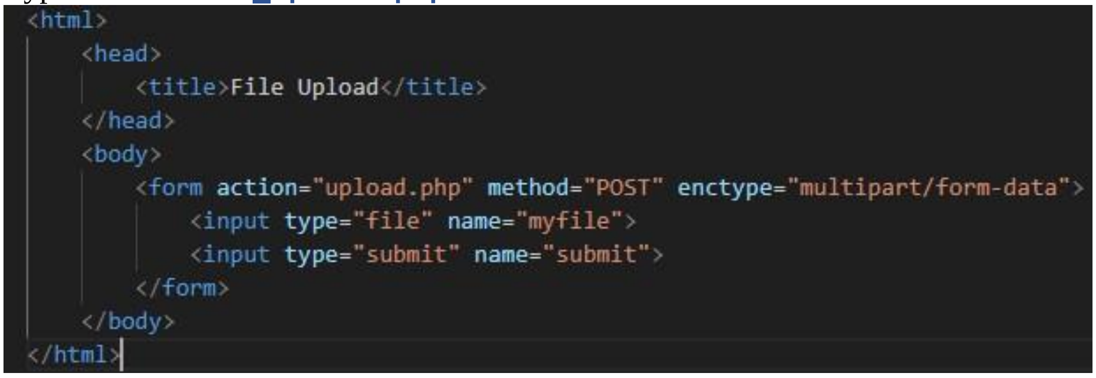
Name: Krisna Prawira Nugraha
NIM: 2341720190
Class/No.: TI-2I/15
Github: https://github.com/CipherPixy032/Web-Design-Jobsheets

Jobsheet-8: PHP - Form Upload, Cookies and Session

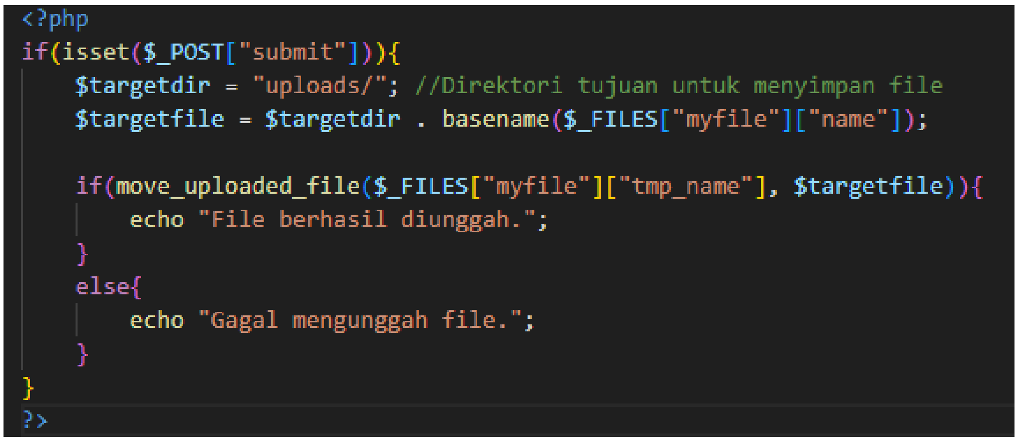
Practical Section 1. Files

Step	Description
1	Create a new file in the `week8` directory, naming it `form_upload.php`.
2	Type into the `form_upload.php` code file below.



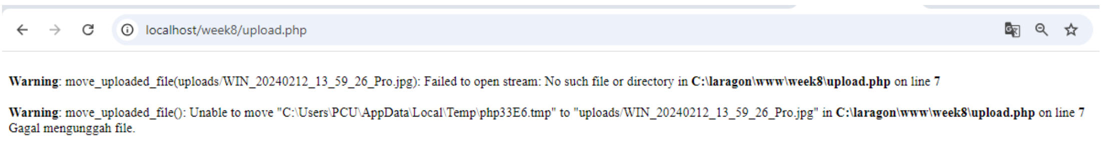
```
<html>
    <head>
        <title>File Upload</title>
    </head>
    <body>
        <form action="upload.php" method="POST" enctype="multipart/form-data">
            <input type="file" name="myfile">
            <input type="submit" name="submit">
        </form>
    </body>
</html>
```

Step	Description
3	Create a new file named `upload.php` that will be used for processing `form_upload.php`.



```
<?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.";
    }
}
?>
```

Step	Description
4	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)

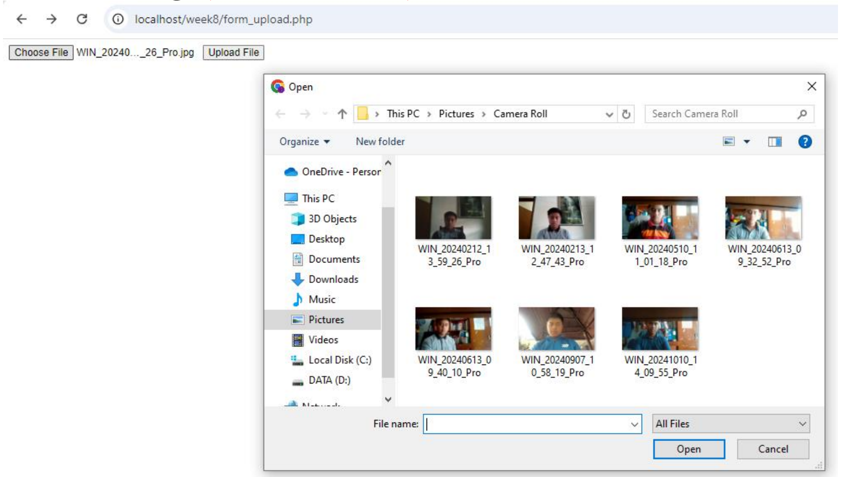


```
localhost/week8/upload.php

Warning: move_uploaded_file(uploads/WIN_20240212_13_59_26_Pro.jpg): Failed to open stream: No such file or directory in C:\laragon\www\week8\upload.php on line 7

Warning: move_uploaded_file(): Unable to move "C:\Users\PCU\AppData\Local\Temp\php33E6.tmp" to "uploads/WIN_20240212_13_59_26_Pro.jpg" in C:\laragon\www\week8\upload.php on line 7
Gagal mengunggah file.
```

Further explanations will be explained in No.2. But in short, the selected file doesn't upload because we didn't make the directory file destination.

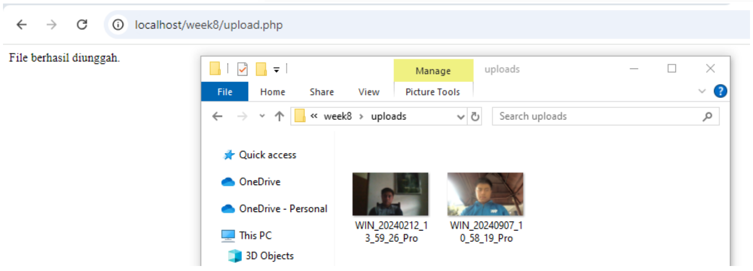
5

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)



```
localhost/week8/form_upload.php
Choose File  WIN_20240..._26_Pro.jpg  Upload File
```

Open
This PC › Pictures › Camera Roll
Search Camera Roll
Organize
New folder
OneDrive - Persor
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
DATA (D:)
WIN_20240212_13_59_26_Pro
WIN_20240213_12_47_43_Pro
WIN_20240510_11_01_18_Pro
WIN_20240613_09_32_52_Pro
WIN_20240613_09_40_10_Pro
WIN_20240907_10_58_19_Pro
WIN_20241010_14_09_55_Pro
File name:
All Files
Open
Cancel

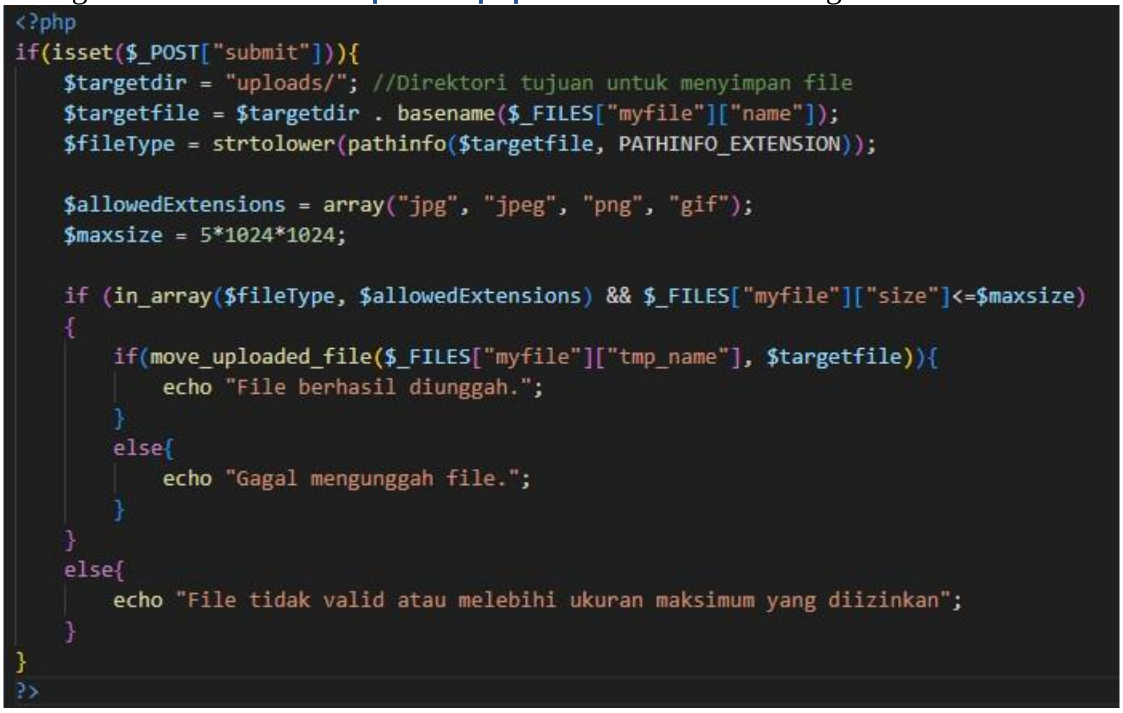


```
localhost/week8/upload.php
File berhasil diunggah.
```

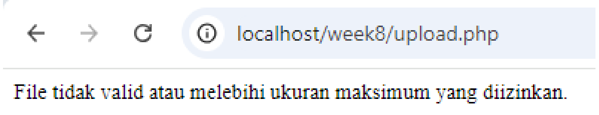
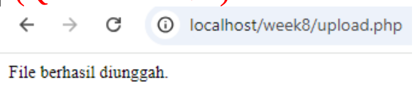
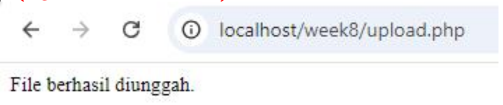
When you select a file and click the submit button, the form sends a POST request with the file data to the server, where the PHP script processes it. The file's temporary location is accessed via $_FILES["fileToUpload"], and the script attempts to move the file to a designated directory (uploads/) using move_uploaded_file(). If the file upload is successful, a message saying "File berhasil diunggah" ("File successfully uploaded") is displayed, otherwise, it shows "Gagal mengunggah file" ("Failed to upload file"). This basic system handles file uploads with minimal validation or error handling.

