



Topic

- The concept of Form Upload with PHP and JQuery.
- Introduction to the concept of cookies.
- Introduction to session concepts.

Objectives

Students are expected to:

1. Students are able to create upload forms using PHP and jQuery.
2. Understand the basic concepts of cookies and its use.
3. Understand the basic concept of session and its use.

Introduction

File Upload

File upload in PHP is the process of uploading files from a user's computer to a web server. File uploads are useful in many cases such as when users want to upload a profile picture, document, or other media to a website. To do this, it can use the HTML element `<input type="file">` along with PHP to manage the upload process.

Here are the general steps to implement file uploads in PHP:

- Create an HTML form to upload a file:

```
<!DOCTYPE html>
<html>
<head>
  <title>File Upload</title>
</head>
<body>
  <form action="upload.php" method="post" enctype="multipart/form-data">
    <input type="file" name="fileToUpload" id="fileToUpload">
    <input type="submit" value="Upload File" name="submit">
  </form>
</body>
</html>
```

- Create a PHP script (**upload.php**) to manage file uploads:

```

<?php
if (isset($_POST["submit"])) {
    $targetDirectory = "uploads/"; // Direktori tujuan untuk menyimpan file
    $targetFile = $targetDirectory . basename($_FILES["fileToUpload"]["name"]);

    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $targetFile)) {
        echo "File berhasil diunggah.";
    } else {
        echo "Gagal mengunggah file.";
    }
}
}

```

Above, we take some important steps:

- Create an HTML form with an element `<input type="file">` that allows users to select the file they want to upload.
- Set the form `enctype` attribute to `"multipart/form-data"` so that the file can be uploaded.
- When the user clicks the "Upload File" button, the form data will be sent to `the upload.php`.
- In `upload.php`, we check if the user has clicked the submit button (`$_POST["submit"]`) and then specify the destination directory to save the file to.
- `move_uploaded_file()` is used to move files from a temp directory to a predefined destination directory.
- The corresponding message (success or failure) will be displayed to the user.

Make sure you have an `uploads` directory on your server, and give it the appropriate permissions so that PHP can store files there.

Practical Section 1. Files

| Step | Description |
|------|---|
| 1 | Create a new file in the <code>week8</code> directory, naming it <code>form_upload.php</code> . |
| 2 | Type into the <code>form_upload.php</code> code file below. <pre> <html> <head> <title>File Upload</title> </head> <body> <form action="upload.php" method="POST" enctype="multipart/form-data"> <input type="file" name="myfile"> <input type="submit" name="submit"> </form> </body> </html> </pre> |
| 3 | Create a new file named <code>upload.php</code> that will be used for processing <code>form_upload.php</code> . |

| | |
|---|---|
| | <pre> <?php if(isset(\$_POST["submit"])){ \$targetdir = "uploads/"; //Direktori tujuan untuk menyimpan file \$targetfile = \$targetdir . basename(\$_FILES["myfile"]["name"]); if(move_uploaded_file(\$_FILES["myfile"]["tmp_name"], \$targetfile)){ echo "File berhasil diunggah."; } else{ echo "Gagal mengunggah file."; } } ?> </pre> |
| 4 | <p>Save the file, then open a browser and run localhost/week8/form_upload.php. Select a file and click the Submit button. Observe what happens and record your understanding. (Question No. 1)</p> |
| 5 | <p>Next, create a folder named uploads in the week8. Re-run localhost/week8/form_upload.php. Select a file and click the Submit button. Observe what happens and record your understanding. (Question No. 2)</p> |
| 6 | <p>Change the contents of the upload.php file with the following code</p> <pre> <?php if(isset(\$_POST["submit"])){ \$targetdir = "uploads/"; //Direktori tujuan untuk menyimpan file \$targetfile = \$targetdir . basename(\$_FILES["myfile"]["name"]); \$fileType = strtolower(pathinfo(\$targetfile, PATHINFO_EXTENSION)); \$allowedExtensions = array("jpg", "jpeg", "png", "gif"); \$maxsize = 5*1024*1024; if (in_array(\$fileType, \$allowedExtensions) && \$_FILES["myfile"]["size"]<=\$maxsize) { if(move_uploaded_file(\$_FILES["myfile"]["tmp_name"], \$targetfile)){ echo "File berhasil diunggah."; } else{ echo "Gagal mengunggah file."; } } else{ echo "File tidak valid atau melebihi ukuran maksimum yang diizinkan"; } } ?> </pre> |
| 7 | <p>Save the file, open a browser and run localhost/week8/form_upload.php. Select a file with the .pdf extension or .docx. Click the Submit button. Observe what happens and record your understanding. (Question No. 3)</p> |

