***Studying***

***Laravel***

* to create laravel project ( composer create-project laravel/laravel project\_name )
* to run laravel project ( php artisan serve )
* to make controller ( php artisan make:controller UserController )
* to make resource controller ( php artisan make:controller UserController --resource )

/\*\*\*\*\*\*\*\*\*\*\*

what is different between controller resource and controller normalize?

- controller resource make default crud function ( create - edit - update - delete )

\*\*\*\*\*\*\*\*\*\*\*/

* to make invoice controller ( php artisan make:controller UserProfileController -i ) => to make single action controller or single action method
* to connect mysql database by laravel and make tables between them :-

1 - go to .env and change DB\_DATABASE from "laravel" to "database name"

2 - open terminal and run this ( php artisan migrate )

* If you would like to see which migrations have run thus far, you may use this (php artisan migrate:status)
* to create table in laravel (php artisan make:migration create\_name\_table\_table)
* to added table from laravel to mysql database write this code in terminal (php artisan migrate)
* to roll back ( the latest ) migration operation use this code (php artisan migrate:rollback)
* to roll back a ( limited number ) of migration operation use this code (php artisan migrate:rollback --step=5)
* to roll back but i don't know the step to use rollback in this case should use reset (php artisan migrate:reset)
* to make rollback and enter data in database but if use this will delete all data from database (php artisan migrate:refresh)
* to drop all tables and migrated another again in database use this code (php artisan migrate:fresh)
* to add column in database (php artisan make:migration add\_column-name\_to\_table-name)
* to edit column in database (php artisan make:migration edit\_column-name\_to\_new-column-name\_table-name)

///////////////////////// Foreign key /////////////////////////

1-

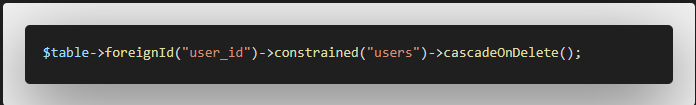
$table->unsignedBigInteger("column\_name");

$table->foreign("column\_name")->references("column inside another table")->on("table\_name");



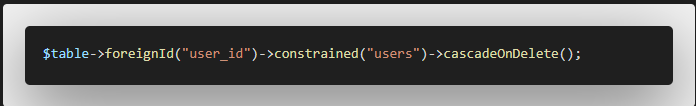
2-

$table->foreignId("user\_id")->constrained("table\_name");



* if user added any posts and you delete this user from database if you want deleted the user's posts

$table->foreignId("user\_id")->constrained("table\_name")->cascadeOnDelete();



* to insert data in database by laravel

DB::table("table\_name")->insert([

"title"=>$my\_request->title,

"body"=>$my\_request->body

]);



* to select or to show data from database by laravel

$posts = DB::table("posts")->get();

return view("posts.index", compact("posts"));



<tbody>

@foreach($posts as $post)

<tr>

<th scope="row">{{$post->id}}</th>

<td>{{$post->title}}</td>

<td>{{$post->body}}</td>

</tr>

@endforeach

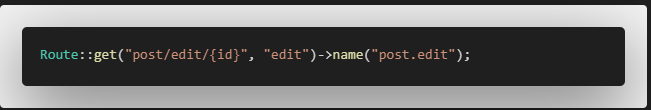
</tbody>



* to get data from database by id

1-

Route::get("post/edit/{id}", [PostController::class, "edit"])->name("post.edit");



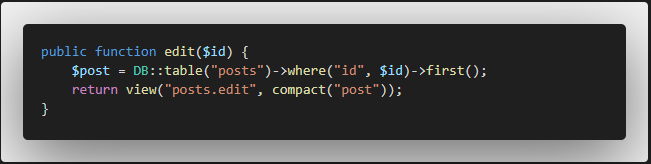
2-

public function edit($id) {

$post = DB::table("posts")->where("id", $id)->first();

return view("posts.edit", compact("post"));

}



3-

<a class="btn btn-secondary" href="{{route('post.edit', $post->id)}}" role="button">Edit</a>

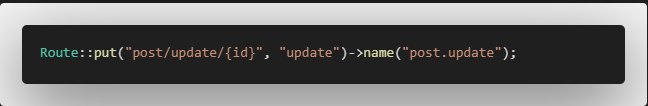


* to update the data from database by laravel

1-

// this is to make update route by put, but if you want use this case you should run this code inside form (@method("put"));

Route::put("post/update/{id}", "update")->name("post.update");



2-

public function update(Request $my\_request, $id) {

$post = DB::table("posts")->where("id", $id)->update([

"title"=>$my\_request->title,

"body"=>$my\_request->body

]);

return redirect()->route("posts");

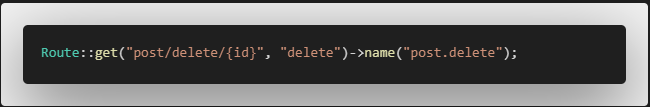
}



* to delete the data from database by laravel

1-

Route::get("post/delete/{id}", "delete")->name("post.delete");



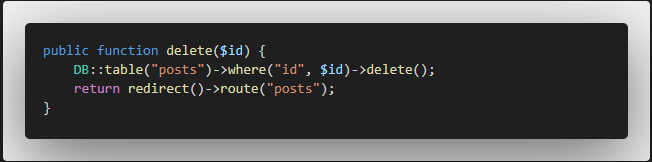
2-

public function delete($id) {

DB::table("posts")->where("id", $id)->delete();

return redirect()->route("posts");

}



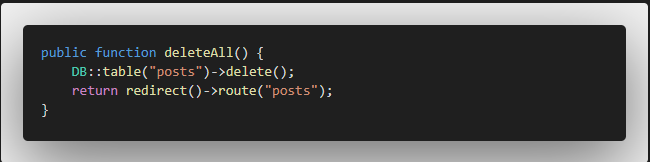
* to delete ( all ) data from database by laravel

public function deleteAll() {

DB::table("posts")->delete();

return redirect()->route("posts");

