**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 :**

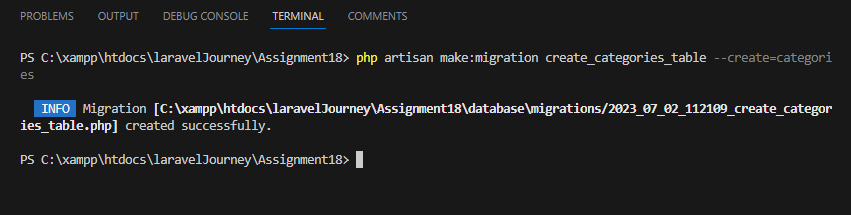
**id (primary key, auto-increment)**

**name (string)**

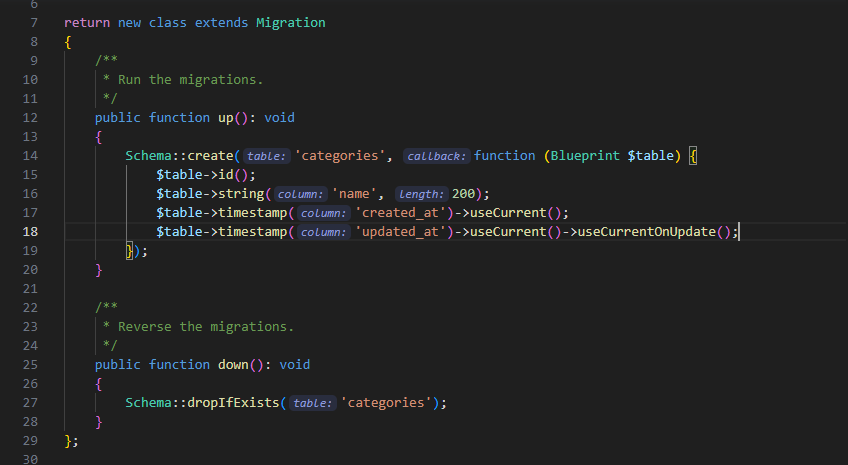
**created\_at (timestamp)**

**updated\_at (timestamp)**

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

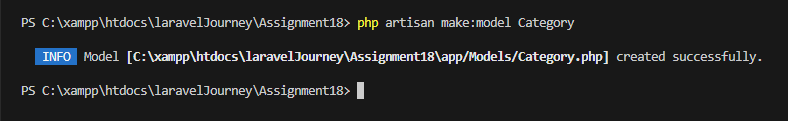


**Then I have add name,created\_at and updated\_at. Here is the another Screenshot below .**

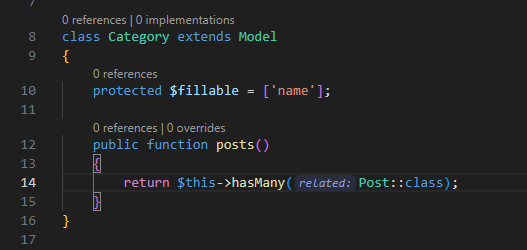


**Task 2 : Create a new model named "Category" associated with the "categories" table. Define the necessary properties and relationships.**

