* **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

Route::get(‘/index’, function(){

return view(‘website.index’);

});

b) About us

Route::get(‘/about’, function(){

return view(‘website.about’);

});

c) Contact us

Route::get(‘/contact’, function(){

return view(‘website.about’);

});

d) Gallery

Route::get(‘/gallery, function(){

return view(‘website.gallery’);

});

e) Registration

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

return view(‘website.registration’);

});

f) Login

Route::get(‘/login, function(){

return view(‘website.login’);

});

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

=> Firstly variable key name define in routes web.php file then we get its value in controller method by corresponding key name.

**In web.php**

Route::get(‘student-detail/{id}/{parameter}’, [StudentController::class, ’detail’]);

**In Controller**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class StudentController extends Controller

{

public function detail($id, $parameter)

{

$id = $id;

$parameter = $parameter;

}

}

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

=> **In web.php**

Route::get('/products/{brand?}', [ProductController::class, 'index']);

**In Controller**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Product;

class ProductController extends Controller

{

public function detail($brand = null)

{

if($brand) {

$products = Product::where('id', '=', $brand)->get();

} else {

$products = Product::all();

}

return view('website.product.index', compact('products'));

}

}

* **Create custom auto using middleware.**

=>

**Step 1 – Creating Middleware**

php artisan make:middleware userafterlogin

**Step 2 – Registering Middleware**

app/Http/Kernel.php

<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel

{

....

protected $routeMiddleware = [

....

'userafterlogin' => \App\Http\Middleware\userAfterLogin::class,

];

}

**Step 3 – Create Code For Middleware**

app/Http/Middleware/userAfterLogin.php

<?php

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

class userAfterLogin

{

/\*\*

\* Handle an incoming request.

\*

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

\* @param \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next

\* @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse

\*/

public function handle(Request $request, Closure $next)

{

if (session()->has('userid')) {

return redirect('/');

}

return $next($request);

}

}

**Step 4 – Using Middleware With Route**

**In web.php**

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

Route::get('/login', [userController::class, 'login'])->middleware('userafterlogin');

* **Generate Resource Controller for employee.**

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

php artisan make:controller EmployeeController --resource

* **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');

}

}

?>

* **How to remove route caching?**

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

php artisan route:cache

* **Create Custom Macro For search User**

=> **In the App\Providers\AppServiceProvider** add a following code :

use Illuminate\Database\Eloquent\Builder;

Builder::macro('search', function(string $attribute, string $searchTerm) {

return $this->where($attribute, 'LIKE', "%{$searchTerm}%");

});

Now, if you want to add a search, **to the User model and every model**, use the macro like :

// searching a single column

User::search('name', $searchTerm)->get();

// searching on multiple columns in one go

User::query()

->search('user', 'moom')

->search('email', 'gmail')

->get();