Website Design and Development

PHP

HTU.
جامعة الحسين التقنية
Al Hussein Technical University

Prepared by Dr. Raneem and Ms. Hana AlRasheed

Super global variables

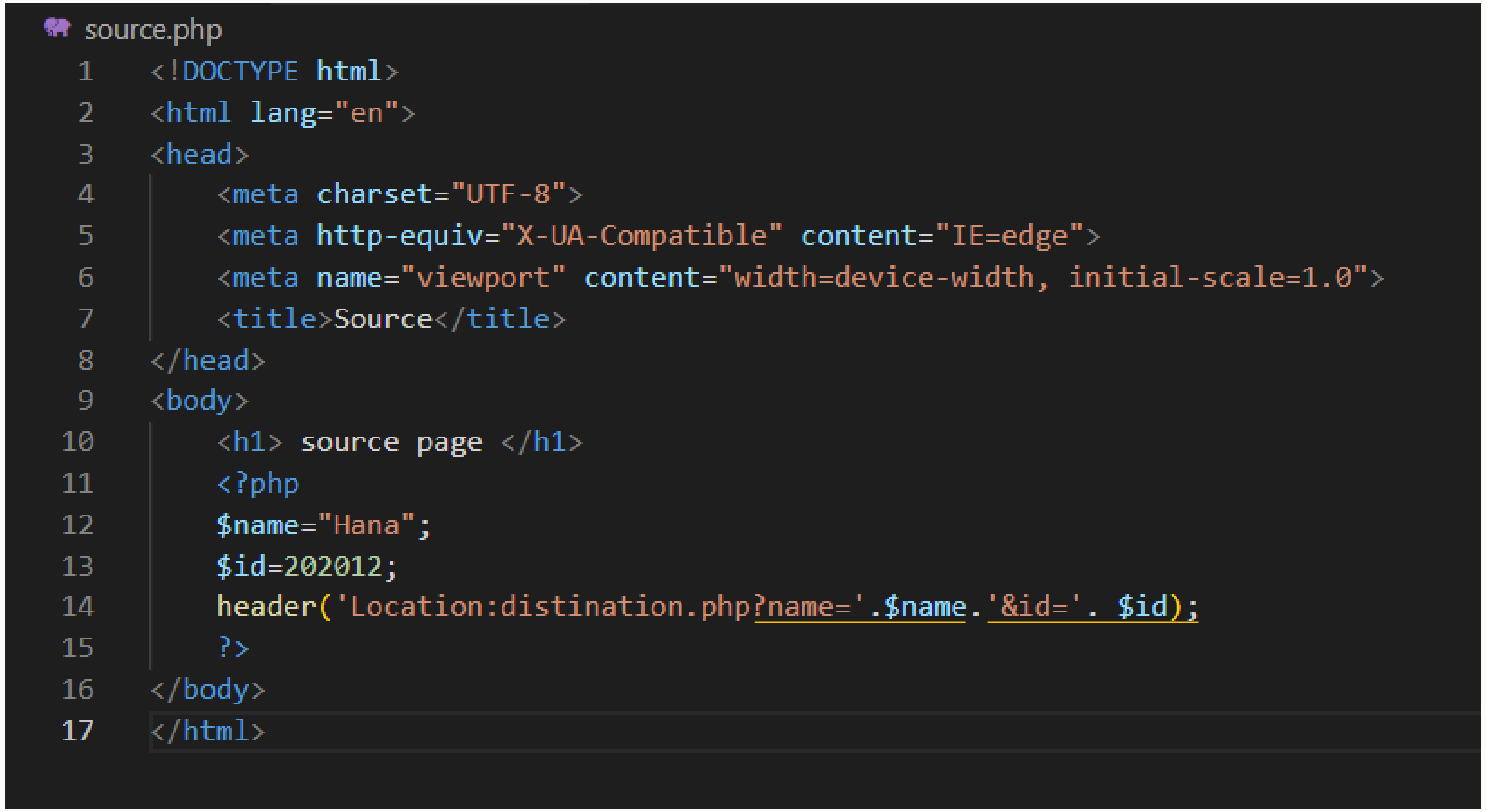
- Super global variables are built-in variables that are always available in all scopes.
- Super global variables create an array (e.g. array(key => value, key2 => value2, key3 => value3, ...)). This array holds key/value pairs.

