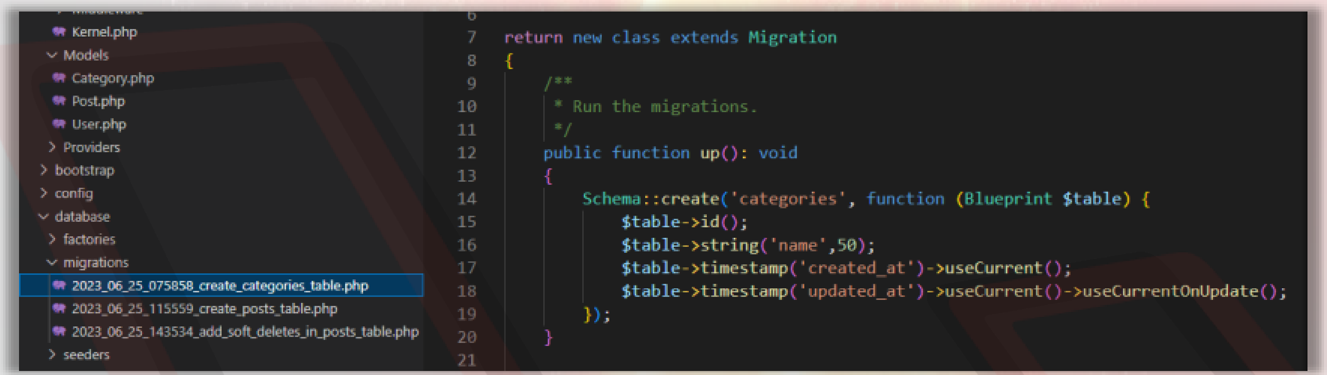
Ostad

Assignment-18

Syed Amir Hamza

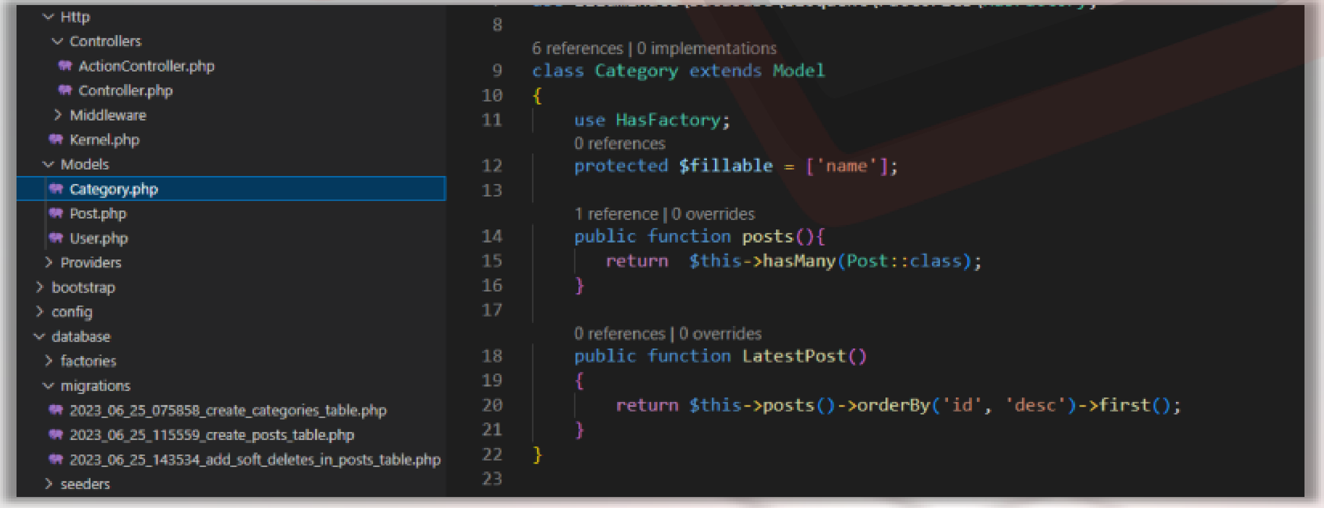
Task 1:
Create a new migration file to add a new table named "categories" to the database. The table should have the following columns:

id (primary key, auto-increment)
name (string)
created_at (timestamp)
updated_at (timestamp)


Kernel.php
Models
Category.php
Post.php
User.php
Providers
bootstrap
config
database
factories
migrations
2023_06_25_075858_create_categories_table.php
2023_06_25_115559_create_posts_table.php
2023_06_25_143534_add_soft_deletes_in_posts_table.php
seeders

