

Assignment – 18

At the beginning I created a laravel application using command “Laravel new Assignment18”

Task 1 : Create a new migration file to add a new table named "categories" to the database. The table should have the following columns :

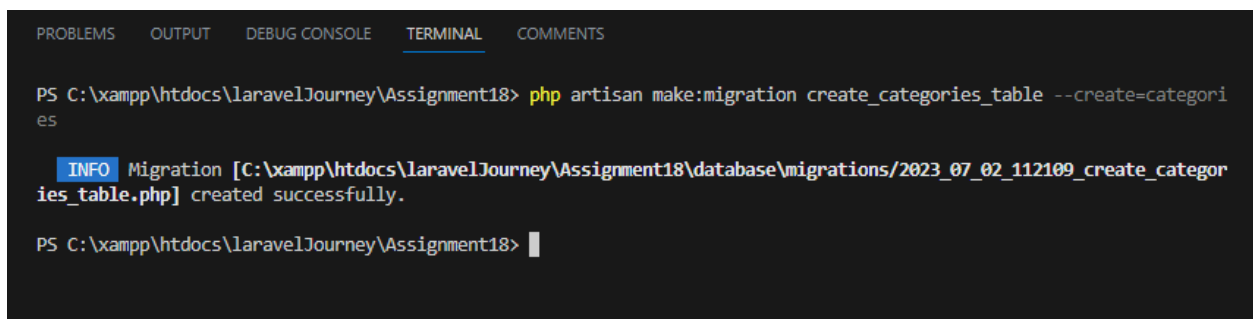
id (primary key, auto-increment)

name (string)

created_at (timestamp)

updated_at (timestamp)

➔ Firstly, I have create a migration name categories table. Here is a screenshot below



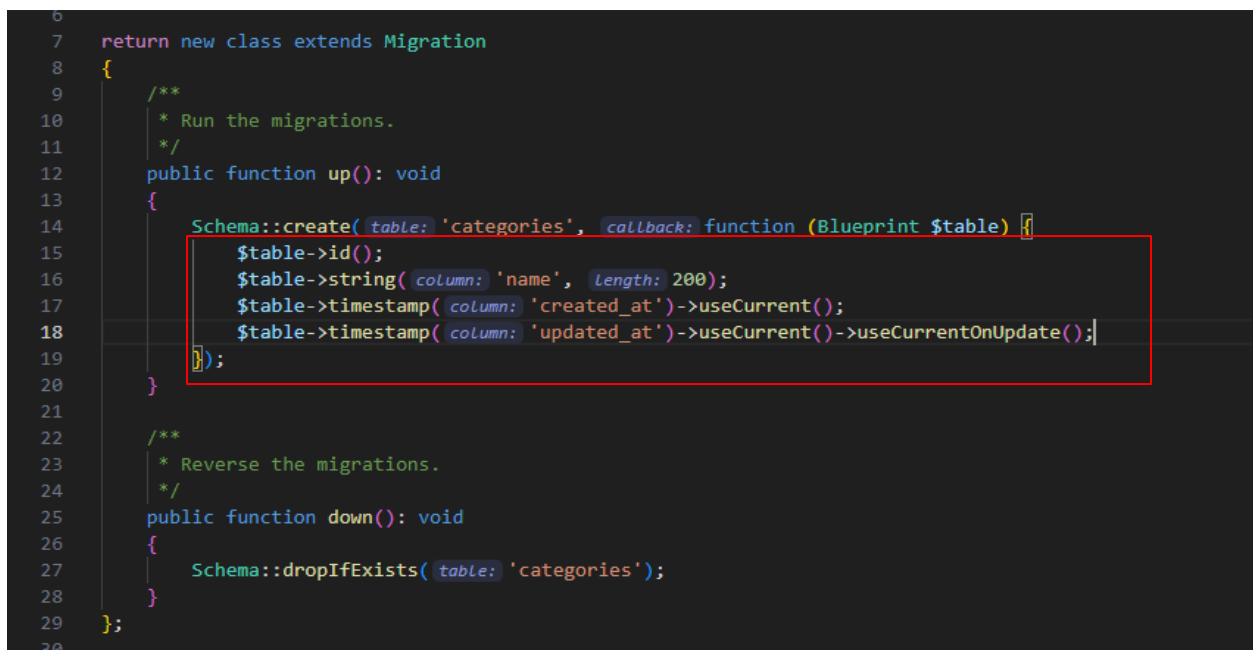
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  COMMENTS

