**Step 11: Implement the File Upload**

In this section we will handle file upload. This is where we will attach images to our posts in the database and display the posts with images in the view files. Let’s get started.

**Update the Post.php Model**

<?php

namespace app\models;

// Include the bootstrap file for global setup

require\_once BASE\_DIR . '/bootstrap.php';

// Define class for managing posts

class Post {

    private $db; // Database connection object

    // Initialize connection in constructor

    public function \_\_construct() {

        $this->db = new \Database; // Create a Database object for connection

    }

   /\*\*

     \* Create a new post with title and content.

     \*

     \* @param string $title The title of the post.

     \* @param string $content The content of the post.

     \*/

    public function createPost($title, $content, $category\_id, $imageFilename = null) {

      // Prepare an SQL statement to insert a new post into the 'posts' table

      $stmt = $this->db->conn->prepare('INSERT INTO posts (title, content, category\_id, image) VALUES (?, ?, ?, ?)');

       // Bind title, content, category, and image to prepared statement parameters

       $stmt->execute([$title, $content, $category\_id, $imageFilename]);

    }

    /\*\*

     \* Retrieve a single post by its ID.

     \*

     \* @param int $id The ID of the post to retrieve.

     \* @return object The fetched post as an object.

     \*/

    public function getPost($id) {

        // Prepare an SQL statement to select a post from the 'posts' table by its ID

        $stmt = $this->db->conn->prepare('SELECT posts.\*, categories.name as category\_name, posts.image as image FROM posts LEFT JOIN categories ON posts.category\_id = categories.id WHERE posts.id = :id');

        // Execute the SQL statement with the provided post ID

        $stmt->execute(['id' => $id]);

        // Return the fetched post as an object

        return $stmt->fetch(\PDO::FETCH\_OBJ);

    }

    /\*\*

     \* Retrieve all posts, ordered by ID in descending order.

     \*

     \* @return array An array of post objects.

     \*/

    public function getPosts() {

        // Prepare an SQL statement to select all posts from the 'posts' table, ordered by ID

        $stmt = $this->db->conn->prepare('SELECT posts.\*, categories.name as category\_name, posts.image as image FROM posts LEFT JOIN categories ON posts.category\_id = categories.id ORDER BY posts.id DESC LIMIT 5');

        // Execute the SQL statement

        $stmt->execute();

        // Return all fetched posts as an array of objects

        return $stmt->fetchAll(\PDO::FETCH\_OBJ);

    }

    // Update an existing post with new title, content and category\_id

    public function updatePost($id, $title, $content, $category\_id, $imageFilename = null) {

        $sql = 'UPDATE posts SET title = :title, content = :content, category\_id = :category\_id, image = :image WHERE id = :id';

        $stmt = $this->db->conn->prepare($sql);

        try {

            $stmt->execute(['title' => $title, 'content' => $content, 'category\_id' => $category\_id,'image' => $imageFilename, 'id' => $id]);

        } catch (PDOException $e) {

            // Log the error

            error\_log("Database error: " . $e->getMessage() . " in " . $e->getFile() . ":" . $e->getLine());

            // Optionally, display a user-friendly error message

            echo "<p>An error occurred while updating the post. Please try again later.</p>";

        }

    }

    public function deletePost($id) {

        $stmt = $this->db->conn->prepare('DELETE FROM posts WHERE id = ?');

        $stmt->execute([$id]);

    }

}

We set the image filename to null when no image has been uploaded.

**Update the AdminController.php**

<?php

namespace app\controllers;

require\_once BASE\_DIR . '/app/models/Post.php';

require\_once BASE\_DIR . '/app/models/Comment.php';

class AdminController {

    private $model;

    public function \_\_construct() {

        $this->model = new \app\models\Post;

    }

    // Read: Display a list of posts

    public function index() {

        // Fetch all posts without pagination

        $posts = $this->model->getPosts();

        // Load the view for displaying posts

        require BASE\_DIR . '/app/views/admin/posts/index.php';

    }

    public function create() {

        // Create an instance of the Category model

        $categoryModel = new \app\models\Category;

        // Fetch all categories

        $categories = $categoryModel->getCategories();

        // Load the view for creating a new post

        require BASE\_DIR . '/app/views/admin/posts/create.php';