Passing variables between pages using Super global variables

- $_GET['varname’] - When the variable is passed from HTML form with “GET” method, or passed in the header location URL
- $_POST['varname']: When the variable is passed from HTML form with “POST” method.
- $_SESSION['varname']: When the variable has been assigned the value from a particular session
- $_COOKIE['varname']: When the variable has been assigned a value from a cookie
- $_REQUEST['varname’]: when the variable is passed from “POST” or “GET” method
- $_FILES['varname']: When the variable is been passed from an uploaded file.

HTU
جامعة الحسين التقنية
Al Hussein Technical University

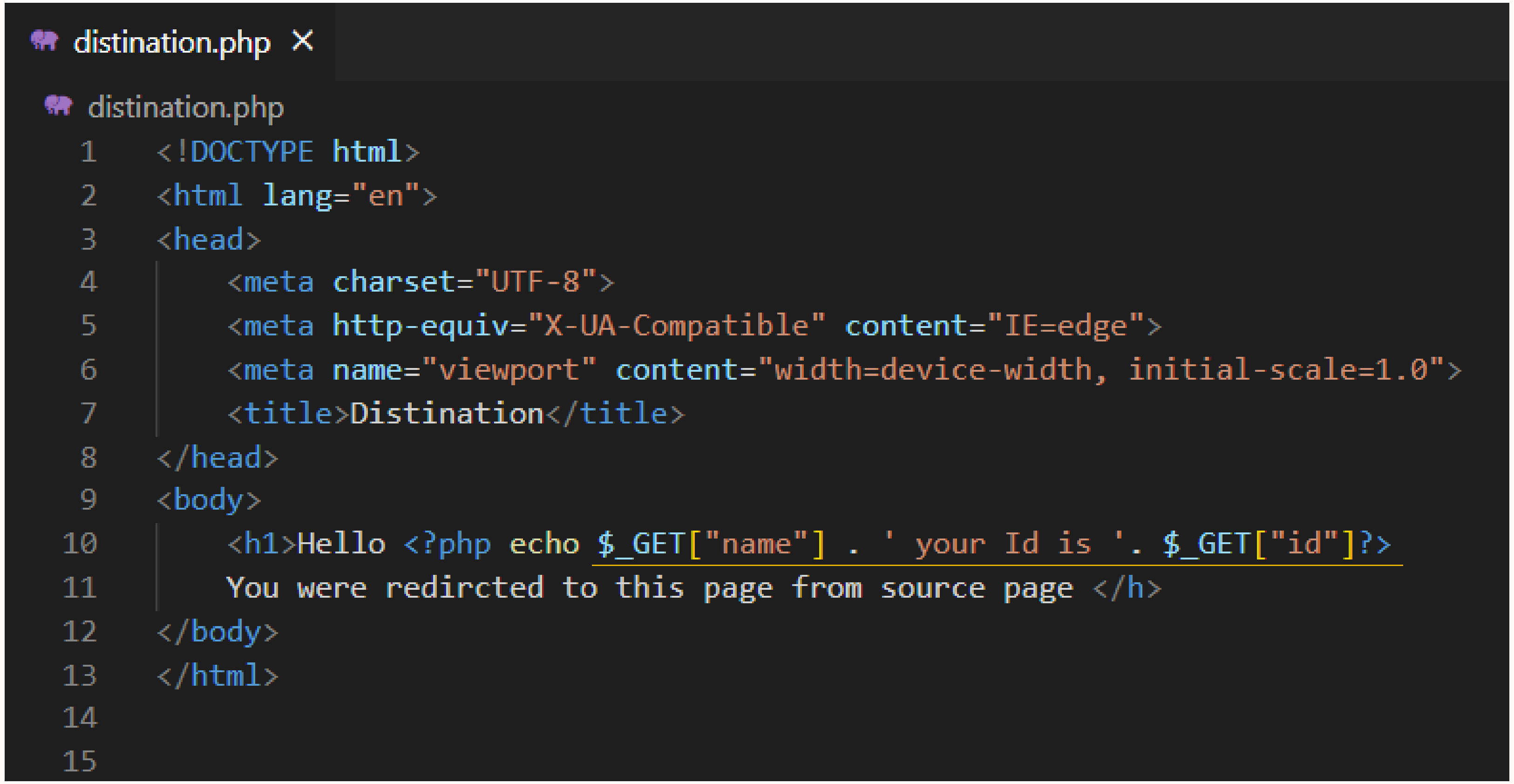
1- pass variable in header URL



source.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Source</title>
</head>
<body>
    <h1> source page </h1>
    <?php
    $name="Hana";
    $id=202012;
    header('Location:distination.php?name='.$name.'&id='. $id);
    ?>
</body>
</html>
```