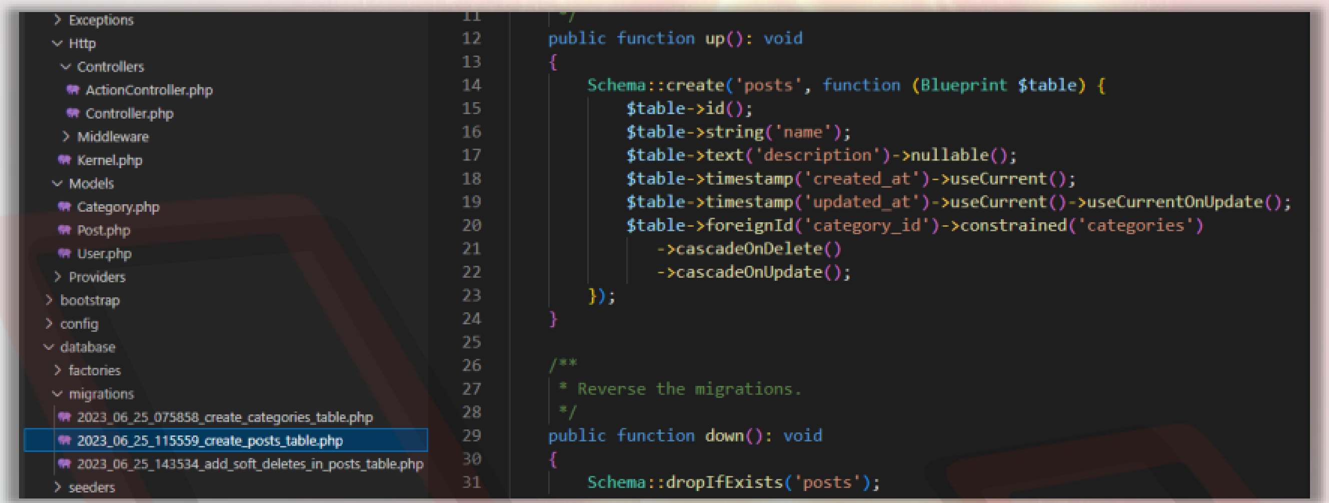
```php
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('categories', function (Blueprint $table) {
            $table->id();
            $table->string('name',50);
            $table->timestamp('created_at')->useCurrent();
            $table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdate();
        });
    }

```

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



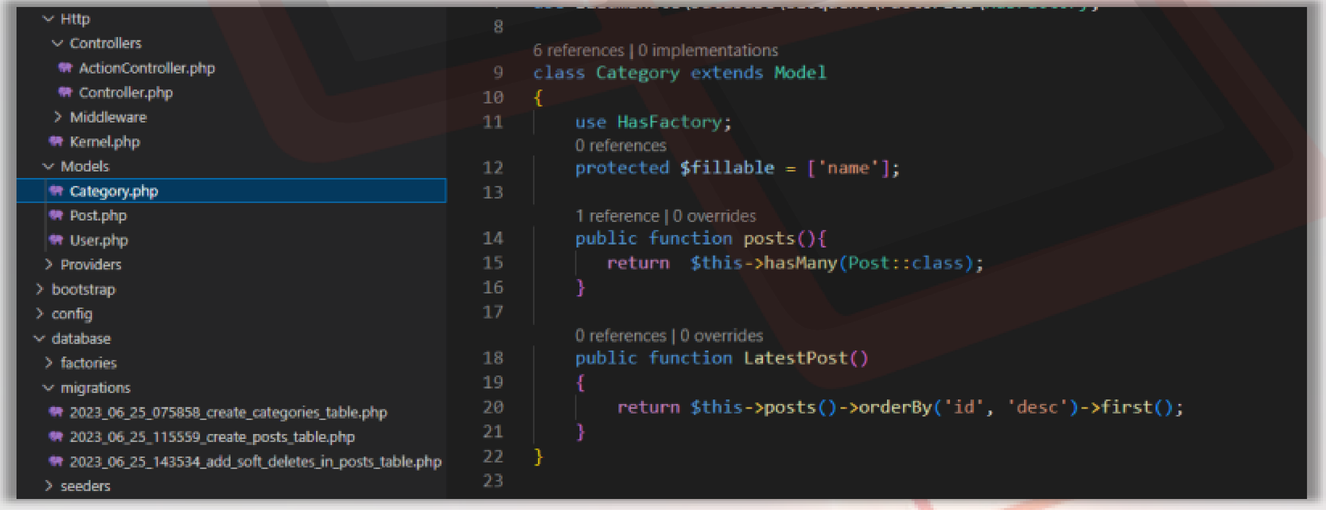
```php

6 references | 0 implementations
class Category extends Model
{
    use HasFactory;
    0 references
    protected $fillable = ['name'];

    1 reference | 0 overrides
    public function posts(){
       return  $this->hasMany(Post::class);
    }

    0 references | 0 overrides
    public function LatestPost()
    {
        return $this->posts()->orderBy('id', 'desc')->first();
    }
}

```

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.