    }

    // Read: Display a single post

    public function show($id) {

        // Fetch the post with the given ID

        $post = $this->model->getPost($id);

        if ($post === null) {

            // Post not found, display 404 page

            require BASE\_DIR . '/app/views/404.php';

            return;

        }

         // Create an instance of the Comment model

         $commentModel = new \app\models\Comment;

         // Fetch the comments for the post

         $comments = $commentModel->getComments($id);

        // Load the view for displaying the single post

        require BASE\_DIR . '/app/views/admin/posts/show.php';

    }

    // Create: Store a new post

    public function store($postData) {

        // Retrieve form data

        $title = $postData['title'];

        $content = $postData['content'];

        $category\_id = $postData['category\_id'];

        $imageFilename = null;  // Default to null if no image is uploaded

        // Check if an image file is uploaded and there are no upload errors

        if (isset($\_FILES['image']) && $\_FILES['image']['error'] === UPLOAD\_ERR\_OK) {

            // Define allowed file types and size

            $allowedTypes = ['image/jpeg', 'image/png', 'image/gif'];

            $maxSize = 5 \* 1024 \* 1024; // 5 MB

            // Extract file information from the uploaded image

            $fileTmpPath = $\_FILES['image']['tmp\_name']; // Temporary path where the uploaded file is stored

            $fileName = $\_FILES['image']['name'];       // Original name of the uploaded file

            $fileType = $\_FILES['image']['type'];       // MIME type of the uploaded file

            $fileSize = $\_FILES['image']['size'];       // Size of the uploaded file in bytes

            // Check file type and size

            if (in\_array($fileType, $allowedTypes) && $fileSize <= $maxSize) {

                // Create a unique file name to avoid overwriting existing files

                $newFileName = uniqid() . '-' . $fileName;

                // Set the upload path for the image

                $uploadPath = 'S:/xampp/htdocs/php8blogapp/public/images/'; // Define the directory where uploaded images will be stored

                $destination = $uploadPath . $newFileName; // Concatenate the upload path with the new file name to create the full destination path

                // Ensure the upload directory exists

                if (!file\_exists($uploadPath)) {

                    // Create the directory if it does not exist, with full read/write/execute permissions (0777) and enable recursion (true)

                    mkdir($uploadPath, 0777, true);

                }

                // Move the uploaded file

                if (move\_uploaded\_file($fileTmpPath, $destination)) {

                    // File uploaded successfully

                    $imageFilename = $newFileName; // Set this for database storage

                } else {

                    // Handle error (file move failed)

                    $errors[] = 'Failed to upload image.';

                }

            } else {

                // Handle invalid file type or size

                $errors[] = 'Invalid file type or size.';

            }

        } else {

            // Handle no file uploaded or upload error

            if ($\_FILES['image']['error'] != UPLOAD\_ERR\_NO\_FILE) {

                $errors[] = 'Error in file upload.';

            }

        }

        try {

            $this->model->createPost($title, $content, $category\_id, $imageFilename);

            // Redirect to the posts page after successful creation

            header('Location: /admin/posts');

            $\_SESSION['message'] = 'Post created successfully!.';

            exit;

        } catch (Exception $e) {

            // If there's a database error, display it

            echo 'Error creating post: ' . $e->getMessage();

            exit;

        }

    }

    public function edit($id) {

        $post = $this->model->getPost($id);

        // Fetch categories

        $categoryModel = new \app\models\Category;

        $categories = $categoryModel->getCategories();

        // Pass post and categories to the view

        require BASE\_DIR . '/app/views/admin/posts/edit.php';

    }

    // Update: Update an existing post

    public function update($postData, $id) {

        $title = $postData['title'];

        $content = $postData['content'];

        $category\_id = $postData['category\_id'];

        // Initialize the image filename from existing data

        $imageFilename = $postData['existing\_image\_filename'] ?? null;

        // Check if an image file is uploaded and there are no upload errors

        if (isset($\_FILES['image']) && $\_FILES['image']['error'] === UPLOAD\_ERR\_OK) {

            // Define allowed file types and size

            $allowedTypes = ['image/jpeg', 'image/png', 'image/gif'];