6

Change the contents of the **upload.php** file with the following code

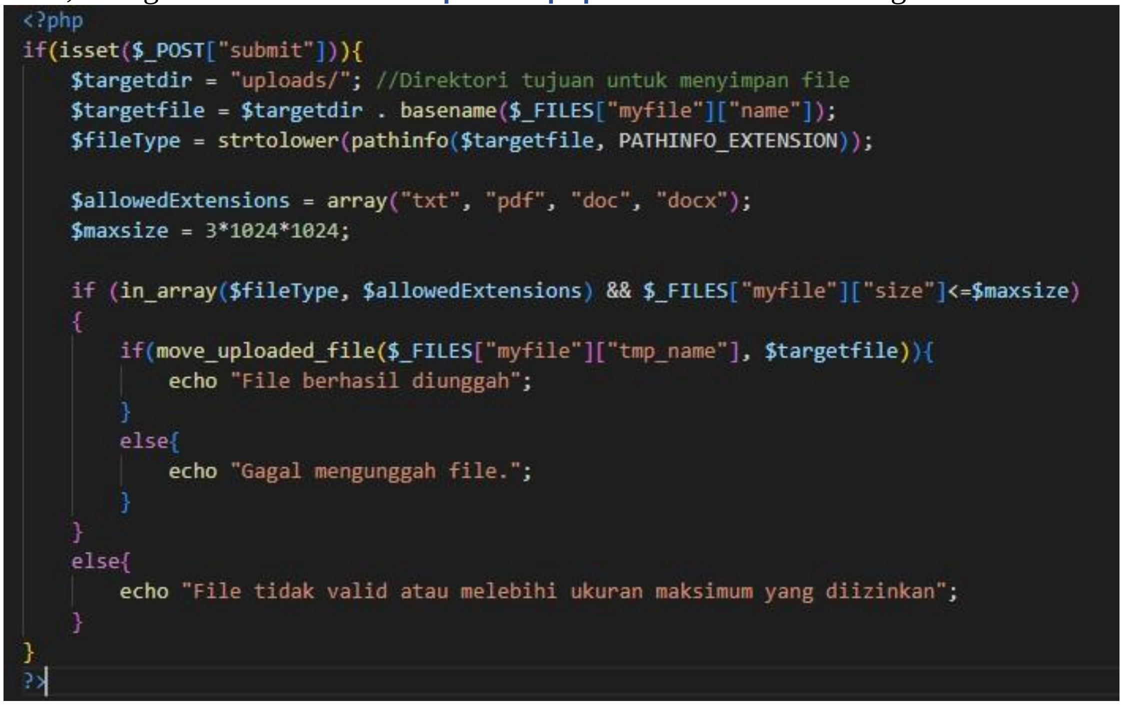


```php
<?php
if(isset($_POST["submit"])){
    $targetdir = "uploads/"; //Direktori tujuan untuk menyimpan file
    $targetfile = $targetdir . basename($_FILES["myfile"]["name"]);
    $fileType = strtolower(pathinfo($targetfile, PATHINFO_EXTENSION));

    $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";
    }
}
?>
```

7	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)  localhost/week8/upload.php File tidak valid atau melebihi ukuran maksimum yang diizinkan. When you select a file and click the submit button, the form sends the file data to the server. The PHP script first retrieves the file and determines its type and size. It checks whether the file has one of the allowed extensions (jpg, jpeg, png, gif) and ensures the file size is within the 5 MB limit. If both conditions are met, the script attempts to move the file to the uploads/ directory. If successful, it shows "File berhasil diunggah" ("File successfully uploaded"). If the file doesn't meet the conditions (invalid file type or exceeds size limit), the script displays "File tidak valid atau melebihi ukuran maksimum yang diizinkan" ("File is invalid or exceeds the allowed size").
8	Next run `localhost/week8/form_upload.php` again. Select a file with the extension .jpg, .jpeg, .png, or .gif. Click the Submit button. Observe what happens and record your understanding. (Question No. 4)  localhost/week8/upload.php File berhasil diunggah. This is the results when the desired file is uploaded.
9	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. (Question No. 5)  localhost/week8/upload.php File berhasil diunggah. When you select a file and click the submit button, the form sends the file data to the server. The script checks if the file has an allowed extension (jpg, jpeg, png, gif) and is within the 5 MB size limit. If both conditions are met, the file is uploaded to the uploads/ directory. If the upload is successful, a confirmation message "File berhasil diunggah" ("File successfully uploaded") appears, along with a thumbnail of the uploaded image displayed below it. The thumbnail is generated using the uploaded file's path with a width of 200 pixels. If the file fails to meet the conditions or can't be uploaded, an error message is shown.

10

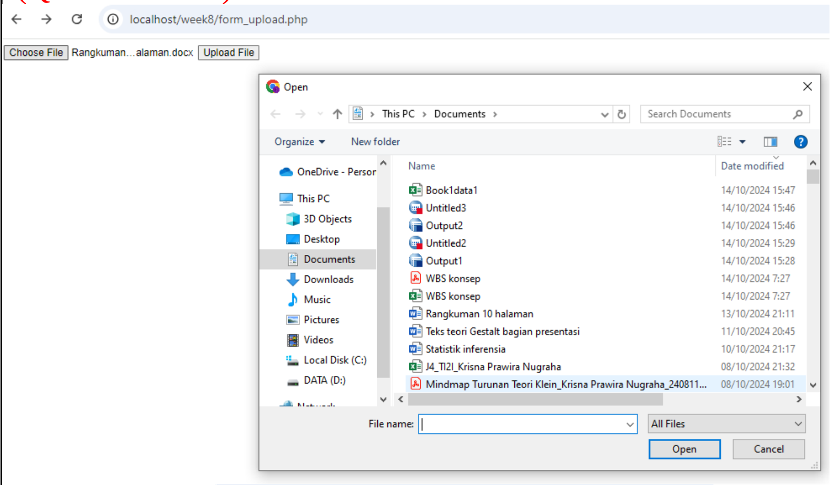
Next, change the contents of the `upload.php` file with the following code.



```php
<?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";
    }
}
?>
```

11

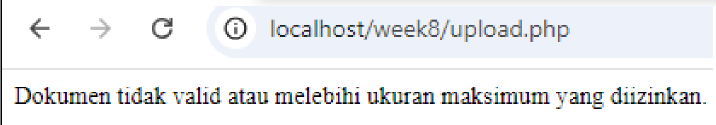
Save the file, open a browser and run `localhost/week8/form_upload.php` Select a file with an extension of .txt, .pdf, .doc, or .docx that is more than 5 MB in size. Click the Submit button. Observe what happens and record your understanding. (Question No. 6)



localhost/week8/form_upload.php

Choose File Rangkuman...alaman.docx Upload File

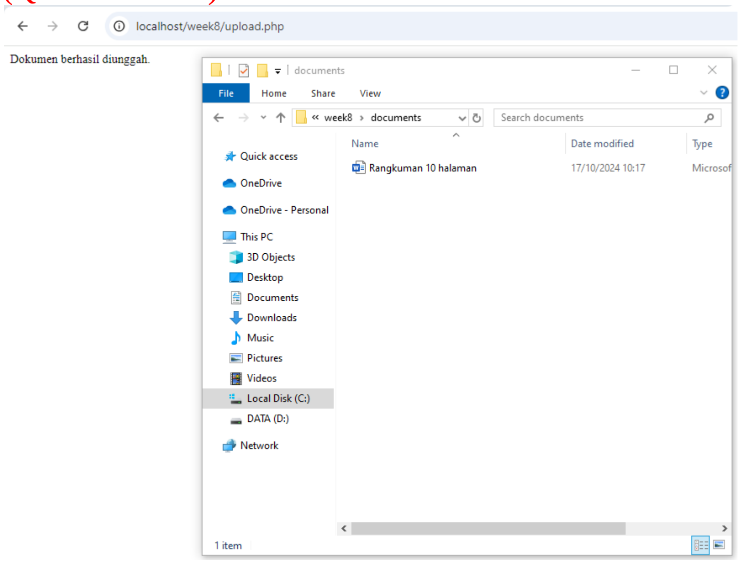
Name	Date modified
Book1data1	14/10/2024 15:47
Untitled3	14/10/2024 15:46
Output2	14/10/2024 15:46
Untitled2	14/10/2024 15:29
Output1	14/10/2024 15:28
WBS konsep	14/10/2024 7:27
WBS konsep	14/10/2024 7:27
Rangkuman 10 halaman	13/10/2024 21:11
Teks teori Gestalt bagian presentasi	11/10/2024 20:45
Statistik inferensia	10/10/2024 21:17
J4_TI2I_Krisna Prawira Nugraha	08/10/2024 21:32
Mindmap Turunan Teori Klein_Krisna Prawira Nugraha_240811...	08/10/2024 19:01



localhost/week8/upload.php

Dokumen tidak valid atau melebihi ukuran maksimum yang diizinkan.