}



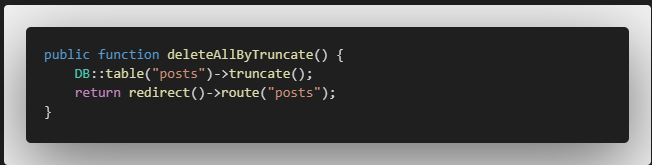
* to delete ( all ) data from database by laravel but in this case “the id will counter from 1”

public function deleteAllTruncate() {

DB::table("posts")->truncate();

return redirect()->route("posts");

}



* to make model (php artisan make:model model\_name)
* to make model and create table in database at the same time (php artisan make:model model\_name --migration)
* to make model and create table and make controller at the same time (php artisan make:model model\_name --migration --controller)
* to make model and create table and make controller and make this controller is resource at the same time (php artisan make:model model\_name --migration –controller --resource)
* if you change table name in database and want use it in model go to model



* If you want don’t used timestamps from table in database must make it

1. Go to table file and delete ( $table->timestamps(); ) from up function
2. Go to model write this inside model ( public $timestamps = false)

* If you want to stop autoincrement for id
  + Go to model file ( public $incrementing = false )
* If you want change the primary key for this table

1. Go to model write this inside model ( protected $primary\_key = “ name” )

* To insert data in database by Laravel for model

1- open web.php

1. Route::post("posts/login", "store")->name("posts.store");

2- open PostController.php

    public function store(Request $request)

    {

// to make new post in database

        $post = new Post(); // to make object from model

        $post->title = $request->title;

        $post->body = $request->body;

        $post->save(); // to save post in database

        if($post->save()):

            return view("posts.create");

        endif;

    }

3- open create.blade.php

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

    @csrf

    <input type="text" name="title" placeholder="Enter Title">

    <input type="text" name="body" placeholder="Enter Body">

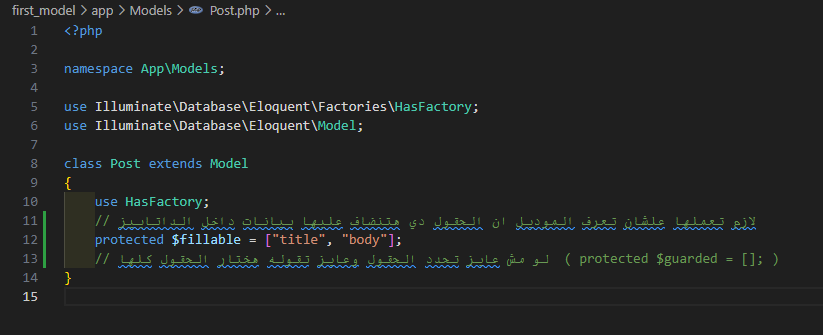
    <input type="submit" value="Login">

</form>

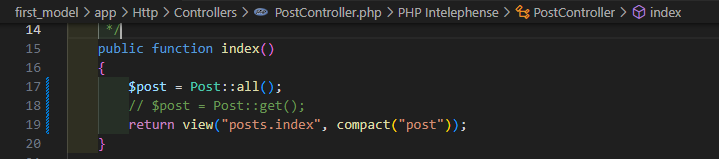
* To insert data in database by Laravel for model another way:-

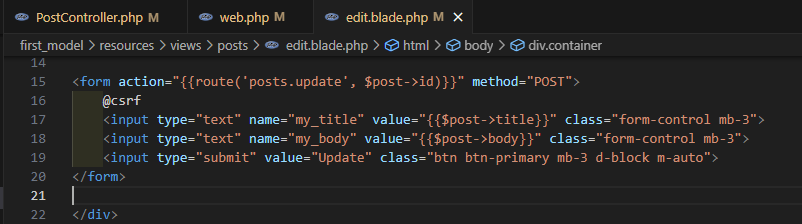
1. open PostController.php

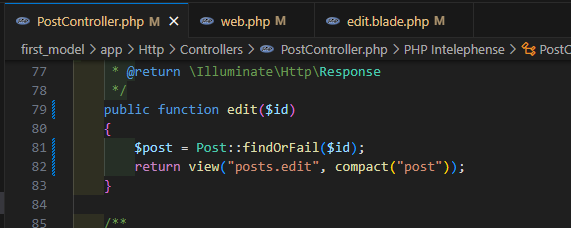
( this step is very important must make it )

1. open model post.php

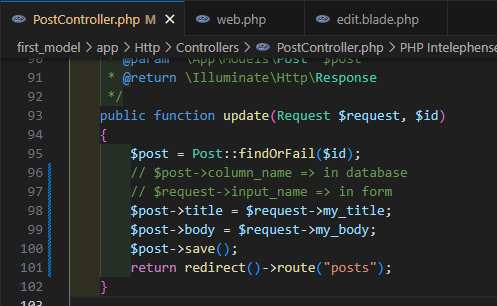
* To show all data from database :-
  + Open PostController File



* To make edit in data :-
  + Open edit file desing
  + Open Post Controller File
    - findOrFail() => is function to check if id is correct or not || is function to check if id is found in database or not and to get data by id



