**Laravel Google ReCaptcha V2 Example Tutorial**

<https://www.itsolutionstuff.com/post/laravel-google-recaptcha-v2-example-tutorialexample.html>

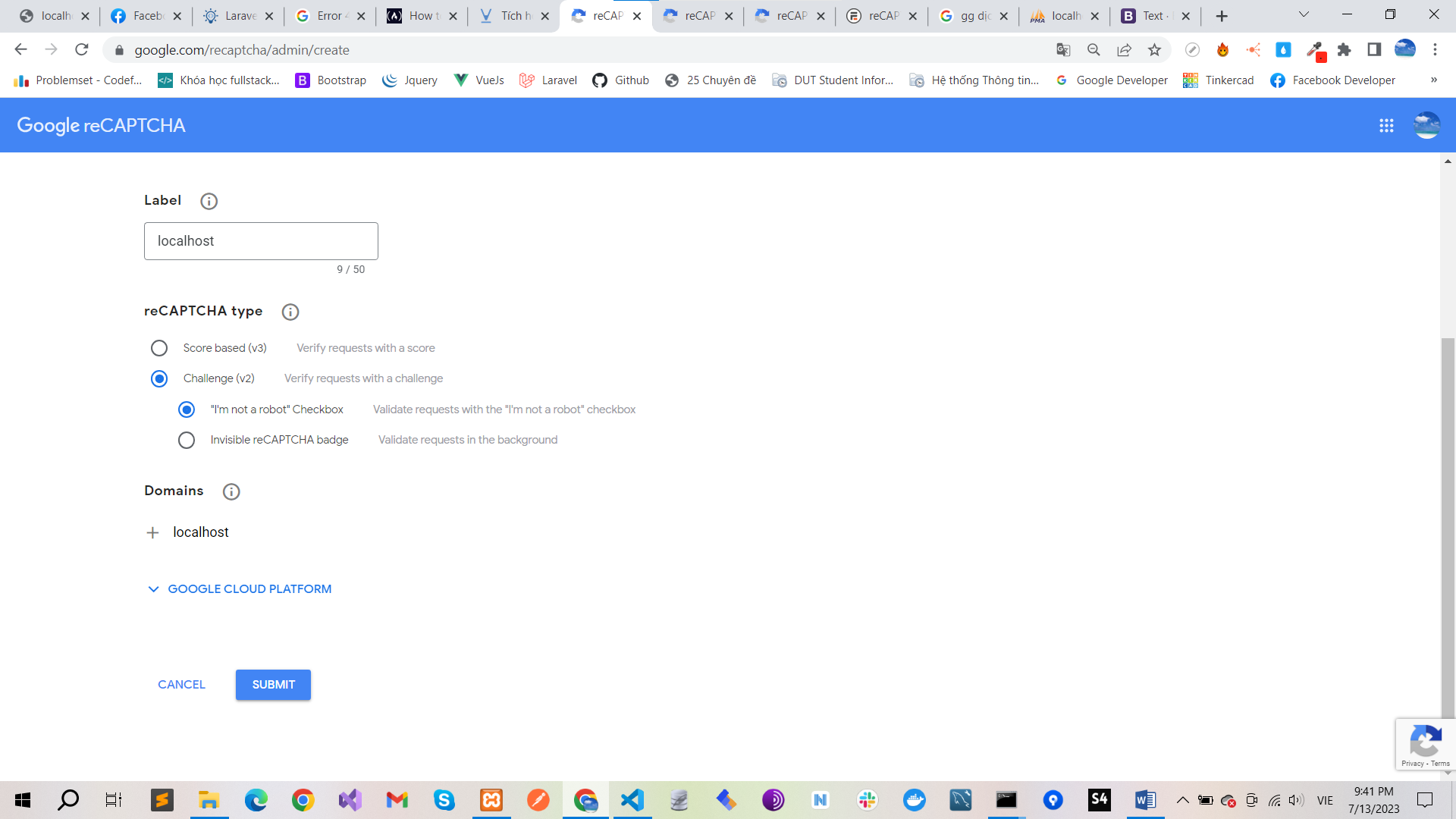
# Laravel Google ReCaptcha V2 Example Tutorial