            $maxSize = 5 \* 1024 \* 1024; // 5 MB

            // Extract file information from the uploaded image

            $fileTmpPath = $\_FILES['image']['tmp\_name']; // Temporary path where the uploaded file is stored

            $fileName = $\_FILES['image']['name'];       // Original name of the uploaded file

            $fileType = $\_FILES['image']['type'];       // MIME type of the uploaded file

            $fileSize = $\_FILES['image']['size'];       // Size of the uploaded file in bytes

            // Check file type and size

            if (in\_array($fileType, $allowedTypes) && $fileSize <= $maxSize) {

                // Create a unique file name to avoid overwriting existing files

                $newFileName = uniqid() . '-' . $fileName;

                // Set the upload path

                $uploadPath = 'S:/xampp/htdocs/php8blogapp/public/images/';

                $destination = $uploadPath . $newFileName;

                // Ensure the upload directory exists

                if (!file\_exists($uploadPath)) {

                    mkdir($uploadPath, 0777, true);

                }

                // Move the uploaded file

                if (move\_uploaded\_file($fileTmpPath, $destination)) {

                    // File uploaded successfully

                    $imageFilename = $newFileName; // Update $imageFilename with new filename for database storage

                } else {

                    // Handle error (file move failed)

                    $errors[] = 'Failed to upload image.';

                }

            } else {

                // Handle invalid file type or size

                $errors[] = 'Invalid file type or size.';

            }

        } elseif ($\_FILES['image']['error'] != UPLOAD\_ERR\_NO\_FILE) {

            $errors[] = 'Error in file upload.';

        }

        try {

            // Call the updatePost method

            $this->model->updatePost($id, $title, $content, $category\_id, $imageFilename);

            // Redirect to the posts page after successful update

            $\_SESSION['message'] = 'Post updated successfully!.';

              header('Location: /admin/posts');

              exit;

        } catch (Exception $e) {

            // If there's a database error, display it

            echo 'Error updating post: ' . $e->getMessage();

            exit;

        }

    }

    // Delete: Remove a post

    public function delete($id) {

        $this->model->deletePost($id);

        // Redirect or display success message

        header('Location: /admin/posts');

    }

}

**Update The View Files**

**admin/posts/create.php**

<!-- admin/posts/create.php -->

<div>

    <!-- View all Post Button -->

    <div style="text-align: right; margin-bottom: 20px;">

        <a href="/admin/posts/">View Posts</a>

    </div>

    <h1>Create Post</h1>

    <form method="POST" action="/admin/posts/store" enctype="multipart/form-data">

        <div>

            <label for="title">Title:</label>

            <input type="text" id="title" name="title">

        </div>

        <div>

            <label for="content">Content:</label>

            <textarea id="content" name="content"></textarea>

        </div>

        <div>

            <label for="image">Image:</label>

            <input type="file" id="image" name="image">

        </div>

        <!-- Populate Categories in this view -->

        <div>

            <label for="category\_id">Category:</label>

            <select id="category\_id" name="category\_id">

                <?php foreach ($categories as $category): ?>

                    <option value="<?php echo $category->id; ?>"><?php echo $category->name; ?></option>

                <?php endforeach; ?>

            </select>

        </div>

        <button type="submit">Create Post</button>

    </form>

</div>

**admin/posts/edit.php**

<div>

    <?php if ($post === null): ?>

        <div>Post not found</div>

        <?php return; ?>

        <?php endif; ?>

      <form method="POST" action="/admin/posts/update/<?php echo $post->id ?>" enctype="multipart/form-data" style="margin-bottom: 20px;">

        <input type="hidden" name="id" value="<?php echo $post->id ?>">

        <div style="margin-bottom: 20px;">

            <label for="title">Title</label>

            <input type="text" id="title" name="title" value="<?php echo htmlspecialchars($post->title, ENT\_QUOTES); ?>">

        </div>

        <div style="margin-bottom: 20px;">

            <label for="content">Content</label>

            <textarea id="content" name="content"><?php echo htmlspecialchars($post->content, ENT\_QUOTES); ?></textarea>

        </div>

        <div>

            <label for="image">Image</label>

            <input type="file" id="image" name="image">

            <?php if (!empty($post->image)): ?>

            <img src="/images/<?php echo htmlspecialchars($post->image, ENT\_QUOTES); ?>" alt="Post Image" style="max-width: 200px;">