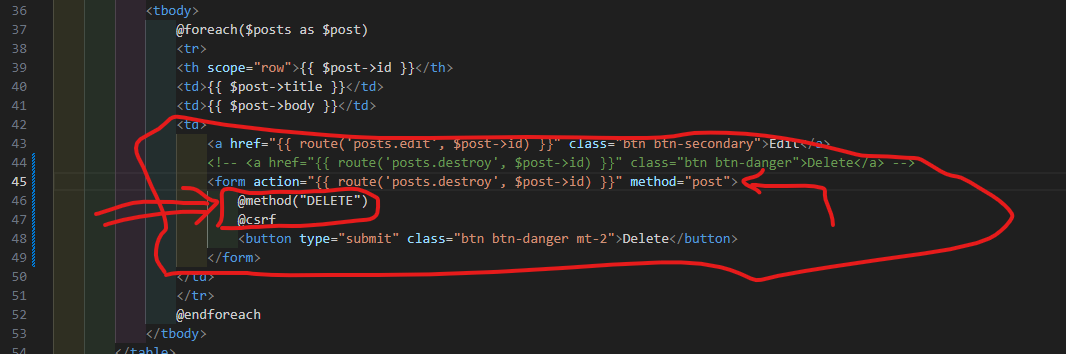
* To make update in data :-
* Route::post("posts/update/{id}", "update")->name("posts.update");
  + Open PostController:-



* To update data in database another way :-



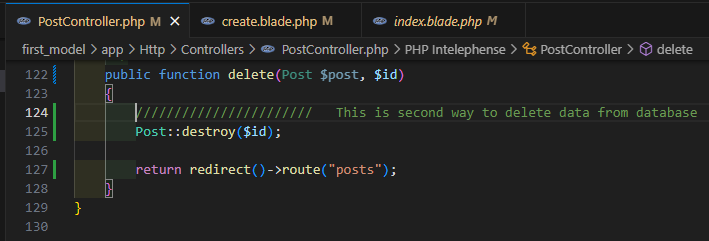
* To delete data from database by Laravel :-

Must make form to choose the way method after delete

* + 1- First way:-



* + 2 – Second way:-



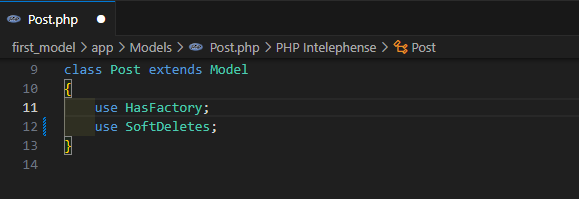
**SoftDelete()**

Used SoftDelete() to know when or to know the time for this post deleted from database

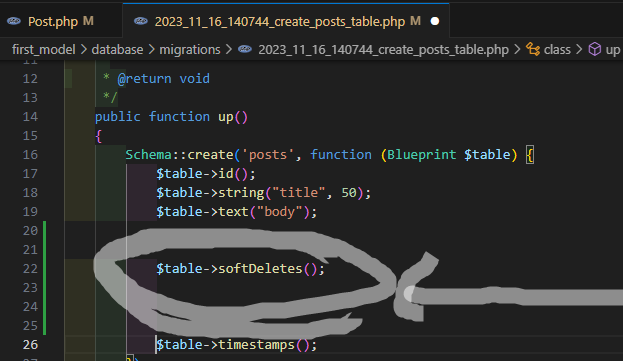
Used when you delete anything and you will needed this at another time

1. Open model file ( post.php )

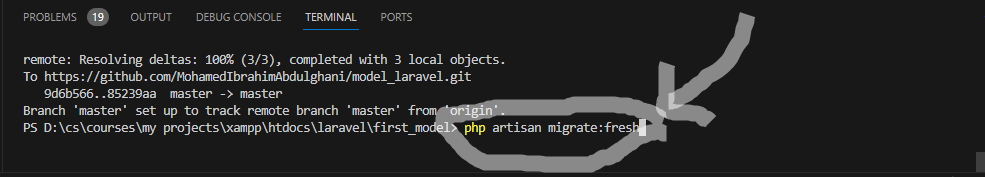
Write in Post class (use SoftDeletes)



1. Open database/migrations/table\_post



1. Open terminal and write this ( php artisan migrate:fresh )



\*-\* If I change in database :-

Open post model file

    // make this when you change table name in database

    protected $table="my\_posts";

    // if i want stop the timestamp()  or don't use timestamp() in my project

    public $timestamp = false;

    // this code to make column primaryKey

    protected $primaryKey = "post\_id";

    // this code to make stop auto increment for id

    public $incrementing = false;

If you deleted item from database and you need show this data inside another table or another page use this ( onlyTrashed()->get() ):

public function show() // لو عايز اعرض بيانات انا حذفتها من الداتا قبل كدة ومحتاج اننا اعرضها في مكان تاني

    {

        $post = Post::onlyTrashed()->get();

        return view("posts/show\_delete", compact("post"));

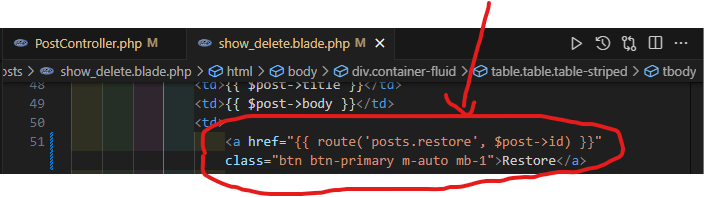
    }

If you want restore the data after deleted from database:

1-

Route::get("posts/restore/{id}", [PostController::class, "restore"])

->name("posts.restore");

2-

3-

    // this function to restore the deleted data from database

    public function restore($id)

    {

// لو عايز اعمل عملية ارجاع للبيانات التي قمت بحذفها من قبل من خلال الداتابيز

        Post::onlyTrashed()->where("id",$id)->restore();

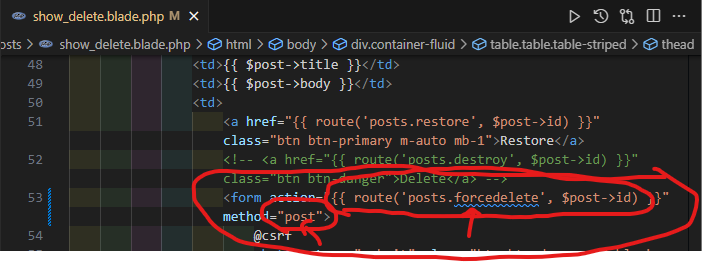
        return redirect()->back();

