Lab 05: Handling File Upload

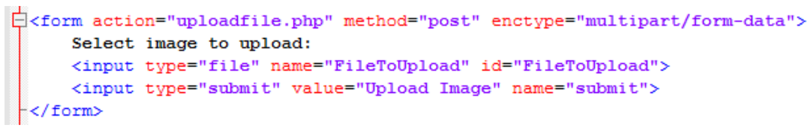
In this lab exercise, you will learn how to handle files that are uploaded by the user. This exercise does not cover handling file upload into a database table (this will be covered in other lab exercise).

1.1 Preparing the HTML Form

Unlike exercises you have done in previous session where data submitted were of a text format, any file being uploaded is of a binary format. Hence, files are generally uploaded using POST method.

In the HTML form tag, you need to include the encoding type: enctype=”multipart/form-data” to reflect the nature of data format being transmitted. The type="file" attribute of the <input> tag shows the input field as a file-select control, with a "Browse" button next to the input control.

HTML form sample is shown in Figure 1 below.



```
<form action="uploadfile.php" method="post" enctype="multipart/form-data">
    Select image to upload:
    <input type="file" name="FileToUpload" id="FileToUpload">
    <input type="submit" value="Upload Image" name="submit">
</form>
```

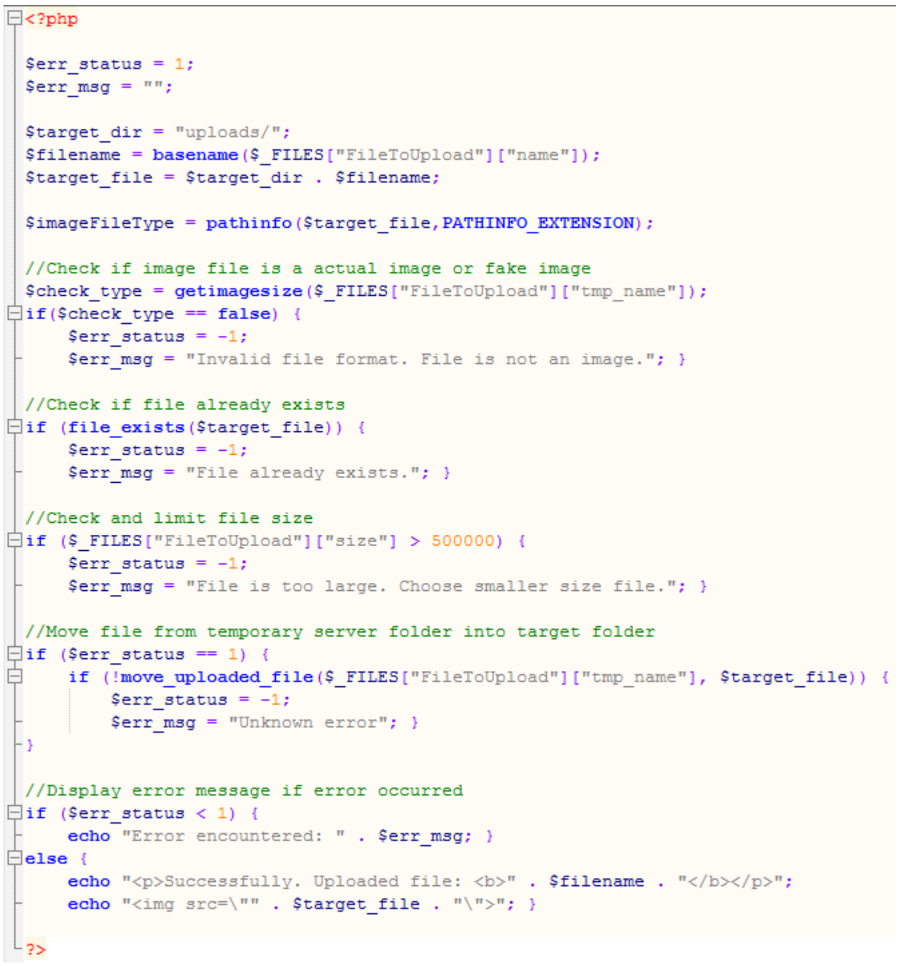
Figure 1: HTML form for uploading a file.

1.2 Writing the PHP Codes to Upload a Single File

Each file uploaded by the user if stored in an array variable containing its binary and file properties. These properties include file name, file size, file type, and more.

Note that uploaded files reside in the application temporary folder, having a temporary name. They are moved into targeted folder using keyword: move_uploaded_file.

