# **Second Task :**Authentic Blog Project with Laravel

**Introduction :**This is my  authentic blog project, here anyone can sign up then can login. When authorized person login whose can create blog post with Laravel

…………………….
……………………………….

## Tools:

- □ **Front-end:** HTML, CSS, Bootstrap
- □ **Back-end:** PHP, MySQL

### Design :

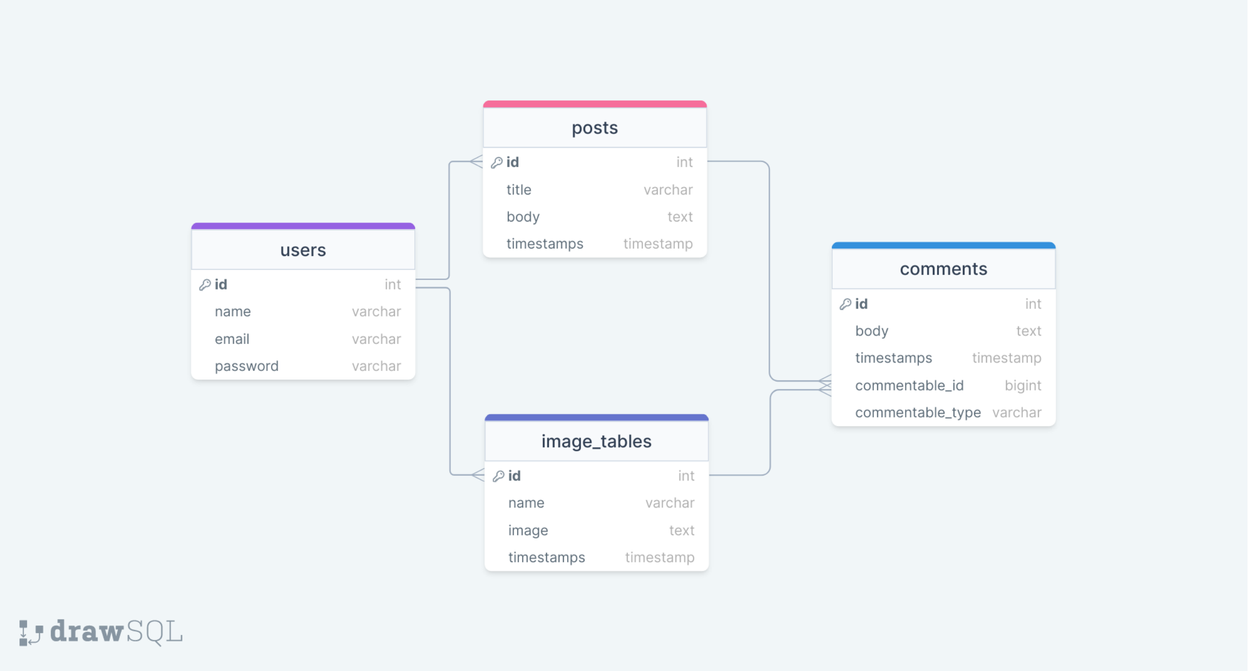
1. Front-end Design:
    - Home page
    - Login page
    - Sign Up page

2. Back-end:

    When any person can signup and signin
    Add a post with title
    Post and title show Home Page
    And post below the comment section
    Anyone can comment any post
    Same time show comment under the post

