

\*\*\* OSTAD Assignment-18 \*\*\*

Question -1 :

Answer :

php artisan make:migration create\_categories\_table --create=categories  
(php artisan command for commandline)

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateCategoriesTable extends Migration
{
    public function up()
    {
        Schema::create('categories', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('categories');
    }
}
```

php artisan migrate //(php artisan command for commandline)

-----

Question -2 :

Answer :

Step 1: Generate the model file

```
php artisan make:model Category (php artisan command for commandline)
```

Step 2: Define the model properties and relationships

```
namespace App;

use Illuminate\Database\Eloquent\Model;

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

    protected $dates = ['created_at', 'updated_at'];
}
```

Step 3: Use the Category model

```
use App\Category;

$category = Category::create([
    'name' => 'Category Name',
]);

$categories = Category::where('name', 'like', '%keyword%')->get();
```

-----

Question -3 :

Answer :

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class AddCategoryIdToPostsTable extends Migration
{
    public function up()
    {
        Schema::table('posts', function (Blueprint $table) {
```

```

        $table->unsignedBigInteger('category_id')->nullable();

        $table->foreign('category_id')
            ->references('id')
            ->on('categories')
            ->onDelete('set null');
    });
}

public function down()
{
    Schema::table('posts', function (Blueprint $table) {
        $table->dropForeign(['category_id']);
        $table->dropColumn('category_id');
    });
}
}

```

-----

Question -4 :

Answer :

```

namespace App;

use Illuminate\Database\Eloquent\Model;

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

    protected $dates = ['created_at', 'updated_at'];

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

```

```

namespace App;

use Illuminate\Database\Eloquent\Model;

class Post extends Model
{
    protected $fillable = ['title', 'content', 'category_id'];

    protected $dates = ['created_at', 'updated_at'];

    public function category()
    {
        return $this->belongsTo(Category::class);
    }
}

// Get the category of a post
$post = Post::find(1);
$category = $post->category;

// Get all posts belonging to a category
$category = Category::find(1);
$posts = $category->posts;

// Create a new post and associate it with a category
$category = Category::find(1);
$post = $category->posts()->create([
    'title' => 'New Post',
    'content' => 'Lorem ipsum dolor sit amet.',
]);

```

-----

Question -5 :

Answer :

```

$posts = Post::with('category')->get();

foreach ($posts as $post) {

```

```
    echo "Post Title: " . $post->title . "\n";
    echo "Category: " . $post->category->name . "\n\n";
}
```

-----

Question -6 :

Answer :

```
namespace App;

use Illuminate\Database\Eloquent\Model;

class Post extends Model
{
    protected $fillable = ['title', 'content', 'category_id'];

    protected $dates = ['created_at', 'updated_at'];

    public function category()
    {
        return $this->belongsTo(Category::class);
    }

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

$category = Category::find(1);
$postCount = Post::getPostsCountByCategory($category->id);

echo "Total Posts in Category: " . $postCount;
```

-----

Question -7 :

Answer :

```
use App\Http\Controllers\PostController;
```

```
Route::delete('/posts/{id}/delete', [PostController::class,  
'destroy']);
```

```
namespace App\Http\Controllers;
```

```
use App\Post;
```

```
use Illuminate\Http\Request;
```

```
class PostController extends Controller
```

```
{
```

```
    public function destroy($id)
```

```
    {
```

```
        $post = Post::findOrFail($id);
```

```
        $post->delete();
```

```
        return redirect()->back()->with('message', 'Post deleted  
successfully.');
```

```
    }
```

```
}
```

```
-----
```

Question -8 :

Answer :

```
public static function getSoftDeletedPosts()
```

```
{
```

```
    return self::onlyTrashed()->get();
```

```
}
```

```
$softDeletedPosts = Post::getSoftDeletedPosts();
```

```
-----
```

Question -9 :

Answer :

```

@extends('layouts.app')

@section('content')
    <h1>All Posts</h1>

    @foreach ($posts as $post)
        <div>
            <h2>{{ $post->title }}</h2>
            <p>{{ $post->content }}</p>
            <p>Category: {{ $post->category->name }}</p>
            <hr>
        </div>
    @endforeach
@endsection

use App\Post;

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

    return view('posts', compact('posts'));
}

-----

```

Question -10 :

Answer :

```

use App\Http\Controllers\CategoryController;

Route::get('/categories/{id}/posts', [CategoryController::class,
'getPostsByCategory']);

namespace App\Http\Controllers;

use App\Category;
use Illuminate\Http\Request;

class CategoryController extends Controller
{

```

```

public function getPostsByCategory($id)
{
    $category = Category::findOrFail($id);
    $posts = $category->posts;

    return view('category_posts', compact('category', 'posts'));
}
}

@extends('layouts.app')

@section('content')
    <h1>{{ $category->name }} Posts</h1>

    @foreach ($posts as $post)
        <div>
            <h2>{{ $post->title }}</h2>
            <p>{{ $post->content }}</p>
            <hr>
        </div>
    @endforeach
@endsection

```

-----

Question -11 :

Answer :

```

public function latestPost()
{
    return $this->hasOne(Post::class)->latest();
}

$category = Category::find(1);
$latestPost = $category->latestPost;

if ($latestPost) {
    echo "Latest Post Title: " . $latestPost->title;
} else {
    echo "No posts found for this category.";
}

```



-----

Question -12 :

Answer :

Step : Write the Blade template

```
@extends('layouts.app')

@section('content')
    <h1>Latest Posts by Category</h1>

    @foreach ($categories as $category)
        <div>
            <h2>{{ $category->name }}</h2>
            @if ($category->latestPost)
                <h3>{{ $category->latestPost->title }}</h3>
                <p>{{ $category->latestPost->content }}</p>
            @else
                <p>No posts found for this category.</p>
            @endif
            <hr>
        </div>
    @endforeach
@endsection
```

Step : Pass data to the Blade template

```
use App\Category;

public function index()
{
    $categories = Category::with('latestPost')->get();

    return view('latest_posts', compact('categories'));
}
```

-----

\*\*\*

Thanks

\*\*\*