**Step 13: Implement Pagination To The Posts Data**

To add pagination which means to display certain number of posts to our pages. We can do it this way.

**Update the Post.php Model**

 public function getPosts($pageNum = 1, $postsPerPage = 5) {

        // Calculate the offset for the SQL query based on the current page number and posts per page

        $offset = ($pageNum - 1) \* $postsPerPage;

        // Prepare an SQL statement to fetch posts with pagination

        // Posts are ordered by their ID in descending order

        // :limit and :offset are placeholders for pagination values

        $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 :limit OFFSET :offset');

        // Bind the limit and offset values to the prepared statement

        // PDO::PARAM\_INT ensures these values are treated as integers

        $stmt->bindValue(':limit', $postsPerPage, \PDO::PARAM\_INT);

        $stmt->bindValue(':offset', $offset, \PDO::PARAM\_INT);

        // Execute the SQL statement

        $stmt->execute();

        // Return all fetched posts as an array of objects

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

    }

**Limit and Offset Explained:**

- In **SQL**, `**LIMIT**` and `**OFFSET**` are clauses used to constrain the number of rows returned by a query.

- `**LIMIT**` specifies the maximum number of rows to return.

- `**OFFSET**` specifies the number of rows to skip before starting to return rows from the query.

**Why Subtract 1 in Offset Calculation:**

- The calculation `**$offset = ($pageNum - 1) \* $postsPerPage**;` is used to determine where to start fetching rows for a given page.

- Pagination typically starts from page 1, but the offset in SQL starts from 0.

- Subtracting 1 adjusts for this difference.

**Example:**

- Imagine you have 5 posts per page (`$postsPerPage = 5`).

- For page 1, you want to start from the very first post (offset 0). The calculation will be `(1 - 1) \* 5 = 0`.

- For page 2, you want to skip the first 5 posts and start from the 6th post. The calculation will be `(2 - 1) \* 5 = 5`.

- Thus, for page 1, the offset is 0 (starting from the first row), and for page 2, the offset is 5 (starting from the 6th row).

Add this method to **Post.php**

 public function getTotalPostsCount() {

        // Prepare an SQL query to count the total number of posts in the database

        $stmt = $this->db->conn->query('SELECT COUNT(\*) FROM posts');

        // Execute the query and return the count result

        // fetchColumn() retrieves the value of the first column in the result set

        return $stmt->fetchColumn();

    }

The rest of the methods remain the same. Let’s ago ahead and update the **AdminController.php and PostController.php** indexes with this code.

**AdminController.php:**

// Read: Display a list of posts

    public function index() {

       // Determine the current page number

       $currentPage = isset($\_GET['page']) ? (int)$\_GET['page'] : 1;

       // Define how many posts you want per page

       $postsPerPage = 5;

       // Fetch posts for the current page

       $posts = $this->model->getPosts($currentPage, $postsPerPage);

       // Calculate the total number of pages

       $totalPosts = $this->model->getTotalPostsCount();

       $totalPages = ceil($totalPosts / $postsPerPage);

        // Load the view for displaying posts

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

    }

**PostController.php**

// Display all posts

    public function index() {

         // Determine the current page number

         $currentPage = isset($\_GET['page']) ? (int)$\_GET['page'] : 1;

         $postsPerPage = 5; // Define how many posts you want per page

         // Fetch posts for the current page

         $posts = $this->model->getPosts($currentPage, $postsPerPage);

         // Calculate the total number of pages

         $totalPosts = $this->model->getTotalPostsCount();

         $totalPages = ceil($totalPosts / $postsPerPage);

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

    }

* **isset($\_GET['page'])**: This checks if the **page** parameter exists in the URL query string. For example, in the URL **http://example.com/posts?page=2**, it checks if **page=2** is present.
* **(int)$\_GET['page']**: This is a type cast to integer. If **page** is present in the URL, its value is converted to an integer. This is important for security and validation purposes to ensure that the value used as the page number is indeed an integer. Type casting helps prevent potential security issues, such as SQL injection when the value is used in a database query.
* **: 1**: This part of the code provides a default value. If **page** is not present in the URL, the current page number is assumed to be 1. This means that by default, or if the **page** parameter is missing, the application will show the first page of the content.

Putting it all together, **$currentPage = isset($\_GET['page']) ? (int)$\_GET['page'] : 1;** means "If **page** is specified in the URL and is a valid number, use it as the current page number; otherwise, default to page 1." This is a succinct way to handle pagination in PHP.

**Update the View Files:**

***admin/posts/index.php***

Place it below the php end foreach

<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; ?>

    <!-- Pagination Controls -->

    <nav aria-label="Page navigation">

        <ul style="list-style: none; padding: 0;">

            <!--we loop through each page number-->

            <?php for ($i = 1; $i <= $totalPages; $i++): ?>

                <!--Check if the loop's current page number ($i, e.g., 3) is the same as the current page ($currentPage, 3)-->

                <!--If yes, then this is the current page, so we could style it differently if needed-->

                <li style="<?php echo $i == $currentPage ? 'font-weight: bold;' : ''; ?>">

                    <!-- returns the page full URL e.g., http://admin/posts?page=2 -->

                    <a href="/admin/posts?page=<?php echo $i; ?>"><?php echo $i; ?></a>

                </li>

            <?php endfor; ?>

        </ul>

    </nav>

</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>

***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; ?>

    <!-- Pagination Controls -->

    <nav aria-label="Page navigation">

        <ul style="list-style: none; padding: 0;">

            <!--we loop through each page number-->

            <?php for ($i = 1; $i <= $totalPages; $i++): ?>

                <!--Check if the loop's current page number ($i, e.g., 3) is the same as the current page ($currentPage, 3)-->

                <!--If yes, then this is the current page, so we could style it differently if needed-->

                <li style="<?php echo $i == $currentPage ? 'font-weight: bold;' : ''; ?>">

                    <!-- returns the page full URL e.g., http://admin/posts?page=2 -->

                    <a href="/?page=<?php echo $i; ?>"><?php echo $i; ?></a>

                </li>

            <?php endfor; ?>

        </ul>

    </nav>

</div>

In conclusion to determine the pagination. Here is how we calculate it mathematical wise.

# Given values for the example

currentPage = 1  # Assuming currentPage is 1 for the example

postsPerPage = 5

totalPosts = 22  # Let's assume there are 22 posts in total

# Calculate the total number of pages

totalPages = ceil(totalPosts / postsPerPage)

totalPages

Result 5.