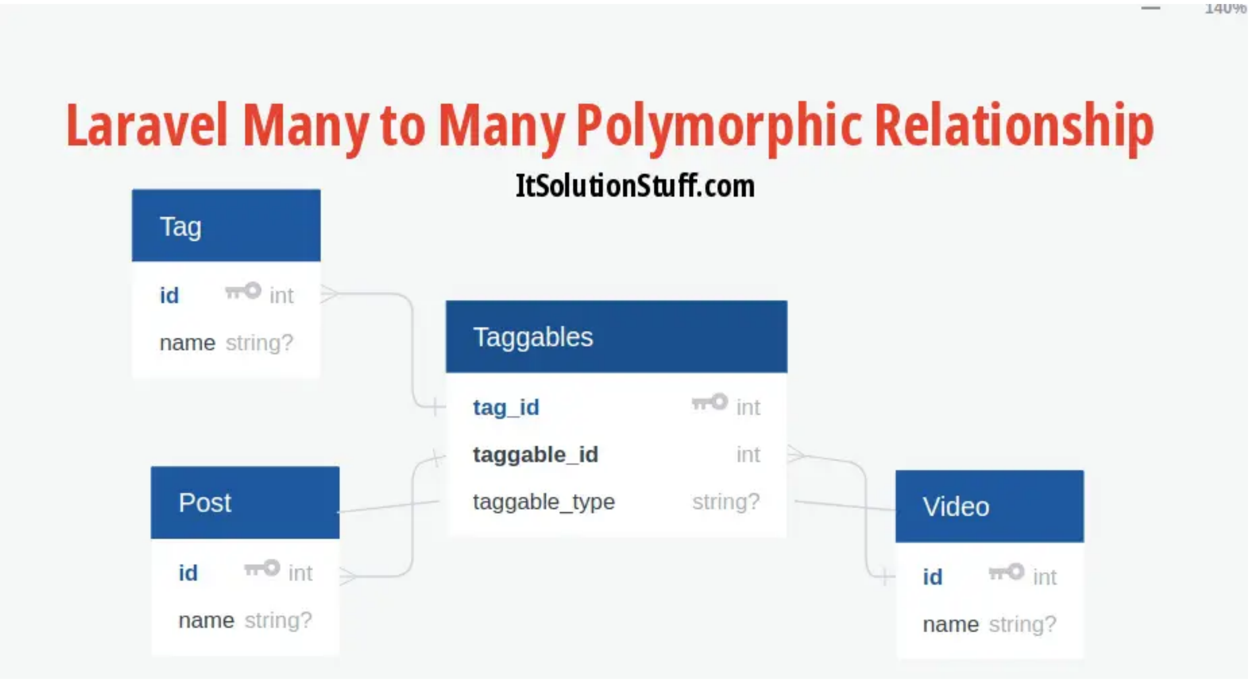
**—-------------------------------------------------------------------------------------------------------**

## Drow SQL



—-----------------------------------------------------------------------------------------------------------------

## **New Learn :** polymorphic relationships in Laravel



### This is the comment table Schema:

```
Schema::create('comments', function (Blueprint $table) {
            $table->increments('id');
            $table->morphs('commentable');
            $table->text('comment')->nullable();
            $table->timestamps();
        });
```

### This is Image Model comment Function

```
public function comments()
    {
        return $this->morphMany(Comment::class,'commentable');
                             //App\Models\ImageModel
    }
```

### This is Post Model comment Function

```
public function comments()
    {
        return $this->morphMany(Comment::class,'commentable');
                             //App\Models\PostModel
    }
```

### How to get All post with comment By HomeController

```
function blogIndex(){
        $post = PostModel::with('comments')->orderBy('id',
'desc')->take(4)->get();
        return view('website.pages.blog_page',compact('post') );
    }
```

### How to get All Image with comment By HomeController

```
    function imageIndex(){
        $image = ImageModel::with('comments')->orderBy('id',
'desc')->take(6)->get();

        return view('website.pages.image_page',compact('image'));
     }
```

**Link:** https://blog.logrocket.com/polymorphic-relationships-laravel/

**—-----------------------------------------------------------------------------------------------------------**

## Toastr JS Alert Notification System for PHP/Laravel

### 1. Install

```
composer require plusemon/notify
```

### 2. Add resources

```
<!DOCTYPE html>
<html>

<head>
```

```
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Laravel Notify</title>
 <link rel="stylesheet"
href="https://cdn.bootcss.com/toastr.js/latest/css/toastr.min.css" />
</head>

<body>

<!-- Main Contents -->
<h1>Laravel Toastr Notify Notification</h1>

<!-- jQuery CDN -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script
>

<!-- Toastr script CDN -->
<script
src="https://cdn.bootcss.com/toastr.js/latest/js/toastr.min.js"></script>

<!-- Toastr Scripts render -->
{!! Notify::message() !!}

</body>

</html>
```