PS C:\xampp\htdocs\laravelJourney\Assignment18> php artisan make:migration create_categories_table --create=categories

INFO Migration [C:\xampp\htdocs\laravelJourney\Assignment18\database\Migrations\2023_07_02_112109_create_categories_table.php] created successfully.

PS C:\xampp\htdocs\laravelJourney\Assignment18>
```

Then I have add name,created_at and updated_at. Here is the another Screenshot below .



```
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12      public function up(): void
13      {
14          Schema::create( table: 'categories', callback: function (Blueprint $table) {
15              $table->id();
16              $table->string( column: 'name', length: 200);
17              $table->timestamp( column: 'created_at')->useCurrent();
18              $table->timestamp( column: 'updated_at')->useCurrent()->useCurrentOnUpdate();
19          });
20      }
21
22      /**
23       * Reverse the migrations.
24       */
25      public function down(): void
26      {
27          Schema::dropIfExists( table: 'categories');
28      }
29  };
30
```

Task 2 : Create a new model named "Category" associated with the "categories" table. Define the necessary properties and relationships.

➔ First I create a command on model. Here is a screenshot below

```
PS C:\xampp\htdocs\laravelJourney\Assignment18> php artisan make:model Category

INFO Model [C:\xampp\htdocs\laravelJourney\Assignment18\app\Models\Category.php] created successfully.

PS C:\xampp\htdocs\laravelJourney\Assignment18>
```

Then I found this

```
7
8 0 references | 0 implementations
9 class Category extends Model
10 {
11     0 references
12     protected $fillable = ['name'];
13
14     0 references | 0 overrides
15     public function posts()
16     {
17         return $this->hasMany(related: Post::class);
18     }
19 }
```

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.

First I ran a command like this –

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS powershell + v [ ] [ ] ... ^ x

PS C:\xampp\htdocs\laravelJourney\Assignment18> php artisan make:migration add_category_id_to_posts_table --table=post

INFO Migration [C:\xampp\htdocs\laravelJourney\Assignment18\database\Migrations\2023_07_02_162501_add_category_id_to_posts_table.php] created successfully.

PS C:\xampp\htdocs\laravelJourney\Assignment18>
```