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

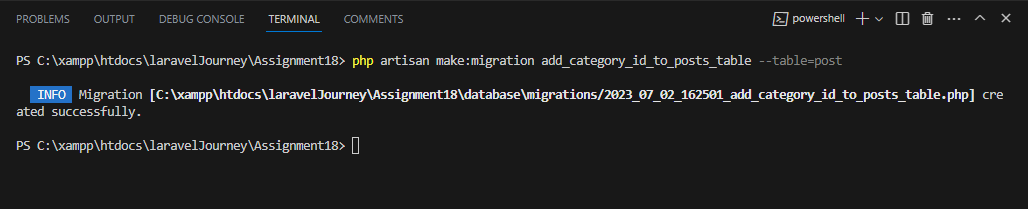


**Then I found this**

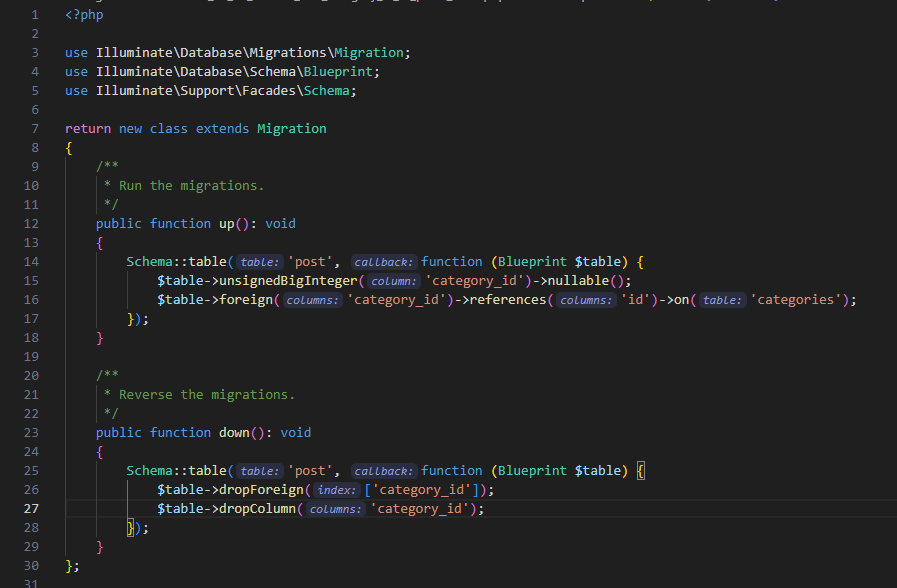


**Task 3 : Write a migration file to add a foreign key constraint to the "posts" table. The foreign key should reference the "categories" table on the "category\_id" column.**

**First I ran a command like this –**

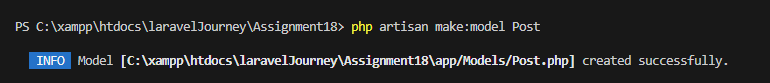


**Then I have got this inside migration**



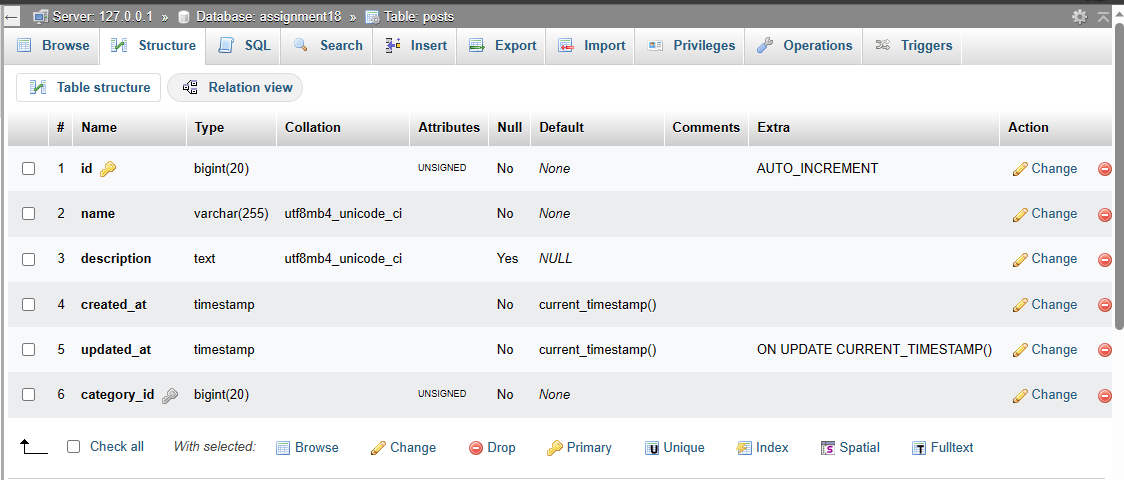
**Task 4: Create a relationship between the "Post" and "Category" models. A post belongs to a category, and a category can have multiple posts.**