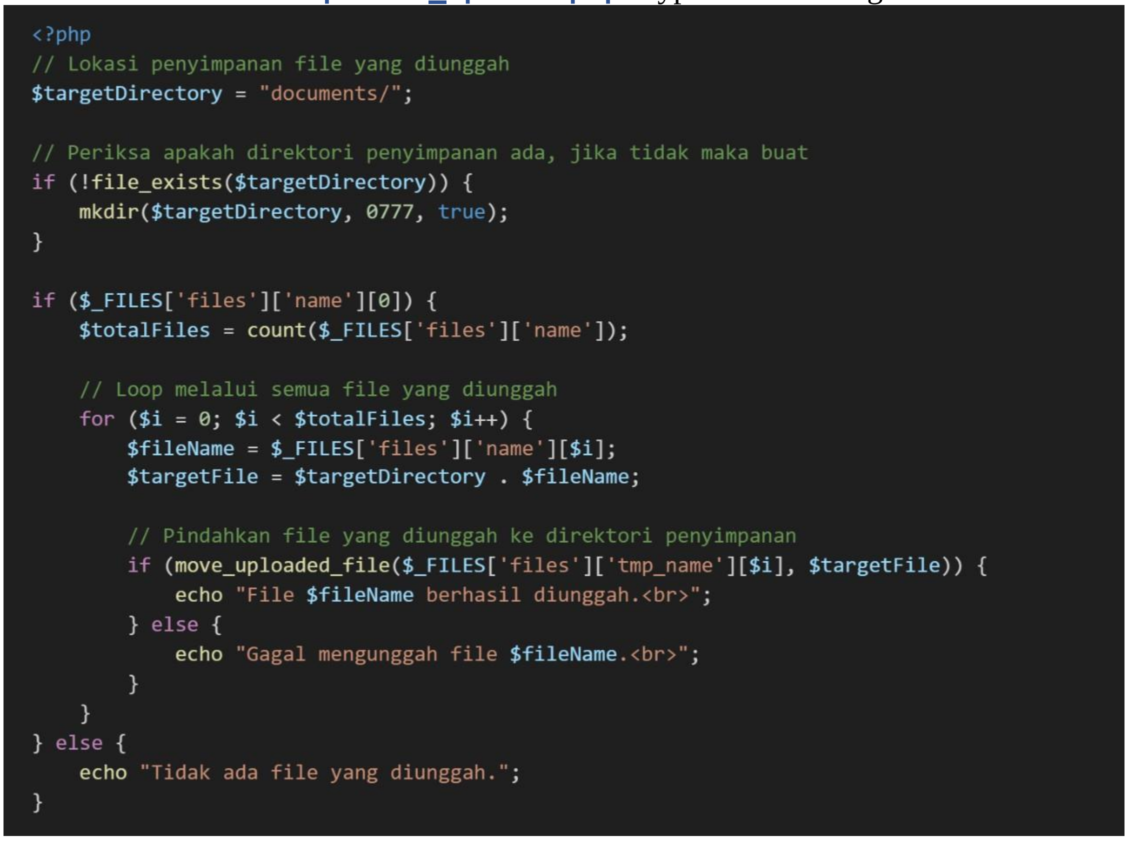
In short, it works like previous questions but it’s for documents instead of picture. As you can see, if the file’s exceeding the size, it won’t upload.

12	Next run **localhost/week8/form_upload.php** again. 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. (Question No. 7)  localhost/week8/upload.php Dokumen berhasil diunggah. documents File Home Share View « week8 › documents Search documents Quick access OneDrive OneDrive - Personal This PC 3D Objects Desktop Documents Downloads Music Pictures Videos Local Disk (C:) DATA (D:) Network Name Date modified Type Rangkuman 10 halaman 17/10/2024 10:17 Microsof 1 item The script validates the uploaded file by checking its extension (limited to txt, pdf, doc, docx) and ensures its size does not exceed 10 MB. If the file meets both criteria, it is moved from the temporary location to the documents/ directory on the server, and the message "Dokumen berhasil diunggah" ("Document successfully uploaded") is displayed. If the file is invalid or too large, or if the upload fails, an error message is shown. This ensures secure and controlled document uploads.

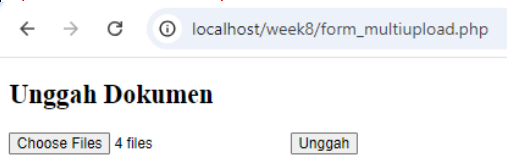
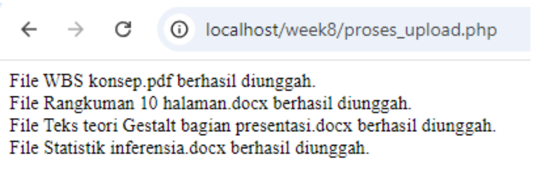
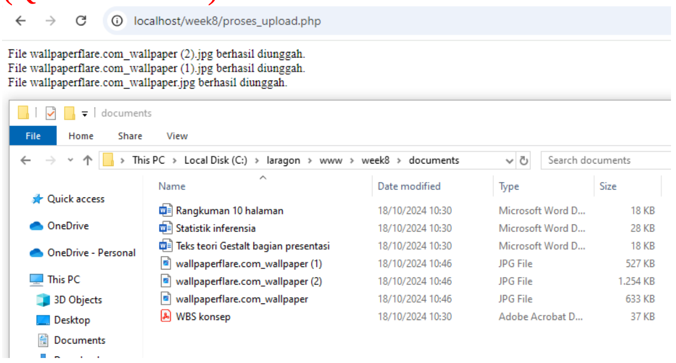
Practical Section 2. Multi Upload File

Step	Description
1	Create a new file named **form_multiupload.php**.
2	Type the following code into **form_multiupload.php.**

```
<!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>
```

3	Create a new file named `proses_upload.php`. Type the following code. 

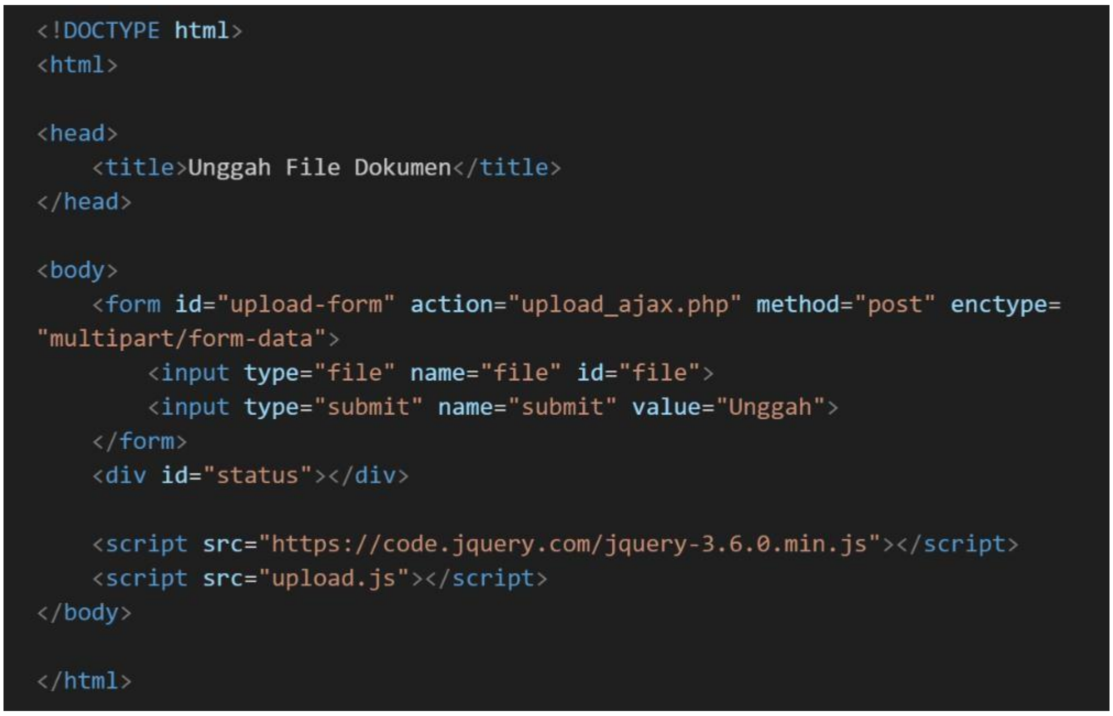
```php
<?php
// Lokasi penyimpanan file yang diunggah
$targetDirectory = "documents/";

// 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.<br>";
        } else {
            echo "Gagal mengunggah file $fileName.<br>";
        }
    }
} else {
    echo "Tidak ada file yang diunggah.";
}
```

4	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. (Question No. 8)  localhost/week8/form_multiupload.php **Unggah Dokumen** Choose Files 4 files Unggah  localhost/week8/proses_upload.php File WBS konsep.pdf berhasil diunggah. File Rangkuman 10 halaman.docx berhasil diunggah. File Teks teori Gestalt bagian presentasi.docx berhasil diunggah. File Statistik inferensia.docx berhasil diunggah. The form_multiupload.php file creates a user interface with a form that allows users to select and upload multiple document files (PDF, DOC, DOCX) at once, using the multiple="multiple" attribute and specifying the file types with the accept attribute. The form submits the selected files via the POST method to proses_upload.php, which handles the server-side processing. In proses_upload.php, a target directory (documents/) is defined for storing the uploaded files, and if the directory does not exist, it is created. The script then checks if files were uploaded, counts the total files, and processes each one by moving it from its temporary location to the target directory.
5	Change the code for multi upload of images. Screen shoot the code changes and provide an explanation of the code. (Question No. 9)  localhost/week8/proses_upload.php File wallpaperflare.com_wallpaper (2).jpg berhasil diunggah. File wallpaperflare.com_wallpaper (1).jpg berhasil diunggah. File wallpaperflare.com_wallpaper.jpg berhasil diunggah. This PC › Local Disk (C:) › laragon › www › week8 › documents Name / Date modified / Type / Size Rangkuman 10 halaman / 18/10/2024 10:30 / Microsoft Word D... / 18 KB Statistik inferensia / 18/10/2024 10:30 / Microsoft Word D... / 28 KB Teks teori Gestalt bagian presentasi / 18/10/2024 10:30 / Microsoft Word D... / 18 KB wallpaperflare.com_wallpaper (1) / 18/10/2024 10:46 / JPG File / 527 KB wallpaperflare.com_wallpaper (2) / 18/10/2024 10:46 / JPG File / 1.254 KB wallpaperflare.com_wallpaper / 18/10/2024 10:46 / JPG File / 633 KB WBS konsep / 18/10/2024 10:30 / Adobe Acrobat D... / 37 KB It features a form that allows users to select and upload several images simultaneously, with accepted file types including .jpg, .jpeg, .png, and .gif. The form uses the multiple="multiple" attribute to enable multi-file selection and submits the selected files via the POST method to the proses_upload.php script for processing. The enctype="multipart/form-data" attribute ensures proper handling of file uploads, while the form's new heading, "Unggah Gambar" (Upload Image), indicates its focus on image file uploads instead of documents.