    }

If you want delete data from database forever and you must don’t use or need this data another time :-

1-

Route::post("posts/forcedelete/{id}", [PostController::class, "forcedelete"])->name("posts.forcedelete");

2-

3-

    // this function to delete data from database forever

    public function forcedelete($id)

    {

// لو انا متاكد اننا مش هحتاج البيانات دي خالص بعد كده هتتمسح نهائي من الداتابيز

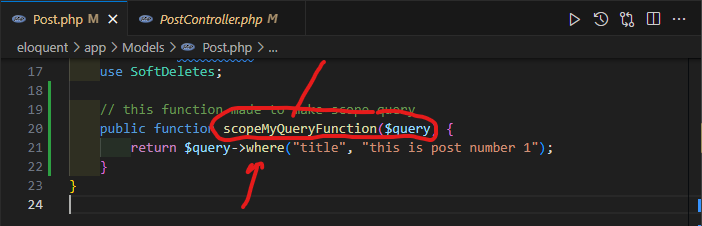
        Post::onlyTrashed()->where("id",$id)->forceDelete();

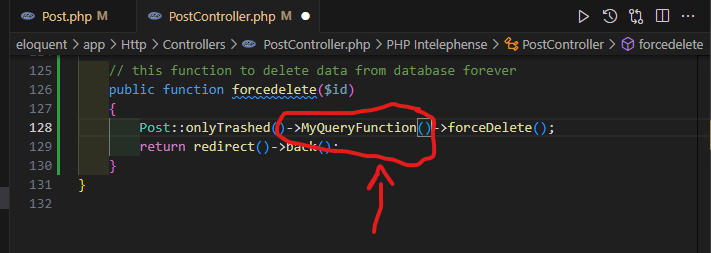
        return redirect()->back();

    }

If you want make query in database and use it more must make scope in model :-

1. Open model name and make function to write query inside this function

Public function scopeScopeName($query) {}

1. Use this scope query inside controller

وهتستخدمها عادي ف اكتر من مكان Function من قبل اسم ال Scope هتحذف كلمة

Validations:

To make validation must made one of two ways:

1. Make validation from controller
2. Make validation from request

First : After insert data inside database you must use this function ( $request->validate([]) )

1. Open postcontroller page

    public function store(Request $request)

    {

        $request->validate([

            "title" => "required",

            "body" => "required"

        ]);

        Post::create([

            "title"=>$request->title,

            "body"=>$request->body

        ]);

        return redirect()->route("posts.index");

    }

2- open create.blade.php page

    <div class="container">

  <form action="{{route('posts.store')}}" method="post" class="p-3 mt-3">

            @csrf

            <h1 class="text-center">Add New Post</h1>

                @if ($errors->any())

                    <div class="alert alert-danger">

                        <ul>

                            @foreach ($errors->all() as $error)

                                <li>{{ $error }}</li>

                            @endforeach

                        </ul>

                    </div>

                @endif

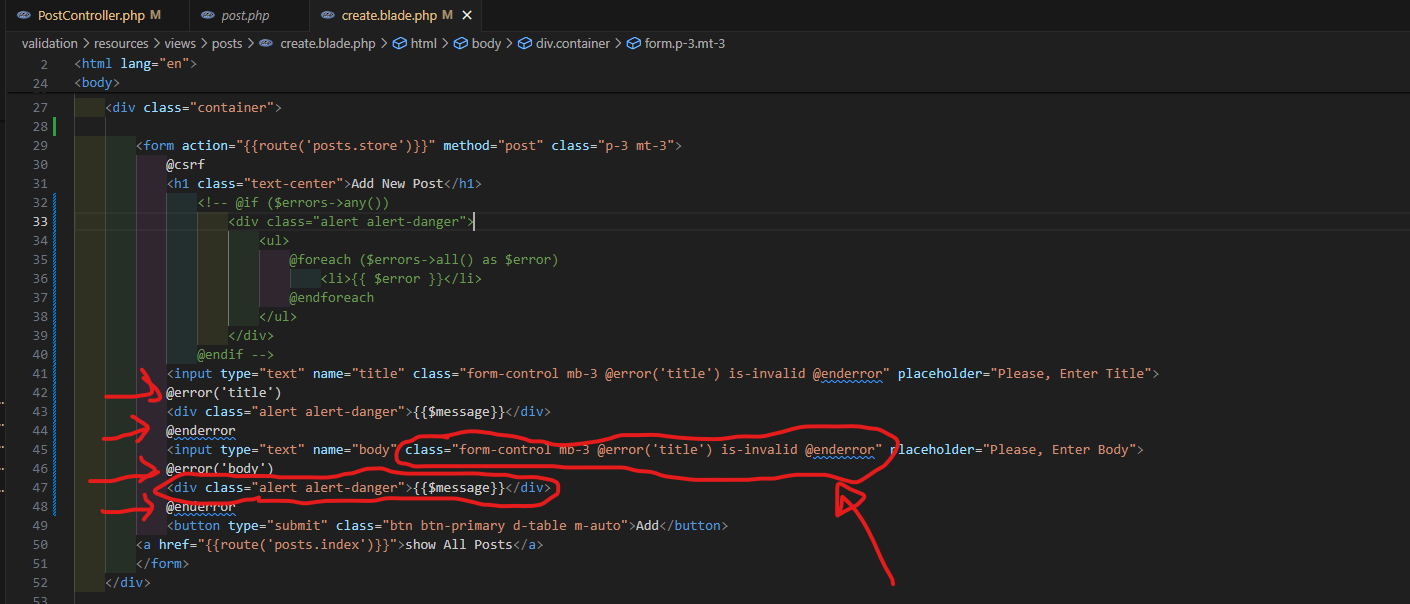
            <input type="text" name="title" class="form-control mb-3" placeholder="Please, Enter Title">

