Submitting Files with PHP Forms

To submit files with a PHP form, you will need to set the enctype attribute of the form to multipart/form-data. This will tell the browser to encode the form data in a special way that allows files to be uploaded.

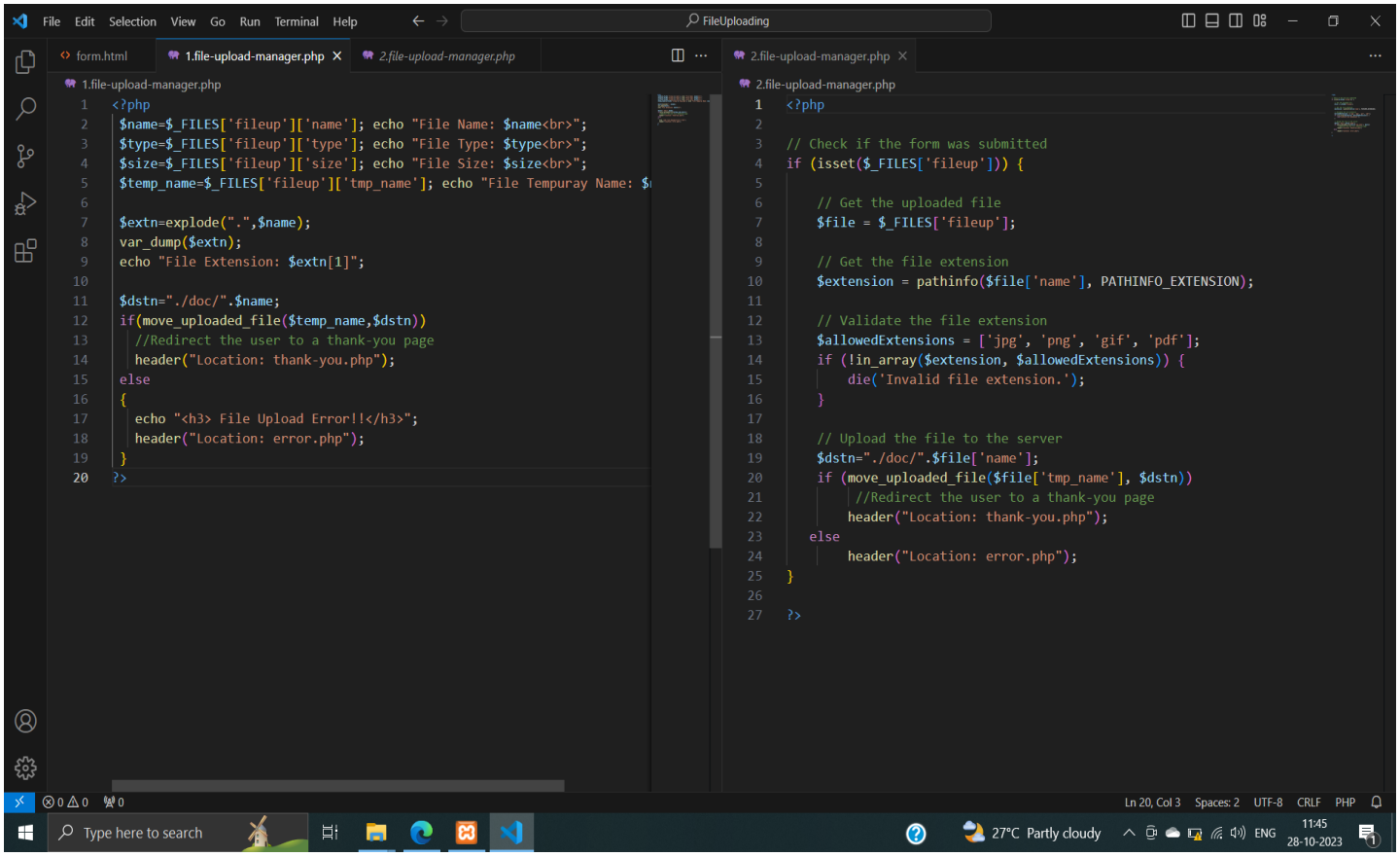
You will also need to add a file input field to the form. This is a special type of input field that allows the user to select files from their computer to upload.

