**Step 9: Associate Posts With Categories**

In this section we will attach the categories we’ve created to our posts. Let’s go ahead and do that.

**Modify Post.php**

<?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) {

        // 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) VALUES (?, ?, ?)');

        // Execute the prepared statement with the provided title and content

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

    }

    /\*\*

     \* 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 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 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) {

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

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

        try {

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

    }

}

**Modify the *app/controllers/AdminController.php***

<?php

namespace app\controllers;

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

        }

        // 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'];

        try {

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

            // 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'];

        try {

            // Call the updatePost method

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

            // 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');

    }

}

**Modify The Following View Files**

***App/views/admin/posts/create***

<!-- 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">

        <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>

        <!-- 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>

***App/views/admin/posts/edit***

<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 ?>" 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>

        <!-- 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>

**<option value="<?php echo $category->id; ?>"**:

* + This creates an option in the dropdown list.
	+ **<?php echo $category->id; ?>** outputs the ID of the category, which is set as the value of the option. When this option is selected and the form is submitted, this ID is sent as part of the form data.

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

* + This is a ternary conditional statement in PHP. It checks if the current category's ID (**$category->id**) matches the category ID of the post (**$post->category\_id**).
	+ If the IDs match, it outputs **selected**, making this option the default selected option in the dropdown. This is typically used to show the current category of a post when editing it.
	+ If the IDs do not match, it outputs an empty string, meaning this option will not be selected by default.

***app/views/admin/posts/index***

<div>

    <!-- Display message -->

    <?php if (isset($\_SESSION['message'])): ?>

        <div id="message" class="<?php echo strpos($\_SESSION['message'], 'error') !== false ? 'message-error' : 'message-success' ?> text-center">

            <?php

                echo $\_SESSION['message'];

                unset($\_SESSION['message']);

            ?>

        </div>

    <?php endif; ?>

      <div style="text-align: right;">

        <!-- Create Post Button -->

        <a href="/admin/posts/create">Create New Post</a>

        <a href="/admin/categories/create">Create Categories</a>

    </div>

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

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

            <div>

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

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

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

                <a href="/admin/posts/edit/<?php echo $post->id; ?>">Edit</a>

                <p><strong>Category:</strong> <?php echo $post->category\_name ? htmlspecialchars($post->category\_name, ENT\_QUOTES) : 'No category'; ?></p>

                <!-- Delete Post -->

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

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

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

                </form>

            </div>

        </div>

    <?php endforeach; ?>

</div>

<style>

.message-success {

color: green;

/\* other styling \*/

}

.message-error {

    color: red;

    background-color: #ffd6d6;

    padding: 10px;

    border: 1px solid red;

    margin-bottom: 20px;

}

</style>