            <input type="text" name="body" class="form-control mb-3" placeholder="Please, Enter Body">

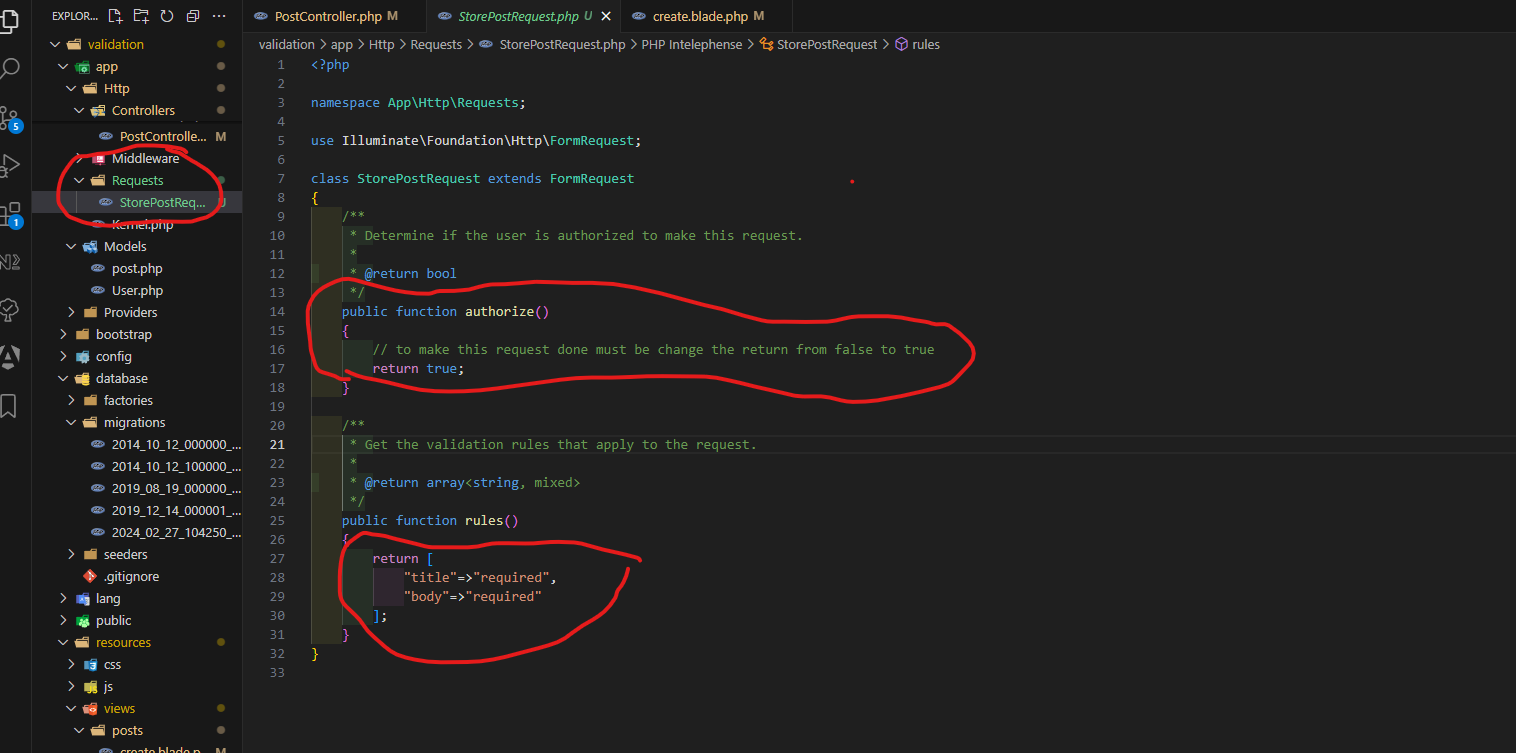
            <button type="submit" class="btn btn-primary d-table m-auto">Add</button>

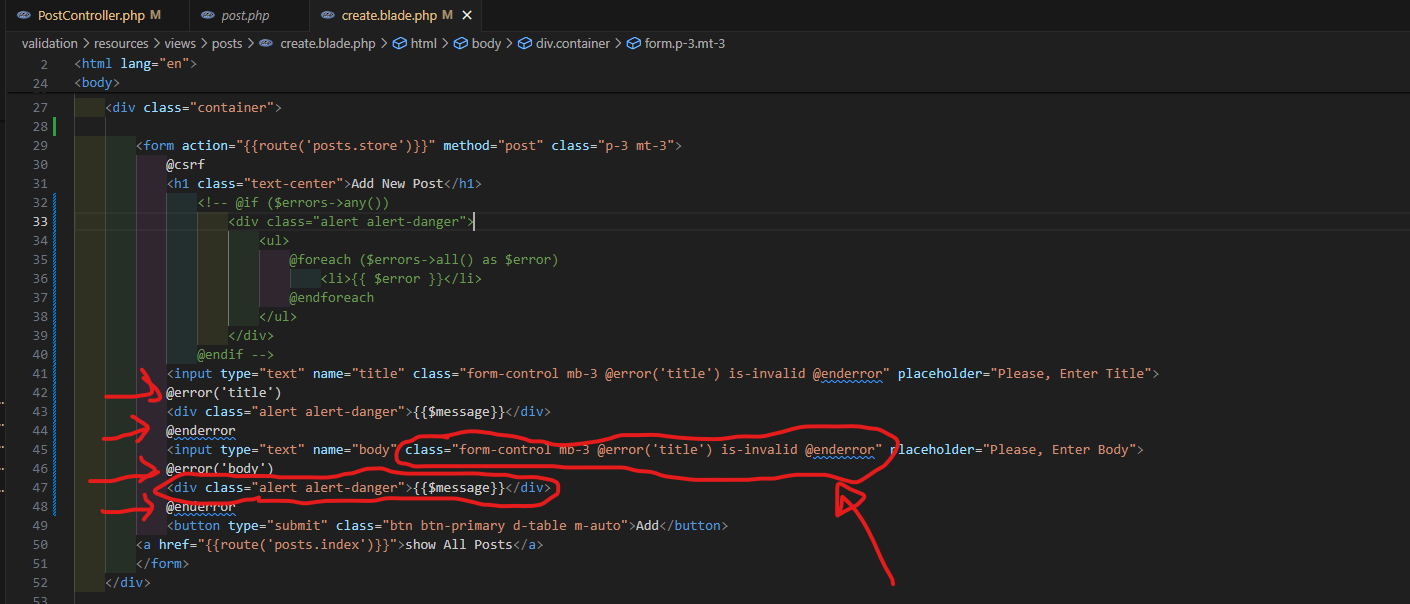
        <a href="{{route('posts.index')}}">show All Posts</a>