2- retrieve variable using $_GET global variable

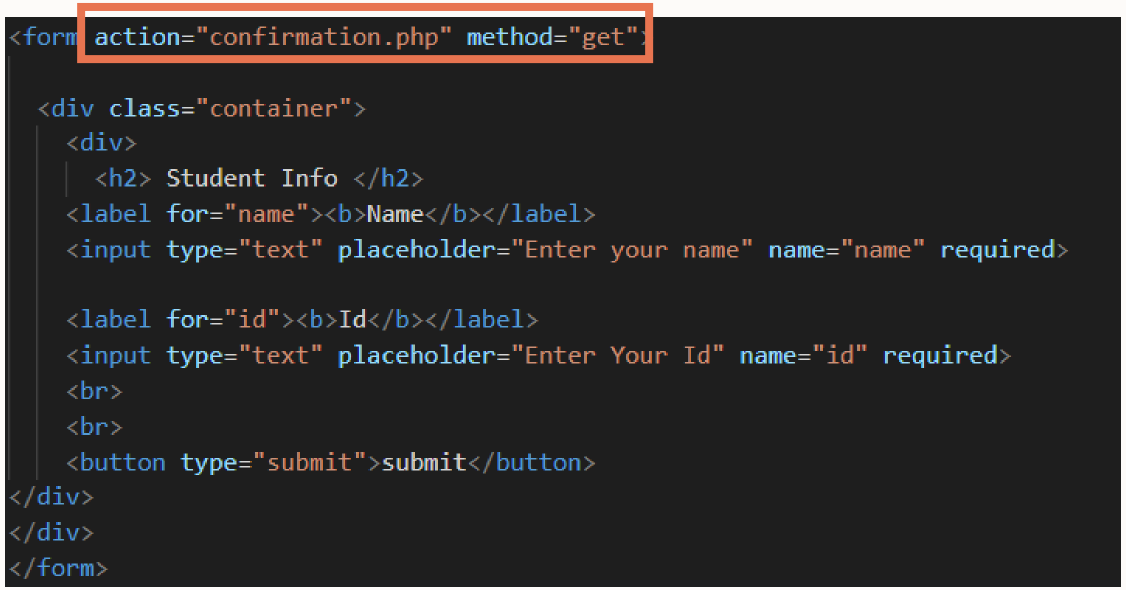


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Distination</title>
</head>
<body>
    <h1>Hello <?php echo $_GET["name"] . ' your Id is '. $_GET["id"]?>
    You were redircted to this page from source page </h>
</body>
</html>


```

Working with forms using get method

- $_GET is an array of variables passed to the current script via the URL parameters.
- GET creates an array (e.g. array(key => value, 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.

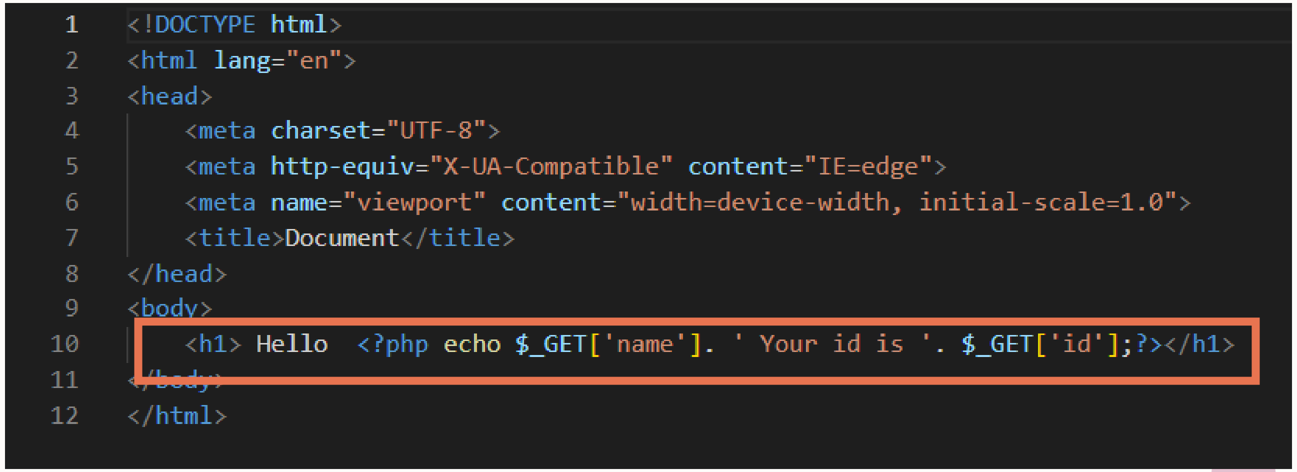


```
<form action="confirmation.php" method="get">

  <div class="container">
    <div>
      <h2> Student Info </h2>
    <label for="name"><b>Name</b></label>
    <input type="text" placeholder="Enter your name" name="name" required>

    <label for="id"><b>Id</b></label>
    <input type="text" placeholder="Enter Your Id" name="id" required>
    <br>
    <br>
    <button type="submit">submit</button>
</div>
</div>
</form>
```

