**Practical – 6**

**Aim: Perform following programs of PHP file handling.**

**Exercise:**

1. **Write a PHP program to create a student registration form with validation for fields including Enrollment No., Full Name, Gender, Mobile, Email, Address, and Profile Photo. Implement validation for Full Name, Mobile, Email, and Profile Photo inputs. Store all submitted data in a CSV file.**

<!DOCTYPE html>

<html>

<head>

<title>Student Registration Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

#container {

background-color: #fff;

padding: 20px;

border-radius: 10px;

box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);

width: 400px;

}

h2 {

text-align: center;

}

input[type="text"], select, input[type="file"] {

width: 100%;

padding: 10px;

margin: 8px 0;

box-sizing: border-box;

border: 1px solid #ccc;

border-radius: 4px;

}

input[type="submit"] {

background-color: #4CAF50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

width: 100%;

}

input[type="submit"]:hover {

background-color: #45a049;

}

.error {

color: red;

font-size: 12px;

}

</style>

</head>

<body>

<div id="container">

<h2>Student Registration Form</h2>

<form method="post" enctype="multipart/form-data">

<input type="text" name="enrollment\_no" placeholder="Enrollment No." required><br>

<input type="text" name="full\_name" placeholder="Full Name" required><br>

<select name="gender" required>

<option value="">Select Gender</option>

<option value="male">Male</option>

<option value="female">Female</option>

</select><br>

<input type="text" name="mobile" placeholder="Mobile" required><br>

<input type="text" name="email" placeholder="Email" required><br>

<input type="text" name="address" placeholder="Address"><br>

<input type="file" name="profile\_photo" accept="image/\*" required><br>

<input type="submit" name="submit" value="Register">

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Validate and store data

$errors = array();

$enrollment\_no = $\_POST['enrollment\_no'];

$full\_name = $\_POST['full\_name'];

$gender = $\_POST['gender'];

$mobile = $\_POST['mobile'];

$email = $\_POST['email'];

$address = $\_POST['address'];

$profile\_photo = $\_FILES['profile\_photo']['name'];

$tmp\_profile\_photo = $\_FILES['profile\_photo']['tmp\_name'];

// Validate full name

if (!preg\_match("/^[a-zA-Z\s]+$/", $full\_name)) {

$errors['full\_name'] = "Invalid full name";

}

// Validate mobile

if (!preg\_match("/^[0-9]{10}$/", $mobile)) {

$errors['mobile'] = "Invalid mobile number";

}

// Validate email

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$errors['email'] = "Invalid email address";

}

// Validate profile photo

$allowed\_extensions = array('jpg', 'jpeg', 'png', 'gif');

$file\_extension = strtolower(pathinfo($profile\_photo, PATHINFO\_EXTENSION));

if (!in\_array($file\_extension, $allowed\_extensions)) {

$errors['profile\_photo'] = "Invalid file format. Allowed formats: jpg, jpeg, png, gif";

}

// If no errors, store data in CSV file

if (empty($errors)) {

$csv\_data = array($enrollment\_no, $full\_name, $gender, $mobile, $email, $address, $profile\_photo);

$file = fopen("student\_data.csv", "a");

fputcsv($file, $csv\_data);

fclose($file);

move\_uploaded\_file($tmp\_profile\_photo, "profile\_photos/$profile\_photo");

echo "<p>Registration successful!</p>";

} else {

// Display validation errors

echo "<div class='error'>";

foreach ($errors as $error) {

echo "<p>$error</p>";

}

echo "</div>";

}

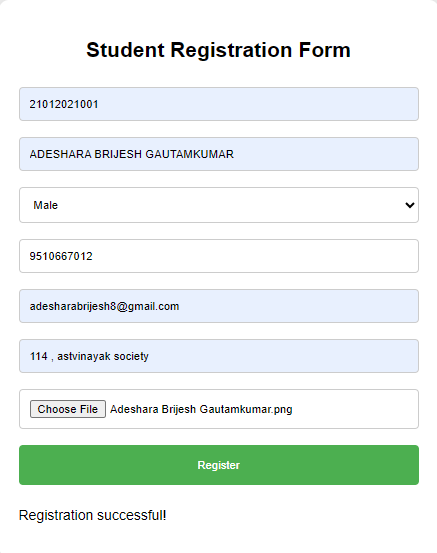
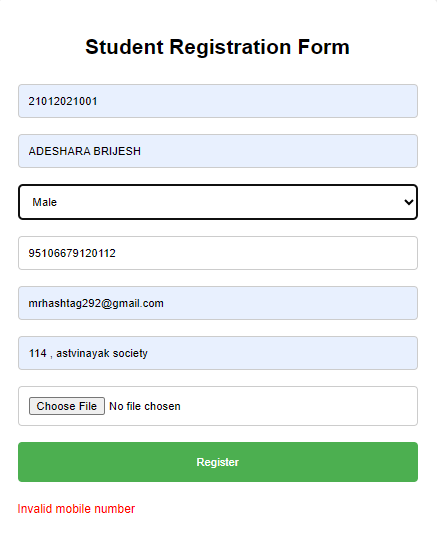
}