            <?php endif; ?>

        </div>

        <!-- Populate Categories in this view -->

        <div>

            <label for="category\_id">Category:</label>

            <select id="category\_id" name="category\_id">

            <?php foreach ($categories as $category): ?>

            <option value="<?php echo $category->id; ?>" <?php echo $category->id == $post->category\_id ? 'selected' : ''; ?>>

            <?php echo $category->name; ?>

            </option>

            <?php endforeach; ?>

            </select>

        </div>

        <button type="submit">Update Post</button>

    </form>

</div>

**Display Posts with images in the View Files**

**Admin Side: (admin/posts/show.php)**

<div>

    <!-- View all Post Button -->

    <div style="text-align: right; margin-bottom: 20px;">

        <a href="/admin/posts/">View Posts</a>

    </div>

    <h2><?php echo htmlspecialchars($post->title, ENT\_QUOTES); ?></h2>

    <!-- Check if the post has an image and display it -->

    <?php if ($post->image): ?>

        <img src="/images/<?php echo htmlspecialchars($post->image, ENT\_QUOTES); ?>" alt="Post Image" style="max-width: 200px;"> <!-- Small preview -->

    <?php endif; ?>

    <p><?php echo htmlspecialchars($post->content, ENT\_QUOTES); ?></p>

    <?php if (!empty($comments)): ?>

    <h3>Comments</h3>

    <?php foreach ($comments as $comment): ?>

        <div class="mb-2">

            <p><?php echo htmlspecialchars($comment->content, ENT\_QUOTES); ?></p>

            <a href="/admin/comments/edit/<?php echo $comment->id; ?>" class="btn btn-warning btn-sm">Edit</a>

            <form action="/admin/comments/delete/<?php echo $comment->id; ?>" method="POST" style="display: inline;">

                <input type="hidden" name="\_method" value="DELETE">

                <button type="submit">Delete</button>

                </form>

        </div>

    <?php endforeach; ?>

    <?php else: ?>

        <p>No comments found in this post.</p>

    <?php endif; ?>

</div>

**Frontend Side: views/posts/index.php**

<?php foreach ($posts as $post): ?>

        <div style="margin-bottom: 20px; margin-top: 20px;">

            <div>

                <h2><?php echo htmlspecialchars($post->title, ENT\_QUOTES); ?></h2>

                 <!-- Check if the post has an image and display it -->

                 <?php if ($post->image): ?>

                        <img src="/images/<?php echo htmlspecialchars($post->image, ENT\_QUOTES); ?>" alt="Post Image">

                    <?php endif; ?>

                <p><?php echo htmlspecialchars($post->content, ENT\_QUOTES); ?></p>

                <div>

                  <a href="/posts/show/<?php echo $post->id; ?>">View</a>

                </div>

            </div>

        </div>

    <?php endforeach; ?>

</div>

**Frontend Side: views/posts/show.php**

<!-- show.php -->

<div>

    <div>

        <div>

            <h2><?php echo htmlspecialchars($post->title, ENT\_QUOTES); ?></h2>

        </div>

        <div>

            <?php if ($post->image): ?>

                <img src="/images/<?php echo htmlspecialchars($post->image, ENT\_QUOTES); ?>" alt="Post Image" class="img-fluid mb-3">

            <?php endif; ?>

        </div>

        <div>

            <p><?php echo htmlspecialchars($post->content, ENT\_QUOTES); ?></p>

        </div>

        <div>

            <h3>Comments</h3>

            <?php foreach ($comments as $comment): ?>

                <div style="margin-bottom: 20px;">

                    <p><?php echo htmlspecialchars($comment->content, ENT\_QUOTES); ?></p>

                </div>

            <?php endforeach; ?>

            <form action="/comments/store" method="post" style="margin-top: 20px;">

                <div style="margin-bottom: 20px;">

                    <label for="content">Content:</label>

                    <textarea id="content" name="content"></textarea>

                </div>

                <input type="hidden" name="post\_id" value="<?php echo $post->id; ?>">

                <button type="submit">Submit</button>

            </form>

        </div>

    </div>

</div>