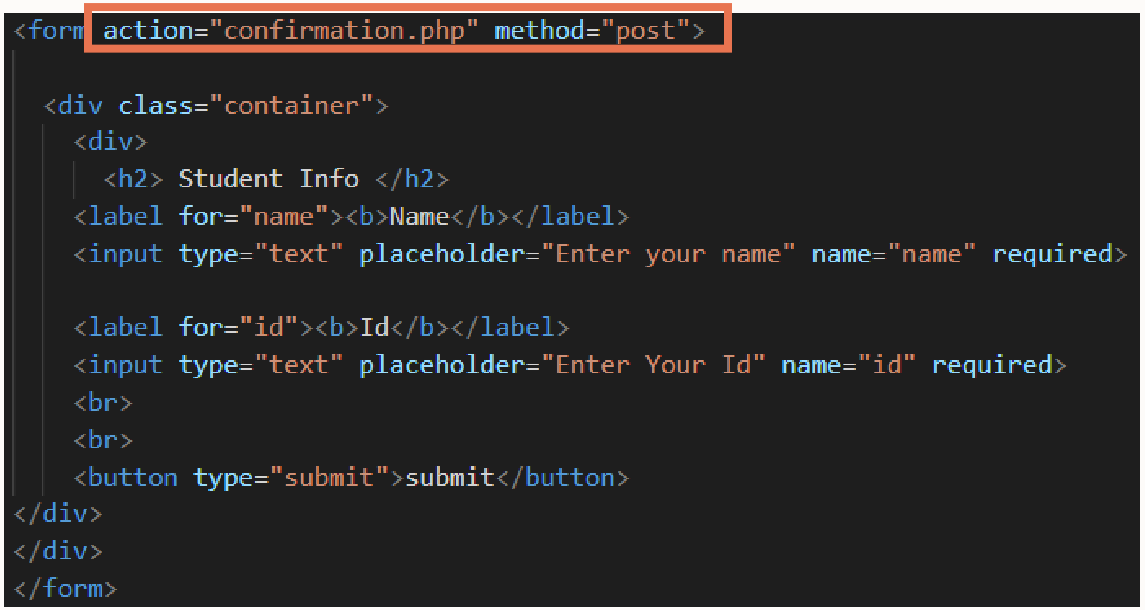
Confirmation.php



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1> Hello  <?php echo $_GET['name']. ' Your id is '. $_GET['id'];?></h1>
</body>
</html>
```

Working with forms using post method

- $_POST is an array of variables passed to the current script via the HTTP POST method.
- POST creates an array (e.g. array(key => value, 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.

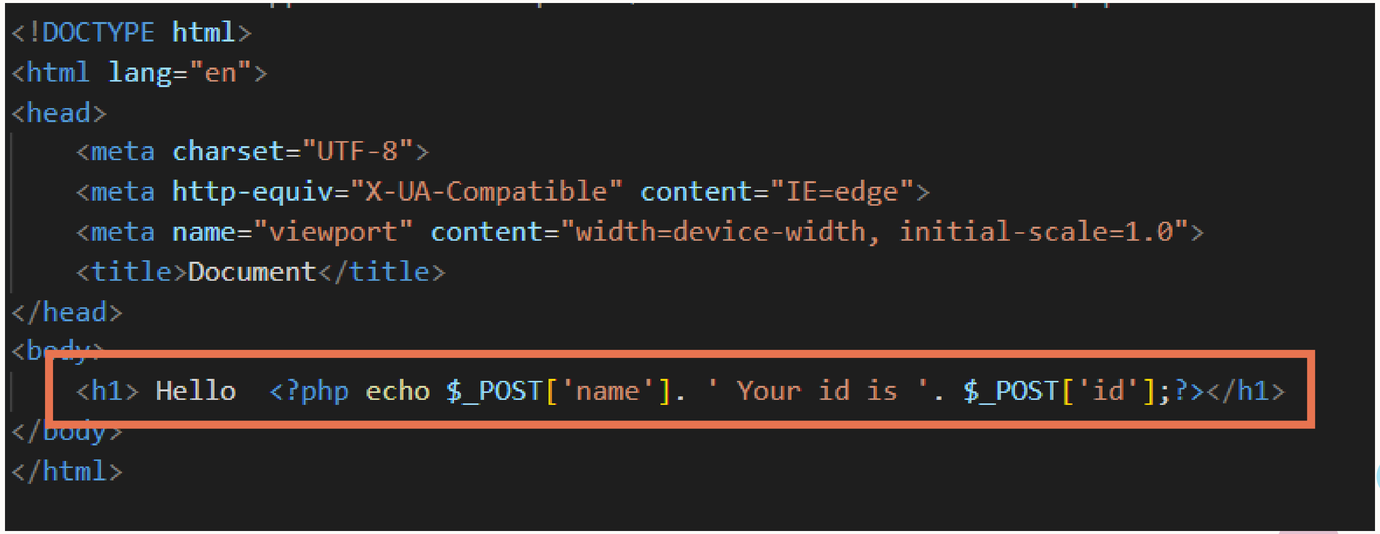


```
<form action="confirmation.php" method="post">

  <div class="container">
    <div>
      <h2> Student Info </h2>
    <label for="name"><b>Name</b></label>
    <input type="text" placeholder="Enter your name" name="name" required>

    <label for="id"><b>Id</b></label>
    <input type="text" placeholder="Enter Your Id" name="id" required>
    <br>
    <br>
