Links 1. :- [How To Send Email To New Users - Vegibit](https://vegibit.com/how-to-send-email-to-new-users/)

[Laravel UI Login Register Email Verification in Laravel 8 (onlinewebtutorblog.com)](https://onlinewebtutorblog.com/laravel-ui-login-register-email-verification-in-laravel-8/)

1. Php artisan make:mail CloudEmailProduct

2.

Once you have your Mailtrap.io account and demo inbox in place, you can populate the .env file of your Laravel installation similar to what you see here:

(do setting according to mailtrap account)

MAIL\_DRIVER=smtp

MAIL\_HOST=smtp.mailtrap.io

MAIL\_PORT=2525

MAIL\_USERNAME=your\_user\_name

MAIL\_PASSWORD=your\_password

MAIL\_ENCRYPTION=tls

MAIL\_FROM\_ADDRESS=hello@welikegames.dev

MAIL\_FROM\_NAME='We Like Games Team'

3.

4. in web.php

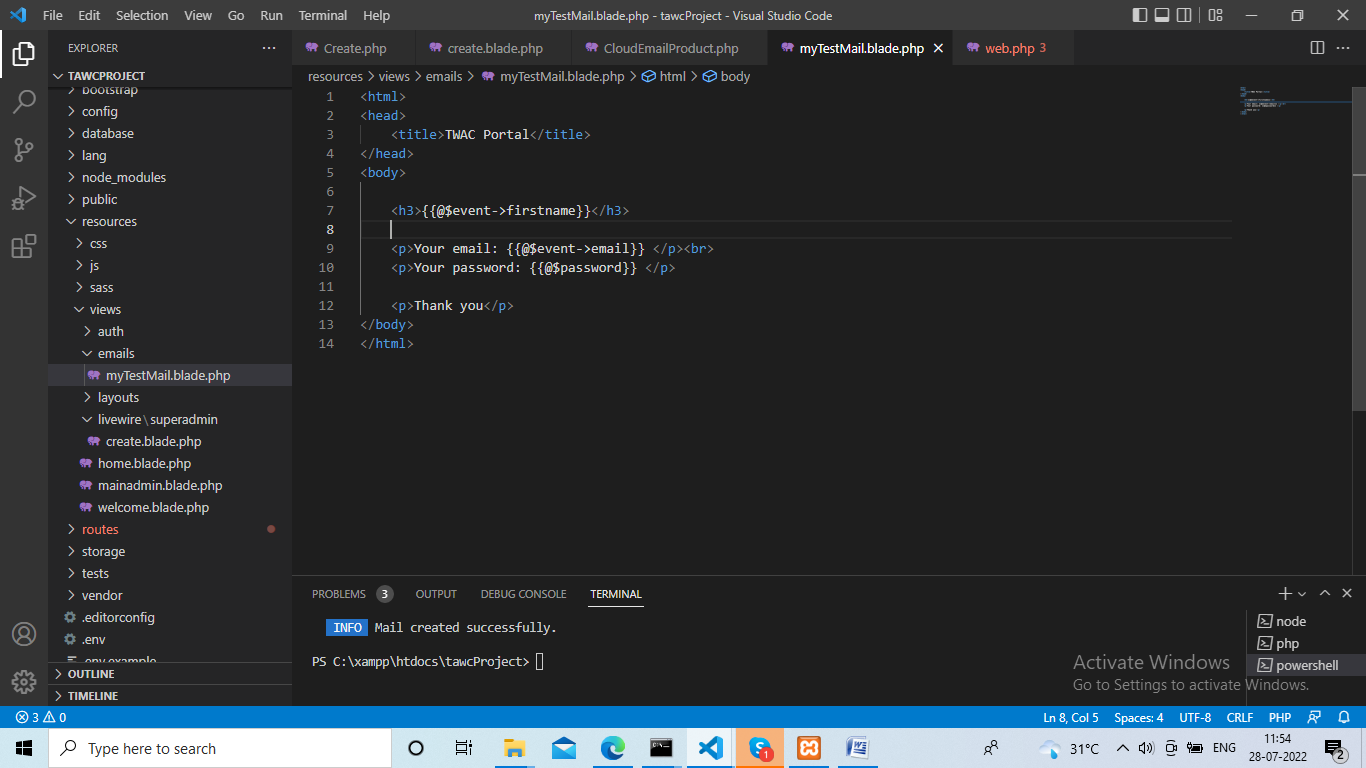
use App\Mail\CloudEmailProduct;

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

    Mail::to(['aman050490@gmail.com'])->send(new CloudEmailProduct);

});

1. In resource 🡪views🡪 emails🡪 mytestmail.blade.php (create emails folder and creater blade file)



1. Controller or component file where you save data in database
2. public function store()
3. {
4. $this->validate();
5. $pass = rand ( 10000000000 , 99999999 );
6. // dd($pass);
7. $event = new User;
8. $event->firstname = $this->firstname;
10. $event->lastname = $this->lastname;
11. $event->organization = $this->organization;
12. $event->email= $this->email;
13. $event->password = Hash::make( $pass);
14. $event->save();
15. // dd($event->password);

18. Mail::to($this->email)->send(new CloudEmailProduct($event,$pass));
19. session()->flash('message','<b>Hi there!</b> Thanks for signup');
20. session()->flash('type','success');
22. $this->resetInput();
24. }

7. app🡪 mail🡪 CloudEmailProduct (this file is created during install of mail)

7. <?php

namespace App\Mail;

use AppUser;

use Illuminate\Bus\Queueable;

use Illuminate\Contracts\Queue\ShouldQueue;

use Illuminate\Mail\Mailable;

use Illuminate\Queue\SerializesModels;

class CloudEmailProduct extends Mailable

{

    use Queueable, SerializesModels;

    public $event;

public $password;

    /\*\*

     \* Create a new message instance.

     \*

     \* @return void

     \*/

    public function \_\_construct( $event, $password)

    {

        $this->event= $event;

        $this->password= $password;

        //

    }

    /\*\*

     \* Build the message.

     \*

     \* @return $this

     \*/

    public function build()

    {

        return $this->subject('Ypur account has been created!!')

        ->view('emails.myTestMail');

        // return $this->view('emails.myTestMail');

    }

}