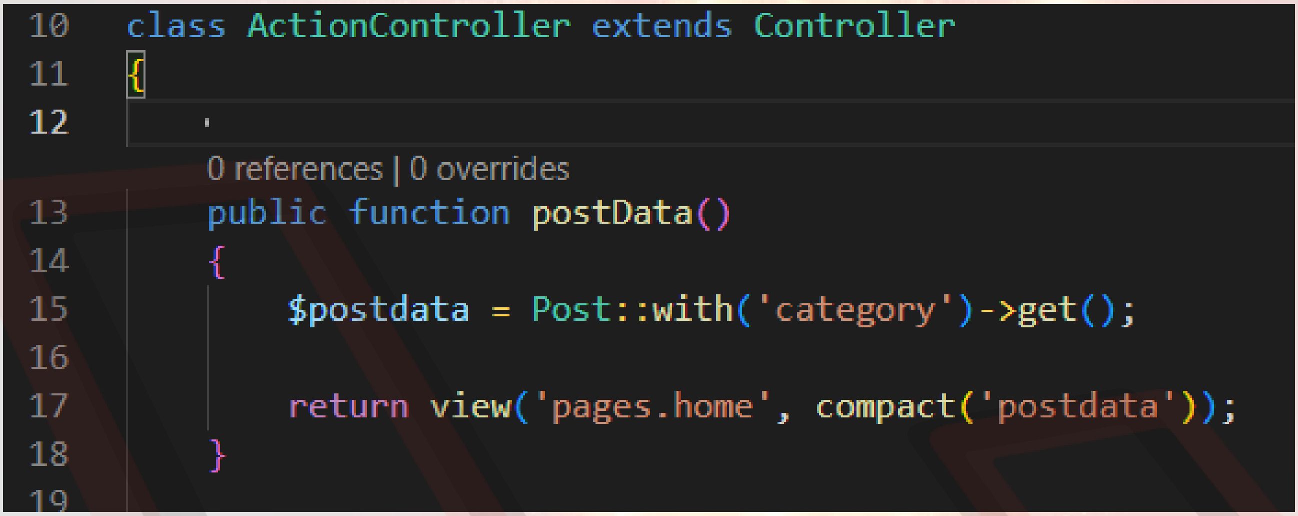
```php
     */
    public function up(): void
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->text('description')->nullable();
            $table->timestamp('created_at')->useCurrent();
            $table->timestamp('updated_at')->useCurrent()->useCurrentOnUpdat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

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('posts');
```

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



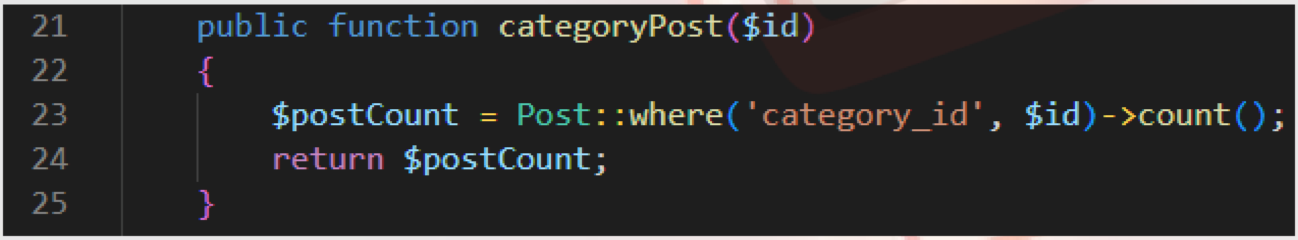
```php

6 references | 0 implementations
class Category extends Model
{
    use HasFactory;
    0 references
    protected $fillable = ['name'];

    1 reference | 0 overrides
    public function posts(){
        return  $this->hasMany(Post::class);
    }

    0 references | 0 overrides
    public function LatestPost()
    {
        return $this->posts()->orderBy('id', 'desc')->first();
    }
}
```