        </form>

Second :

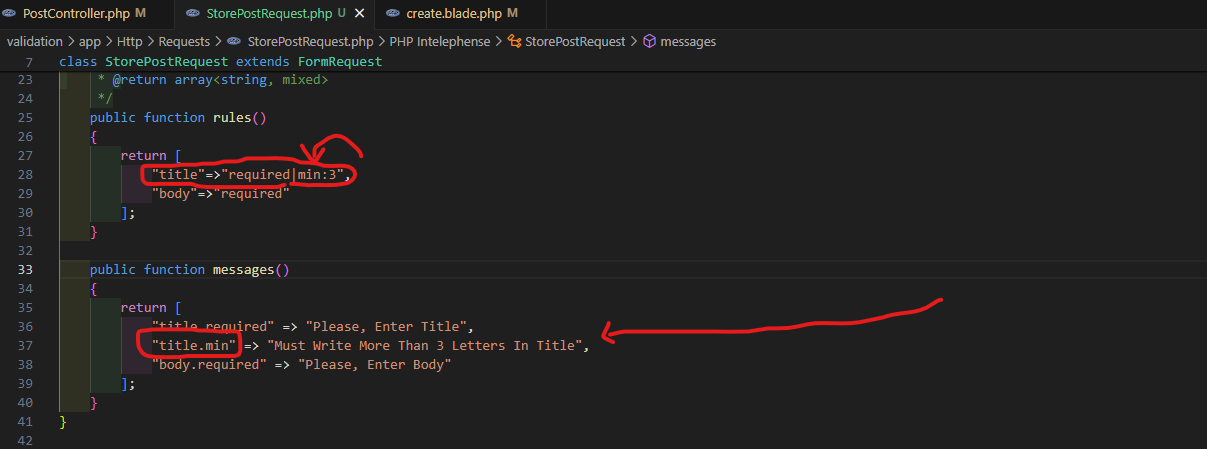
The second way to make request validation :-

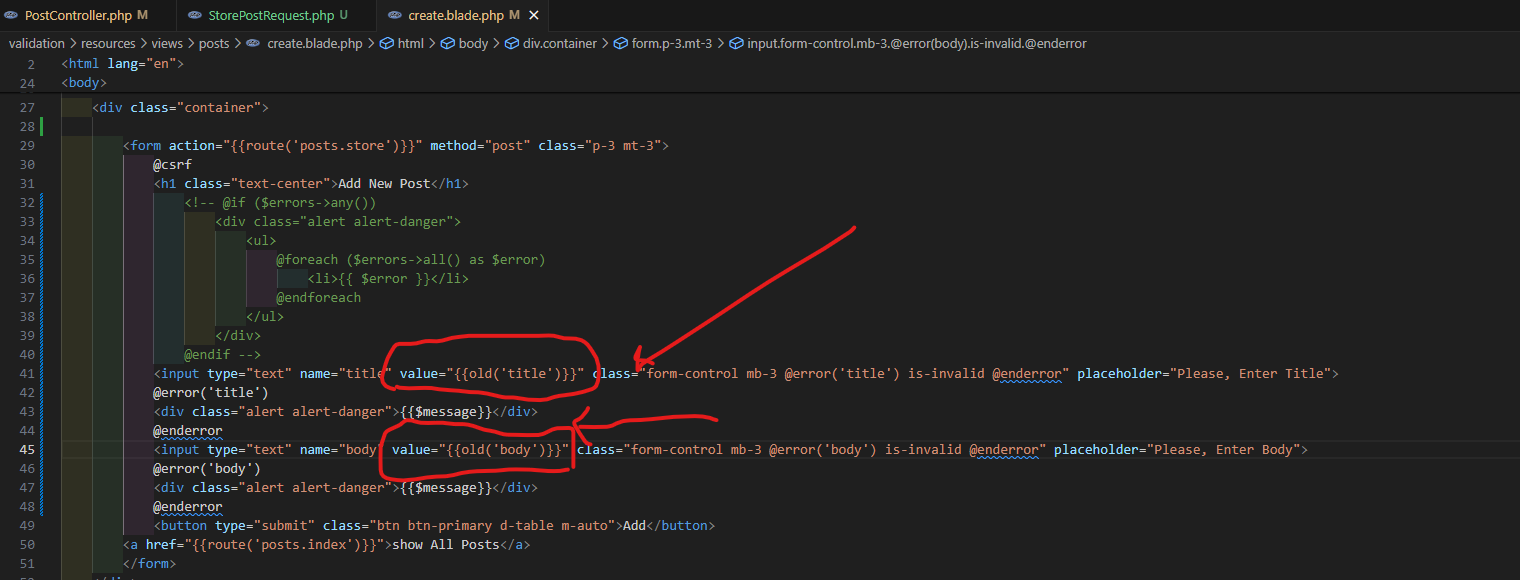
* Run this in terminal ( php artisan make:request RequestName )
* علشان يشتغل معاكReturn from false to true inside authorize() functionأولا لازم تغير