| | |
|----|---|
| | |
| 8 | <p>Next run localhost/week8/form_upload.php again.</p> <p>Select a file with the extension .jpg, .jpeg, .png, or .gif. Click the Submit button. Observe what happens and record your understanding.</p> <p>(Question No. 4)</p> |
| 9 | <p>Add script from step 6 to display thumbnail image files with a width of 200 and height following the changes automatically after the image file is successfully uploaded. Screen shoot the additional code. Explain your understanding after adding the program code.</p> <p>(Question No. 5)</p> |
| 10 | <p>Next, change the contents of the upload.php file with the following code.</p> <pre> <?php if(isset(\$_POST["submit"])){ \$targetdir = "uploads/"; //Direktori tujuan untuk menyimpan file \$targetfile = \$targetdir . basename(\$_FILES["myfile"]["name"]); \$fileType = strtolower(pathinfo(\$targetfile, PATHINFO_EXTENSION)); \$allowedExtensions = array("txt", "pdf", "doc", "docx"); \$maxsize = 3*1024*1024; if (in_array(\$fileType, \$allowedExtensions) && \$_FILES["myfile"]["size"]<=\$maxsize) { if(move_uploaded_file(\$_FILES["myfile"]["tmp_name"], \$targetfile)){ echo "File berhasil diunggah"; } else{ echo "Gagal mengunggah file."; } } else{ echo "File tidak valid atau melebihi ukuran maksimum yang diizinkan"; } } ?> </pre> |
| 11 | <p>Save the file, open a browser and run localhost/week8/form_upload.php</p> <p>Select a file with an extension of .txt, .pdf, .doc, or .docx that is more than 5 MB in size. Click the Submit button. Observe what happens and record your understanding.</p> <p>(Question No. 6)</p> |

| | |
|----|---|
| 12 | <p>Next run localhost/week8/form_upload.php again.</p> <p>Select a file with the extension .txt, .pdf, .doc, or .docx that is less than 3 MB in size. Click the Submit button. Observe what happens and record your understanding.</p> <p>(Question No. 7)</p> |
|----|---|

Practical Section 2. Multi Upload File

| Step | Description |
|------|---|
| 1 | Create a new file named form_multiupload.php . |
| 2 | <p>Type the following code into form_multiupload.php.</p> <pre><!DOCTYPE html> <html> <head> <title>Multiupload Dokumen</title> </head> <body> <h2>Unggah Dokumen</h2> <form action="proses_upload.php" method="post" enctype="multipart/form-data"> <input type="file" name="files[]" multiple="multiple" accept=".pdf, .doc, .docx" <input type="submit" value="Unggah" /> </form> </body> </html></pre> |
| 3 | <p>Create a new file named proses_upload.php. Type the following code.</p> <pre><?php // Lokasi penyimpanan file yang diunggah \$targetDirectory = "documents/"; // Periksa apakah direktori penyimpanan ada, jika tidak maka buat if (!file_exists(\$targetDirectory)) { mkdir(\$targetDirectory, 0777, true); } if (\$_FILES['files']['name'][0]) { \$totalFiles = count(\$_FILES['files']['name']); // Loop melalui semua file yang diunggah for (\$i = 0; \$i < \$totalFiles; \$i++) { \$fileName = \$_FILES['files']['name'][\$i]; \$targetFile = \$targetDirectory . \$fileName; // Pindahkan file yang diunggah ke direktori penyimpanan if (move_uploaded_file(\$_FILES['files']['tmp_name'][\$i], \$targetFile)) { echo "File \$fileName berhasil diunggah. "; } else { echo "Gagal mengunggah file \$fileName. "; } } } else { echo "Tidak ada file yang diunggah."; }</pre> |

| | |
|---|---|
| 4 | <p>Save the file, open the browser and run localhost/week8/form_multiupload.php. Select multiple files at once to upload. What do you understand from the script in the file? Record your understanding.</p> <p>(Question No. 8)</p> |
| 5 | <p>Change the code for multi upload of images. Screen shoot the code changes and provide an explanation of the code.</p> <p>(Question No. 9)</p> |