Practical Section 3. Upload Files with PHP and Jquery

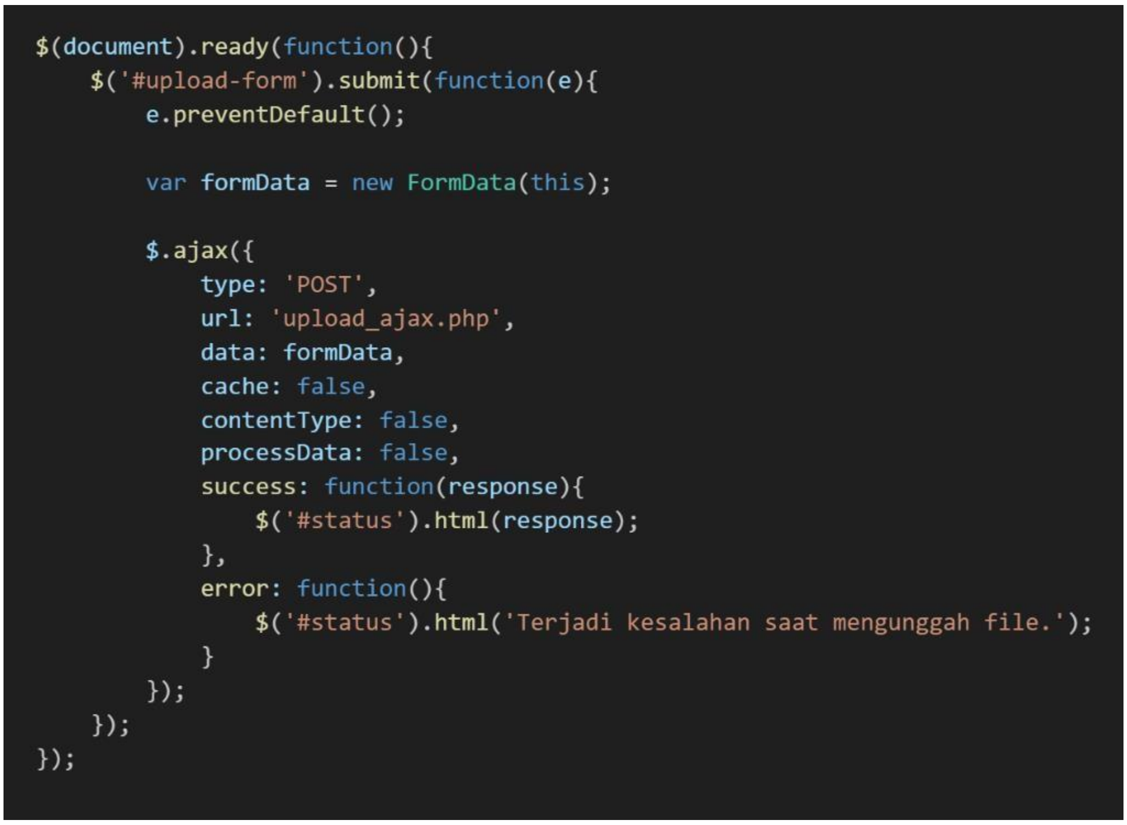
Step	Description
1	Create a new file named `form_upload_ajax.php`

2 Type the following code on the **`form_upload_ajax.php`**



```
<!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>
```

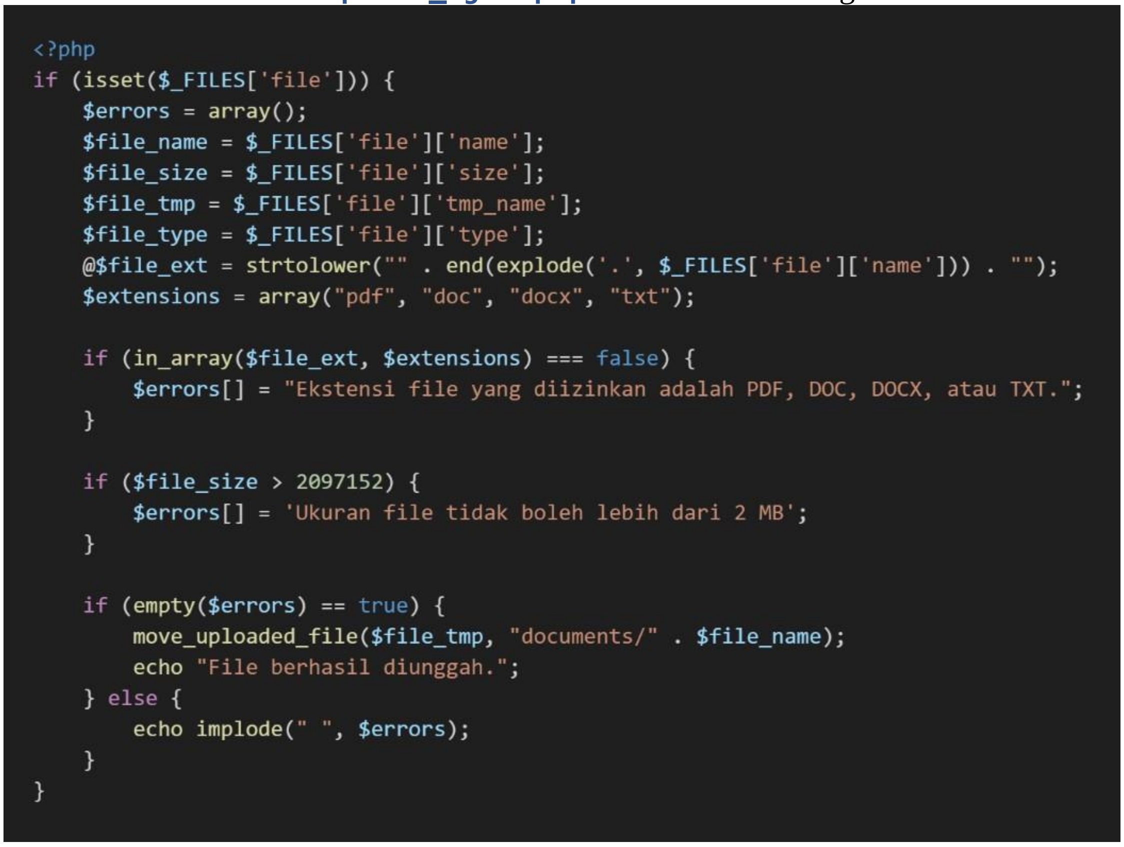
3 Create a new file named **`upload.js`**. Write the following code.



```
$(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.

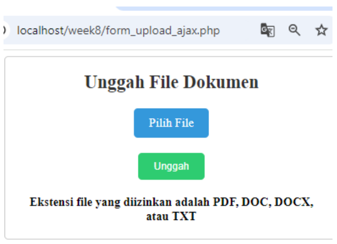


```
<?php
if (isset($_FILES['file'])) {
    $errors = array();
    $file_name = $_FILES['file']['name'];
    $file_size = $_FILES['file']['size'];
    $file_tmp = $_FILES['file']['tmp_name'];
    $file_type = $_FILES['file']['type'];
    @$file_ext = strtolower("" . end(explode('.', $_FILES['file']['name']))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

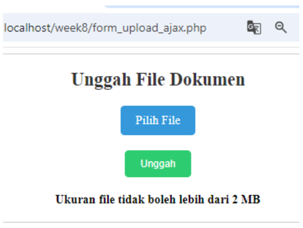


localhost/week8/form_upload_ajax.php

Unggah File Dokumen

Ekstensi file yang diizinkan adalah PDF, DOC, DOCX, atau TXT

- Upload a PDF file that is > 4 MB in size.

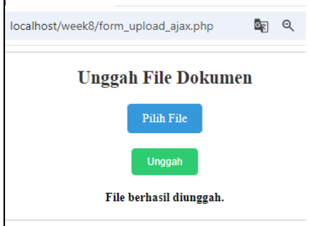


localhost/week8/form_upload_ajax.php

Unggah File Dokumen

Ukuran file tidak boleh lebih dari 2 MB

- Upload .docx files with a size of < 2 MB.



localhost/week8/form_upload_ajax.php

Unggah File Dokumen

File berhasil diunggah.

Observe what is happening and explain your understanding.
(Question No. 10)
This code creates a simple file upload system using HTML, jQuery, and PHP. The front-end (in form_upload_ajax.php) provides a form where users can select a file, with the "Upload" button initially disabled. Once a file is selected, the button is enabled, and the form can be submitted via AJAX. The jQuery script in upload.js captures the form submission, prevents the default page reload, and sends the file to the server using a FormData object. The server-side PHP script (upload_ajax.php) processes the file, checking its extension (allowing only PDF, DOC, DOCX, and TXT) and ensuring it is under 2MB. If validation passes, the file is saved in a "documents/" directory, and a success message is returned. Otherwise, error messages are displayed. The result is an asynchronous, user-friendly file upload process with real-time validation feedback.

