

Name: Sharmin Kabir Kripa

Module: 18

Task 1:

Create a new migration file to add a new table named "categories" to the database. The table should have the following columns:

id (primary key, auto-increment)

name (string)

created_at (timestamp)

updated_at (timestamp)

Artisan Command for creating file:

php artisan make:migration create_Categories_table

Migration file:

```
*/
public function up(): void
{
    Schema::create('categories', function (Blueprint $table) {
        $table->id();
        $table->string('name',50);
        $table->timestamp('created_at')->useCurrent();
        $table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdate();
    });
}
```

2. Task 2:

Create a new model named "Category" associated with the "categories" table. Define the necessary properties and relationships.

Defining Relationship:

```
<?php

namespace App\Models;

use App\Models\Post;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```

class Category extends Model
{
    use HasFactory;
    protected $fillable = ['name'];

    public function posts(){
        return $this->hasMany(Post::class);
    }
}

```

Task 3:

Write a migration file to add a foreign key constraint to the "posts" table. The foreign key should reference the "categories" table on the "category_id" column.

Posts table Migration file:

```

public function up(): void
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->text('description')->nullable();
        $table->timestamp('created_at')->useCurrent();
        $table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdate();
        $table->foreignId('category_id')->constrained('categories')
            ->cascadeOnDelete()
            ->cascadeOnUpdate();
    });
}

```

Task 4:

Create a relationship between the "Post" and "Category" models. A post belongs to a category, and a category can have multiple posts.

Post Model:

```

<?php

namespace App\Models;

use App\Models\Category;

```

```

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\SoftDeletes;

class Post extends Model
{
    use HasFactory, SoftDeletes;
    protected $fillable = ['name', 'description', 'category_id'];
    public function category()
    {
        return $this->belongsTo(Category::class);
    }
}

```

Task 5:

Write a query using Eloquent ORM to retrieve all posts along with their associated categories. Make sure to eager load the categories to optimize the query.

Controller method:

```

//Task 5
public function postData()
{
    $postdata = Post::with('category')->get();
    return $postdata;
}

```

Task 6

Implement a method in the "Post" model to get the total number of posts belonging to a specific category. The method should accept the category ID as a parameter and return the count.

Route:

```

Route::controller(AdminController::class)->group(function () {

    Route::get('/categoryPost/{category_id}', 'categoryPost');

});

```

Implemented Method in Post Model:

```
public static function totalPostsCountByCategory($categoryId)
{
    return self::where('category_id', $categoryId)->count();
}
```

Controller method for testing In Postman:

```
public function categoryPost($category_id)
{
    $postCount = Post::totalPostsCountByCategory($category_id);
    return response()->json(['Count' => $postCount]);
}
```

Postman Output:

The screenshot shows the Postman interface for a GET request. The URL is `{{baseurl}}/categoryPost/:category_id`. The path variable `category_id` is set to `3`. The response status is `200 OK` with a time of `481 ms` and a size of `1.11 KB`. The response body is a JSON object: `{\"Count\": 3}`.

Key	Value	Description
category_id	3	Description

Key	Value	Description
category_id	3	Description

Body: `{\"Count\": 3}`

Task 7:

Create a new route in the web.php file to handle the following URL pattern: `/posts/{id}/delete`. Implement the corresponding controller method to delete a post by its ID. Soft delete should be used.

Route:

```
Route::controller(AdminController::class)->group(function () {  
  
    Route::get('/posts/{id}/delete', 'softDelete');  
});
```

Controller method:

```
//Task 7  
public function softDelete($id)  
{  
    $softDelete = Post::findOrFail($id)->delete();  
    if ($softDelete) {  
        return 'successfully soft deleted';  
    } else {  
        return 'failed to softdelete';  
    }  
}
```

Postman Output:

The screenshot shows a Postman interface for a GET request. The URL is `{{baseurl}}/posts/id/delete`. The 'Path Variables' section shows `id` with the value `18`. The response status is `200 OK` with a body of `successfully soft deleted`.

Key	Value	Description
id	18	Description

Key	Value	Description
Key	Value	Description

Body: `successfully soft deleted`

Task 8:

Implement a method in the "Post" model to get all posts that have been soft deleted. The method should return a collection of soft deleted posts.

Route:

```
Route::controller(AdminController::class)->group(function () {
    Route::get('/softData', 'softData');
});
```

Implemented Method in Post Model:

```
public static function softDeletedData()
{
    return self::onlyTrashed()->get();
}
```

Controller method for testing In Postman:

```
public function softData()
{
    $softData = Post::softDeletedData();
    return $softData;
}
```

Postman output:

The screenshot shows the Postman interface with a GET request to `{{baseurl}}/softData` executed successfully. The response status is 200 OK, with a time of 405 ms and a size of 4.31 KB. The response body is displayed in JSON format, showing an array of two objects representing soft-deleted posts.

```
{
  "id": 6,
  "name": "Post 1",
  "description": "\r\nImagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop.\r\n\r\nAs the wind whispers through the tall grass, the figure contemplates the ",
  "created_at": "2023-06-25T18:18:26.000000Z",
  "updated_at": "2023-06-27T10:48:05.000000Z",
  "category_id": 1,
  "deleted_at": "2023-06-25T14:43:00.000000Z"
},
{
  "id": 7,
  "name": "Post 2",
  "description": "\r\nImagine a picturesque landscape with vibrant colors. where the golden sun dips below the
```

Task 9:

Write a Blade template to display all posts and their associated categories. Use a loop to iterate over the posts and display their details.