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



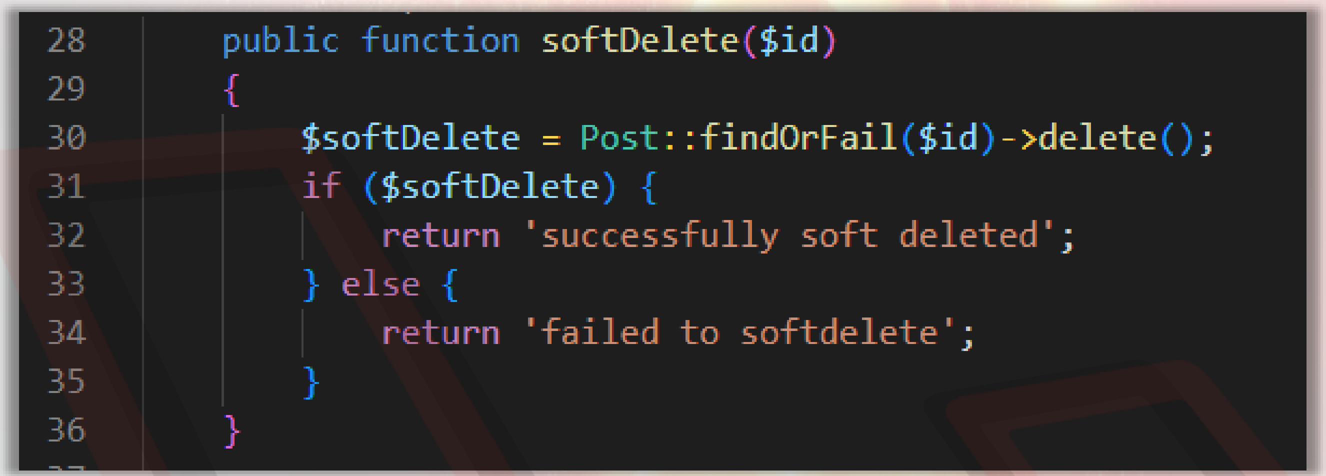
```
class ActionController extends Controller
{

    public function postData()
    {
        $postdata = Post::with('category')->get();

        return view('pages.home', compact('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.



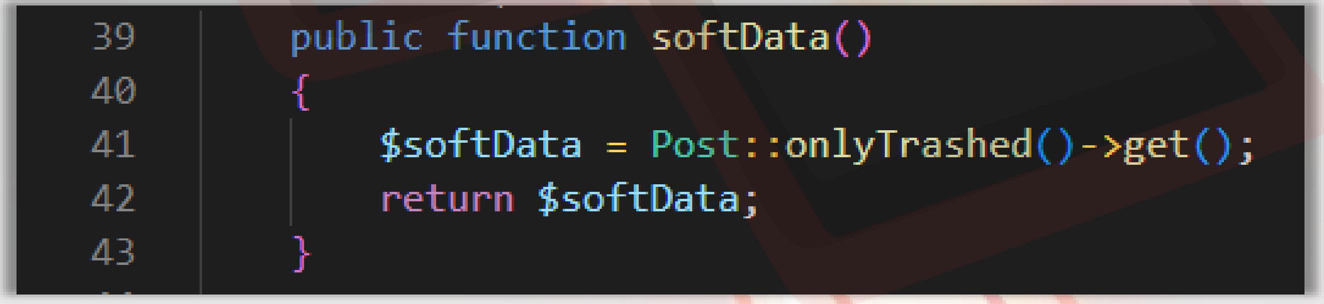
```
    public function categoryPost($id)
    {
        $postCount = Post::where('category_id', $id)->count();