6

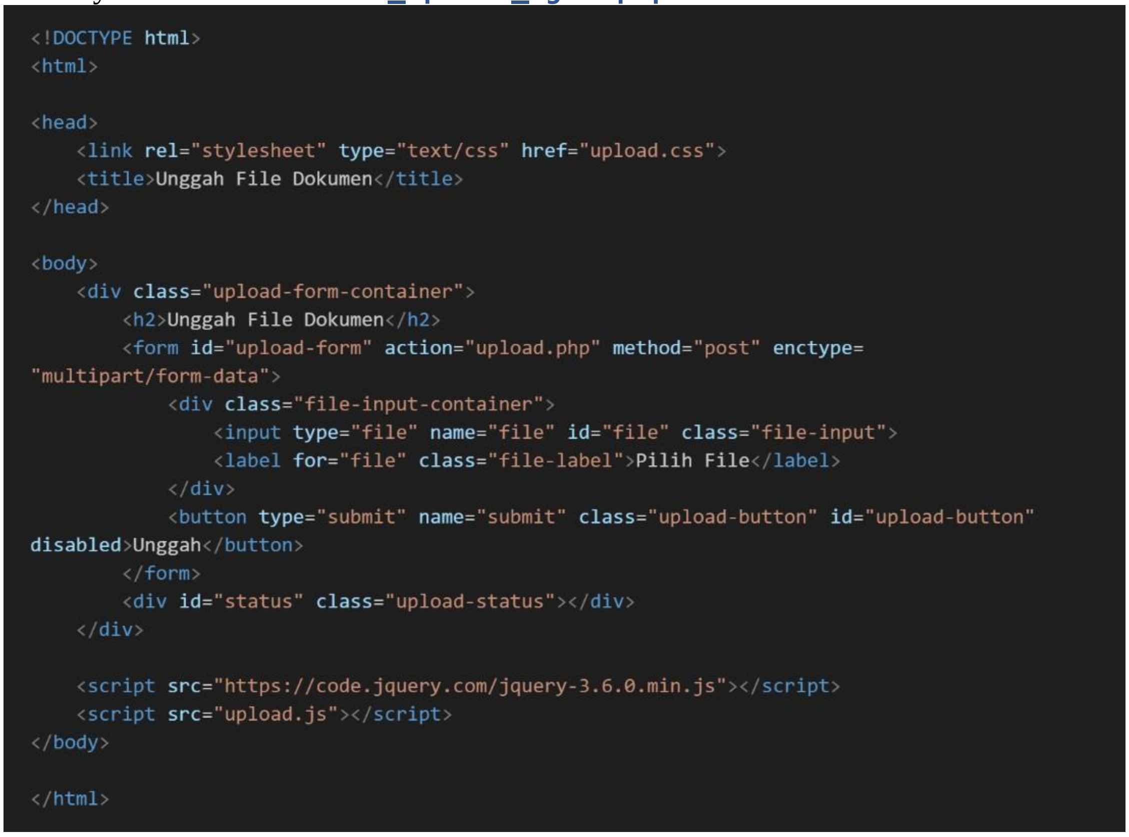
Change the code to be able to do multi-upload image files.
Screenshot the code changes and explain the code.
(Question No. 11)
Here is the updated code (upload_ajax.php) to do multiupload image file:

```php
<?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'];
        $file_ext = strtolower(pathinfo($file_name,
PATHINFO_EXTENSION)); // Perbaiki bagian ini

        $extensions = array("jpg", "jpeg", "png", "gif");

        if (!in_array($file_ext, $extensions)) {
            $errors[] = "Ekstensi file yang diizinkan adalah JPG, JPEG,
PNG, atau GIF";
        }

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

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

The changes in this PHP code modify the types of files that are allowed to be uploaded. Previously, the code allowed files with extensions like PDF, DOC, DOCX, and TXT. In the updated version, it restricts the allowed file extensions to image formats: JPG, JPEG, PNG, and GIF. This is reflected in the $extensions array, which now contains these image file types, and the error message has been updated to inform users that only image files with these extensions are accepted.

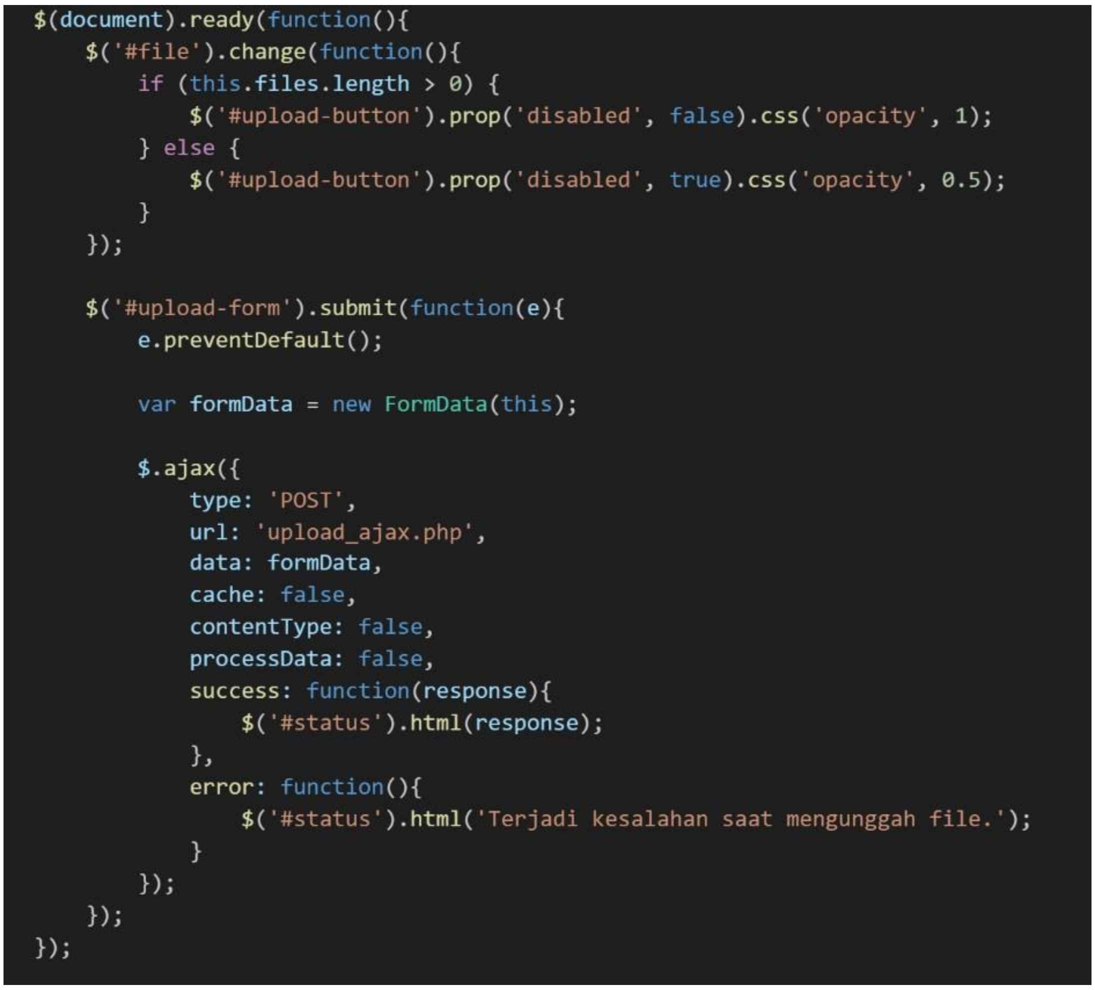
Practical Section 4. Decorate Upload Files

Step	Description

1	Modifiy the code of file `form_upload_ajax.php` in Practical Section 3 

```html
<!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>
```

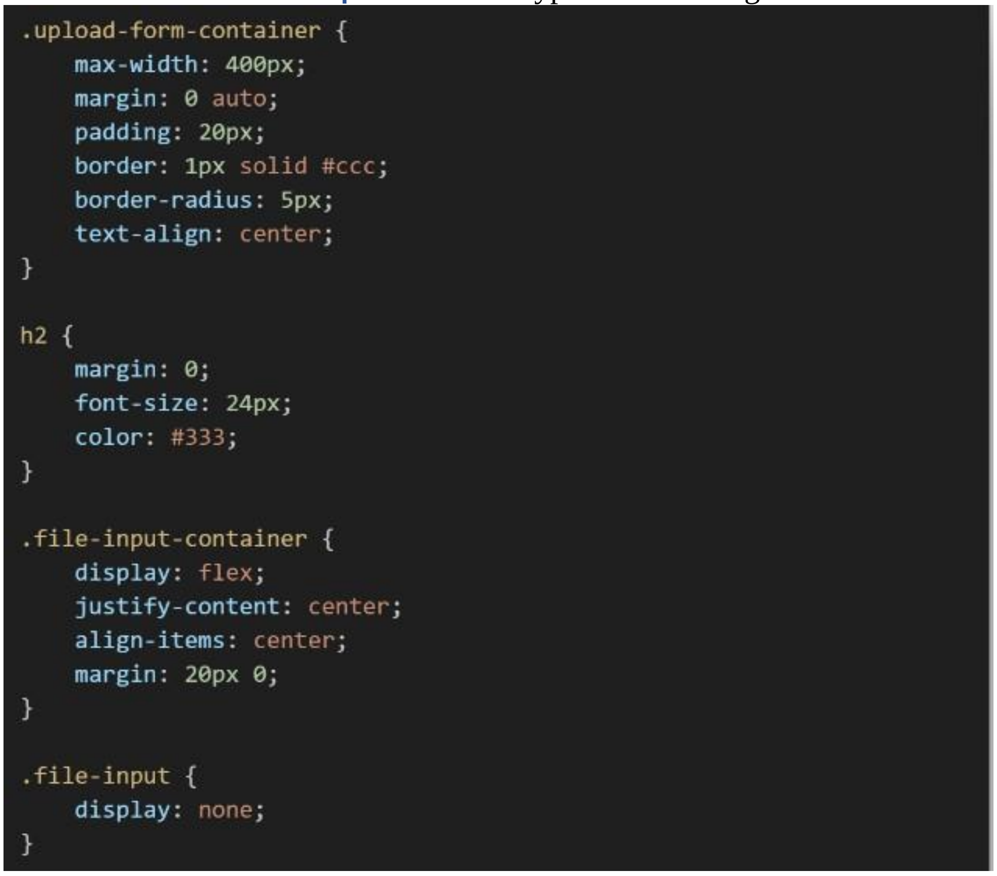
2	Also modify the `upload.js` file as shown in the following code.



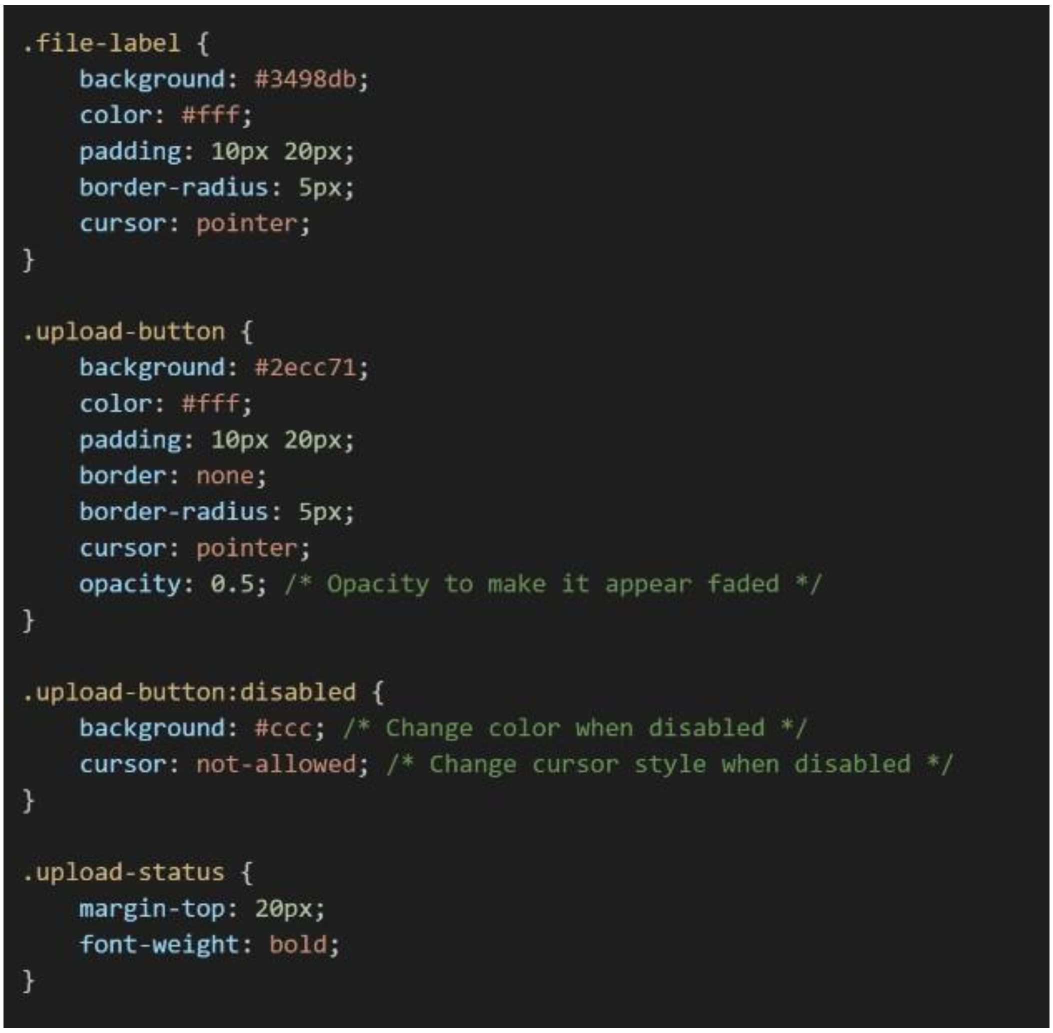
```
$(document).ready(function(){
    $('#file').change(function(){
        if (this.files.length > 0) {
            $('#upload-button').prop('disabled', false).css('opacity', 1);
        } else {
            $('#upload-button').prop('disabled', true).css('opacity', 0.5);
        }
    });