\*-\* If you want customize the error message :-

Make message function inside request file

\*-\* طب لو انا عايز اعمل اكتر من شرط

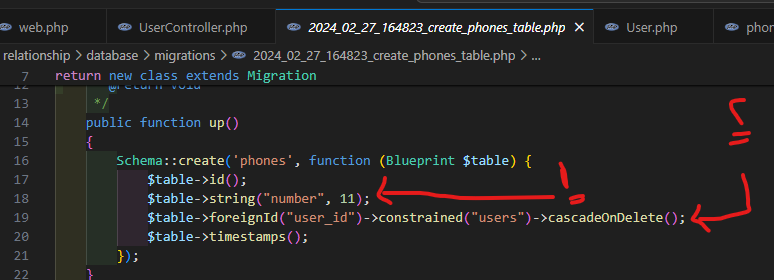
\*-\* valueلو انت دخلت بيانات غلط وبيانات صح وعايز البيانات الصح تبقي ذي ماهي ماتتمسحش هتستخدم ال

**Relationships**

* One to one
* One to many
* Many to many

1. **One to one**

) Usersاللي هتربطة بالجدول الأساسي ( ( Phones اول حاجه لازم تعمل الجدول (

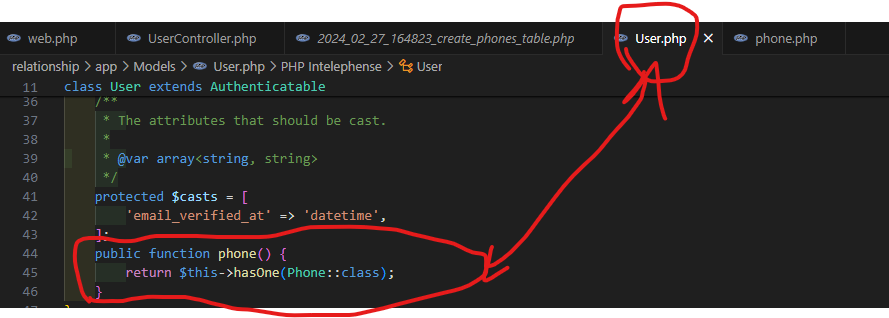
( php artisan make:model phone -m )

model for user and model for phoneتاني حاجة لازم تعمل

php artisan make:model user

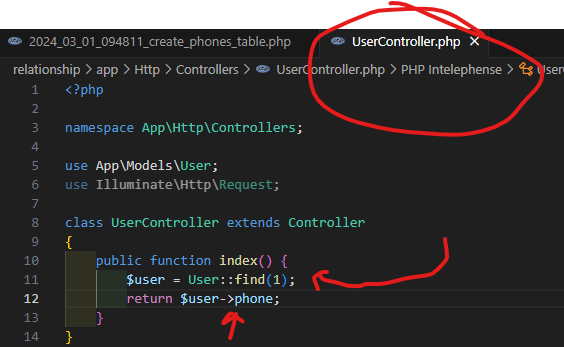
php artisan make:model phone

user model تالت حاجه هتكتب داخل ال



خاص بهPhone له User انت كده بتعرفه ان لكل

علشان تقدر تعرض بيانات جدول عن طريق جدول تاني Relationship one to one\*--\* خلاصة الكلام دة انت بتعمل

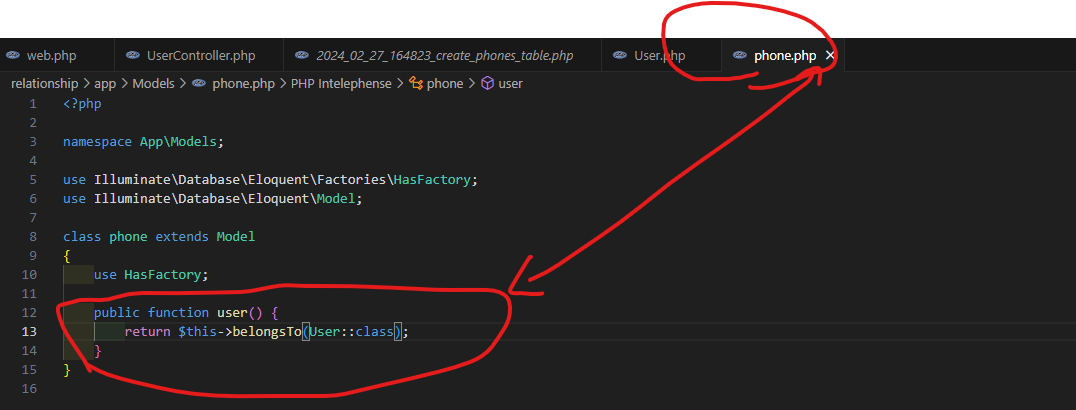
من خلال الربط بين الجدولين دول

رقم 1 \* خليته يعرض بيانات جدول عن طريق جدول تاني بسبب الربط اللي بينهم \*User كدة خليته يعرض الفون الخاص بال

