**Batch: A3 Roll No.: 16010121045**

**Experiment No. 06**

|  |
| --- |
| **TITLE: Demonstrate the File Uploading in PHP** |

**AIM:** To understand various file operations and file upload in PHP

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:** Process external files in PHP.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem Statement: Description of the application implemented with output**:

To design and implement a blog site that allows multiple users to create and manage

their own blogs, while also providing functionality for users to view, search, and

comment on other blogs. The site will support CRUD operations for each entity, along

with the support for querying of blogs by title or content, and all blogs and comments

associated with a specific user and all comments associated with a specific blog.

**PHP Basic Concepts Learned With Syntax**

connection.php (code to connect php to the database)

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "blogsite";

*// Create connection*

$conn = mysqli\_connect($servername, $username, $password, $dbname);

*// Check connection*

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

?>

**Description of the application implemented with output**:

The create page in the blog-site allows users to create a new blog of their own which is stored in the sql database.

html form (to get the input from the user)

<form *action*="../scripts/upload.php" *method*="POST" *enctype*="multipart/form-data">

<input *type*="text" *name*="title" *class*="input" *placeholder*="Write your title here......">

<div *class*="blog-img">

<label *for*="file" *id*="output-label">

<img *src*="../images/img-upload.svg" *alt*="Image-upload" *id*="output">

<span></span>

</label>

<input *type*="file" *accept*="image/\*" *name*="image" *id*="file" *onchange*="loadFile(event)" *style*="display: none; width: 100vh;">

</div>

<textarea *name*="content" *id*="content" *class*="input" *placeholder*="Write your content here......"></textarea>

<input *type*="text" *name*="hashtags" *id*="hashtags" *autocomplete*="off" *placeholder*="Type your hashtags and press space">

<div *class*="tag-container">

</div>

<button *id*="publish-btn" *type*="submit" *class*="btn">PUBLISH</button>

<button *type*="submit" *class*="neu-btn">SAVE DRAFT</button>

</form>

javascript (to store the data temporarily in the browser cache)

var loadFile = function(event) {

var image = document.getElementById('output');

var label = document.getElementById('output-label');

label.innerHTML = "Upload another Image";

image.style.width = "100%";

image.src = URL.createObjectURL(event.target.files[0]);

};

*// Hashtags*

let input, hashtagArray, container, t;

input = document.querySelector('#hashtags');

container = document.querySelector('.tag-container');

hashtagArray = [];

input.addEventListener('keyup', () => {

*if* (event.which == 32 && input.value.length > 0 && input.value.trim() !== '') {

var text = document.createTextNode(input.value);

var p = document.createElement('p');

container.appendChild(p);

p.appendChild(text);

p.classList.add('tag');

hashtagArray.push(input.value);

console.log(hashtagArray)

input.value = '';

let deleteTags = document.querySelectorAll('.tag');

*for* (let i = 0; i < deleteTags.length; i++) {

deleteTags[i].addEventListener('click', () => {

container.removeChild(deleteTags[i]);

hashtagArray.splice(i, 1);

});

}

}

});

document.getElementById('publish-btn').addEventListener('click', function(event) {

event.preventDefault(); *// Prevent the default form submission*

let hiddenInput = document.createElement('input');

hiddenInput.type = 'hidden';

hiddenInput.name = 'hashtags';

hiddenInput.value = JSON.stringify(hashtagArray);

*this*.parentNode.appendChild(hiddenInput);

*this*.parentNode.submit();

});

php (to store the images in local storage)

<?php

*//connect to database*

*include*('../scripts/connection.php');

*//start the session*

session\_start();

*//query to get the max blogid*

$query = "SELECT MAX(blogid) as maxblogid FROM blogs";

$result = mysqli\_query($conn, $query);

$row = mysqli\_fetch\_assoc($result);

*//increment the userid by 1*

$maxid = $row['maxblogid'];

$newid = $maxid + 1;

*//get data from form*

$title = $\_POST["title"];

$content = $\_POST["content"];

$date = date('Y-m-d');

$image = $\_FILES['image']['name'];

$target = "../images/blog\_data/".basename($newid). ".png";

move\_uploaded\_file($\_FILES['image']['tmp\_name'], $target);

echo "$target";

$hashtags = json\_decode($\_POST['hashtags']);

$serializedhash = serialize($hashtags);

$sql = "INSERT INTO blogs (title, content, blogid, userid,date\_of\_upload,blog\_image,hashtags)

VALUES ('$title', '$content', '$newid', '$\_SESSION[user\_id]','$date','$image','$serializedhash')";

*if* (mysqli\_query($conn, $sql)) {

header("Location: ../pages/home.php");

}

*//close connection*

mysqli\_close($conn);

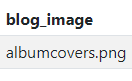
?>

**OUTPUT**

images stored in local storage with the image name as the blogid



path of the images stored in the sql database.



**Post Lab Questions:**

**1. Perform following file operations with respect to problem statement/application:**

Open, read and write with various modes, append, delete and downloading file.

<?php

*// File paths*

$filename = 'example.txt';

$downloadFile = 'example.pdf';

*// Open file for reading*

$file = fopen($filename, 'r');

*if* ($file) {

*// Read file content*

$content = fread($file, filesize($filename));

echo "File content: $content<br>";

*// Close file*

fclose($file);

}

*// Open file for writing*

$file = fopen($filename, 'w');

*if* ($file) {

*// Write content to file*

$content = 'This is some new content.';

fwrite($file, $content);

*// Close file*

fclose($file);

}

*// Open file for appending*

$file = fopen($filename, 'a');

*if* ($file) {

*// Append content to file*

$content = 'This content will be appended.';

fwrite($file, $content);

*// Close file*

fclose($file);

}

*// Delete file*

*if* (file\_exists($filename)) {

unlink($filename);

echo "File deleted successfully.<br>";

}

*// Download file*

*if* (file\_exists($downloadFile)) {

header('Content-Type: application/pdf');

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

readfile($downloadFile);

}