Practical Section 3. Upload Files with PHP and JQuery

| Step | Description |
|------|--|
| 1 | Create a new file named form_upload_ajax.php |
| 2 | <p>Type the following code on the form_upload_ajax.php</p> <pre> <!DOCTYPE html> <html> <head> <title>Unggah File Dokumen</title> </head> <body> <form id="upload-form" action="upload_ajax.php" method="post" enctype= "multipart/form-data"> <input type="file" name="file" id="file"> <input type="submit" name="submit" value="Unggah"> </form> <div id="status"></div> <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> <script src="upload.js"></script> </body> </html> </pre> |
| 3 | Create a new file named upload.js . Write the following code. |

```

$(document).ready(function(){
    $('#upload-form').submit(function(e){
        e.preventDefault();

        var formData = new FormData(this);

        $.ajax({
            type: 'POST',
            url: 'upload_ajax.php',
            data: formData,
            cache: false,
            contentType: false,
            processData: false,
            success: function(response){
                $('#status').html(response);
            },
            error: function(){
                $('#status').html('Terjadi kesalahan saat mengunggah file.');

```

4

Create a new file named **upload_ajax.php**. Write the following code.

```

<?php
if (isset($_FILES['file'])) {
    $errors = array();
    $file_name = $_FILES['file']['name'];
    $file_size = $_FILES['file']['size'];
    $file_tmp = $_FILES['file']['tmp_name'];
    $file_type = $_FILES['file']['type'];
    @ $file_ext = strtolower(" " . end(explode('.', $_FILES['file']['name']))) . " ");
    $extensions = array("pdf", "doc", "docx", "txt");

    if (in_array($file_ext, $extensions) === false) {
        $errors[] = "Ekstensi file yang diizinkan adalah PDF, DOC, DOCX, atau TXT.";
    }

    if ($file_size > 2097152) {
        $errors[] = 'Ukuran file tidak boleh lebih dari 2 MB';
    }

    if (empty($errors) == true) {
        move_uploaded_file($file_tmp, "documents/" . $file_name);
        echo "File berhasil diunggah.";
    } else {
        echo implode(" ", $errors);
    }
}

```

5

Save the file, then open a browser and run **localhost/week8/form_upload_ajax.php**.

- Upload a file in the form of an image.
- Upload a PDF file that is > 4 MB in size.
- Upload .docx files with a size of < 2 MB.

| | |
|---|--|
| | Observe what is happening and explain your understanding. (Question No. 10) |
| 6 | Change the code to be able to do multi-upload image files. Screenshot the code changes and explain the code. (Question No. 11) |

Practical Section 4. Decorate Upload Files

| Step | Description |
|------|--|
| 1 | <p>Modify the code of file <code>form_upload_ajax.php</code> in Practical Section 3</p> <pre> <!DOCTYPE html> <html> <head> <link rel="stylesheet" type="text/css" href="upload.css"> <title>Unggah File Dokumen</title> </head> <body> <div class="upload-form-container"> <h2>Unggah File Dokumen</h2> <form id="upload-form" action="upload.php" method="post" enctype= "multipart/form-data"> <div class="file-input-container"> <input type="file" name="file" id="file" class="file-input"> <label for="file" class="file-label">Pilih File</label> </div> <button type="submit" name="submit" class="upload-button" id="upload-button" disabled>Unggah</button> </form> <div id="status" class="upload-status"></div> </div> <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script> <script src="upload.js"></script> </body> </html> </pre> |
| 2 | Also modify the <code>upload.js</code> file as shown in the following code. |


```

$(document).ready(function(){
    $('#file').change(function(){
        if (this.files.length > 0) {
            $('#upload-button').prop('disabled', false).css('opacity', 1);
        } else {
            $('#upload-button').prop('disabled', true).css('opacity', 0.5);
        }
    });

    $('#upload-form').submit(function(e){
        e.preventDefault();

        var formData = new FormData(this);

        $.ajax({
            type: 'POST',
            url: 'upload_ajax.php',
            data: formData,
            cache: false,
            contentType: false,
            processData: false,
            success: function(response){
                $('#status').html(response);
            },
            error: function(){
                $('#status').html('Terjadi kesalahan saat mengunggah file.');
```