B1 : Vào <https://www.google.com/recaptcha/admin/site/656341338#list>

Hoặc và thẳng create : <https://www.google.com/recaptcha/admin/create>

B2 : **Label : localhost** . **reCAPTCHA type chọn v2 ‘I am not robot’** . **Domains : localhost**

* Submit

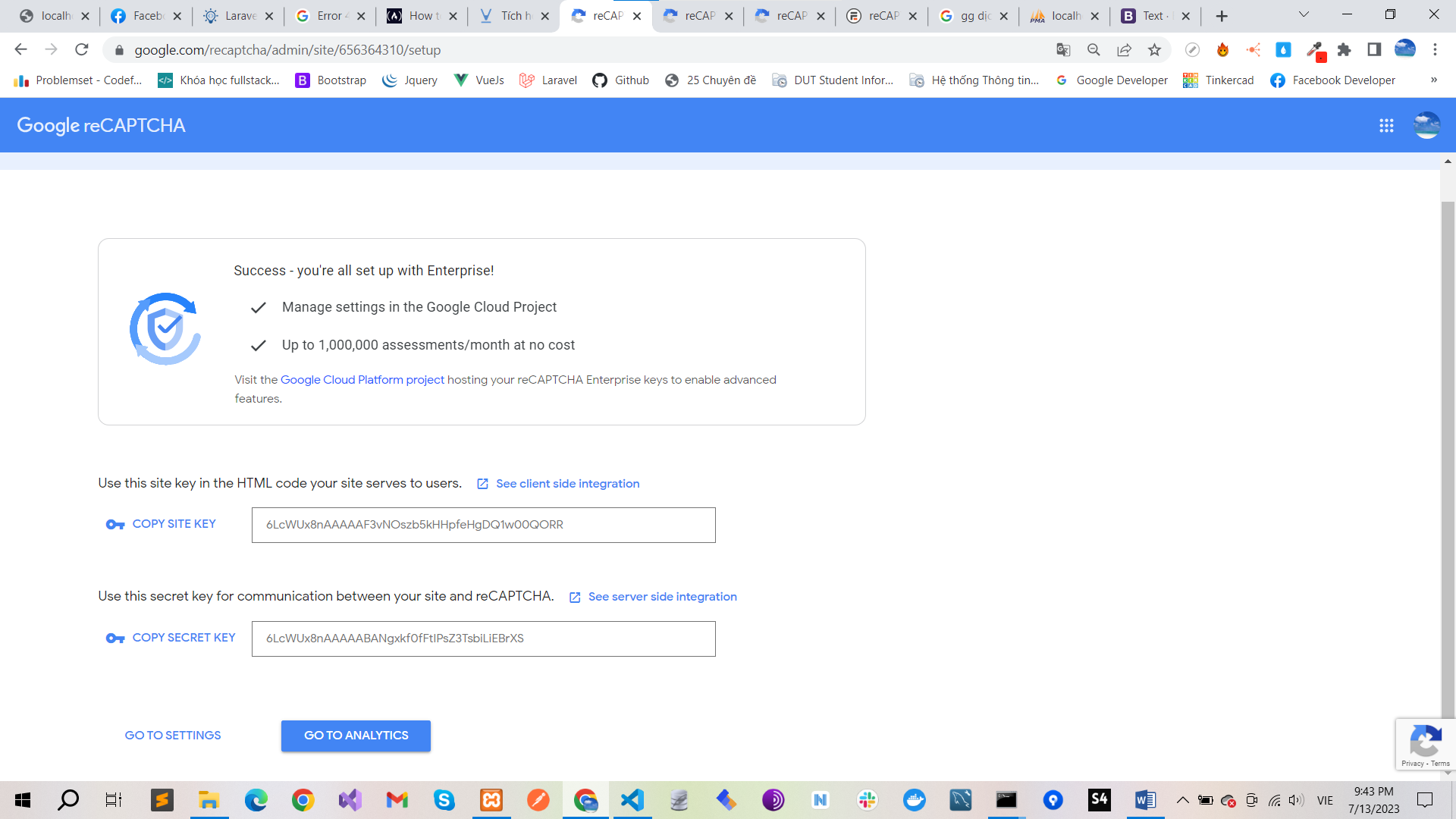


Ta sẽ có được :