    $('#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.');
            }
        });
    });
});
```

3

Create a new file named **`upload.css`**. Type the following code:



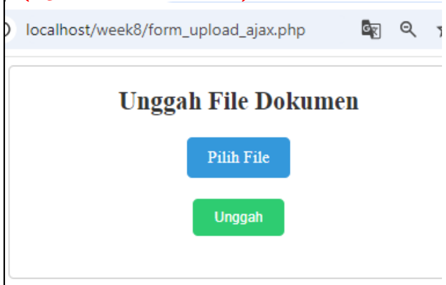
```
.upload-form-container {
    max-width: 400px;
    margin: 0 auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    text-align: center;
}

h2 {
    margin: 0;
    font-size: 24px;
    color: #333;
}

.file-input-container {
    display: flex;
    justify-content: center;
    align-items: center;
    margin: 20px 0;
}

.file-input {
    display: none;
}
```



```css
.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;
}
```

4

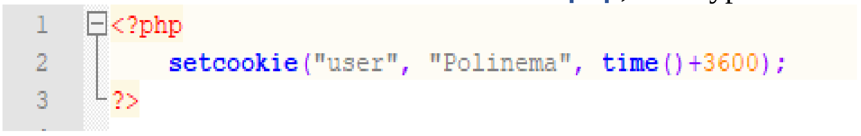
Save the file. Open a browser and run `localhost/week8/form_upload_ajax.php`.
What do you understand from the program code above? Record your understanding. (Question No. 12)

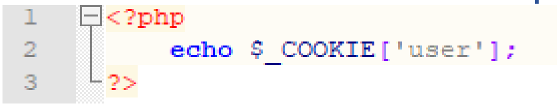


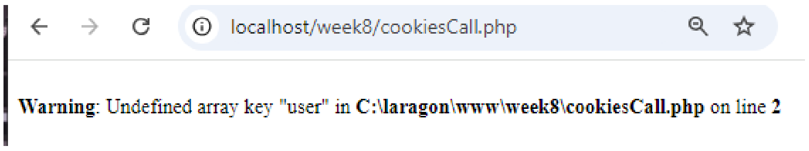
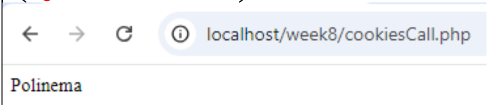
The .upload-form-container centers the form, with a 400px width, padding, a border, and rounded corners. The h2 element is styled with a larger font and dark gray color. The .file-input-container uses flexbox to center its content, while .file-input hides the actual file input element, making the .file-label act as a clickable, button-like area with a blue background and white text. The .upload-button is styled with a green background, white text, and rounded corners, but appears faded and disabled initially due to reduced opacity. When the button is disabled, it turns gray with a not-allowed cursor. Lastly, the .upload-status section displays status messages with bold text and margin for spacing after an upload attempt.

Practical Section 5. Creating *Cookies*

Step	Description

1	Create a new file named `cookiesCreate.php`, then type the following code.
	
	`<?php`
	`    setcookie("user", "Polinema", time()+3600);`
	`?>`

2	Create a new file named `cookiesCall.php`, then type the following code.
	
	`<?php`
	`    echo $_COOKIE['user'];`
	`?>`

3	Open a *browser* and run the program code in step 2 by typing `localhost/week8/cookiesCall.php`
4	Observe and explain your observations (Question No. 13)  localhost/week8/cookiesCall.php **Warning**: Undefined array key "user" in **C:\laragon\www\week8\cookiesCall.php** on line **2** When I open cookiesCall.php without first creating the cookie using cookiesCreate.php, the "user" cookie does not exist. Since no cookie has been set, $_COOKIE['user'] will not contain a value, leading to an undefined or empty output and error message.
5	Open a *browser* 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)  localhost/week8/cookiesCall.php Polinema When I open cookiesCreate.php, a "user" cookie with the value "Polinema" is created and stored in the browser for 1 hour. Upon reopening cookiesCall.php, the script will now find the "user" cookie in the $_COOKIE array, and the output will display "Polinema".
8	*Restart* 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) When I restart my computer and restart Apache, the cookie still persist because cookies are stored on the client (browser) side, not on the server. Then I reopen cookiesCall.php, it still display "Polinema" as long as the cookie has not expired (which occurs 1 hour after it was created). However, if the browser has been cleared or the cookie has expired, the value won't be available, and $_COOKIE['user'] will be undefined again.

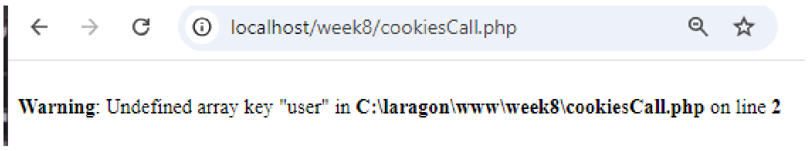
Practical Section 6. Deleting the Value of *Cookies*

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 `cookiesDel.php` name, then type the following code.
	`<?php`
	`    setcookie("user", "Polinema", time()-3600);`
	`?>`

2	Open a *browser* and run the program code by typing `localhost/week8/cookiesDel.php`
3	Open a *browser* and run the program code from the part 5 Practical Section by typing `localhost/week8/cookiesCall.php`

4	Observe and describe the results from steps 2 and 3, then draw conclusions. (Question No. 16)
	
	`localhost/week8/cookiesCall.php`
	`Warning: Undefined array key "user" in C:\laragon\www\week8\cookiesCall.php on line 2`
	When you open cookiesDel.php, the script deletes the "user" cookie by setting its expiration time to a point in the past, causing the browser to remove the cookie. After that, if you open cookiesCall.php, the cookie will no longer exist in the $_COOKIE array. As a result, cookiesCall.php will not be able to retrieve the "user" cookie's value, leading to either no output or an undefined index error

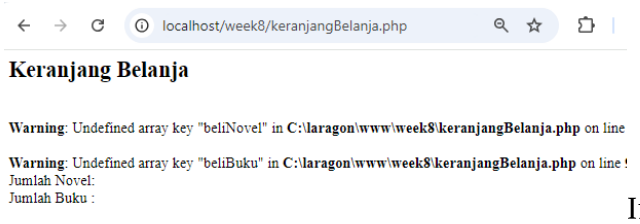
Practical Section 7. Application of *Cookies* to the Shopping Cart Feature

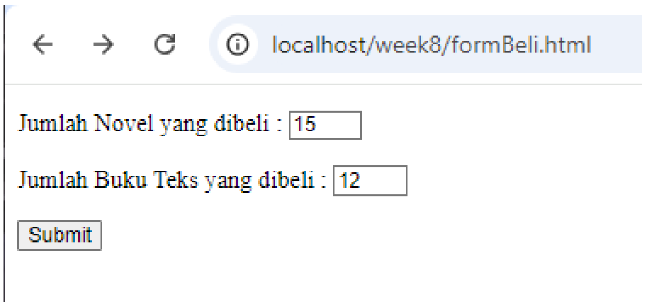
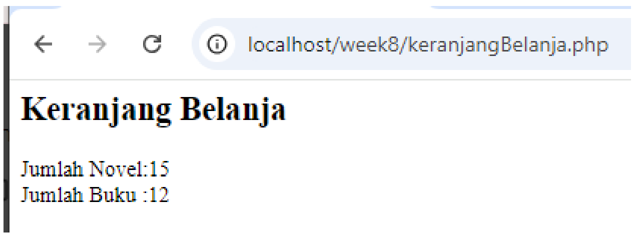
One example of the use of *cookies* is the "shopping cart" feature on the online store web application. The shopping cart contains the items that the user will buy. *Cookies* are used to remember the number of items selected by the user. Here is an example of the use *of cookies* in the shopping cart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

Step	Description

1	Create a new file with **`formBeli.html`** name, then type the following code.
	`<html>`
	`    <head>`
	`    </head>`
	`    <body>`
	`        <form action="prosesBeli.php" method="POST">`
	`            <p> Jumlah Novel yang dibeli :`
	`            <input type="text" name="beliNovel" value= "0" size="2"> </p>`
	`            <p> Jumlah Buku Teks yang dibeli :`
	`            <input type="text" name="beliBuku" value= "0" size="2"> </p>`
	`            <input type="submit">`
	`        </form>`
	`    </body>`
	`</html>`

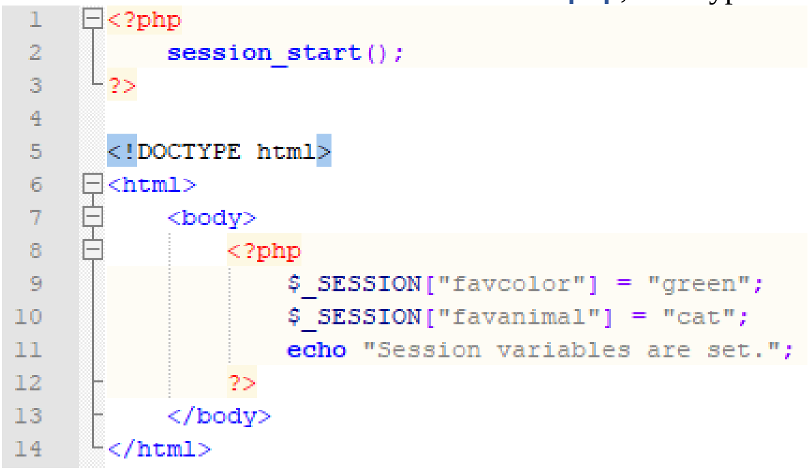
2	Create a new file named **`prosesBeli.php`**, then type the following code.