Then I have got this inside migration

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::table('post', callback: function (Blueprint $table) {
15             $table->unsignedBigInteger('category_id')->nullable();
16             $table->foreign('category_id')->references('id')->on('categories');
17         });
18     }
19
20     /**
21      * Reverse the migrations.
22      */
23     public function down(): void
24     {
25         Schema::table('post', callback: function (Blueprint $table) {
26             $table->dropForeign(['category_id']);
27             $table->dropColumn('category_id');
28         });
29     }
30 };
```

Task 4: Create a relationship between the "Post" and "Category" models. A post belongs to a category, and a category can have multiple posts.

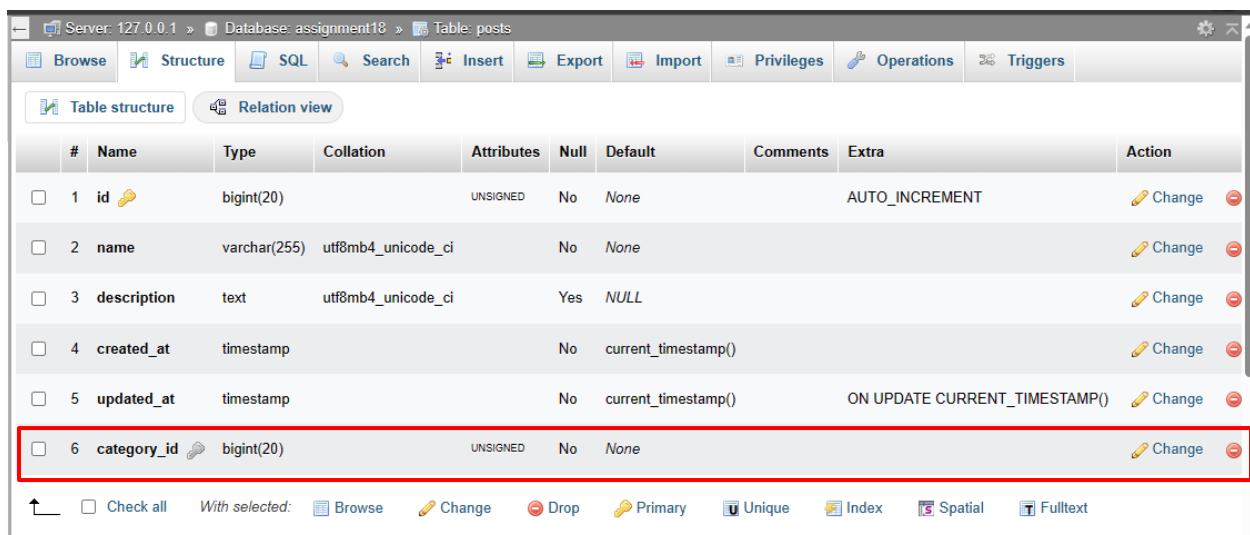
Firstly I have created a model name "Post"

```
PS C:\xampp\htdocs\laravelJourney\Assignment18> php artisan make:model Post
```

INFO Model [C:\xampp\htdocs\laravelJourney\Assignment18\app\Models\Post.php] created successfully.

Then I have created Post table and Then have a relationship with the post and category models

```
7   return new class extends Migration
8   {
9       /**
10        * Run the migrations.
11        */
12        public function up(): void
13        {
14            Schema::create( table: 'posts', callback: function (Blueprint $table) {
15                $table->id();
16                $table->string( column: 'name');
17                $table->text( column: 'description')->nullable();
18                $table->timestamp( column: 'created_at')->useCurrent();
19                $table->timestamp( column: 'updated_at')->useCurrent()->useCurrentOnUpdate();
20                $table->foreignId( column: 'category_id')->constrained( table: 'categories')
21                    ->cascadeOnDelete()
22                    ->cascadeOnUpdate();
23            });
24        }
25
26        /**
27         * Reverse the migrations.
28         */
29        public function down(): void
30        {
31            Schema::dropIfExists( table: 'posts');
32        }
33    };
34
35
```



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		No	None			Change
<input type="checkbox"/>	3 description	text	utf8mb4_unicode_ci		Yes	NULL			Change
<input type="checkbox"/>	4 created_at	timestamp			No	current_timestamp()			Change
<input type="checkbox"/>	5 updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change
<input type="checkbox"/>	6 category_id	bigint(20)		UNSIGNED	No	None			Change

Model Relationship :

```
1  <?php
2
3  namespace App\Models;
4
5  use App\Models\Category;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\Factories\HasFactory;
8  use Illuminate\Database\Eloquent\SoftDeletes;
9
10 9 references | 0 implementations
11 class Post extends Model
12 {
13     use HasFactory, SoftDeletes;
14     0 references
15     protected $fillable = ['name', 'description', 'category_id'];
16     0 references | 0 overrides
17     public function category()
18     {
19         return $this->belongsTo(related: Category::class);
20     }
21 }
```

