Module 7

Laravel

1) Create following Routes (for frontend side application).

a) Home page

b) About us

c) Contact us

d) Gallery

e) Registration

f) Login

a) Home page

Ans. Route::get(‘/homepage’, function()

{

return view (‘website.homepage’);

});

b) About us

Ans.Route::get(‘/about, function()

{

return view (‘website.about);

});

c) Contact us

Ans. Route::get(‘/contact, function()

{

return view (‘website.contact);

});

d) Gallery

Ans.Route::get(‘/gallery, function()

{

return view (‘website.gallery);

});

e) Registration

Ans.Route::get(‘/registration, function()

{

return view (‘website.registration);

});

f) Login

Ans.Route::get(‘/login, function()

{

return view (‘website.login);

});

2). How to pass Multiple Variable in route?

Ans. Route::get('/user/{id}/{name}', function ($id, $name) {

return "User ID: " . $id . " and Name: " . $name;

});3). How to pass variable which can be null in Route?

Ans. Route::get('/user/{id}/{name?}', function ($id, $name = null) {

return "User ID: " . $id . " and Name: " . ($name ?: "not provided");

});4). Create custom auto using middleware.

Ans.Step 1: Create Middleware

Open the terminal and execute the following command to create custom middleware in laravel 8. So let’s open your command prompt and execute below command on it to learn PHP middleware:

After successfully create middleware, go to app/http/kernel.php and register your custom middleware here:

app/Http/Kernel.php

<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel

{

....

/\*\*

\* The application's route middleware.

\*

\* These middleware may be assigned to groups or used individually.

\*

\* @var array

\*/

protected $routeMiddleware = [

....

'checkStatus' => \App\Http\Middleware\CheckStatus::class,

];

}

Step 2: Implement Logic In Middleware:

After successfully register your middleware in laravel project, go to app/http/middleware and implement your logic here :

app/Http/Middleware/CheckStatus.php

<?php

namespace App\Http\Middleware;

use Closure;

classCheckStatus

{

/\*\*

\* Handle an incoming request.

\*

\* @param \Illuminate\Http\Request $request

\* @param \Closure $next

\* @return mixed

\*/

public function handle($request, Closure $next)

{

if (auth()->user()->status == 'active') {

return $next($request);

}

return response()->json('Your account is inactive');

}

}

?>

Step 3:Add Route

Simply create a laravel route and use custom middleware with routes for filter every HTTP request:

routes/web.php

use App\Http\Controllers\HomeController;

use App\Http\Middleware\CheckStatus;

Route::middleware([CheckStatus::class])->group(function(){

Route::get('home', [HomeController::class,'home']);

Step 4: Add Method In Controller

Create one method name home and add this method on HomeController.php file, which is placed on \*\*app/Http/Controllers/ \*\*directory Welare learn the middleware tutorial:

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

classHomeController extends Controller

{

public function home()

{

dd('You are active');

}

}

?>

5). Generate Resource Controller for employee.

Ans. Php artisan make:controller Employee--resource

main :Php artisan make:controller Employee--resource --model=User

6). In Employee Controller’s action Call Middleware.

Ans.

7). How to remove route caching?

Ans. php artisan route:clear

8). Create Custom Macro For search User.

Ans.To create a custom macro for searching users in Laravel, you can extend the functionality of Eloquent models using macros. Here's how you can define a custom macro for searching users:

1.Open your Laravel application and navigate to the App\Providers\AppServiceProvider.php file.

2.Within the boot() method of the AppServiceProvider class, define your custom macro for searching users.

3.Assuming you have a User model, you can add a macro to search users based on specific criteria.

<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;

use App\Models\User;

use Illuminate\Database\Eloquent\Builder;

classAppServiceProvider extends ServiceProvider

{

/\*\*

\* Bootstrap any application services.

\*

\* @return void

\*/

public function boot()

{

// Define custom macro for searching users by name

User::macro('searchByName', function ($name) {

return $this->where('name', 'like', "%$name%");

});

}

/\*\*

\* Register any application services.

\*

\* @return void

\*/

public function register()

{

//

}

}

After defining the macro, you can use it in your application like this:

use App\Models\User;

// Search users by name

$users = User::searchByName('John')->get();

Remember to import the necessary classes at the top of your file:

use App\Models\User;

use Illuminate\Database\Eloquent\Builder;