Practical:10

Aim: Buil d a C MS w ith Larevel.

Step 1:

Setting up laravel

```
composer create-project --prefer-dist laravel/laravel cms
```

Configure Database: Update the '.env' file with your database credentials.

Create Database Tables: Use Laravel migrations to define your database schema.

```
php artisan make:migration create_posts_table
```

Update the migration file with the necessary columns and then run:

```
php artisan migrate
```

Step 2: Authentication.

Install Laravel UI (Optional): Laravel UI provides a simple way to scaffold authentication and frontend components.

```
composer require laravel/ui
php artisan ui vue --auth
npm install && npm run dev
```

Step 3: CRUD Operations.

Generate Models and Controllers: Use Artisan commands to generate models and controllers.

```
php artisan make:model Post -m
php artisan make:controller PostController –resource
```

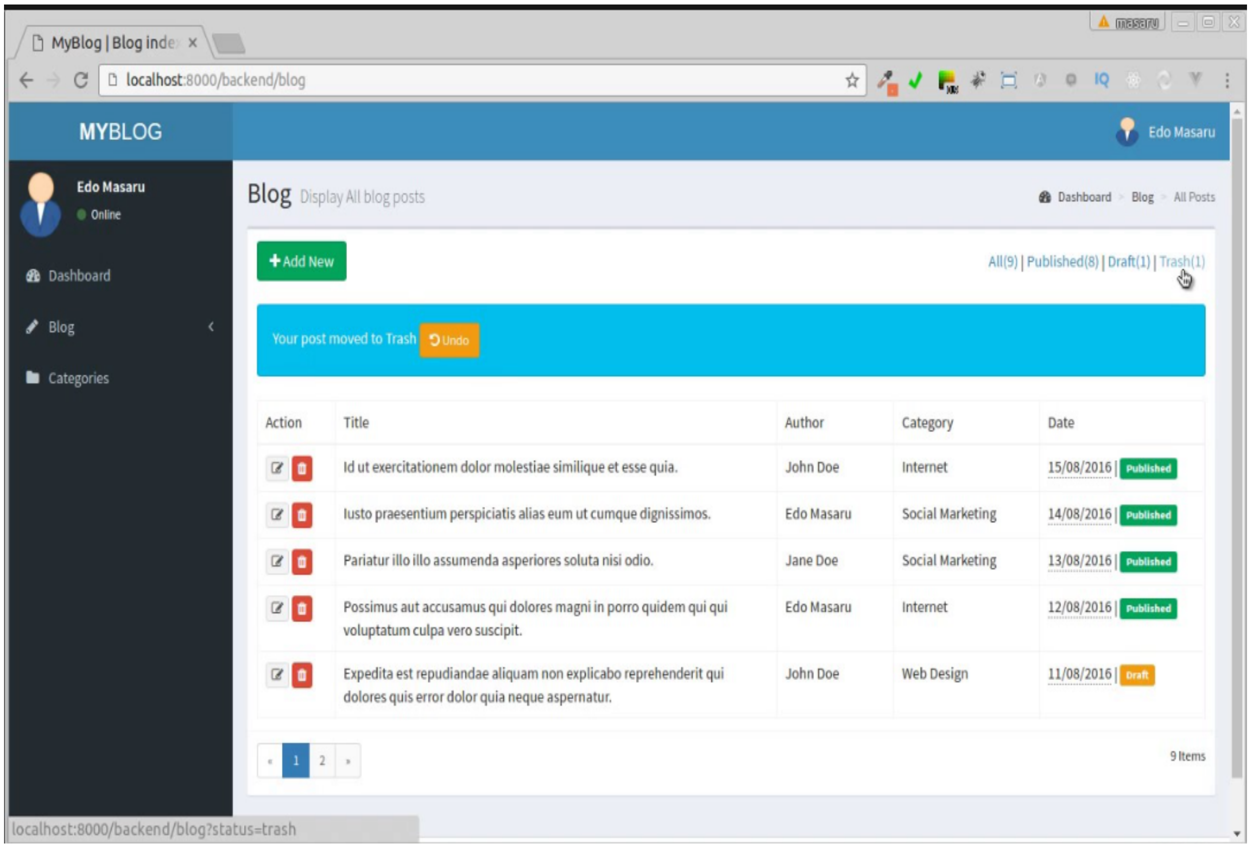
Define Routes: Define routes for CRUD operations in routes/web.php.