SITE KEY : **6LcWUx8nAAAAAF3vNOszb5kHHpfeHgDQ1w00QORR**

SECRET KEY : **6LcWUx8nAAAAABANgxkf0fFtIPsZ3TsbiLiEBrXS**

2023 Category : Laravel



B3 : Thêm vào .env của laravel :

# Google Capcha

GOOGLE\_RECAPTCHA\_KEY=6LfVKR8nAAAAAB9oHO-h\_S4DEi6dK0gArr3E6WzE

GOOGLE\_RECAPTCHA\_SECRET=6LfVKR8nAAAAANKaCmS2frdoPhmu3sU8q3GY4C\_c

B4 : C:\Users\ADMIN\Downloads\ProLaravel\bloglaravel>php artisan make:rule ReCaptcha

Rule created successfully.

Code vào đó : C:\Users\ADMIN\Downloads\ProLaravel\bloglaravel\app\Rules\ReCaptcha.php

<?php

namespace App\Rules;

use Illuminate\Contracts\Validation\Rule;

use Illuminate\Support\Facades\Http;

class ReCaptcha implements Rule

{

    /\*\*

     \* Create a new rule instance.

     \*

     \* @return void

     \*/

    public function \_\_construct()

    {

    }

    /\*\*

     \* Determine if the validation rule passes.

     \*

     \* @param  string  $attribute

     \* @param  mixed  $value

     \* @return bool

     \*/

    public function passes($attribute, $value)

    {

        $response = Http::get("https://www.google.com/recaptcha/api/siteverify",[

            'secret' => env('GOOGLE\_RECAPTCHA\_SECRET'),

            'response' => $value

        ]);

        return $response->json()["success"];

    }

    /\*\*

     \* Get the validation error message.

     \*

     \* @return string

     \*/

    public function message()

    {

        return 'The google recaptcha is required.';

    }

}

B5 : Thêm link js vào head của master.blade.php

    <script src='https://www.google.com/recaptcha/api.js'></script>

B6 : Thêm vào form submit nào đó mà mình muốn có recaptcha google . Ví dụ là Login

                      <div class="ml-3 g-recaptcha" data-sitekey="{{ env('GOOGLE\_RECAPTCHA\_KEY') }}"></div>

                      @if ($errors->has('g-recaptcha-response'))