Once the form has been submitted, the uploaded files will be available in the $_FILES superglobal variable.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
</head>
<body>
                <!-- 2.file-upload-manager.php  -->
    <form action="2.file-upload-manager.php" method="post"
enctype="multipart/form-data">
        <label for="fileSelect">File Upload: </label>
        <input type="file" name="fileup" id="fileupload"></br></br>
        <input type="submit" name="submit" value="Upload"></form></br>
    </form>
</body>
</html>
```

CCreate a PHP script (`file-upload-manager.php`) that will handle the file upload. This script will check if the file is of an allowed type, move it to a specific directory, and then redirect the user to a success or error page.



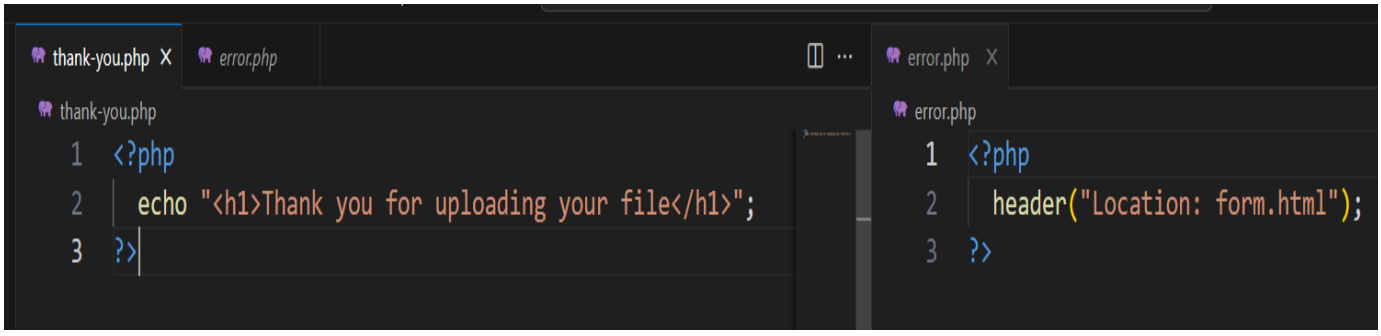

```php
<?php



// Check if the form was submitted

if (isset($_FILES['fileup'])) {

    // Get the uploaded file

    $file = $_FILES['fileup'];

    // Get the file extension

    $extension = pathinfo($file['name'], PATHINFO_EXTENSION);

    // Validate the file extension

    $allowedExtensions = ['jpg', 'png', 'gif', 'pdf'];

    if (!in_array($extension, $allowedExtensions)) {
```

```
        die('Invalid file extension.');

    }

    // Upload the file to the server

    $dstn="./doc/".$file['name'];

    if (move_uploaded_file($file['tmp_name'], $dstn))

         //Redirect the user to a thank-you page

        header("Location: thank-you.php");

   else

        header("Location: error.php");

}

?>
```

Redirecting Forms After Submission

To redirect a form after submission in PHP, you can use the header() function. This function sends a header to the browser instructing it to redirect to a new URL.

For example, the following code will redirect the user to the thank-you.php page after they submit the form:



It is important to note that you should redirect the user before you output any content to the browser. This is because the browser will ignore any content that is output after the header()

function is called.

PHP File Submission and Redirect Form:

HTML Form: Create an HTML form with the `<form>` element, allowing users to select and upload files using `<input type="file">`.

Form Submission: Set the form's `action` attribute to a PHP script that will handle the file upload. Use `enctype="multipart/form-data"` to support file uploads.

PHP Handling Script: In the PHP script (`upload.php`), use `$_FILES` to access the uploaded file information. Check file type and size, and define the allowed file types.

Validation: Check if the uploaded file is of an allowed type and size. Display error messages or proceed with file processing accordingly.

File Upload: If validation is successful, move the file from its temporary location to a designated directory using `move_uploaded_file`.

Success Page: After a successful file upload, redirect users to a success page (e.g., `success.php`) using `header("Location: success.php")`.

Error Handling: Create an error page (`error.php`) to handle cases where file uploads fail. Display appropriate error messages.

File Storage: Ensure the "uploads" directory is created to store the uploaded files.

Benefits of Redirecting Forms in PHP

- It can improve the user experience by providing feedback on the success or failure of the form submission.
- It can prevent the user from resubmitting the form accidentally.
- It can protect the form from being submitted multiple times.
- It can be used to redirect the user to a different page depending on the outcome of the form submission.

For example, if the user is submitting a contact form, you could redirect them to a confirmation page after the form has been submitted successfully. Or, if the user is submitting a payment form, you could redirect them to a payment success page after the

payment has been processed.

Types of Forms That Can Be Submitted in PHP

- Plain text forms
- Forms with file uploads
- Contact forms
- Login forms
- Payment forms
- Survey forms