Blade File:

```
<section class="py-5">
    <div class="container px-5 mb-5" >
        <div class="text-center mb-5">
            <h1 class="display-5 fw-bolder mb-0"><span class="text-gradient d-inline">All Post With Category</span></h1>
        </div>
        <div class="row gx-5 justify-content-center">
            <div id="project-list" class="col-lg-11 col-xl-9 col-xxl-8">
                @foreach ( $postdata as $data )
                    <div class="card overflow-hidden shadow rounded-4 border-0 mb-5">
                        <div class="card-body p-0">
                            <div class="d-flex align-items-center">
                                <div class="p-5">
                                    <h2 class="fw-bolder"> {{ $data->name }}</h2>
                                    <h4 class="fw-bolder"> {{ $data->category['name'] }}</h4>
                                    <p>{{ $data->description }}</p>
                                </div>
                                <div class="p-5" style="background-color: #f8d7da; padding: 10px; text-align: right; font-size: 0.9em; font-weight: normal;">
                                    You, 5 days ago • All functionalites done
                                </div>
                            </div>
                        </div>
                    </div>
                @endforeach
            </div>
        </div>
    </div>
</section>
```


SRZ

postDatecategories

All Post With Category

Post 3

Technology

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Post 4

Food

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Post 5

Sports

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Post 6

Sports

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Post 7

Food

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Post 8

Food

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Post 13

Fashion

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Post 14

Travel

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Post 15

Fashion

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Post 16

Travel

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Copyright © Your Website 2024

Task 10:

Create a new route in the web.php file to handle the following URL pattern: `"/categories/{id}/posts"`. Implement the corresponding controller method to retrieve all posts belonging to a specific category. The category ID should be passed as a parameter to the method.

Route:

```
Route::controller(AdminController::class)->group(function () {  
  
    Route::get('/categories/{id}/posts', 'specificCatPost');  
});
```

Controller method:

```
//Task 10  
public function specificCatPost($id)  
{  
    try {  
        $category = Category::findOrFail($id);  
        return $category->posts;  
    } catch (\Exception $e) {  
        return 'Category not found.';  
    }  
}
```

Postman output:

The screenshot shows the Postman interface with a GET request to `{{baseurl}}/categories/3/posts`. The response is a JSON array of two post objects. The first post has an ID of 9 and a detailed description. The second post has an ID of 12.

Key	Value	Description
id	3	Description

```
{  
  "id": 9,  
  "name": "Post 4",  
  "description": "\\Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop.\\r\\n\\r\\nAs the wind whispers through the tall grass, the figure contemplates the ",  
  "created_at": "2023-06-25T18:18:27.000000Z",  
  "updated_at": "2023-06-27T10:48:06.000000Z",  
  "category_id": 3,  
  "deleted_at": null  
},  
  {  
    "id": 12,
```

Task 11:

Implement a method in the "Category" model to get the latest post associated with the category. The method should return the post object.

Route:

```
Route::controller(AdminController::class)->group(function () {  
    Route::get('/categories/{id}/latestpost', 'latestPost');  
});
```

Implemented Method in Category Model:

```
public function LatestPost()  
{  
    return $this->posts()->orderBy('id', 'desc')->first();  
}
```

Controller method for testing In Postman:

```
public function latestPost($id)  
{  
    $category = Category::findOrFail($id)->LatestPost();  
    return $category;  
}
```

Postman output:

The screenshot shows the Postman interface with a GET request to `{{baseurl}}/categories/categoryId/latestpost`. The path variable `categoryId` is set to `2`. The response status is `200 OK` with a time of `396 ms` and a size of `1.61 KB`. The response body is displayed in JSON format:

```
{  
  "id": 11,  
  "name": "Post 6",  
  "description": "\nImagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop.\n\nAs the wind whispers through the tall grass, the figure contemplates the ",  
  "created_at": "2023-06-25T18:18:27.000000Z",  
  "updated_at": "2023-06-27T10:48:06.000000Z",  
  "category_id": 2,  
  "deleted_at": null  
}
```

Task 12:

Write a Blade template to display the latest post for each category. Use a loop to iterate over the categories and display the post details.

```
<section class="py-5">
  <div class="container px-5 mb-5">
    <div class="text-center mb-5">
      <h1 class="display-5 fw-bolder mb-0"><span class="text-gradient d-inline">Latest Post For Each Category</span></h1>
    </div>
    <div class="row gx-5 justify-content-center">
      <div id="project-list" class="col-lg-11 col-xl-9 col-xxl-8">
        @foreach ($categories as $category)
          <div class="card overflow-hidden shadow rounded-4 border-0 mb-5">
            <div class="card-body p-0">
              <div class="d-flex align-items-center">
                <div class="p-5">
                  <h2>{{ $category->name }}</h2>
                  @if ($category->LatestPost())
                    <div>
                      <h3>{{ $category->LatestPost()->name }}</h3>
                      <p>{{ $category->LatestPost()->description }}</p>
                    </div>
                  @else
                    <p>No posts found for this category.</p>
                  @endif
                </div>
              </div>
            </div>
          </div>
        @endforeach
      </div>
    </div>
  </div>
</section>
```

Output:

Latest Post For Each Category

Technology

Post 3

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Sports

Post 6

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Food

Post 8

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Fashion

Post 15

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the

Travel

Post 16

"Imagine a picturesque landscape with vibrant colors, where the golden sun dips below the horizon, casting a warm glow on the rolling hills. In this mesmerizing setting, a solitary figure stands atop a cliff, their silhouette perfectly framed against the breathtaking backdrop. As the wind whispers through the tall grass, the figure contemplates the