                          <span class="text-danger">{{ $errors->first('g-recaptcha-response') }}</span>

                      @endif

B7 : Login của BlogAuthController.php : Thêm vào validate là được

            $request->validate([

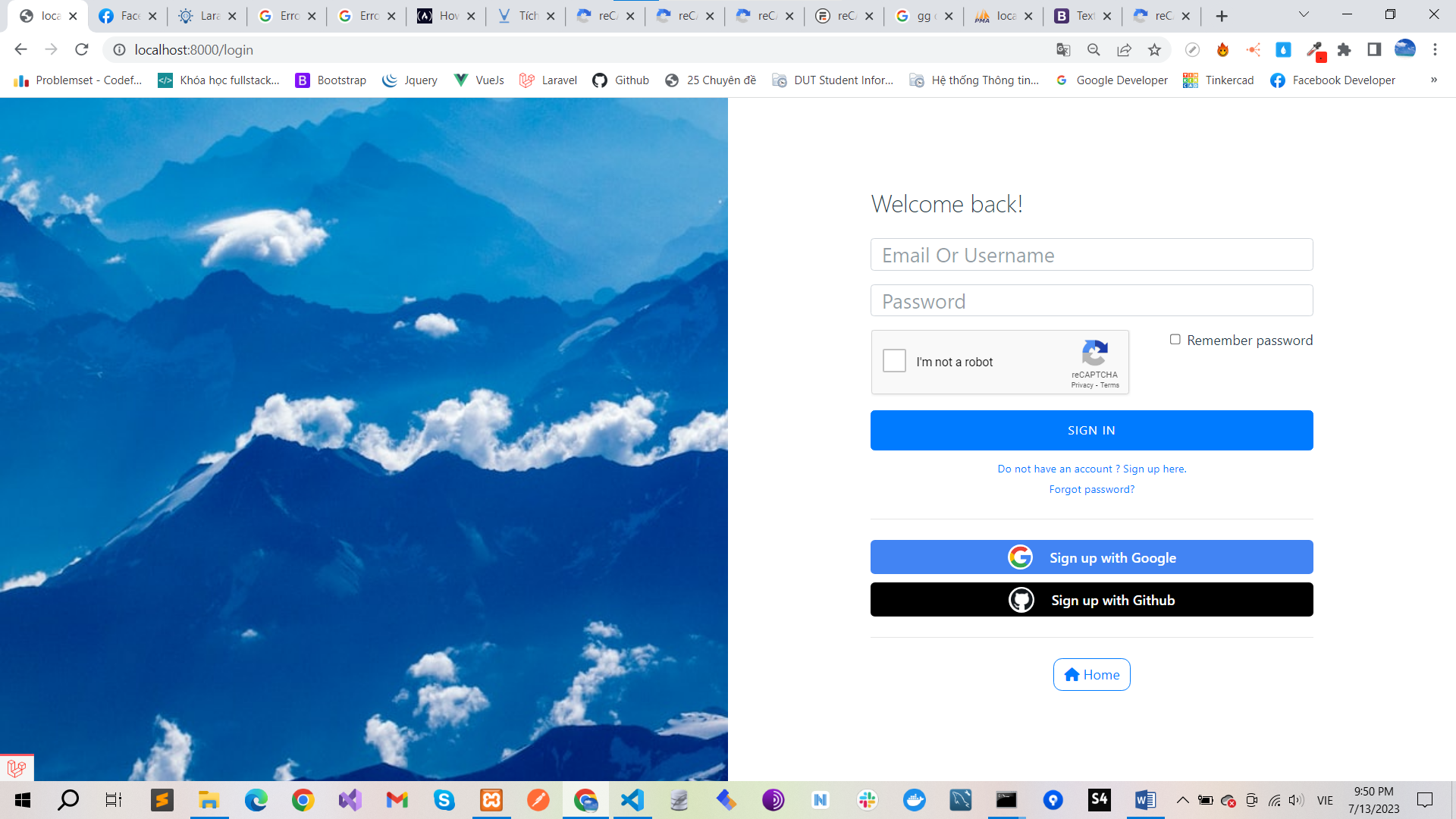
                'email' => 'required|email',

                'password' => 'required',

                'g-recaptcha-response' => ['required', new ReCaptcha]

            ]);

Kết quả :



Hello,

Today, laravel google recaptcha v2 example is our main topic. if you have question about how to use google recaptcha v2 in laravel then I will give simple example with solution. if you want to see example of how to add google recaptcha v2 in laravel then you are a right place. I’m going to show you about integrate google recaptcha v2 in laravel.

You can use this example with laravel 6, laravel 7, laravel 8, laravel 9 and laravel 10 version.

As we know, When we created contact us form or feedback form or any form that access by publicly Then we have so many submission from spam. They submit those form using curl request or something. So we can not get real user from those forms. Here, i we will use google recaptcha v2 to prevent spam submission. we will create contact us form and use google recaptcha v2 and prevent them. So that way you have only real users and real feedback you will get.

Let's see the below steps and make it done.

**Step 1: Install Laravel**

This step is not required; however, if you have not created the laravel app, then you may go ahead and execute the below command:

composer create-project laravel/laravel example-app

**Step 2: Add Google API Key**

In this step we need to set google site key and secret key. If you don't have site key and secret key then you can create from here. First click on this link : [Recaptcha Admin](https://www.google.com/recaptcha/admin/site/536053053)

Ok, after sucessfully register you can get site key and secret key from like bellow preview.

