**Laravel**

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

a) Home page

b) About us

c) Contact us

d) Gallery

e) Registration

f) Login

* Route::get(‘home’, function()){

Return view(‘fronted.home\_page’);

});

* Route::get(‘about, function()){

Return view(‘fronted.aboutus);

});

* Route::get(‘contact, function()){

Return view(‘fronted.contact);

});

* Route::get(‘gallery, function()){

Return view(‘fronted.gallery);

});

* Route::get(‘Registration, function()){

Return view(‘fronted. Registration);

});

* Route::get(‘Login, function()){

Return view(‘fronted. Login);

});

1. How to pass Multiple Variable in route?

* Route::get('student-detail/{id}/{parameter}',[StudentController::class,'detail'])
* ->name('student.detail');
* Route::get('student-detail/{id?}/{parameter?}',[StudentController::class,'detail'])
* ->name('student.detail');

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

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

return $name;

});

1. Create custom auto using middleware

We can create a custom middleware using below artisan command, so open up your terminal and run the command:

php artisan make:middleware YourMiddlewareName

This command will create a file in app/Http/Middleware directory, it should look something like this

app/Http/Middleware/YourMiddlewareName.php

1. Generate Resource Controller For employee.

* php artisan make:controller employee –resource.

 This will create our resource controller with all the methods we need.

1. In Employee Controller‟s action Call Middleware

Employee middleware in $routeMiddleware in your App\Http\Kernel.php

protected $routeMiddleware = [ ‘emp\_name’ => 'App\Http\Middleware\ emp\_name Middleware', ];

Enable middleware in your controller, using middleware's key, not class name:

$this->middleware(emp\_name);

1. How to remove route caching?

* Laravel Clear Route Cache

php artisan route:cache

1. Create Custom Macro For search User

* // 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();

* use Illuminate\Database\Eloquent\Builder;

// ...

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

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

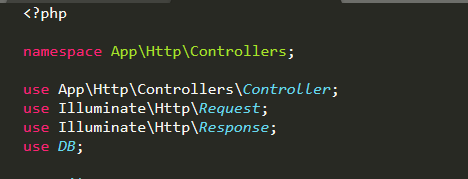
});

**Migration**

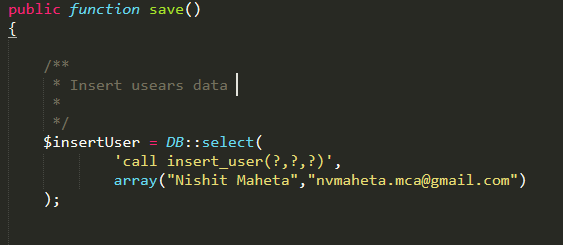
* 1. How to do config database in laravel
* Configuration
* Read/write connections
* Running queries
* Database transaction
* Accessing connections
* Query logging
  1. Call MySQLi Store Procedure from Laravel.

**Step 1:**Create create Stored Procedure. Follow the link for detail explanation.

**Step 2:**Include [laravel’s DB](https://laravel.com/docs/5.8/database) class to the Controller / Model.



**Step 2:**User **DB::select**to call Stored Procedure. As per the image, calling insert\_user Stored Procedure which has two parameters (Name, Email ).



The first parameter uses **call**keyword to execute Stored Procedure and the second parameter take data in an array format.

* 1. Apply Curd Operation through Query Builder for Employee Management
* use Illuminate\Support\Facades\DB;

// fetch all data

function view\_user()

{

$data= DB::table('users')->get();

return view('view\_user',['alldata'=>$data]);

}

// fetch where data

function view\_user()

{

$data= DB::table('users')

->where(‘uid’,10)

->get();

return view('view\_user',['alldata'=>$data]);

}

* 1. Create All Migration for Employee management
* php artisan make:migration employees

Php artisan make:model employee

php artisan migrate