**Firstly I have created a model name “Post”**

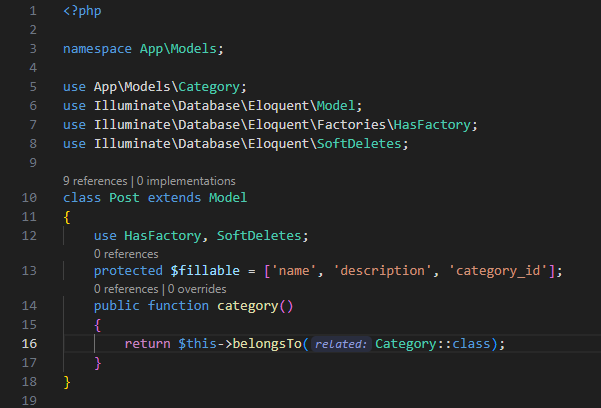


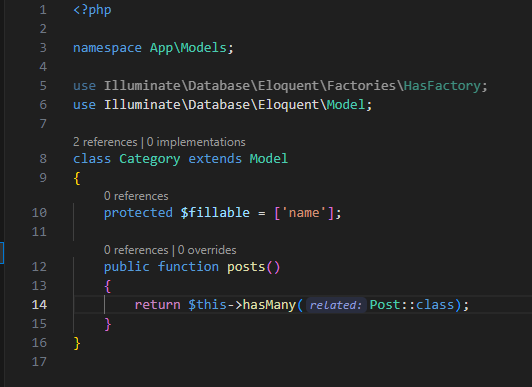
**Then I have created Post table and Then have a relationship with the post and category models**





**Model RelationShip :**

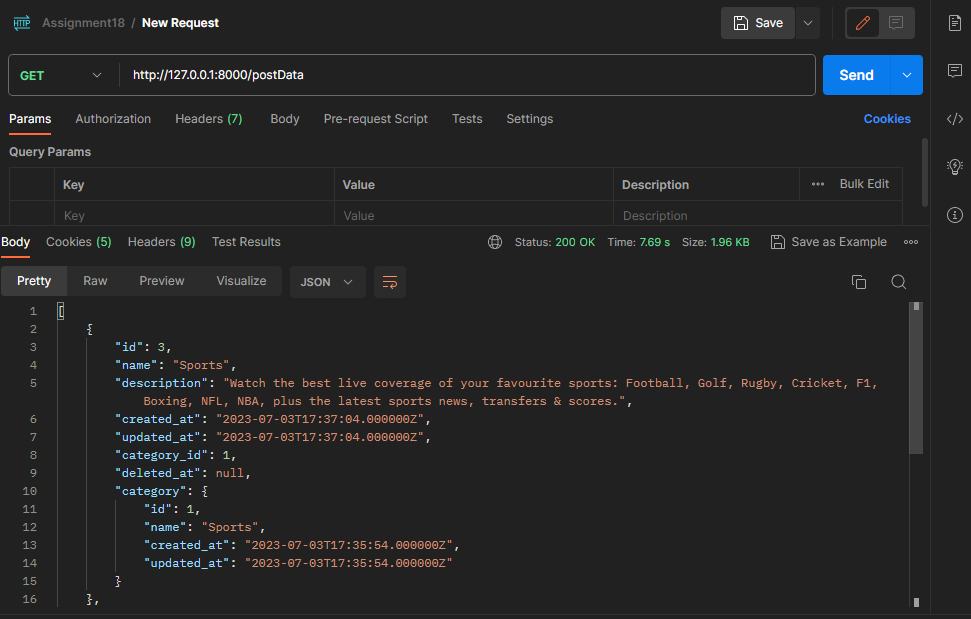




**Task 5 : Write a query using Eloquent ORM to retrieve all posts along with their associated categories. Make sure to eager load the categories to optimize the query.**

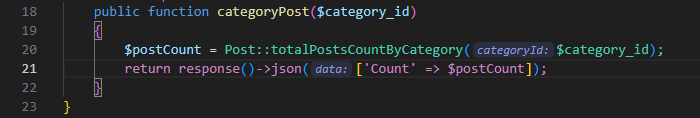
**Here I implement a controller Method :**



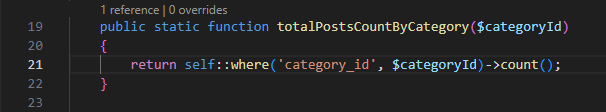


**Task 6 : Implement a method in the "Post" model to get the total number of posts belonging to a specific category. The method should accept the category ID as a parameter and return the count.**