?>

</div>

</body>

</html>

**** ****

****

1. **Create a form with a file input field to allow users to upload files. Implement server-side PHP code to handle the uploaded file:**
2. **Validate the file type and size to prevent malicious uploads and ensure server resource management.**
3. **Move the uploaded file to a secure location on the server.**
4. **Provide feedback to the user on the success or failure of the upload process.**

<!DOCTYPE html>

<html>

<head>

<title>File Upload Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

#container {

background-color: #fff;

padding: 20px;

border-radius: 10px;

box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);

width: 400px;

}

h2 {

text-align: center;

}

input[type="file"] {

width: 100%;

padding: 10px;

margin: 8px 0;

box-sizing: border-box;

border: 1px solid #ccc;

border-radius: 4px;

}

input[type="submit"] {

background-color: #4CAF50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

width: 100%;

}

input[type="submit"]:hover {

background-color: #45a049;

}

.error {

color: red;

font-size: 12px;

}

.success {

color: green;

font-size: 14px;

font-weight: bold;

text-align: center;

}

</style>

</head>

<body>

<div id="container">

<h2>File Upload Form</h2>

<form method="post" enctype="multipart/form-data">

<input type="file" name="uploaded\_file" required><br>

<input type="submit" name="submit" value="Upload File">

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Check if file is uploaded successfully

if (isset($\_FILES["uploaded\_file"]) && $\_FILES["uploaded\_file"]["error"] == UPLOAD\_ERR\_OK) {

$uploaded\_file\_name = $\_FILES["uploaded\_file"]["name"];

$uploaded\_file\_tmp = $\_FILES["uploaded\_file"]["tmp\_name"];

$uploaded\_file\_size = $\_FILES["uploaded\_file"]["size"];

$uploaded\_file\_type = $\_FILES["uploaded\_file"]["type"];

// Validate file type

$allowed\_types = array("image/jpeg", "image/png", "image/gif", "application/pdf");

if (!in\_array($uploaded\_file\_type, $allowed\_types)) {

echo "<p class='error'>Error: Only JPG, PNG, GIF, or PDF files are allowed.</p>";

exit;

}

// Validate file size (limit to 5MB)

$max\_file\_size = 5 \* 1024 \* 1024; // 5MB in bytes

if ($uploaded\_file\_size > $max\_file\_size) {

echo "<p class='error'>Error: File size exceeds the maximum limit of 5MB.</p>";

exit;

}

// Move the uploaded file to a secure location

$upload\_dir = "uploads/";

$destination\_file = $upload\_dir . basename($uploaded\_file\_name);

if (move\_uploaded\_file($uploaded\_file\_tmp, $destination\_file)) {

echo "<p class='success'>File uploaded successfully.</p>";

} else {

echo "<p class='error'>Error: Failed to move the uploaded file.</p>";

}

} else {

echo "<p class='error'>Error: No file uploaded or an error occurred during upload.</p>";

}

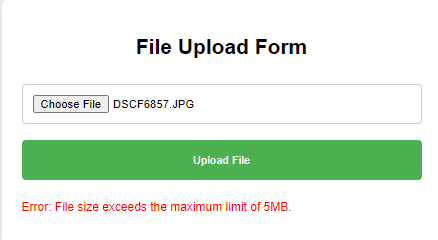
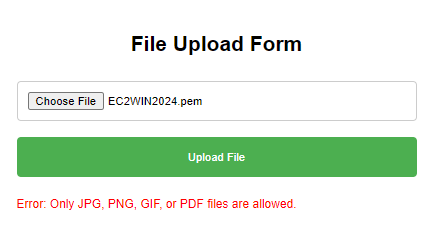
}

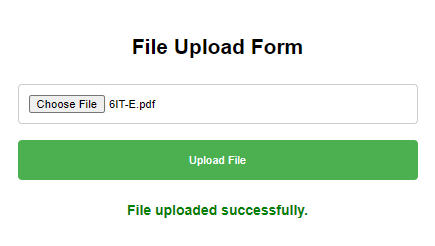
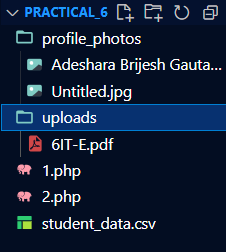
?>

</div>

</body>

</html>

**** ****

**** ****

1. **Develop functionality to manage uploaded files:**
   1. **List uploaded files with options for viewing, downloading, or deleting them.**

**Manage\_files.php**

<!DOCTYPE html>

<html>

<head>

<title>Manage Uploaded Files</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

margin: 0;

padding: 20px;