PHP codes sample is shown in Figure 2 below.



```php
<?php

$err_status = 1;
$err_msg = "";

$target_dir = "uploads/";
$filename = basename($_FILES["FileToUpload"]["name"]);
$target_file = $target_dir . $filename;

$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);

//Check if image file is a actual image or fake image
$check_type = getimagesize($_FILES["FileToUpload"]["tmp_name"]);
if($check_type == false) {
    $err_status = -1;
    $err_msg = "Invalid file format. File is not an image."; }

//Check if file already exists
if (file_exists($target_file)) {
    $err_status = -1;
    $err_msg = "File already exists."; }

//Check and limit file size
if ($_FILES["FileToUpload"]["size"] > 500000) {
    $err_status = -1;
    $err_msg = "File is too large. Choose smaller size file."; }

//Move file from temporary server folder into target folder
if ($err_status == 1) {
    if (!move_uploaded_file($_FILES["FileToUpload"]["tmp_name"], $target_file)) {
        $err_status = -1;
        $err_msg = "Unknown error"; }
}

//Display error message if error occurred
if ($err_status < 1) {
    echo "Error encountered: " . $err_msg; }
else {
    echo "<p>Successfully. Uploaded file: <b>" . $filename . "</b></p>";
    echo "<img src=\"" . $target_file . "\">"; }

?>
```

Figure 2: Sample codes to upload an image file to server folder and subsequently displaying it on the web browser. It checks several condition by using error status.

1.3 Error Messages during File Upload

During file upload process, PHP returns an appropriate error code along with the file array. The error code can be found in the error segment of the file array:
$_FILES[“<userfile>”][“error”]

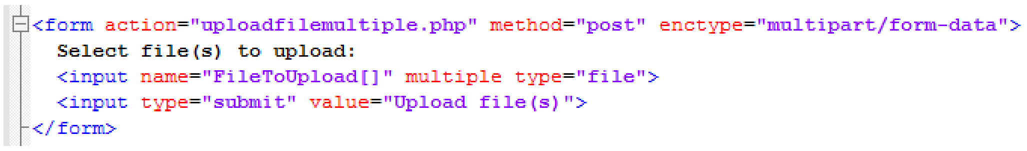
Below are the error codes and their corresponding meanings:

- UPLOAD_ERR_OK
  Value: 0; There is no error, the file uploaded with success.
- UPLOAD_ERR_INI_SIZE
  Value: 1; The uploaded file exceeds the upload_max_filesize directive in php.ini.
- UPLOAD_ERR_FORM_SIZE
  Value: 2; The uploaded file exceeds the MAX_FILE_SIZE directive that was specified in the HTML form.

- UPLOAD_ERR_PARTIAL
  Value: 3; The uploaded file was only partially uploaded.
- UPLOAD_ERR_NO_FILE
  Value: 4; No file was uploaded.
- UPLOAD_ERR_NO_TMP_DIR
  Value: 6; Missing a temporary folder. Introduced in PHP 5.0.3.
- UPLOAD_ERR_CANT_WRITE
  Value: 7; Failed to write file to disk. Introduced in PHP 5.1.0.
- UPLOAD_ERR_EXTENSION
  Value: 8; A PHP extension stopped the file upload.

1.4 Uploading Multiple Files

Multiple files can be uploaded at once. At the HTML page, add a keyword multiple at the input tag. Input name needs to be an array format, otherwise only first file will be transmitted. Refer to Figure 3 below for an example how to format the HTML form.



```
<form action="uploadfilemultiple.php" method="post" enctype="multipart/form-data">
  Select file(s) to upload:
  <input name="FileToUpload[]" multiple type="file">
  <input type="submit" value="Upload file(s)">
</form>
```