    <button type="submit">submit</button>
</div>
</div>
</form>
```

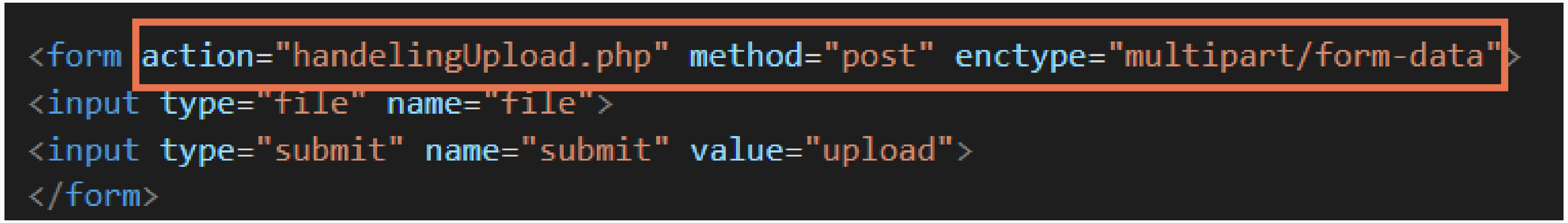
Confirmation.php



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1> Hello  <?php echo $_POST['name']. ' Your id is '. $_POST['id'];?></h1>
</body>
</html>
```

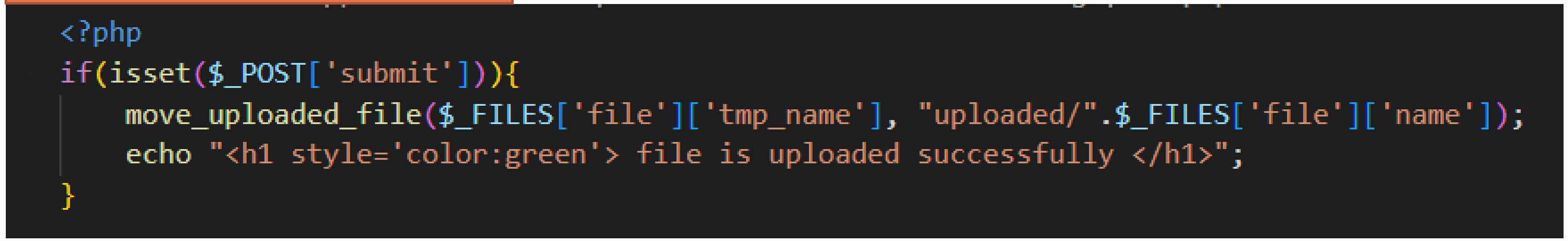
PHP File Upload

- Make sure that the form uses method="post"
- The form also needs the following attribute and value: enctype="multipart/form-data". This attribute specifies which content-type to use when submitting the form
- The input tag with type="file" attribute shows the input field as a file-select control, with a "Browse" button next to the input control.



```
<form action="handelingUpload.php" method="post" enctype="multipart/form-data">
<input type="file" name="file">
<input type="submit" name="submit" value="upload">
</form>
```

