**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 Routes

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

return view('website.home');

});

b) About us page Routes

Route::get(‘about’, function(){

return view(‘website.about’);

});

c) Contact us page Routes

Route::get(‘contact’, function(){

return view(‘website.contact’);

});

d) Gallery page Routes

Route::get(‘gallery’, function(){

return view(‘website.gallery’);

});

e) Registration page Routes

Route::get(‘registration’, function(){

return view(‘website.registration’);

});

f) Login page Routes

Route::get(‘login’, function(){

return view(‘website.login’);

});

2. How to pass Multiple Variable in route?

Route::get(‘users/{user\_id}/posts/{post\_id}’,[userController::class, ‘show’])->name(“users.posts.show”);

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

=>Route::get(‘/student/{products?}’,[student\_controller::class,’index’]);

4. Create custom auto using middleware.

=>Step1:create a middleware –>

Php artisan make:middleware customerafterlogin;

Step2:Registering middleware->

app/Http/Kernel.php

namespace App\Http;

<?php

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel

{

....

protected $routeMiddleware = [

....

'customerafterlogin' => \App\Http\Middleware\employeeAfterLogin::class,

];

}

Step3:Create Code For Middleware

<?php

namespace App\Http\Middleware;

use Closure;

class CustomMiddleware

{

public function handle($request, Closure $next)

{

// Perform some logic before the request reaches its destination

$response = $next($request);

// Perform some logic after the request has been handled

$response->header('X-Custom-Header', 'Hello from Custom Middleware');

return $response;

}

}

**Step 4 – Using Middleware With Route**

**In web.php**

Route::get('/register', [employeeController::class, 'index'])->middleware('employeeafterlogin');

Route::get('/login', [employeeController::class, 'login'])->middleware('employeeafterlogin')

(4) Generate Resource Controller for employee.

=>We can create resource controller for employee by using following command

=>php artisan make:controller EmployeeController --resource

(5)In Employee Controller‟s action Call Middleware

=> <?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class EmployeeController extends Controller

{

// Add middleware in constructor of controller

public function \_\_construct()

{

$this->middleware('auth');

}

}

?>

(6)How to remove route caching?

=> To remove route cache in Laravel use the given below command:

=>php artisan route:cache