Now open .env file and add this two variable

**.env**

GOOGLE\_RECAPTCHA\_KEY=6Lc9hfMfAAAAAMNLm5\_2P4jbzVSNWxaF0vgk

GOOGLE\_RECAPTCHA\_SECRET=6Lc9hfMfAAAAAAdBOKGoz88JVgc5LwMO9

Read Also: [How to Remove Column from Table in Laravel Migration?](https://www.itsolutionstuff.com/post/how-to-remove-column-from-table-in-laravel-migrationexample.html)

**Step 3: Create Routes**

In this step, we will create two routes GET and POST. so let's add it.

**routes/web.php**

<?php

**use** Illuminate\Support\Facades\Route;

**use** App\Http\Controllers\ContactController;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::**get**('contact-us', [ContactController::**class**, 'index']);

Route::post('contact-us', [ContactController::**class**, 'store'])->name('contact.us.store');

**Step 4: Create Validation Rule**

In this step, we will create new "ReCaptcha" validation rule that will check user is real or not using google recaptcha v2 API. so let's run below command and update rule validation file.

php artisan make:rule ReCaptcha

**app/Rules/ReCaptcha.php**

<?php

**namespace** App\Rules;

**use** Illuminate\Contracts\Validation\Rule;

**use** Illuminate\Support\Facades\Http;

**class** ReCaptcha **implements** Rule

{

/\*\*

\* Create a new rule instance.

\*

\* @return void

\*/

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

{

}

/\*\*

\* Determine if the validation rule passes.

\*

\* @param string $attribute

\* @param mixed $value

\* @return bool

\*/

**public** **function** passes($attribute, $value)

{

$response = Http::**get**("https://www.google.com/recaptcha/api/siteverify",[

'secret' => env('GOOGLE\_RECAPTCHA\_SECRET'),

'response' => $value

]);

**return** $response->json()["success"];

}

/\*\*

\* Get the validation error message.

\*

\* @return string

\*/

**public** **function** message()

{

**return** 'The google recaptcha is required.';

}

}

**Step 5: Create Controller**

In this step, we have to create new controller as ContactController and we have also need to add two methods index() and store() on that controller. We will use recaptcha validation on store method. so let's update follow code:

**app/Http/Controllers/ContactController.php**

<?php

**namespace** App\Http\Controllers;

**use** Illuminate\Http\Request;

**use** App\Rules\ReCaptcha;

**class** ContactController **extends** Controller

{

/\*\*

\* Write code on Method

\*

\* @return response()

\*/

**public** **function** index()

{

**return** view('contactForm');

}

/\*\*

\* Write code on Method

\*

\* @return response()

\*/

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

{

$request->validate([

'name' => 'required',

'email' => 'required|email',

'phone' => 'required|digits:10|numeric',

'subject' => 'required',

'message' => 'required',

'g-recaptcha-response' => ['required', **new** ReCaptcha]

]);

$input = $request->all();

/\*------------------------------------------

--------------------------------------------

Write Code for Store into Database

--------------------------------------------

--------------------------------------------\*/

dd($input);

**return** redirect()->back()->**with**(['success' => 'Contact Form Submit Successfully']);

}

}

**Step 6: Create View File**

In Last step, let's create contactForm.blade.php(resources/views/contactForm.blade.php) for create form with google recaptcha v2 and put following code:

**resources/views/contactForm.blade.php**

<!DOCTYPE html>

**<html>**

**<head>**

**<title>**Laravel Google ReCaptcha V2 Example - ItSolutionStuff.com**</title>**

**<meta** **charset**="utf-8"**>**

**<meta** **http-equiv**="X-UA-Compatible" **content**="IE=edge"**>**

**<meta** **name**="viewport" **content**="width=device-width, initial-scale=1"**>**

**<link** **rel**="stylesheet" **href**="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.5.2/css/bootstrap.min.css" **/>**