**First I made a function inside ActionController :**



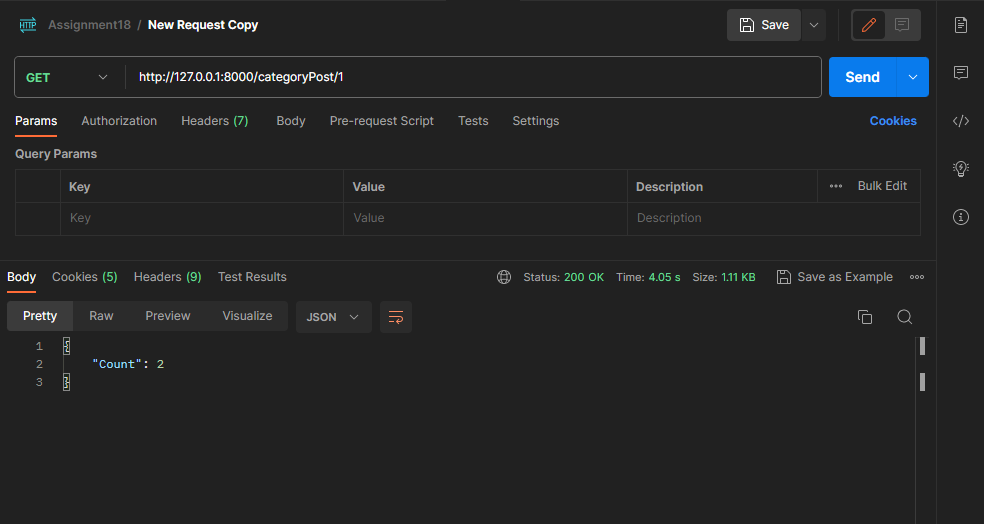
**Implement a Method in Model :**



**Finally I created a Route for this :**

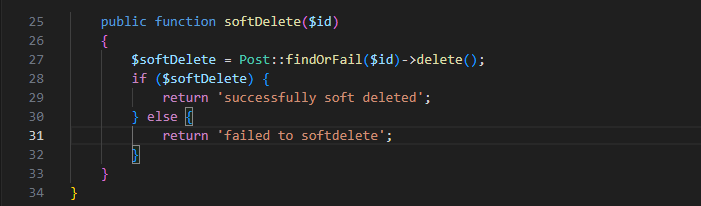


**Output By Postman :**



**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.**

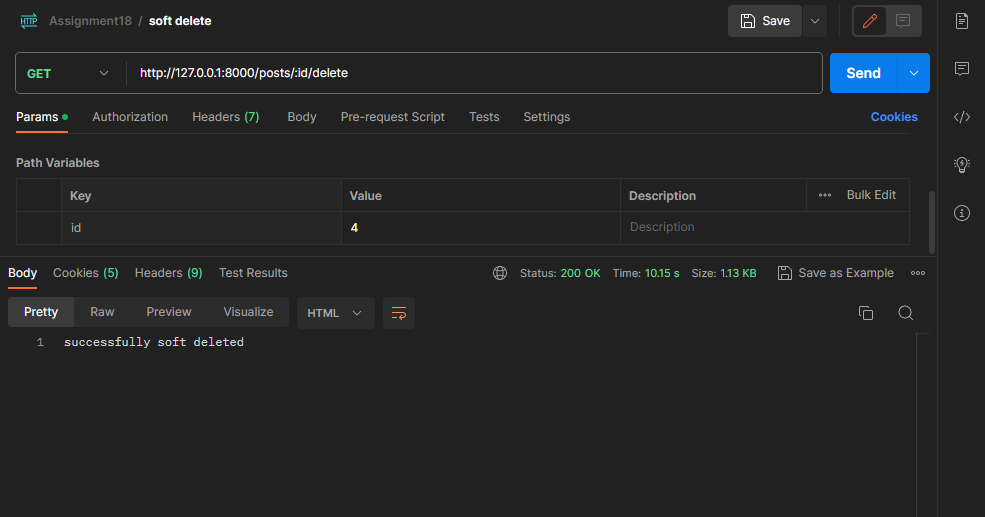
**Controller Method :**



**Route screenshot given bellow:**

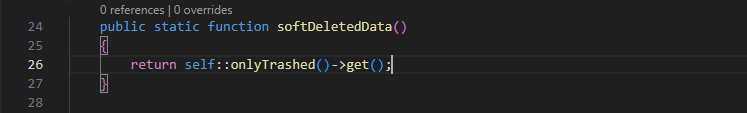


**Output By Postman :**

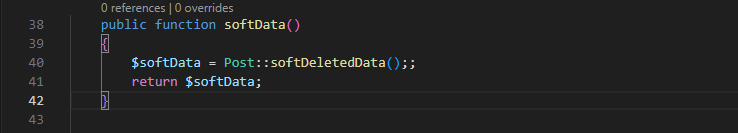


**Task 8 : Implement a method in the "Post" model to get all posts that have been soft deleted. The method should return a collection of soft deleted posts.**

