td>First name: </td><td><!-- echo first name --></td></tr>
                    <tr><td>Last name: </td><td><!-- echo last name--></td></tr>
                    <tr><td>Gender: </td><td><!-- echo male or female --></td></tr>
                    <tr><td>Following course: </td><td><!-- echo course --></td></tr>
          </table>
          </body>
</html>
```

Capturing form data with PHP

Task 03:

- Open the registration.html URL in your Web browser.
- Fill in the fields in the form, then click the Send Details button.
- If all goes well, you should see a page displaying the data that you just entered



localhost/webtech/process_registration.php

Thank You

Thank you for registering. Here is the information you submitted.

First name:	Rohit
Last name:	Liza
Gender:	Female
Following course:	Computer Science

Capturing form data with PHP

Dealing Securely with form data

- Display the password that the users entered is not right way
- Bad idea to pass any user - entered data — such as the values in $_GET and $_POST straight through to a statement like echo() or print() for displaying in a Web page. You should never trust user input on a public Web site; a malicious user might be trying to break into the site.
- You should always validate (that is, check) or filter user input to make sure it ’ s safe before you display it in a Web page

Capturing form data with PHP

Handling empty form fields

- **Try:**

```
<?php
                        First Name: echo $_POST["firstNames"];
?>
```

Capturing form data with PHP

Handling empty form fields

- **Try:**

```
<?php
                        echo $_POST["firstNames"];
?>
```

Warning: Undefined array key “firstNames"
in **C:\xampp\htdocs\webtech\process_registration.php**
on line **10**

Capturing form data with PHP

Handling empty form fields

You can do this using PHP functions such as **isset()** or **array_key_exists()**

```
<?php
		if(isset($_POST["firstName"]))
				echo $_POST["firstName"];
?>
```

To check whether a variable is empty, you could PHP function : **empty()**