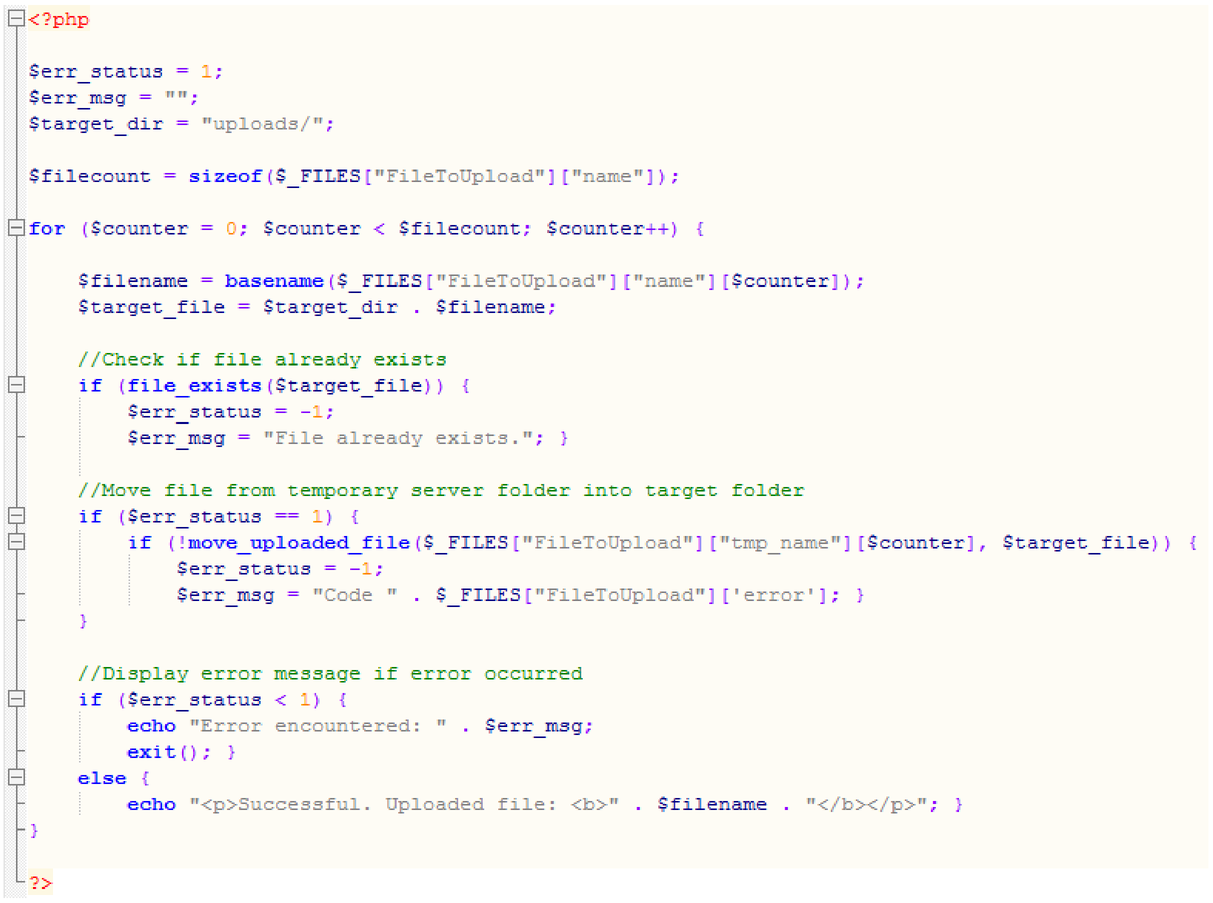
Figure 3: HTML form to upload multiple files at once. Keyword multiple is used.

Similar to single file upload, multiple files upload uses the same method. The only thing differs is that in multiple files upload, PHP stores the uploaded files in an array.

Format of the array: $_FILES [“<HTML input tag name>”] [“<property>”] [<index number>]

Example: $_FILES ["FileToUpload"] ["name"] [0]

Figure 4 below shows an example of how to get the number of files and perform a loop to upload each file.



```php
<?php

$err_status = 1;
$err_msg = "";
$target_dir = "uploads/";

$filecount = sizeof($_FILES["FileToUpload"]["name"]);

for ($counter = 0; $counter < $filecount; $counter++) {

    $filename = basename($_FILES["FileToUpload"]["name"][$counter]);
    $target_file = $target_dir . $filename;

    //Check if file already exists
    if (file_exists($target_file)) {
        $err_status = -1;
        $err_msg = "File already exists."; }

    //Move file from temporary server folder into target folder
    if ($err_status == 1) {
        if (!move_uploaded_file($_FILES["FileToUpload"]["tmp_name"][$counter], $target_file)) {
            $err_status = -1;
            $err_msg = "Code " . $_FILES["FileToUpload"]['error']; }
    }

    //Display error message if error occurred
    if ($err_status < 1) {
        echo "Error encountered: " . $err_msg;
        exit(); }
    else {
        echo "<p>Successful. Uploaded file: <b>" . $filename . "</b></p>"; }
}

?>
```

Figure 4: PHP code to upload multiple files.

You tasks:

1. Create a page that allows the user to select and upload a single file.
2. Also allows the user to choose destination name (name to save as in the server).
3. Provide options (combo box / radio button / check box) for the user to overwrite if file already exists.
4. Display on web browser the link to download the uploaded file.
5. Repeat the above for multiple files upload.

-- END --