**<script** **src**='https://www.google.com/recaptcha/api.js'**></script>**

**</head>**

**<body>**

**<div** **class**="container"**>**

**<div** **class**="row mt-5 mb-5"**>**

**<div** **class**="col-10 offset-1 mt-5"**>**

**<div** **class**="card"**>**

**<div** **class**="card-header bg-primary"**>**

**<h3** **class**="text-white"**>**Laravel Google ReCaptcha V2 Example - ItSolutionStuff.com**</h3>**

**</div>**

**<div** **class**="card-body"**>**

**<form** **method**="POST" **action**="{{ route('contact.us.store') }}"**>**

{{ csrf\_field() }}

**<div** **class**="row"**>**

**<div** **class**="col-md-6"**>**

**<div** **class**="form-group"**>**

**<strong>**Name:**</strong>**

**<input** **type**="text" **name**="name" **class**="form-control" **placeholder**="Name" **value**="{{ old('name') }}"**>**

@if ($errors->has('name'))

**<span** **class**="text-danger"**>**{{ $errors->first('name') }}**</span>**

@endif

**</div>**

**</div>**

**<div** **class**="col-md-6"**>**

**<div** **class**="form-group"**>**

**<strong>**Email:**</strong>**

**<input** **type**="text" **name**="email" **class**="form-control" **placeholder**="Email" **value**="{{ old('email') }}"**>**

@if ($errors->has('email'))

**<span** **class**="text-danger"**>**{{ $errors->first('email') }}**</span>**

@endif

**</div>**

**</div>**

**</div>**

**<div** **class**="row"**>**

**<div** **class**="col-md-6"**>**

**<div** **class**="form-group"**>**

**<strong>**Phone:**</strong>**

**<input** **type**="text" **name**="phone" **class**="form-control" **placeholder**="Phone" **value**="{{ old('phone') }}"**>**

@if ($errors->has('phone'))

**<span** **class**="text-danger"**>**{{ $errors->first('phone') }}**</span>**

@endif

**</div>**

**</div>**

**<div** **class**="col-md-6"**>**

**<div** **class**="form-group"**>**

**<strong>**Subject:**</strong>**

**<input** **type**="text" **name**="subject" **class**="form-control" **placeholder**="Subject" **value**="{{ old('subject') }}"**>**

@if ($errors->has('subject'))

**<span** **class**="text-danger"**>**{{ $errors->first('subject') }}**</span>**

@endif

**</div>**

**</div>**

**</div>**

**<div** **class**="row"**>**

**<div** **class**="col-md-12"**>**

**<div** **class**="form-group"**>**

**<strong>**Message:**</strong>**

**<textarea** **name**="message" **rows**="3" **class**="form-control"**>**{{ old('message') }}**</textarea>**

@if ($errors->has('message'))

**<span** **class**="text-danger"**>**{{ $errors->first('message') }}**</span>**

@endif

**</div>**

**</div>**

**</div>**

**<div** **class**="row"**>**

**<div** **class**="col-md-12"**>**

**<div** **class**="form-group"**>**

**<strong>**ReCaptcha:**</strong>**

**<div** **class**="g-recaptcha" **data-sitekey**="{{ env('GOOGLE\_RECAPTCHA\_KEY') }}"**></div>**

@if ($errors->has('g-recaptcha-response'))

**<span** **class**="text-danger"**>**{{ $errors->first('g-recaptcha-response') }}**</span>**

@endif

**</div>**

**</div>**

**</div>**

**<div** **class**="form-group text-center"**>**

**<button** **class**="btn btn-success btn-submit"**>**Submit**</button>**

**</div>**

**</form>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**</body>**

**</html>**

**Run Laravel App:**

All the required steps have been done, now you have to type the given below command and hit enter to run the Laravel app:

php artisan serve

Now, Go to your web browser, type the given URL and view the app output:

Read Also: [Laravel Send an Email on Error Exceptions Tutorial](https://www.itsolutionstuff.com/post/laravel-send-an-email-on-error-exceptions-tutorialexample.html)

http://localhost:8000/contact-us

**Output: You Email Look Like This**

I hope it can help you...