## 3. Usage

```
<!--  Example   -->
Notify::success('You have successfully added Notify alert system');
notify()->info('Info message with helper function');

<!--  Using Facade    -->
Notify::info('message', 'title', ['options']);
Notify::success('message', 'title', ['options']);
Notify::warning('message', 'title', ['options']);
Notify::error('message', 'title', ['options']);

<!--   Clear   -->
Notify::clear();
```

return Notify::info('Info redirect back with message by helper function')->back(); return notify()->info('Info redirect with message by helper function')->redirect('/');

### You have successfully installed notify system. 😃

## Configuration

Add the service provider to `config/app.php`

```
Plusemon\Notify\NotifyServiceProvider::class,
```

Optionally include the Facade in config/app.php if you'd like.

```
'Notify'  => Plusemon\Notify\Facades\Notify::class,
```

## Options

Publish the config file

```
php artisan vendor:publish --provider=Plusemon\Notify\NotifyServiceProvider
```

**Link**:https://github.com/plusemon/notify

**—-------------------------------------------------------------------------------------------------------------**

**For Authentication 🙂**

For authentication use Starter Kits Laravel Breeze
Create profile page
Create  Update user info page
Reset a new password with given current password
Successfully updated user info with return message

—------------------------------------------------------------------------------------------------------------

## Email verification in laravel - how to verify user after login

1. **Step:**

   Go to user Model then implements MustVerifyEmail add to class User extends Authenticatable. Then use

   use Illuminate\Contracts\Auth\MustVerifyEmail;

```
        use Illuminate\Contracts\Auth\MustVerifyEmail;
```

```
        class User extends Authenticatable implements
MustVerifyEmail
        {
```

2. **Step**:

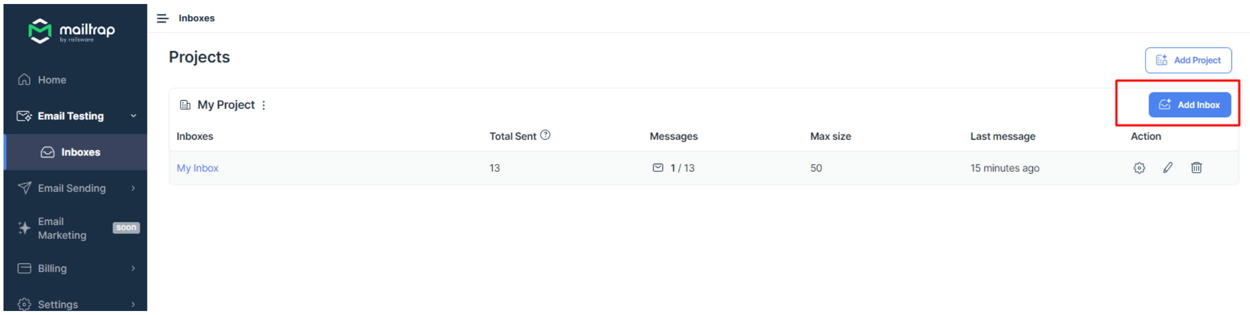
   Go to routes folder and web.php dashboard name middleware function auth bestie add 'verified'

```
Route::get('/dashboard', function () {
     return view('admin.home.index');

})->middleware(['auth', 'verified'])->name('dashboard');
```