        return $postCount;
    }
```

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.



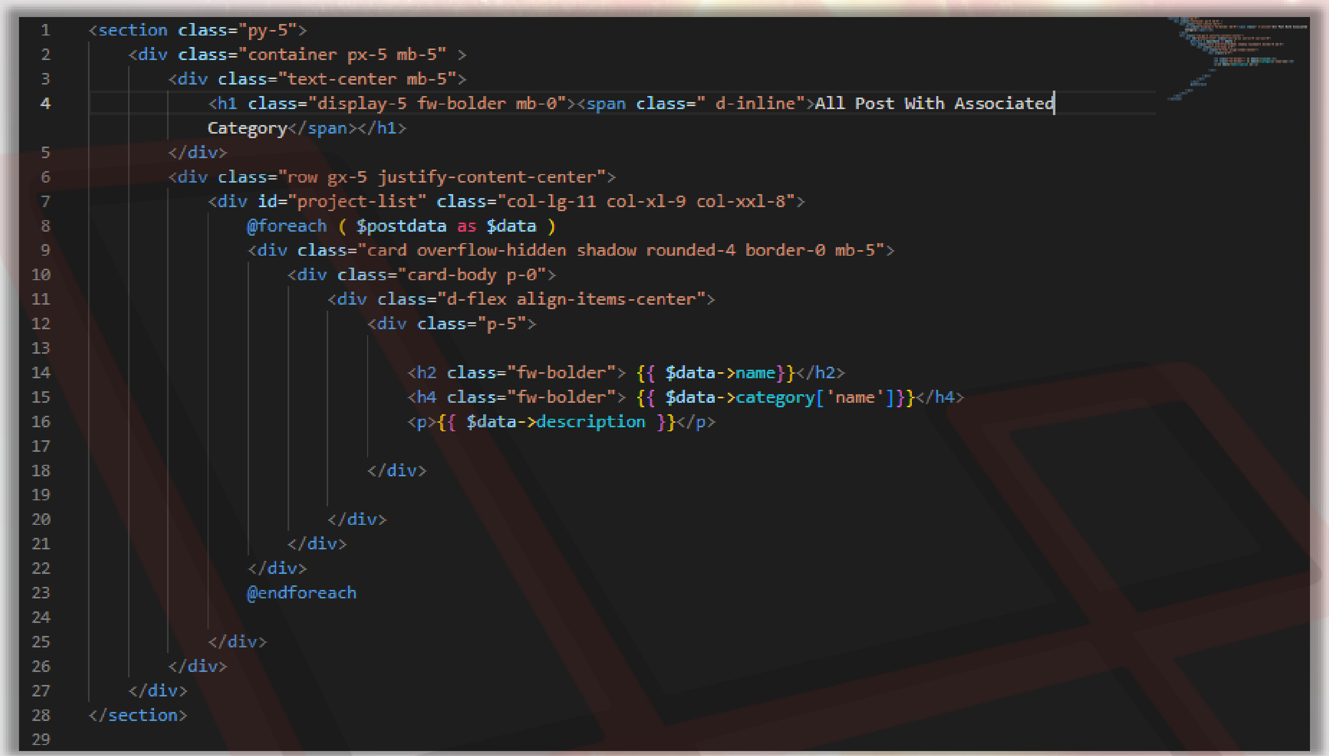
```php
    public function softDelete($id)
    {
        $softDelete = Post::findOrFail($id)->delete();
        if ($softDelete) {
            return 'successfully soft deleted';
        } else {
            return 'failed to softdelete';
        }
    }
```

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.

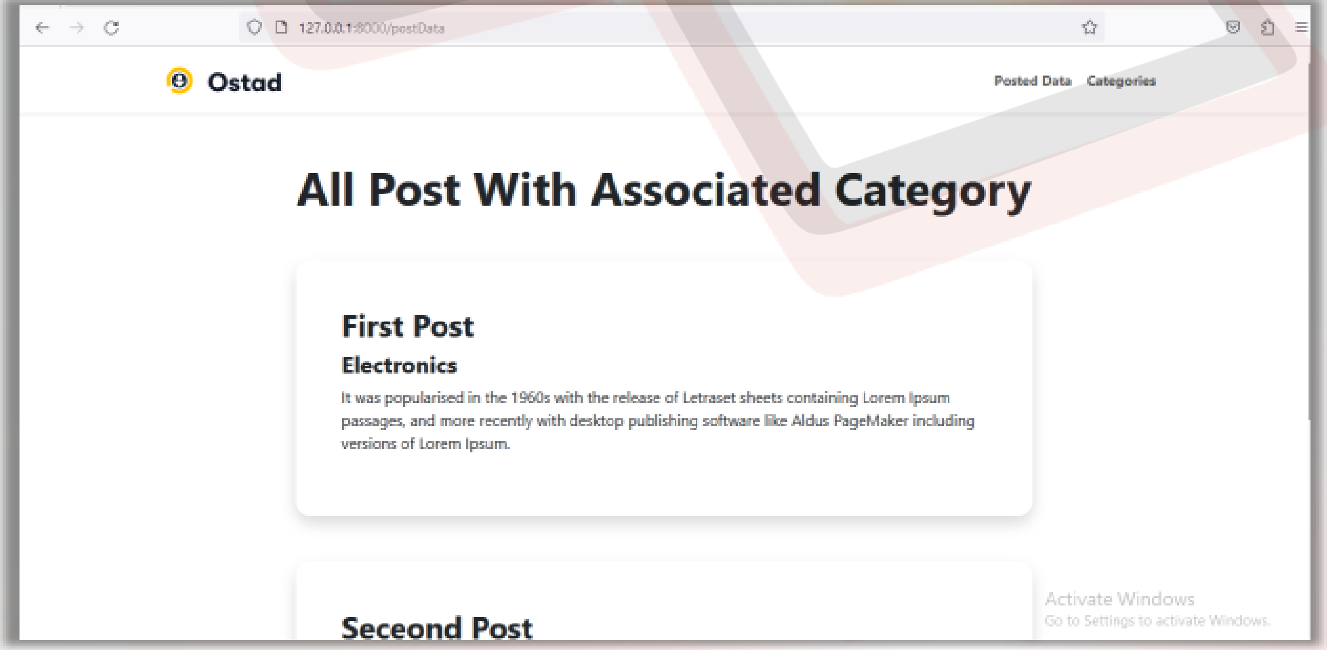