| | |
|---|--|
| | <pre> .file-label { background: #3498db; color: #fff; padding: 10px 20px; border-radius: 5px; cursor: pointer; } .upload-button { background: #2ecc71; color: #fff; padding: 10px 20px; border: none; border-radius: 5px; cursor: pointer; opacity: 0.5; /* Opacity to make it appear faded */ } .upload-button:disabled { background: #ccc; /* Change color when disabled */ cursor: not-allowed; /* Change cursor style when disabled */ } .upload-status { margin-top: 20px; font-weight: bold; } </pre> |
| 4 | <p>Save the file. Open a browser and run localhost/week8/form_upload_ajax.php.</p> <p>What do you understand from the program code above? Record your understanding. (Question No. 12)</p> |

Introduction to Cookies

A *cookie* is a value that is sent and embedded *by a server* on the *client's computer*. *Cookies* are small files that are generally less than 100 bytes, so they won't affect *your browsing* speed. *Cookies* contain Description relating to the user. *Cookies* are useful for making it easier for users, such as "remembering" users every time they visit the same website.

The Description stored in *cookies* is simple data about user habits. A simple example is when a user visits a website that offers a view with English and Indonesian versions. If the user selects Indonesian, *the cookies* will automatically save the configuration. This configuration will be remembered, so that if the user visits the website again, the user does not need to select the language type again.

Cookies have an expiration date, meaning that any data stored on the user's computer could at some point be lost or destroyed. Based on their age, *cookies* are divided into two types, namely *session cookies* and *persistent cookies*. *Session cookies* are a type of *cookie* that is stored temporarily and will

disappear when the user closes *the browser*. *These session cookies* are usually used in the "shopping cart" feature of online shopping sites. *Persistent cookies* are a type of *cookie* that is permanently stored on the computer until the user decides to delete it. *Persistent cookies* are generally used in the "*remember me*" feature when logging in.

PHP has functions that can be used to create and retrieve cookie values. Cookies must be declared before the page is displayed, which means they are written before the **<html> tag**. *Cookies* are created using **the setcookie()** function. Here is the basic syntax of **the setcookie()** function:

```
<?php
    setcookie(name, value, expire);
?>
```

Description:

- **name**, is the name of *the cookie* or variable that identifies *the cookie*
- **value**, contains the value stored in *the cookie*
- **expire**, is the period of time *that cookies* will be stored on the computer

After knowing how to create *cookies*, then the value of the cookies that have been created can be retrieved using the **\$_COOKIE** variable. The following is the syntax for retrieving the value *of cookies*:

```
<?php
    $_COOKIE['user'];
?>
```

Practical Section 5. Creating Cookies

| Step | Description |
|------|---|
| 1 | Create a new file named cookiesCreate.php , then type the following code. <pre>1 <?php 2 setcookie("user", "Polinema", time()+3600); 3 ?></pre> |
| 2 | Create a new file named cookiesCall.php , then type the following code. <pre>1 <?php 2 echo \$_COOKIE['user']; 3 ?></pre> |
| 3 | Open a <i>browser</i> and run the program code in step 2 by typing localhost/week8/cookiesCall.php |
| 4 | Observe and explain your observations (Question No. 13) |
| 5 | Open a <i>browser</i> and run the program code step 1 by typing localhost/week8/cookiesCreate.php |

| | |
|----|---|
| 6 | Repeat step 3. |
| 7 | Observe and explain the results displayed (Question No. 14) |
| 8 | <i>Restart</i> your computer. |
| 9 | Once the computer is turned on, restart Apache on the laragon. |
| 10 | Open the same browser as before then repeat step 3. |
| 11 | Observe and explain the results displayed. (Question No. 15) |

Practical Section 6. Deleting the Value of *Cookies*

In this Practical Section, it will be discussed how to delete the value of *cookies*. If in the previous Practical Section the *cookies* were set with an *expiration* `time()+3600`, then to delete the cookie value is as follows:

| Step | Description |
|------|---|
| 1 | Create a new file with <code>cookiesDel.php</code> name, then type the following code. <pre> 1 <?php 2 setcookie("user", "Polinema", time()-3600); 3 ?> </pre> |
| 2 | Open a <i>browser</i> and run the program code by typing <code>localhost/week8/cookiesDel.php</code> |
| 3 | Open a <i>browser</i> and run the program code from the part 5 Practical Section by typing <code>localhost/week8/cookiesCall.php</code> |
| 4 | Observe and describe the results from steps 2 and 3, then draw conclusions. (Question No. 16) |