Implement CRUD Functionality: Implement CRUD functionality in your controller.

Step 4: views.
Step 5: Authentication Middleware
Step 6: Frontend
Step 7: Additional Features
Step 8: Testing
Step 9: Deployment
Step 10: Maintenance

OUTPUT:-

Exampole (Build blogs + CMS).



MyBlog | Blog inde ×

localhost:8000/backend/blog

MYBLOG — Edo Masaru

Edo Masaru
Online

Dashboard
Blog
Categories

Blog Display All blog posts — Dashboard > Blog > All Posts

+ Add New — All(9) | Published(8) | Draft(1) | Trash(1)

Your post moved to Trash Undo

Action	Title	Author	Category	Date
	Id ut exercitationem dolor molestiae similique et esse quia.	John Doe	Internet	15/08/2016 | Published
	Iusto praesentium perspiciatis alias eum ut cumque dignissimos.	Edo Masaru	Social Marketing	14/08/2016 | Published
	Pariatur illo illo assumenda asperiores soluta nisi odio.	Jane Doe	Social Marketing	13/08/2016 | Published
	Possimus aut accusamus qui dolores magni in porro quidem qui qui voluptatum culpa vero suscipit.	Edo Masaru	Internet	12/08/2016 | Published
	Expedita est repudiandae aliquam non explicabo reprehenderit qui dolores quis error dolor quia neque aspernatur.	John Doe	Web Design	11/08/2016 | Draft

« 1 2 » — 9 Items

localhost:8000/backend/blog?status=trash

<u>CONCLUSION:-</u>Building a CMS with Laravel provides a robust framework for efficient content management, offering scalability, customization, and security. Leveraging Laravel's ecosystem facilitates the development of feature-rich CMS platforms tailored to diverse project requirements, ensuring a seamless user experience and streamlined content management.

Practical:09

Aim: De monstrate How It Has Made: Laravel Router

Step 1:

Create a Controller

First, let's create a controller that will handle the incoming requests. In your terminal, run the following command to generate a controller:

```
php artisan make:controller MyController
```

Step 2:

Define Routes

Next, you need to define routes in the 'routes/web.php' file. Open the file and define your routes. Here's an example:

```
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\MyController;
Route::get('/hello', [MyController::class, 'hello']);
Route::get('/goodbye', [MyController::class, 'goodbye']);
```

Step 3:

Implement Controller Methods

Now, let's implement the methods in the MyController controller. Open MyController.php in the app/Http/Controllers directory and define the hello and goodbye methods:

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class MyController extends Controller
{
   public function hello()
   {
      return "Hello, World!";
   }
   public function goodbye()
   {
      return "Goodbye, World!";
   }
}
```

Step 4:

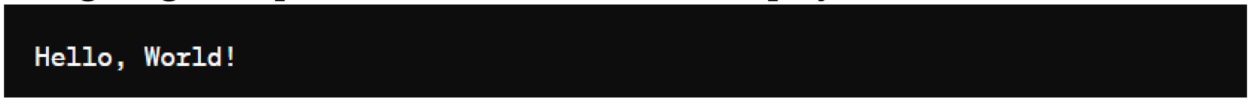
Run the Application

That's it! You have created a basic router in Laravel. Now, you can run your Laravel application using the following command:

```
php artisan serve
```

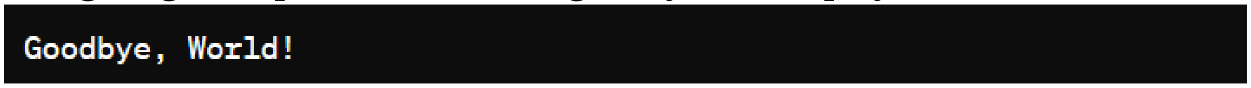
OUTPUT:-

Navigating to http://localhost:8000/hello will display:



```
Hello, World!
```

Navigating to http://localhost:8000/goodbye will display:



```
Goodbye, World!
```