

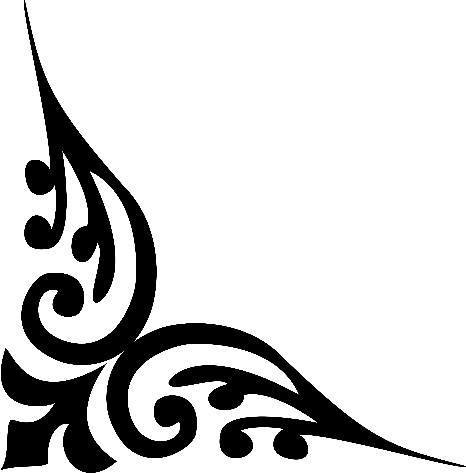
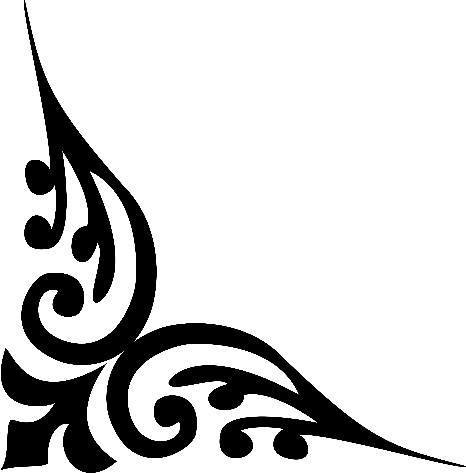
MODULE – 7(Php introduction of laravel php framework industry)

**MODULES:-[7]**

**Submitted to :-**

**Mr. Raj Nagar**

**Submitted by :-**

  **Janvi Panchal**

**MODULES:-[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**

**Ans:-**

1. **Home page**

Route::get('Home page',function () {

return view('admin/dashboard');

});

**b) About us**

Route::get('About us',function () {

return view('admin/dashboard');

});

1. **Contact us**

Route::get('Contact us',function () {

return view('admin/dashboard');

});

1. **Gallery**

Route::get('Gallery',function () {

return view('admin/dashboard');

});

**e) Registration**

Route::get('Registration',function () {

return view('admin/dashboard');

});

1. **Login**

Route::get('Login',function () {

return view('admin/dashboard');

});

1. **How to pass Multiple Variable in route?**

**Ans:-**

Route::get('/user/{id}/profile/{username}', 'UserController@profile')->name('user.profile');

1. **How to pass variable which can be null in Route?**

**Ans:-**

Route::get('example/{id?}', function ($id = null) {

// Your route logic here

});

1. **Create custom auto using middleware.**

**Ans:-**

**Step1 :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:

**php artisan make:middleware CheckStatus**

**<?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;

class CheckStatus

{

/\*\*

\* 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 We lare learn the middleware tutorial:

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller

{

public function home()

{

dd('You are active');

}

}

?>

1. **Generate Resource Controller for employee.**

**Ans:-**

**Php artisan make:controller EmployeeController** **–resource**

**main : Php artisan make:controller EmployeeController --resource –**

**model=Employee**

1. **In Employee Controller’s action Call Middleware.**

**Ans:-**

**<?php**

**namespace App\Http\Controllers;**

**use Illuminate\Http\Request;**

**use App\Http\Controllers\Controller;**

**class EmployeeController extends Controller**

**{**

**public function \_\_construct()**

**{**

**$this->middleware('your-middleware-name')->only('yourAction');**

**}**

**public function yourAction(Request $request)**

**{**

**// Your controller action logic here**

**}**

**}**

1. **How to remove route caching?**

**Ans:-**

php artisan route:clear