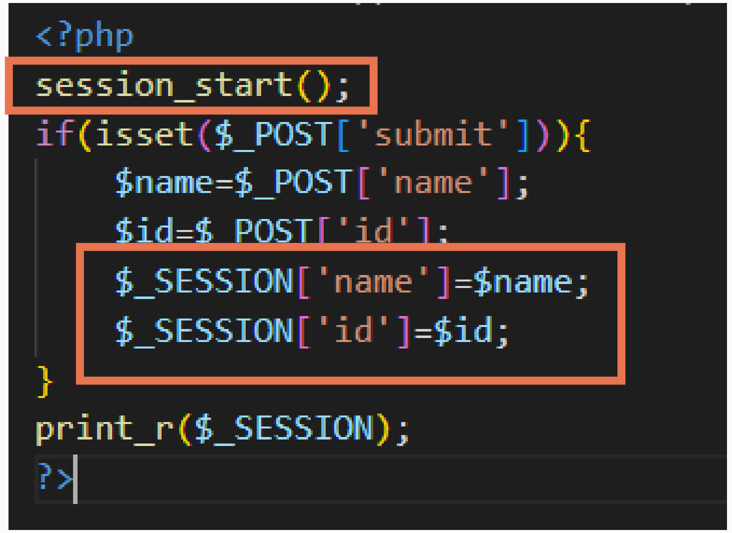
handelingUpload.php



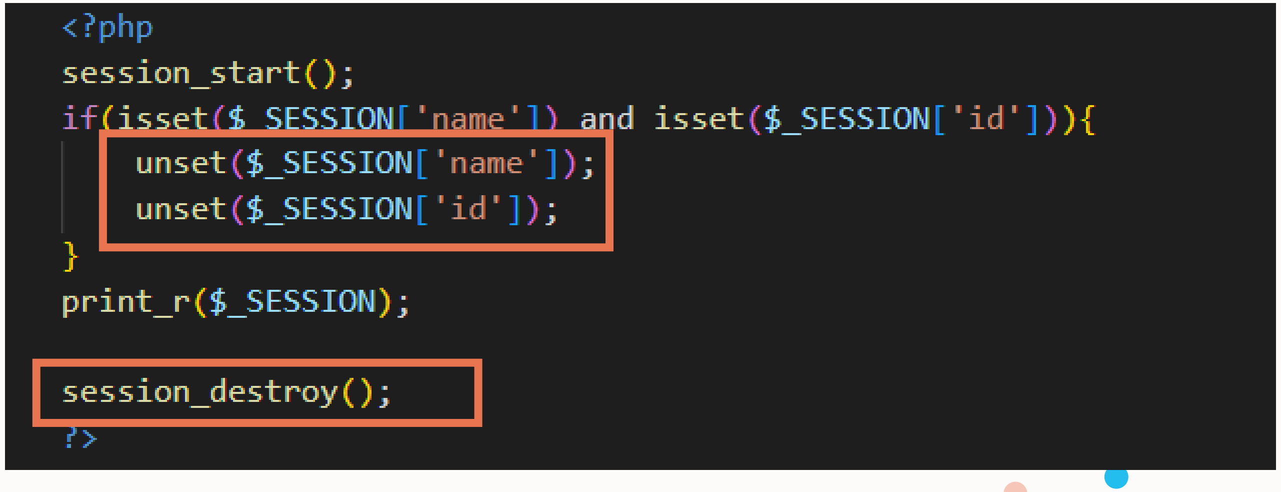
```
<?php
if(isset($_POST['submit'])){
    move_uploaded_file($_FILES['file']['tmp_name'], "uploaded/".$_FILES['file']['name']);
    echo "<h1 style='color:green'> file is uploaded successfully </h1>";
}
```