```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  2 references | 0 implementations
9  class Category extends Model
10 {
11     0 references
12     protected $fillable = ['name'];
13
14     0 references | 0 overrides
15     public function posts()
16     {
17         return $this->hasMany(related: Post::class);
18     }
19 }
```

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.

Here I implement a controller Method :

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Post;
6  use Illuminate\Http\Request;
7
8  2 references | 0 implementations
9  class ActionController extends Controller
10 {
11     0 references | 0 overrides
12     public function postData()
13     {
14         $postdata = Post::with('relations: 'category')->get();
15         return $postdata;
16     }
17 }
```

Assignment18 / New Request

GET http://127.0.0.1:8000/postData Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies (5) Headers (9) Test Results Status: 200 OK Time: 7.69 s Size: 1.96 KB Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 3,
3   "name": "Sports",
4   "description": "Watch the best live coverage of your favourite sports: Football, Golf, Rugby, Cricket, F1,
5     Boxing, NFL, NBA, plus the latest sports news, transfers & scores.",
6   "created_at": "2023-07-03T17:37:04.000000Z",
7   "updated_at": "2023-07-03T17:37:04.000000Z",
8   "category_id": 1,
9   "deleted_at": null,
10  "category": {
11    "id": 1,
12    "name": "Sports",
13    "created_at": "2023-07-03T17:35:54.000000Z",
14    "updated_at": "2023-07-03T17:35:54.000000Z"
15  }
16 }
```

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.

First I made a function inside ActionController :

```
18 public function categoryPost($category_id)
19 {
20     $postCount = Post::totalPostsCountByCategory( categoryId: $category_id);
21     return response()->json( data: ['Count' => $postCount]);
22 }
23 }
```

Implement a Method in Model :

```
1 reference | 0 overrides
19 public static function totalPostsCountByCategory($categoryId)
20 {
21     return self::where('category_id', $categoryId)->count();
22 }
23 }
```

Finally I created a Route for this :

```
26 Route::get( uri: '/categoryPost/{category_id}', action: 'categoryPost');
```

Output By Postman :

The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:8000/categoryPost/1`. The request is successful, returning a 200 OK status with a response time of 4.05 s and a size of 1.11 KB. The response body is displayed in JSON format as `\"Count\": 2`.

Key	Value	Description
Key	Value	Description

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.

Controller Method :

```
25     public function softDelete($id)
26     {
27         $softDelete = Post::findOrFail($id)->delete();
28         if ($softDelete) {
29             return 'successfully soft deleted';
30         } else {
31             return 'failed to softdelete';
32         }
33     }
34 }
```

Route screenshot given bellow:

```
28     Route::get(uri: '/posts/{id}/delete', action: 'softDelete');
```

Output By Postman :

The screenshot shows the Postman interface for a GET request. The URL is `http://127.0.0.1:8000/posts/id/delete` with a path variable `id` set to `4`. The response status is `200 OK` and the body is `successfully soft deleted`.

Key	Value	Description
id	4	Description

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.

Implement Method In Post Model :

```
0 references | 0 overrides
24 public static function softDeletedData()
25 {
26     return self::onlyTrashed()->get();
27 }
28
```

Implement Controller Method :

```
0 references | 0 overrides
38 public function softData()
39 {
40     $softData = Post::softDeletedData();
41     return $softData;
42 }
43
```

Implement Route :