```php
    public function softData()
    {
        $softData = Post::onlyTrashed()->get();
        return $softData;
    }
```

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.



```
<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=" d-inline">All Post With Associated 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>
                    </div>
                </div>
                @endforeach

            </div>
        </div>
    </div>
</section>
```



All Post With Associated Category

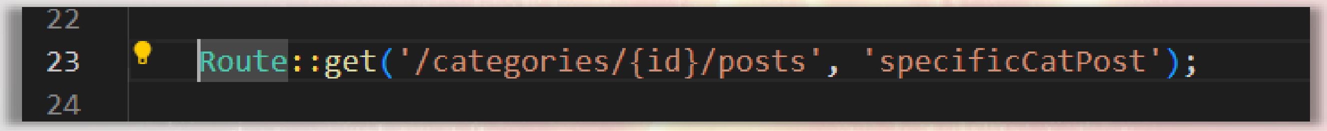
First Post

Electronics

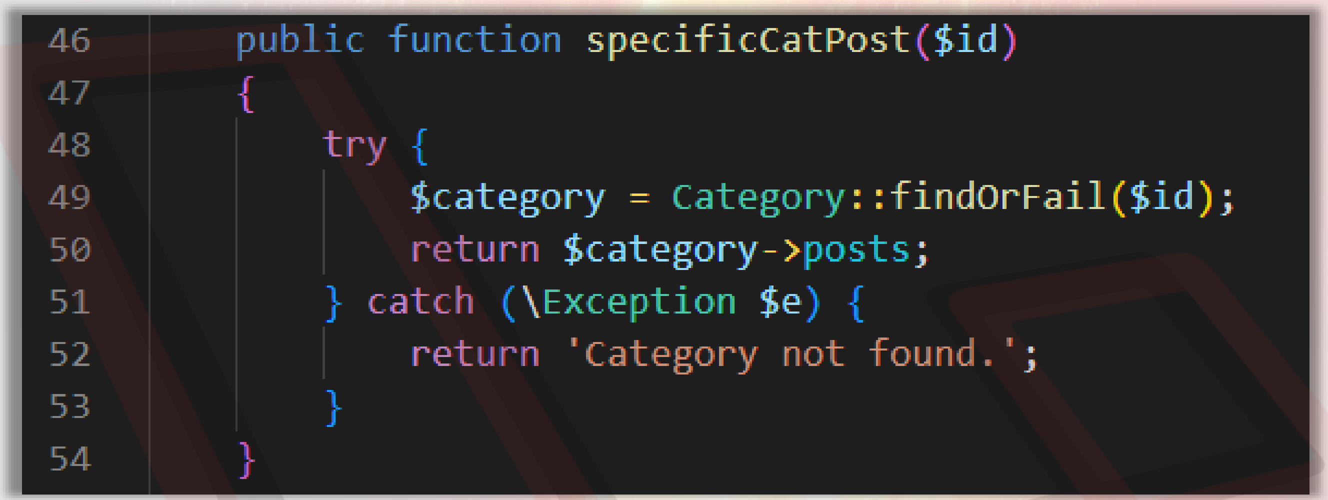
It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Seceond Post

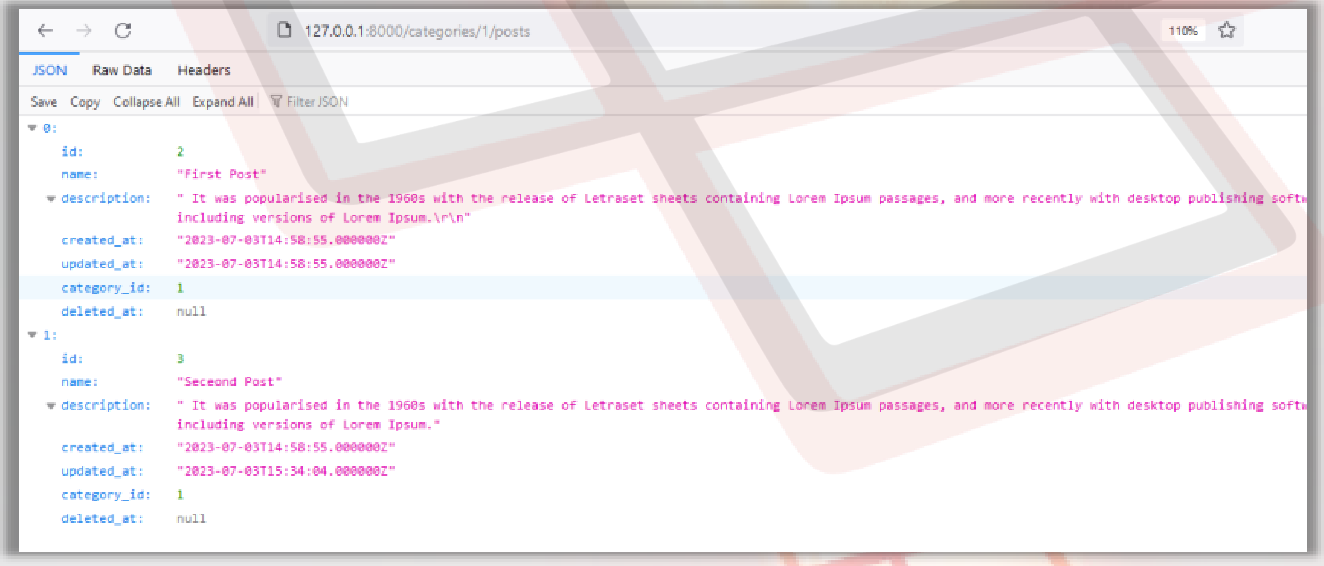
Task 10:
Create a new route in the web.php file to handle the following URL pattern: "/categories/{id}/posts". Implement the corresponding controller method to retrieve all posts belonging to a specific category. The category ID should be passed as a parameter to the method.