**Implement Method In Post Model :**



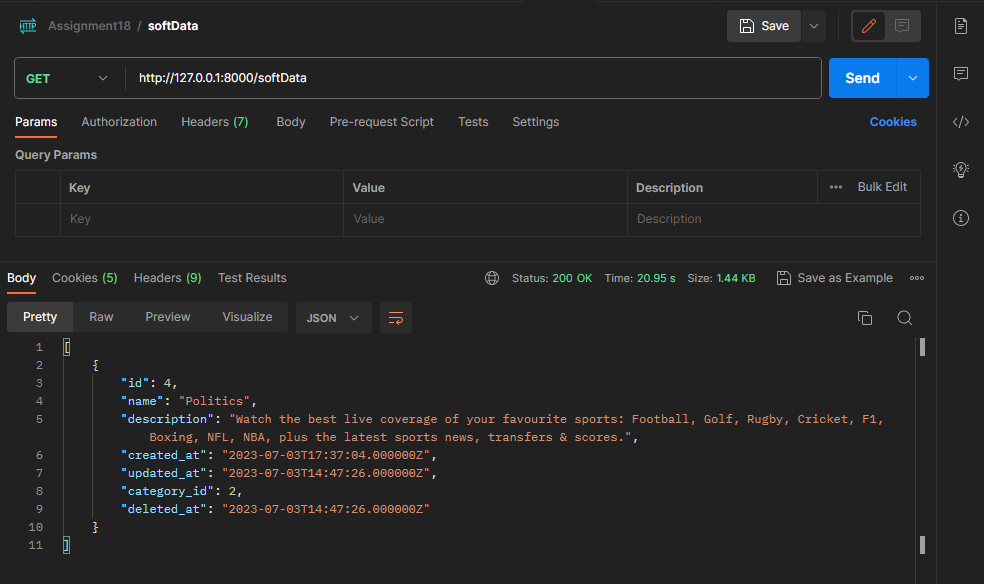
**Implement Controller Method :**



**Implement Route :**



**Output By Postman :**

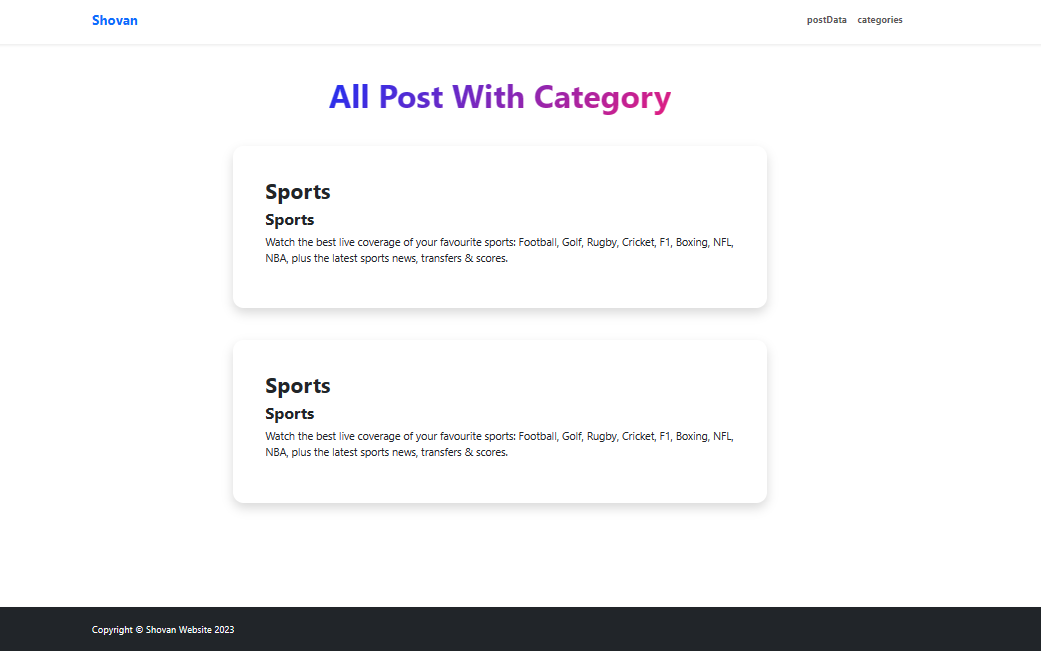


**Task 9 : Write a Blade template to display all posts and their associated categories. Use a loop to iterate over the posts and display their details.**