Practical Section 7. Application of *Cookies* to the Shopping Cart Feature

One example of the use of *cookies* is the "shopping cart" feature on the online store web application. The shopping cart contains the items that the user will buy. *Cookies* are used to remember the number of items selected by the user. Here is an example of the use of *cookies* in the shopping cart feature:

| Step | Description |
|------|---|
| 1 | <p>Create a new file with formBeli.html name, then type the following code.</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <form action="prosesBeli.php" method="POST"> 6 <p> Jumlah Novel yang dibeli : 7 <input type="text" name="beliNovel" value= "0" size="2"> </p> 8 <p> Jumlah Buku Teks yang dibeli : 9 <input type="text" name="beliBuku" value= "0" size="2"> </p> 10 <input type="submit"> 11 </form> 12 </body> 13 </html> </pre> |
| 2 | <p>Create a new file named prosesBeli.php, then type the following code.</p> <pre> 1 <?php 2 if(isset(\$_POST["beliNovel"]) && isset(\$_POST["beliBuku"])){ 3 setcookie("beliNovel", \$_POST["beliNovel"]); 4 setcookie("beliBuku", \$_POST["beliBuku"]); 5 header("location:keranjangBelanja.php"); 6 } 7 ?> </pre> |
| 3 | <p>Create a new file named keranjangBelanja.php, then type the following code.</p> <pre> 1 <html> 2 <head> 3 </head> 4 <body> 5 <h2> Keranjang Belanja </h2> 6 7 <?php 8 \$beliNovel=\$_COOKIE['beliNovel']; 9 \$beliBuku = \$_COOKIE['beliBuku']; 10 11 echo "Jumlah Novel:" . \$beliNovel . " "; 12 echo "Jumlah Buku :" . \$beliBuku ; 13 ?> 14 </body> 15 </html> </pre> |
| 4 | <p>Open a browser and run the program code step 3 by typing localhost/week8/keranjangBelanja.php</p> |
| 5 | <p>Observe and explain the results displayed. (Question No. 17)</p> |
| 6 | <p>Run the program code step 1 by typing localhost/week8/formBeli.html</p> |
| 7 | <p>Fill in the number of novels and textbooks you want to buy and then click the "submit" button.</p> |
| 8 | <p>Observe and explain the results displayed. (Question No. 18)</p> |

| | |
|----|---|
| | |
| 9 | Close <i>the browser</i> then reopen <i>the browser</i> then re-run the program code step 3 by typing localhost/week8/keranjangBelanja.php |
| 10 | Observe and explain the results displayed. (Question No. 19) |

Session Introduction

Session is one of the facilities in PHP that is used to store data as a result of being stored in variables, so that the data can be accessed by the user as long as the *session* variable is not emptied or deleted. *Sessions* are almost the same in principle as *cookies*, the difference is the position of the data storage. If *cookies* store data on the *client*, the *session data* will be stored on the server side. So *sessions* are relatively safer to use to store confidential value variables such as *usernames* and *passwords* at the time of login. Examples of using *sessions* include storing valid login Description in only one session and storing records of ordering goods in the *e-commerce/online* transaction system. The following table describes the comparison of *cookies* and *sessions*:

Comparison of *Cookies* and *Sessions*

| | Storing Description | Storage Location | Can be summoned | Global | Can be called up at another time | Browser Restricted | User editable |
|----------------|---------------------|------------------|-----------------|--------|----------------------------------|--------------------|---------------|
| <i>Cookies</i> | Yes | <i>Client</i> | Yes | Yes | Yes | Yes | Yes |
| <i>Session</i> | Yes | <i>Server</i> | Yes | Yes | Not | Not | Not |

To create a *session* in PHP the **session_start()** function is used. Here is the basic syntax for starting a *session*:

```
<?php
    session_start();
?>
```

Once *the session* starts, the *session* variable can be used. The variable in question is the PHP global variable **\$_SESSION**. Here's the syntax of using the **\$_SESSION** global variable:

```
<?php
    $_SESSION['nameVariable'];
?>
```