```
30 Route::get('/softData', 'softData');
```

Output By Postman :

The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:8000/softData`. The status is 200 OK, and the response is a JSON object representing a soft-deleted post.

Key	Value	Description
Key	Value	Description

Body (Pretty):

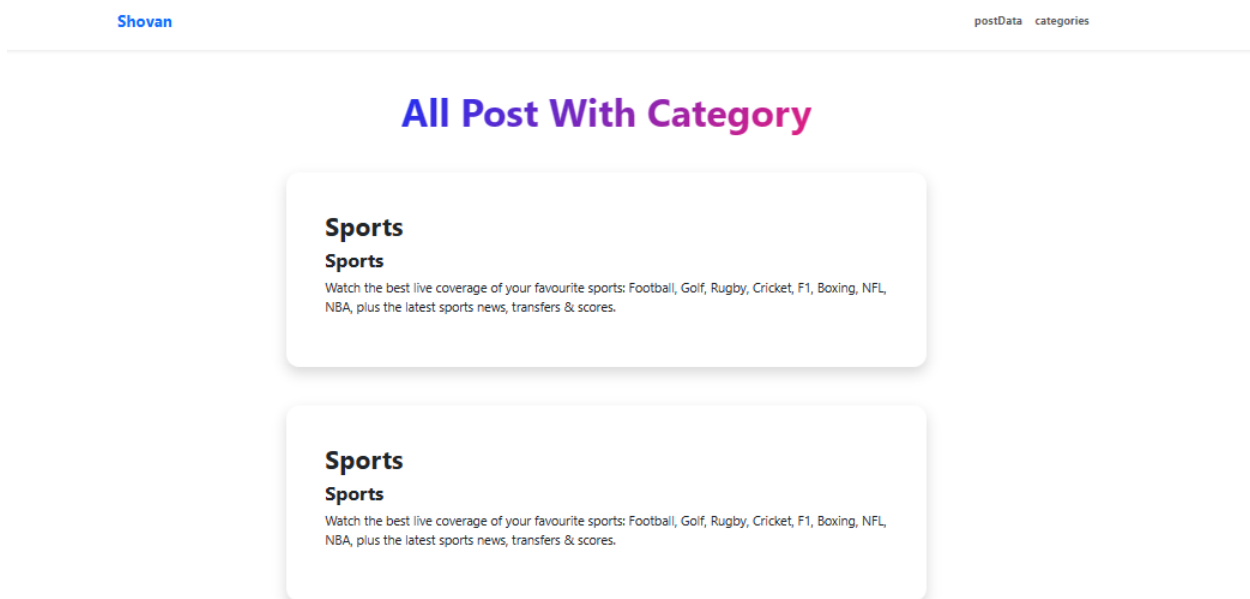
```
{
  "id": 4,
  "name": "Politics",
  "description": "Watch the best live coverage of your favourite sports: Football, Golf, Rugby, Cricket, F1, Boxing, NFL, NBA, plus the latest sports news, transfers & scores.",
  "created_at": "2023-07-03T17:37:04.000000Z",
  "updated_at": "2023-07-03T14:47:26.000000Z",
  "category_id": 2,
  "deleted_at": "2023-07-03T14:47:26.000000Z"
}
```


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 :