Working with PHP $_SESSION

- A session is a way to store information (in variables) to be used across multiple pages.
- Session variables hold information about one single user, and are available to all pages in one application.
- By default, session data is stored in files on the server.
- By default, session variables last until the user closes the browser.
- The session_start() function must be the very first thing in your document before any HTML tags.



```
<?php
session_start();
if(isset($_POST['submit'])){
    $name=$_POST['name'];
    $id=$_POST['id'];
    $_SESSION['name']=$name;
    $_SESSION['id']=$id;
}
print_r($_SESSION);
?>
```

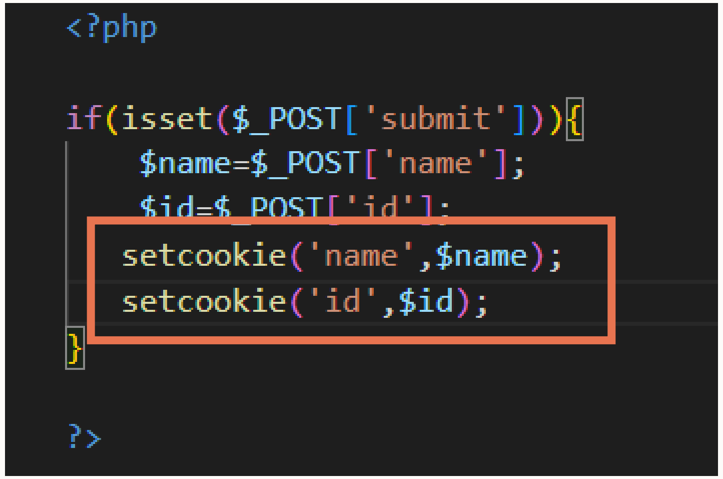


```
<?php
session_start();
if(isset($_SESSION['name']) and isset($_SESSION['id'])){
    unset($_SESSION['name']);
    unset($_SESSION['id']);
}
print_r($_SESSION);

session_destroy();
?>
```

Working with PHP $_COOKIE

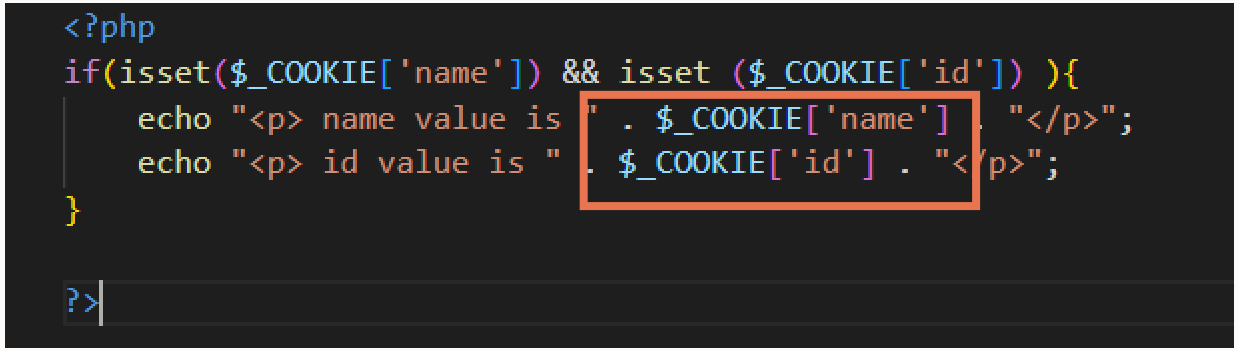
- A cookie is a file that is embedded on the user's computer.
- Each time the same computer requests a page with a browser, it will send the cookie too.
- A cookie is created with the setcookie() function.
- We then retrieve the value of the cookie "user" using the global variable $_COOKIE



```php
<?php

if(isset($_POST['submit'])){
    $name=$_POST['name'];
    $id=$_POST['id'];
   setcookie('name',$name);
   setcookie('id',$id);
}

?>
```



```php
<?php
if(isset($_COOKIE['name']) && isset ($_COOKIE['id']) ){
    echo "<p> name value is " . $_COOKIE['name'] . "</p>";
    echo "<p> id value is " . $_COOKIE['id'] . "</p>";
}

?>
```

In the Videos, we will elaborate examples on different ways for passing variables between pages.

Thank you

HTU.
جامعة الحسين التقنية
Al Hussein Technical University

Prepared by Dr. Raneem and Ms. Hana AlRasheed