}

h2 {

text-align: center;

}

table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

.actions {

display: flex;

justify-content: center;

}

.actions a {

margin-right: 10px;

}

</style>

</head>

<body>

<h2>Uploaded Files</h2>

<?php

// Directory where uploaded files are stored

$upload\_dir = "uploads/";

// Check if the directory exists

if (is\_dir($upload\_dir)) {

// Open directory

if ($dh = opendir($upload\_dir)) {

// Display table header

echo "<table>";

echo "<tr><th>Filename</th><th>Size</th><th>Actions</th></tr>";

// Read files from the directory

while (($file = readdir($dh)) !== false) {

// Skip directories and hidden files

if ($file != "." && $file != "..") {

// Get file size

$file\_size = filesize($upload\_dir . $file);

// Format file size

$size\_units = array('B', 'KB', 'MB', 'GB', 'TB');

$size\_index = 0;

while ($file\_size >= 1024 && $size\_index < count($size\_units) - 1) {

$file\_size /= 1024;

$size\_index++;

}

$formatted\_size = round($file\_size, 2) . " " . $size\_units[$size\_index];

// Display file details in table rows

echo "<tr>";

echo "<td>$file</td>";

echo "<td>$formatted\_size</td>";

echo "<td class='actions'><a href='uploads/$file' target='\_blank'>View</a> <a href='download.php?file=$file'>Download</a> <a href='delete.php?file=$file'>Delete</a></td>";

echo "</tr>";

}

}

// Close directory and table

closedir($dh);

echo "</table>";

} else {

echo "<p>Error: Unable to open directory.</p>";

}

} else {

echo "<p>Error: Upload directory does not exist.</p>";

}

?>

</body>

</html>

**Download.php**

<?php

// File to be downloaded

$file = $\_GET['file'];

// Path to the file

$file\_path = "uploads/" . $file;

// Check if the file exists

if (file\_exists($file\_path)) {

// Set headers to force download

header("Content-type: application/octet-stream");

header("Content-Disposition: attachment; filename=\"$file\"");

// Read the file and output its contents

readfile($file\_path);

exit;

} else {

echo "Error: File not found.";

}

?>

**Delete.php**

<?php

$file = $\_GET['file'];

$file\_path = "uploads/" . $file;

if (file\_exists($file\_path)) {

if (unlink($file\_path)) {

echo "File deleted successfully.";

} else {

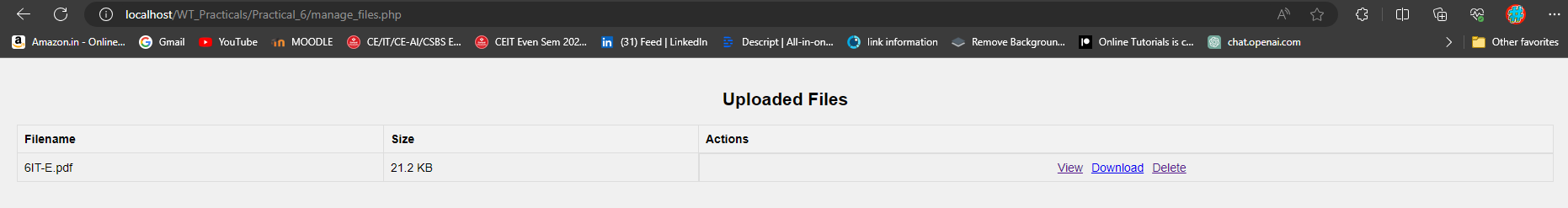
echo "Error: Failed to delete file.";

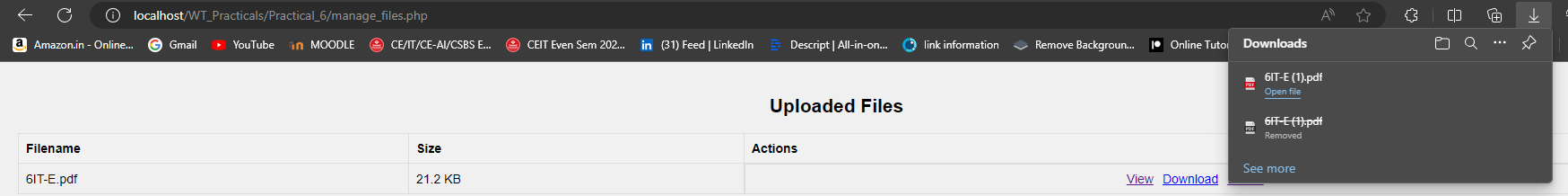
}

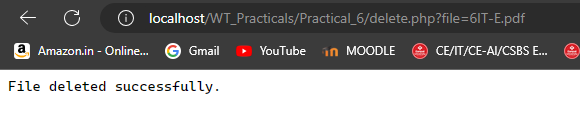
} else {

echo "Error: File not found."; }

?>

****

****

****