	`<?php`
	`    if(isset($_POST["beliNovel"]) && isset($_POST["beliBuku"])){`
	`        setcookie("beliNovel", $_POST["beliNovel"]);`
	`        setcookie("beliBuku", $_POST["beliBuku"]);`
	`        header("location:keranjangBelanja.php");`
	`    }`
	`?>`

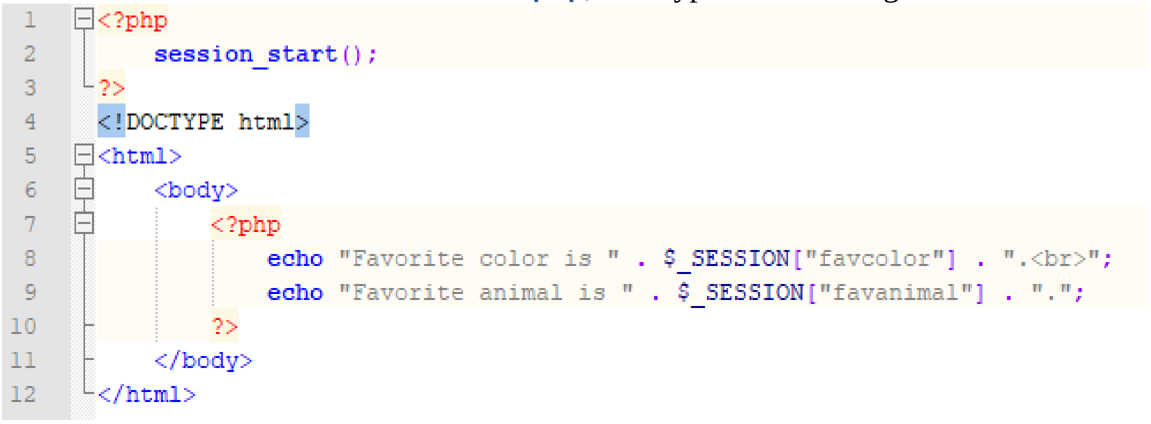
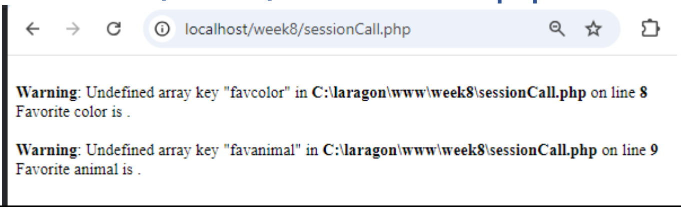
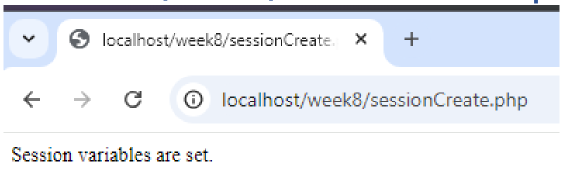
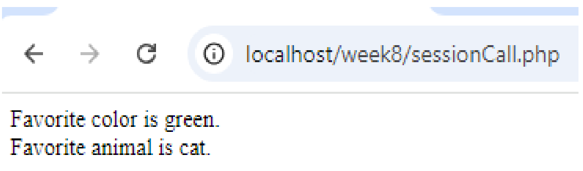
3	Create a new file named **`keranjangBelanja.php`**, then type the following code. `<html>` `    <head>` `    </head>` `    <body>` `        <h2> Keranjang Belanja </h2>` `        <?php` `            $beliNovel=$_COOKIE['beliNovel'];` `            $beliBuku = $_COOKIE['beliBuku'];` `            echo "Jumlah Novel:" . $beliNovel ." ";` `            echo "Jumlah Buku :" . $beliBuku ;` `        ?>` `    </body>` `</html>`
4	Open *a browser* and run the program code step 3 by typing **`localhost/week8/keranjangBelanja.php`**
5	Observe and explain the results displayed. (Question No. 17)  localhost/week8/keranjangBelanja.php **Keranjang Belanja** **Warning**: Undefined array key "beliNovel" in **C:\laragon\www\week8\keranjangBelanja.php** on line **Warning**: Undefined array key "beliBuku" in **C:\laragon\www\week8\keranjangBelanja.php** on line Jumlah Novel: Jumlah Buku : If you access keranjangBelanja.php

	without first submitting the form in formBeli.html and processing the data in prosesBeli.php, it will lead to errors or undefined behavior. This is because the cookies "beliNovel" and "beliBuku" are set in prosesBeli.php after the form submission, and without these cookies, keranjangBelanja.php will encounter issues when trying to access them. If error reporting is enabled, "Undefined index" warnings will appear, while if it's suppressed, the page will display empty or missing values.
6	Run the program code step 1 by typing `localhost/week8/formBeli.html`
7	Fill in the number of novels and textbooks you want to buy and then click the "submit" button.
8	Observe and explain the results displayed. (Question No. 18)  localhost/week8/formBeli.html Jumlah Novel yang dibeli : 15 Jumlah Buku Teks yang dibeli : 12 When you run formBeli.html in the browser, fill out the form with the number of novels and textbooks you want to buy, and click "submit," the data is sent to prosesBeli.php. This script checks if the form data is set, then uses the setcookie function to store the values of "beliNovel" and "beliBuku" in cookies. After setting the cookies, it redirects to keranjangBelanja.php, where the values stored in the cookies are retrieved and displayed
9	Close *the browser* then reopen *the browser* then re-run the program code step 3 by typing `localhost/week8/keranjangBelanja.php`
10	Observe and explain the results displayed. (Question No. 19)  localhost/week8/keranjangBelanja.php **Keranjang Belanja** Jumlah Novel:15 Jumlah Buku :12 When you close and reopen the browser, then re-run keranjangBelanja.php, the values previously stored in the cookies (such as "beliNovel" and "beliBuku") will still be accessible, assuming the cookies have not expired or been deleted. Since cookies are stored on the user's computer and persist across browser sessions by default (until their expiration time or manual deletion), the script in keranjangBelanja.php will retrieve the previously saved values from the cookies and display them.

Practical Section 8. Creating a *Session*

Step	Description

1	Create a new file named `sessionCreate.php`, then type the following code.
	
	`<?php`
	`    session_start();`
	`?>`
	
	`<!DOCTYPE html>`
	`<html>`
	`    <body>`
	`        <?php`
	`            $_SESSION["favcolor"] = "green";`
	`            $_SESSION["favanimal"] = "cat";`
	`            echo "Session variables are set.";`
	`        ?>`
	`    </body>`
	`</html>`

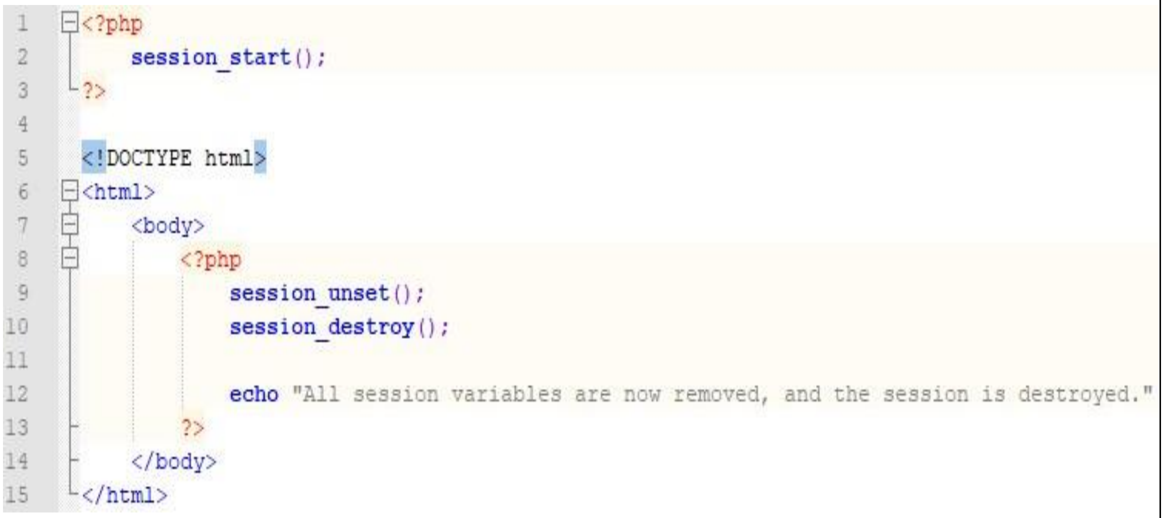
2	Create a new file named `sessionCall.php`, then type the following code.  `<?php` `    session_start();` `?>` `<!DOCTYPE html>` `<html>` `    <body>` `        <?php` `            echo "Favorite color is " . $_SESSION["favcolor"] . ". ";` `            echo "Favorite animal is " . $_SESSION["favanimal"] . ".";` `        ?>` `    </body>` `</html>`
3	Open a *browser* and run the program code in step 2 by typing `localhost/week8/sessionCall.php`  localhost/week8/sessionCall.php **Warning**: Undefined array key "favcolor" in **C:\laragon\www\week8\sessionCall.php** on line **8** Favorite color is . **Warning**: Undefined array key "favanimal" in **C:\laragon\www\week8\sessionCall.php** on line **9** Favorite animal is .
4	Open a *browser* and run the program code step 1 by typing `localhost/week8/sessionCreate.php`  localhost/week8/sessionCreate.php Session variables are set.
5	Repeat step 3  localhost/week8/sessionCall.php Favorite color is green. Favorite animal is cat.

6	Observe and explain the results displayed (Question No. 20) When you first run sessionCall.php without executing sessionCreate.php, the session variables have not been set, so you will encounter errors or undefined behavior as the script attempts to access undefined session variables. After running sessionCreate.php, which starts the session and sets the variables favcolor and favanimal to "green" and "cat," respectively, the message "Session variables are set" is displayed. Upon reopening sessionCall.php, the session variables are now accessible, and the page will display "Favorite color is green." and "Favorite animal is cat." since the session has been properly initialized.

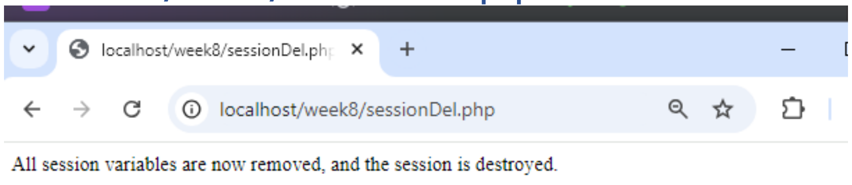
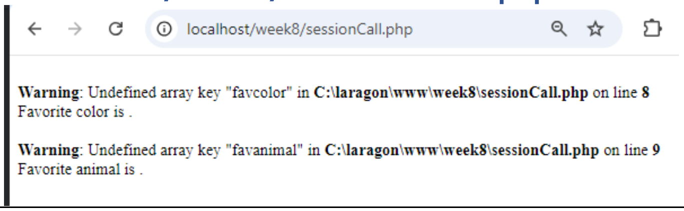
Practical Section 9. Removing Session Values

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.

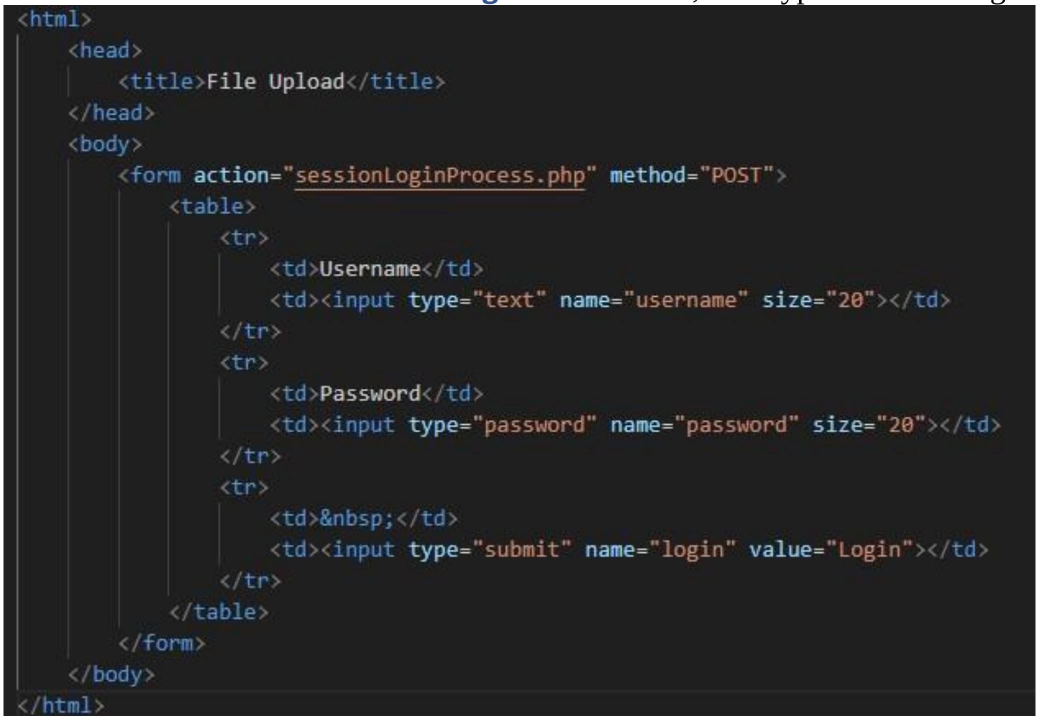