```php

Route::get('/categories/{id}/posts', 'specificCatPost');

```

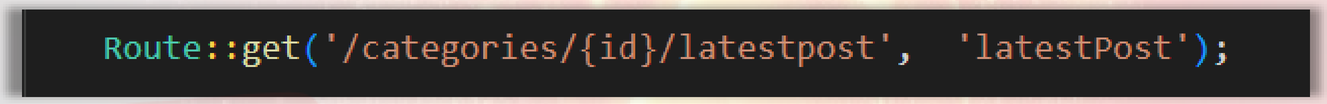


```php
    public function specificCatPost($id)
    {
        try {
            $category = Category::findOrFail($id);
            return $category->posts;
        } catch (\Exception $e) {
            return 'Category not found.';
        }
    }
```

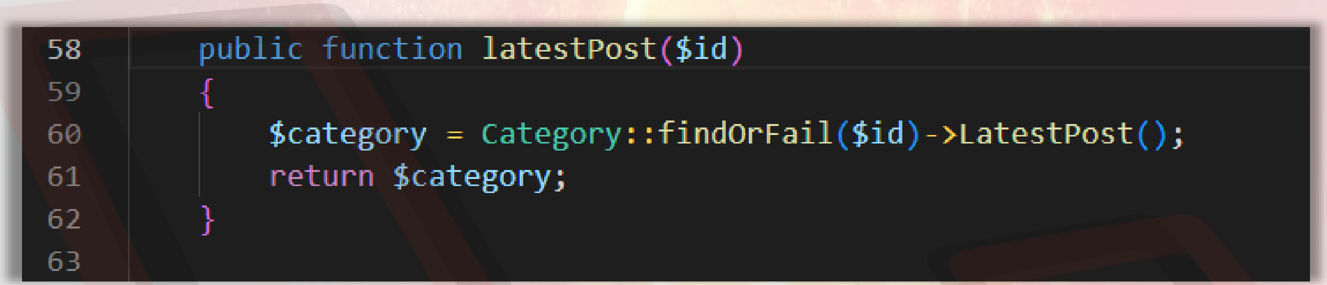


```
▼ 0:
    id:             2
    name:           "First Post"
  ▼ description:    " It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing softw
                    including versions of Lorem Ipsum.\r\n"
    created_at:     "2023-07-03T14:58:55.000000Z"
    updated_at:     "2023-07-03T14:58:55.000000Z"
    category_id:    1
    deleted_at:     null
▼ 1:
    id:             3
    name:           "Seceond Post"
  ▼ description:    " It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing softw
                    including versions of Lorem Ipsum."
    created_at:     "2023-07-03T14:58:55.000000Z"
    updated_at:     "2023-07-03T15:34:04.000000Z"
    category_id:    1
    deleted_at:     null
```

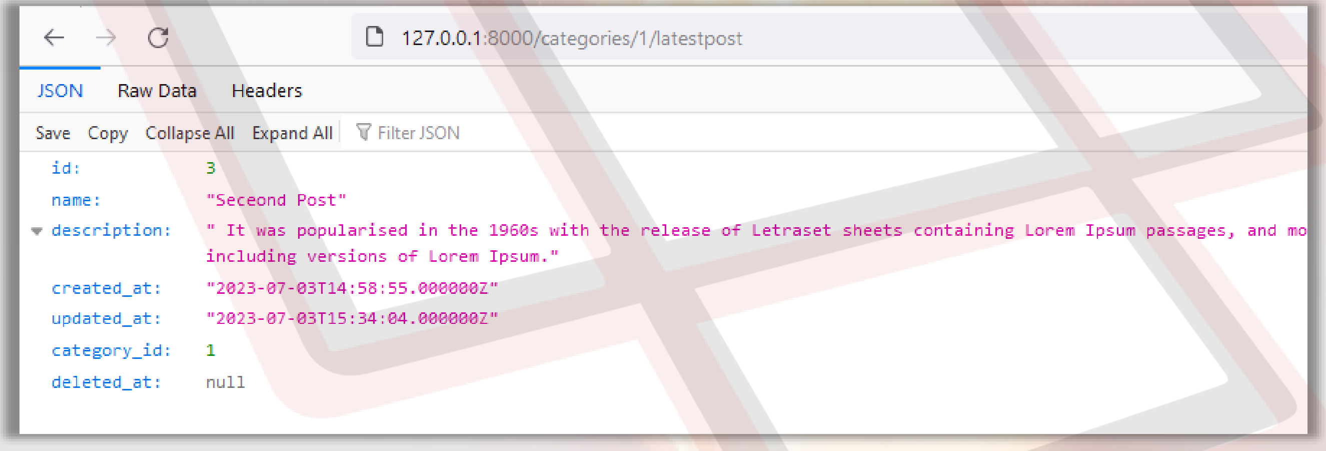
Task 11:
Implement a method in the "Category" model to get the latest post associated with the category. The method should return the post object.



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

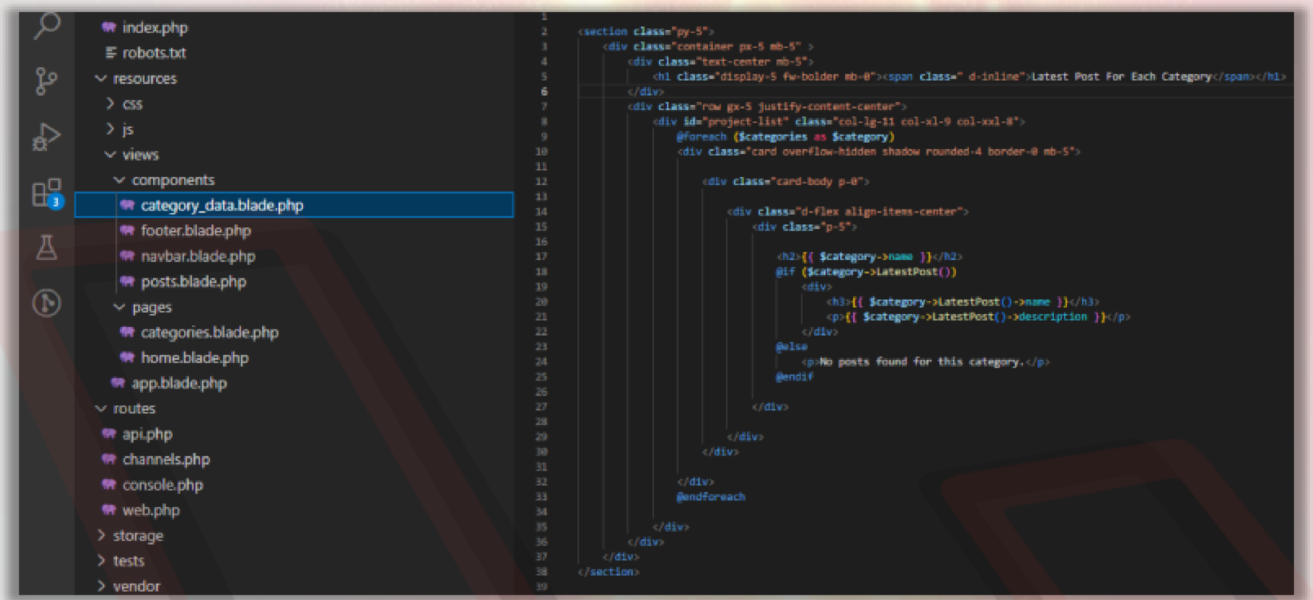