Practical Section 8. Creating a Session

| Step | Description |
|------|--|
| 1 | Create a new file named sessionCreate.php , then type the following code. <pre>1 <?php 2 session_start(); 3 ?> 4 5 <!DOCTYPE html> 6 <html> 7 <body> 8 <?php 9 \$_SESSION["favcolor"] = "green"; 10 \$_SESSION["favanimal"] = "cat"; 11 echo "Session variables are set."; 12 ?> 13 </body> 14 </html></pre> |
| 2 | Create a new file named sessionCall.php , then type the following code. <pre>1 <?php 2 session_start(); 3 ?> 4 <!DOCTYPE html> 5 <html> 6 <body> 7 <?php 8 echo "Favorite color is " . \$_SESSION["favcolor"] . " "; 9 echo "Favorite animal is " . \$_SESSION["favanimal"] . "."; 10 ?> 11 </body> 12 </html></pre> |
| 3 | Open a <i>browser</i> and run the program code in step 2 by typing localhost/week8/sessionCall.php |
| 4 | Open a <i>browser</i> and run the program code step 1 by typing localhost/week8/sessionCreate.php |
| 5 | Repeat step 3 |
| 6 | Observe and explain the results displayed (Question No. 20) |

Practical Section 9. Removing Session Values

PHP provides a **session_destroy()** function that can be used to delete *sessions*.

| Step | Description |
|------|---|
| 1 | Create a new file named sessionDel.php , then type the following code. |

| | |
|---|---|
| | <pre> 1 <?php 2 session_start(); 3 ?> 4 5 <!DOCTYPE html> 6 <html> 7 <body> 8 <?php 9 session_unset(); 10 session_destroy(); 11 12 echo "All session variables are now removed, and the session is destroyed." 13 ?> 14 </body> 15 </html> </pre> |
| 2 | Open a <i>browser</i> and run the program code by typing localhost/week8/sessionDel.php |
| 3 | Open a <i>browser</i> and run the program code from the Practical Section 8 by typing localhost/week8/sessionCall.php |
| 4 | Observe and explain the results displayed. (Question No. 21) |

Practical Section 10. Implementation of *Session* on the Login Feature

| Step | Description |
|------|--|
| 1 | <p>Create a new file named sessionLoginForm.html, then type the following code</p> <pre> <html> <head> <title>File Upload</title> </head> <body> <form action="sessionLoginProcess.php" method="POST"> <table> <tr> <td>Username</td> <td><input type="text" name="username" size="20"></td> </tr> <tr> <td>Password</td> <td><input type="password" name="password" size="20"></td> </tr> <tr> <td>&nbsp;</td> <td><input type="submit" name="login" value="Login"></td> </tr> </table> </form> </body> </html> </pre> |
| 2 | Create a file named sessionLoginProcess.php , then type the following code. |

| | |
|---|--|
| | <pre> <?php \$username = \$_POST['username']; \$password = \$_POST['password']; if(\$username=="admin" && \$password=="1234"){ session_start(); \$_SESSION["username"] = \$username; \$_SESSION["status"] = 'login'; echo "Anda berhasil login. Silahkan menuju Halaman Home"; } else{ echo "Gagal login. Silahkan login lagi Halaman Login"; } ?> </pre> |
| 3 | <p>Create a file named homeSession.php, then type the following code.</p> <pre> <html> <head> <title>Halaman Home</title> </head> <body> <?php session_start(); if(\$_SESSION['status']=='login'){ echo "Selamat datang " . \$_SESSION['username']; } Logout <?php } else{ echo "Anda belum login, silahkan"; } Login <?php } ?> </body> </html> </pre> |
| 4 | <p>Create a file named sessionLogout.php, then type the following code.</p> <pre> <?php session_start(); session_destroy(); echo "Anda berhasil logout"; ?> </pre> |
| 5 | <p>Open a <i>browser</i> and run the program code by typing localhost/week8/sessionLoginForm.html</p> |
| 6 | <p>Log in using your email username and password "0000".</p> |
| 7 | <p>Observe and explain the results displayed (Question No. 22)</p> |
| 8 | <p>Re-run the program code by typing localhost/week8/sessionLoginForm.html Log in using the username "admin" and password "1234".</p> |

| | |
|----|--|
| 9 | Observe and explain the results displayed (Question No. 23) |
| 10 | Describe the sequence of the process from login to logout (also mention the order in which the files are processed) (Question No. 24) |