```
resources > views > components > posts.blade.php > section.py-5 > div.container.px-5.mb-5 > div.row.gx-5.justify-content-center >
1 <section class="py-5">
2   <div class="container px-5 mb-5">
3     <div class="text-center mb-5">
4       <h1 class="display-5 fw-bolder mb-0"><span class="text-gradient d-inline">All Post With Category</span></h1>
5     </div>
6     <div class="row gx-5 justify-content-center">
7       <div id="project-list" class="col-lg-11 col-xl-9 col-xxl-8">
8         @foreach ( $postdata as $data )
9           <div class="card overflow-hidden shadow rounded-4 border-0 mb-5">
10            <div class="card-body p-0">
11              <div class="d-flex align-items-center">
12                <div class="p-5">
13
14                  <h2 class="fw-bolder"> {{ $data->name }}</h2>
15                  <h4 class="fw-bolder"> {{ $data->category['name'] }}</h4>
16                  <p>{{ $data->description }}</p>
17
18                </div>
19              </div>
20            </div>
21          </div>
22        </div>
23      @endforeach
24    </div>
25  </div>
26 </section>
27
28
```

Output :



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.

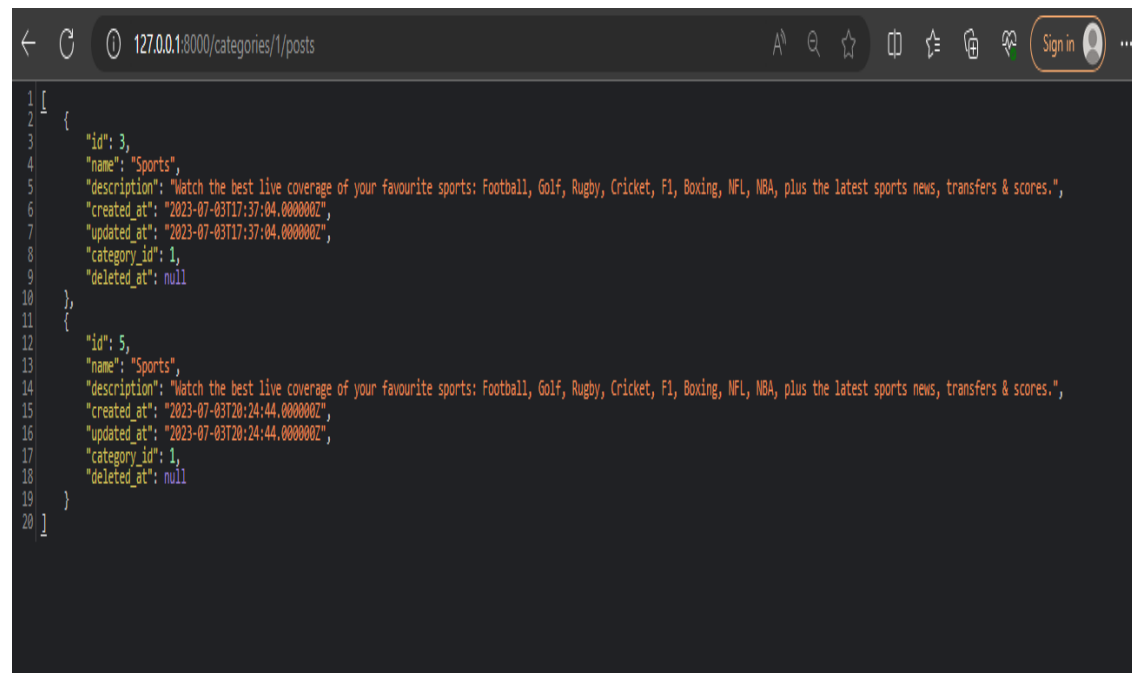
Route :

```
34 Route::get( uri: '/categories/{id}/posts', action: 'specificCatPost');
```

Controller Method :

```
55 public function specificCatPost($id)
56 {
57     try {
58         $category = Category::findOrFail($id);
59         return $category->posts;
60     } catch (\Exception $e) {
61         return 'Category not found.';
62     }
63 }
64
```

Output :



```
1 [
2   {
3     "id": 3,
4     "name": "Sports",
5     "description": "Watch the best live coverage of your favourite sports: Football, Golf, Rugby, Cricket, F1, Boxing, NFL, NBA, plus the latest sports news, transfers & scores.",
6     "created_at": "2023-07-03T17:37:04.000000Z",
7     "updated_at": "2023-07-03T17:37:04.000000Z",
8     "category_id": 1,
9     "deleted_at": null
10  },
11  {
12    "id": 5,
13    "name": "Sports",
14    "description": "Watch the best live coverage of your favourite sports: Football, Golf, Rugby, Cricket, F1, Boxing, NFL, NBA, plus the latest sports news, transfers & scores.",
15    "created_at": "2023-07-03T20:24:44.000000Z",
16    "updated_at": "2023-07-03T20:24:44.000000Z",
17    "category_id": 1,
18    "deleted_at": null
19  }
20 ]
```

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.

Implemented method in Category model :