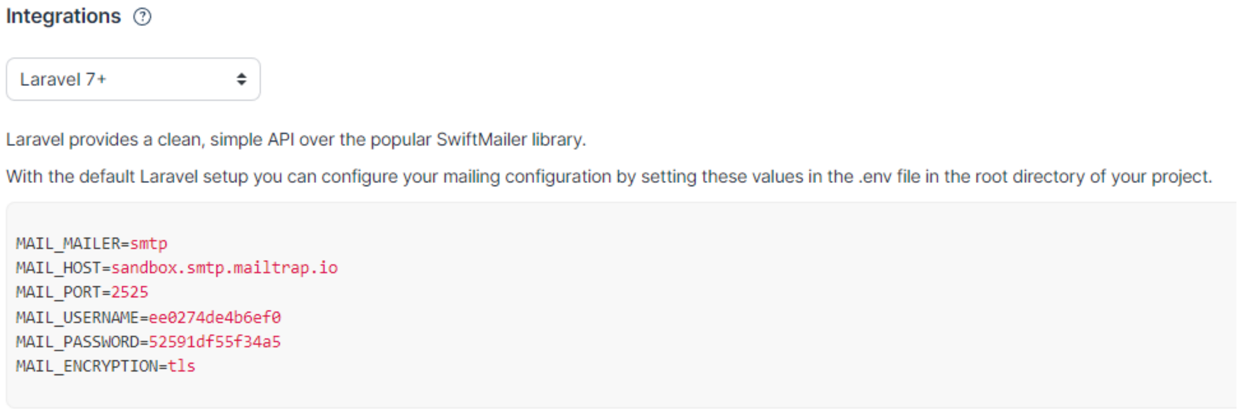
3. **Step**:

   Go to [https://mailtrap.io/](https://mailtrap.io/) url then signup. Click Add index button.



| Inboxes | Total Sent | Messages | Max size | Last message | Action |
|---|---|---|---|---|---|
| My Inbox | 13 | 1 / 13 | 50 | 15 minutes ago | |

Then go to integration cURL
Select Laravel 7+

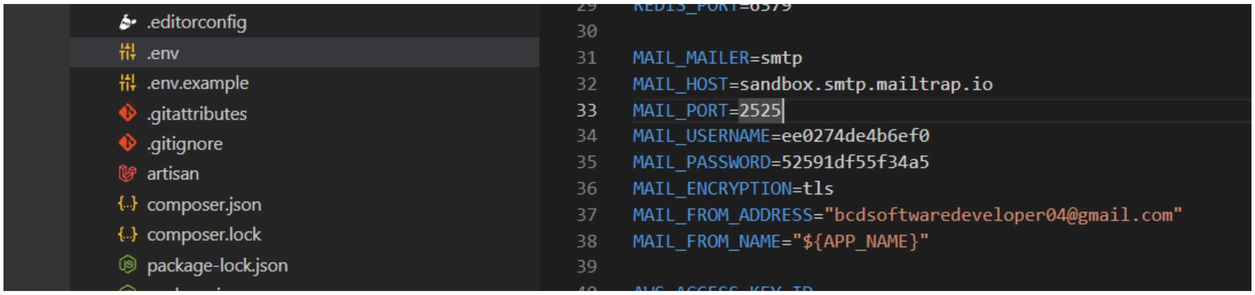


Laravel provides a clean, simple API over the popular SwiftMailer library.

With the default Laravel setup you can configure your mailing configuration by setting these values in the .env file in the root directory of your project.

```
MAIL_MAILER=smtp
MAIL_HOST=sandbox.smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=ee0274de4b6ef0
MAIL_PASSWORD=52591df55f34a5
MAIL_ENCRYPTION=tls
```

Copy in this code and past .evn file here..



```
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=sandbox.smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=ee0274de4b6ef0
MAIL_PASSWORD=52591df55f34a5
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="bcdsoftwaredeveloper04@gmail.com"
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID
```

And also add `MAIL_FROM_ADDRESS="Enter your Mailtrap sign-up gmail"`

.
.
.
Now ready to email verification.

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

# Forget Password

First of all login.blade file add ankor tag this name Forget Password with route. Then i can custom forget-password.blade file. ……………………………………………… ……………………………………………………………