```
    public function latestPost($id)
    {
        $category = Category::findOrFail($id)->LatestPost();
        return $category;
    }

```

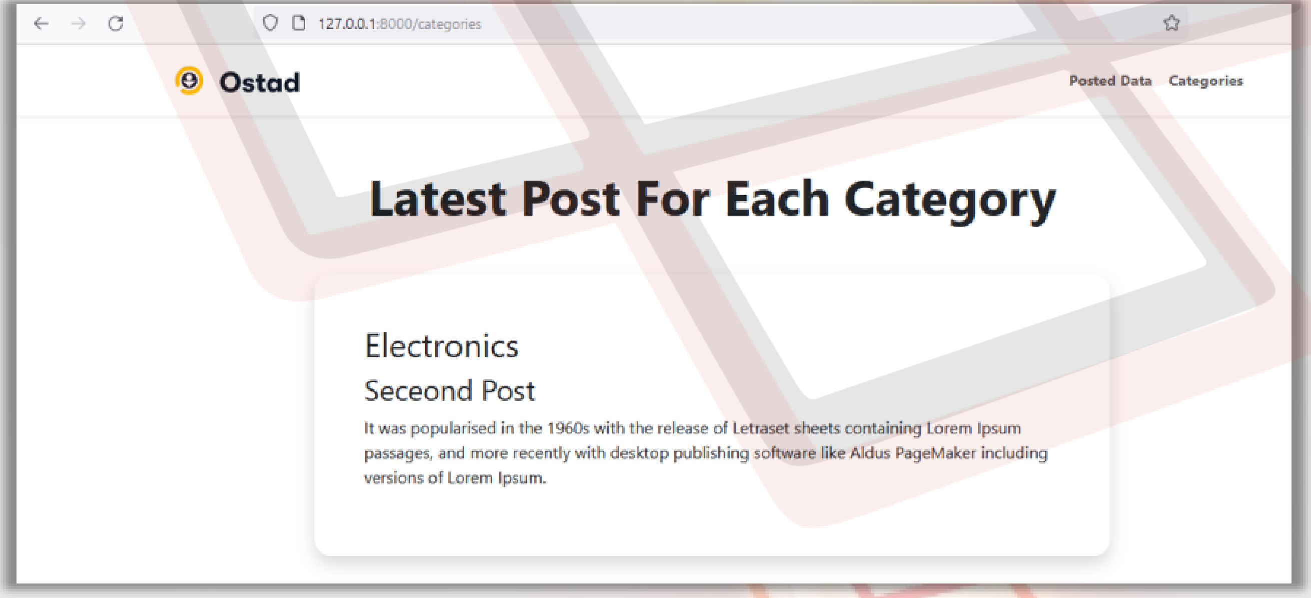


```
id:           3
name:         "Seceond Post"
description:  " It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and mo
              including versions of Lorem Ipsum."
created_at:   "2023-07-03T14:58:55.000000Z"
updated_at:   "2023-07-03T15:34:04.000000Z"
category_id:  1
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=" 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>
```



Latest Post For Each Category

Electronics

Seceond Post

It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.