```
17 public function LatestPost()  
18 {  
19     return $this->posts()->orderBy('id', 'desc')->first();  
20 }
```

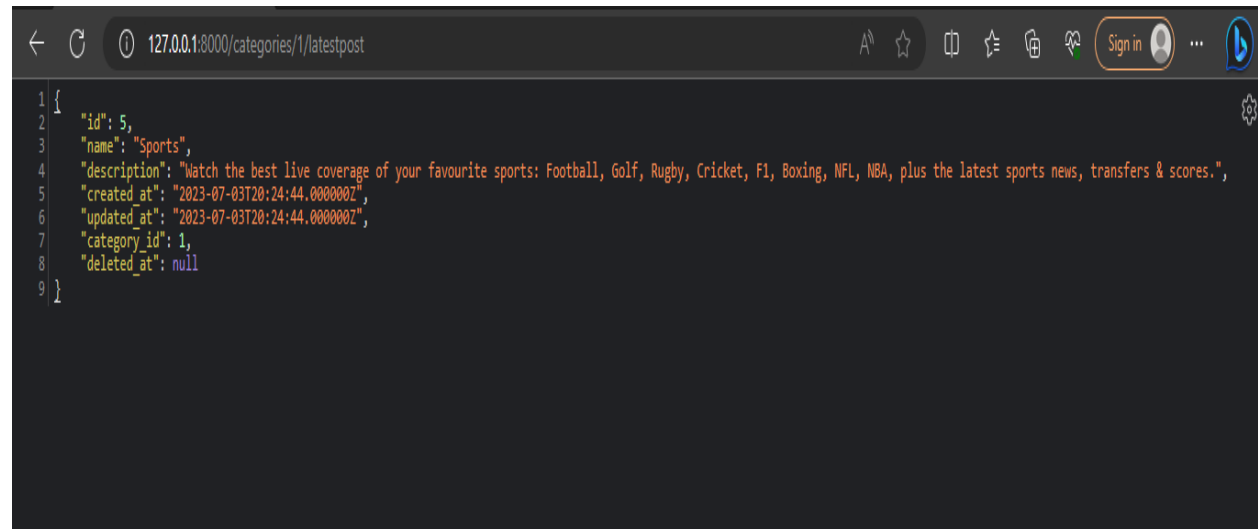
Implement Route :

```
36 Route::get('categories/{id}/latestpost', 'action: latestPost');
```

Implement Controller :

```
67 public function latestPost($id)  
68 {  
69     $category = Category::findOrFail($id)->LatestPost();  
70     return $category;  
71 }
```

Output :



```
1 {  
2     "id": 5,  
3     "name": "Sports",  
4     "description": "Watch the best live coverage of your favourite sports: Football, Golf, Rugby, Cricket, F1, Boxing, NFL, NBA, plus the latest sports news, transfers & scores.",  
5     "created_at": "2023-07-03T20:24:44.000000Z",  
6     "updated_at": "2023-07-03T20:24:44.000000Z",  
7     "category_id": 1,  
8     "deleted_at": null  
9 }
```

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.

Blade file :

```
resources > views > components > category_data.blade.php > section.py-5 > div.container.px-5.mb-5 > div.row.gx-5.justify-content-center
1 <section class="py-5">
2 <div class="container px-5 mb-5">
3 <div class="text-center mb-5">
4 <h1 class="display-5 fw-bolder mb-0"><span class="text-gradient d-inline">Latest Post For Each Category</span></h1>
5 </div>
6 <div class="row gx-5 justify-content-center">
7 <div id="project-list" class="col-lg-11 col-xl-9 col-xxl-8">
8 @foreach ($categories as $category)
9 <div class="card overflow-hidden shadow rounded-4 border-0 mb-5">
10 <div class="card-body p-0">
11 <div class="d-flex align-items-center">
12 <div class="p-5">
13 <h2>{{ $category->name }}</h2>
14 @if ($category->LatestPost())
15 <div>
16 <h3>{{ $category->LatestPost()->name }}</h3>
17 <p>{{ $category->LatestPost()->description }}</p>
18 </div>
19 @else
20 <p>No posts found for this category.</p>
21 @endif
22 </div>
23 </div>
24 </div>
25 </div>
26 @endforeach
27 </div>
28 </div>
29 </section>
30
```

Implement Controller Method :

```
74 public function CategoriesLatestPosts()
75 {
76     $categories = Category::all();
77
78     return view('pages.categories', data: compact('categories'));
79 }
80
```

Output :