**Blade File :**



**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.**

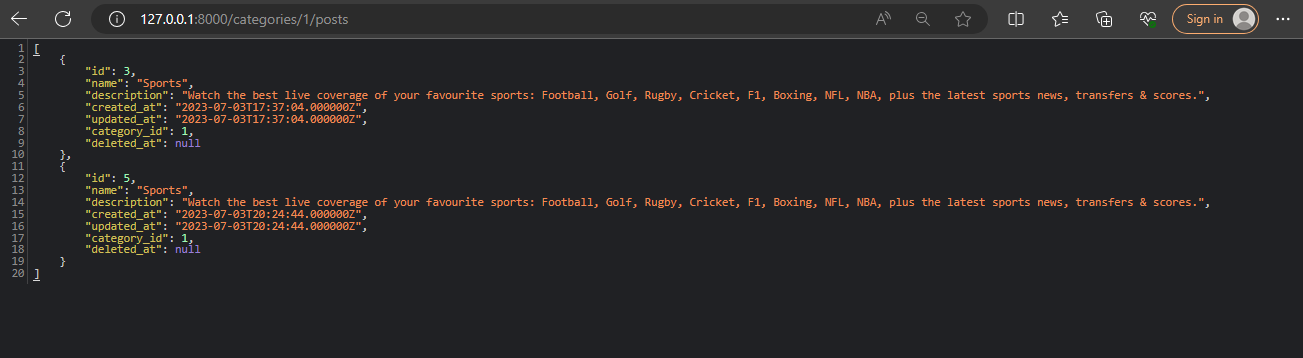
**Route :**



**Controller Method :**

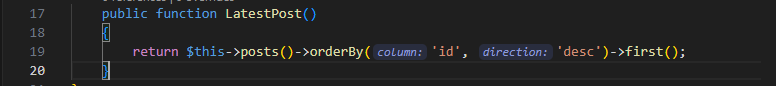


**Output :**



**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.**

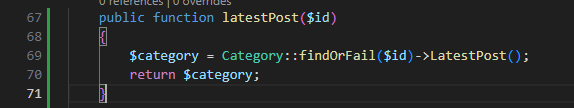
**Implemented method in Category model :**



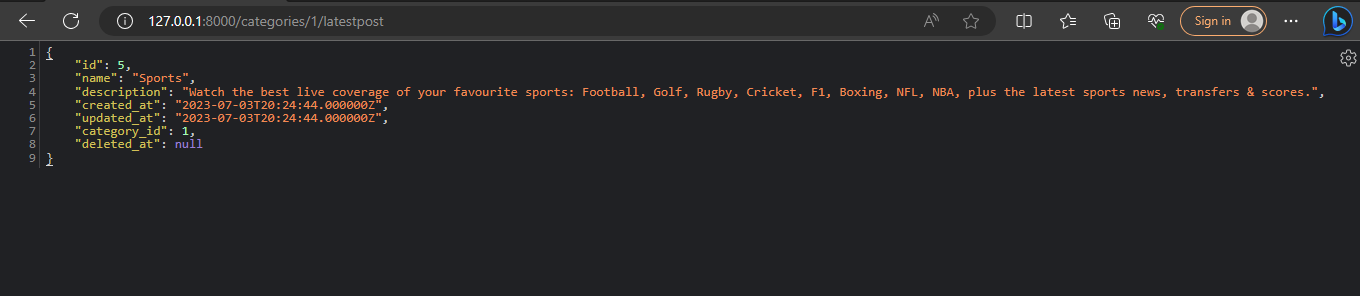
**Implement Route :**



**Implement Controller :**



**Output :**

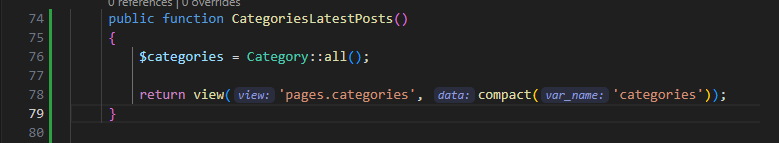


**Task 12 : Write a Blade template to display the latest post for each category. Use a loop to iterate over the categories and display the post details.**

**Blade file :**



**Implement Controller Method :**



**Output :**