```
<?php
    session_start();
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            session_unset();
            session_destroy();

            echo "All session variables are now removed, and the session is destroyed."
        ?>
    </body>
</html>
```

Step	Description
2	Open a *browser* and run the program code by typing `localhost/week8/sessionDel.php`  localhost/week8/sessionDel.ph localhost/week8/sessionDel.php All session variables are now removed, and the session is destroyed.
3	Open a *browser* and run the program code from the Practical Section 8 by typing `localhost/week8/sessionCall.php`  localhost/week8/sessionCall.php **Warning**: Undefined array key "favcolor" in **C:\laragon\www\week8\sessionCall.php** on line **8** Favorite color is . **Warning**: Undefined array key "favanimal" in **C:\laragon\www\week8\sessionCall.php** on line **9** Favorite animal is .
4	Observe and explain the results displayed. (Question No. 21) When you run sessionDel.php, it starts the session, removes all session variables using session_unset(), and destroys the session with session_destroy(), displaying the message "All session variables are now removed, and the session is destroyed." After

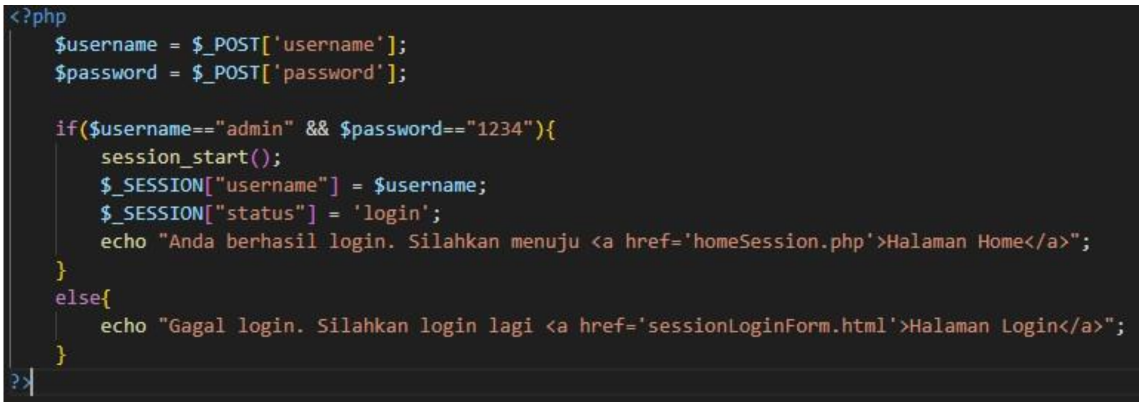
	this, if you run sessionCall.php, the session variables favcolor and favanimal no longer exist because the session was terminated. As a result, sessionCall.php will produce errors or display nothing

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

Step	Description
1	Create a new file named **`sessionLoginForm.html`**, then type the following code 

```
<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>
```

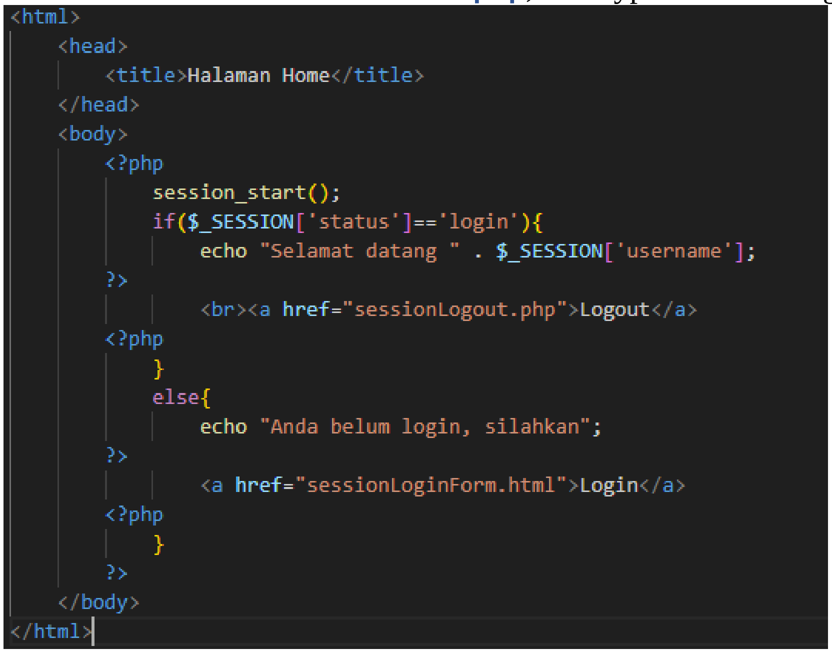
Step	Description
2	Create a file named **`sessionLoginProcess.php`**, then type the following code.



```php
<?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 <a href='homeSession.php'>Halaman Home</a>";
    }
    else{
        echo "Gagal login. Silahkan login lagi <a href='sessionLoginForm.html'>Halaman Login</a>";
    }
?>
```

3

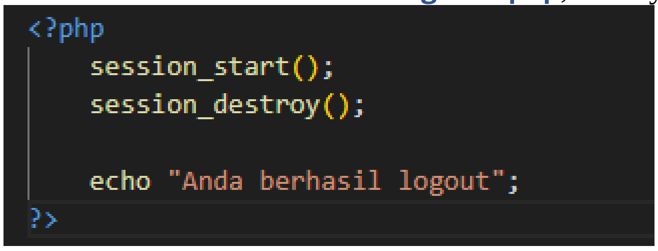
Create a file named **homeSession.php**, then type the following code.



```html
<html>
    <head>
        <title>Halaman Home</title>
    </head>
    <body>
        <?php
            session_start();
            if($_SESSION['status']=='login'){
                echo "Selamat datang " . $_SESSION['username'];
        ?>
                <br><a href="sessionLogout.php">Logout</a>
        <?php
            }
            else{
                echo "Anda belum login, silahkan";
        ?>
                <a href="sessionLoginForm.html">Login</a>
        <?php
            }
        ?>
    </body>
</html>
```

4

Create a file named **sessionLogout.php**, then type the following code.



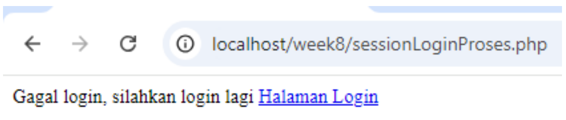
```php
<?php
    session_start();
    session_destroy();

    echo "Anda berhasil logout";
?>
```

5

Open a *browser* and run the program code by typing **localhost/week8/sessionLoginForm.html**

6

Log in using your email username and password "0000".



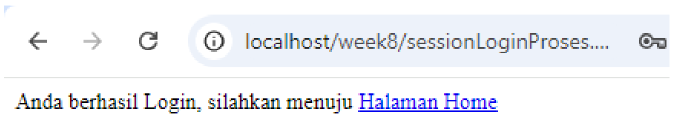
localhost/week8/sessionLoginProses.php

Gagal login, silahkan login lagi Halaman Login

7

Observe and explain the results displayed
(Question No. 22)

The form submits the POST data to sessionLoginProses.php. In sessionLoginProses.php, the username and password are compared with hardcoded values: the correct username is "admin" and the correct password is "1234". Since we entered incorrect credentials (email username and "0000"), it will fail.

8	Re-run the program code by typing `localhost/week8/sessionLoginForm.html` Log in using the username "admin" and password "1234".  localhost/week8/sessionLoginProses.... Anda berhasil Login, silahkan menuju Halaman Home
9	Observe and explain the results displayed (Question No. 23) When you re-run sessionLoginForm.html and log in using the correct username "admin" and password "1234", the following happens: 1. The form sends the correct POST data to sessionLoginProses.php. 2. The script checks if the submitted username and password match the correct hardcoded values. 3. Since the credentials are correct, the script will starts a session using session_start(). Then sets session variables $_SESSION["username"] = $username; and $_SESSION["status"] = "login";. You will see the success message afterwards
10	Describe the sequence of the process from login to logout (also mention the order in which the files are processed) (Question No. 24) Sequence of the process from login to logout: 1. Login Form Display (sessionLoginForm.html): The user first interacts with the sessionLoginForm.html, which contains a form asking for a username and password. Once the form is filled out and submitted, it sends the data to sessionLoginProses.php via the POST method. 2. Login Processing (sessionLoginProses.php): ○ This script receives the submitted username and password. ○ It checks if the submitted values match the correct hardcoded credentials (admin and 1234). ○ If the credentials are correct, the script starts a session using session_start(), sets session variables, and displays a message indicating successful login along with a link to the home page (homeSession.php). ○ If the credentials are incorrect, the user is redirected to the login page with a failure message. 3. Home Page (homeSession.php): ○ When the user clicks the link to homeSession.php, the script checks if the session is active and if the session variable $_SESSION['status'] is set to "login". ○ If the session is valid, the user is greeted with a welcome message that includes the username stored in the session, and a "Logout" link is displayed. ○ If the session is not valid, the user is prompted to log in again. 4. Logout (sessionLogout.php): ○ When the user clicks the "Logout" link, it directs them to sessionLogout.php. ○ This script starts the session, destroys all session data using session_destroy(), and displays a message indicating that the user has successfully logged out. So in summary, the order look likes this: • Step 1: User opens sessionLoginForm.html and submits the form. • Step 2: sessionLoginProses.php processes the login request. • Step 3: On successful login, homeSession.php is accessed, which checks session

	status. • Step 4: User logs out through sessionLogout.php, which destroys the session and confirms logout.