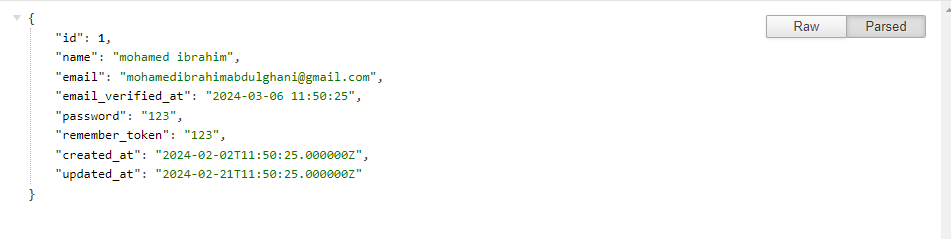
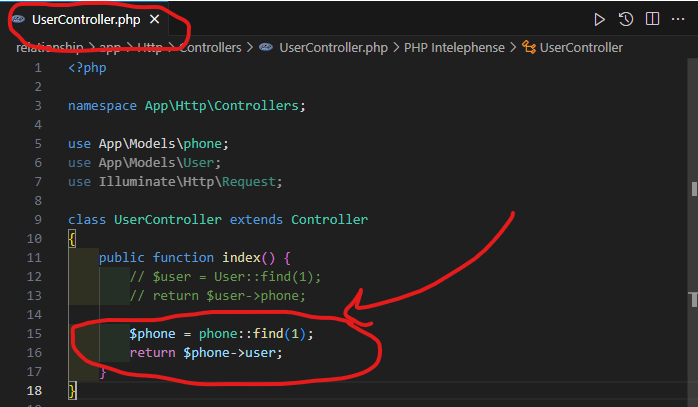


\*-\* افرض انت عايز تجيب العكس؟

)Phone من خلال جدول ال User يعني انت عايز تجيب بيانات المستخدم من خلال رقم تليفونه ( عايز تظهر بيناتات جدول ال

phone model - هتكتب داخل ال

User بينتمي لل Phone انت كده بتعرفه ان

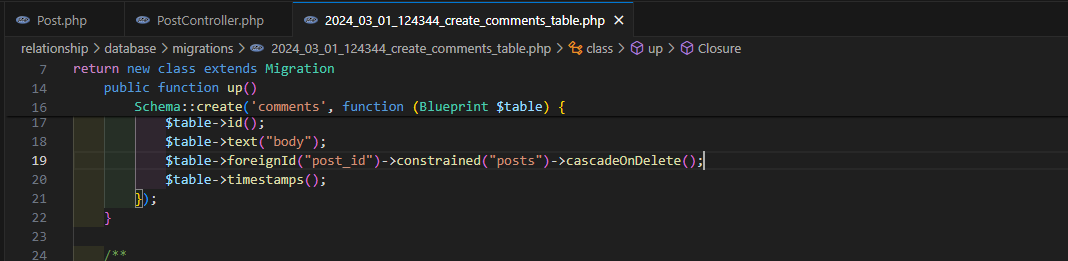
انت عايز تجيب بيناتات مستخدم معين عن طريق رقم تليفونه

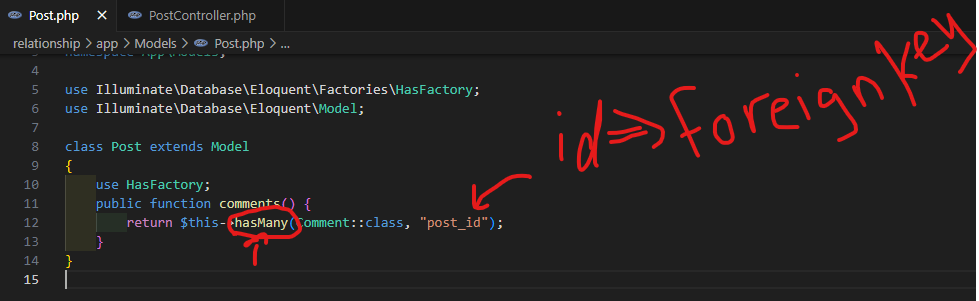
1. **One to many**

والعلاقة بينهم هتتبني علي أساس :-Comments وجدول عبارة عن Posts لو انت عامل جدولين جدول عبارة عن

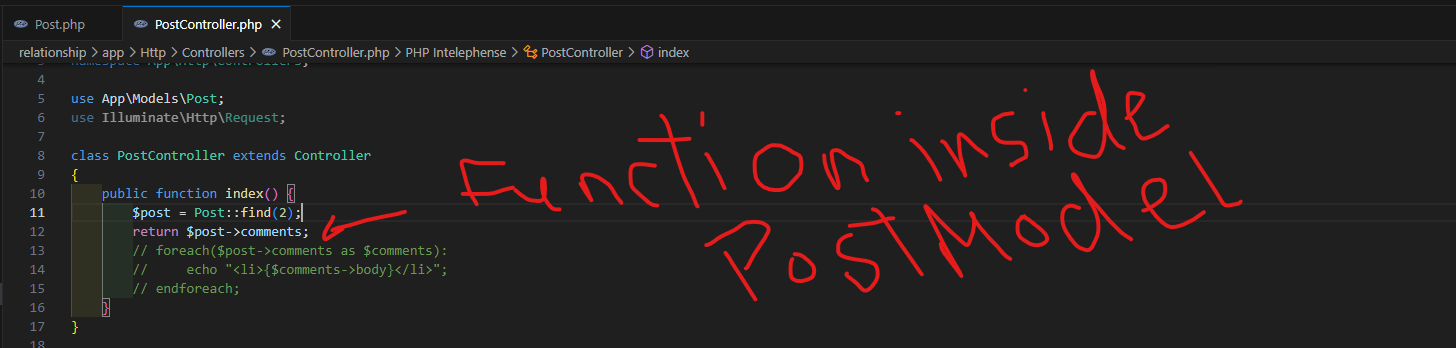
( One To Many ) علاقه Comments هيحتوي علي اكتر من Post ان كل

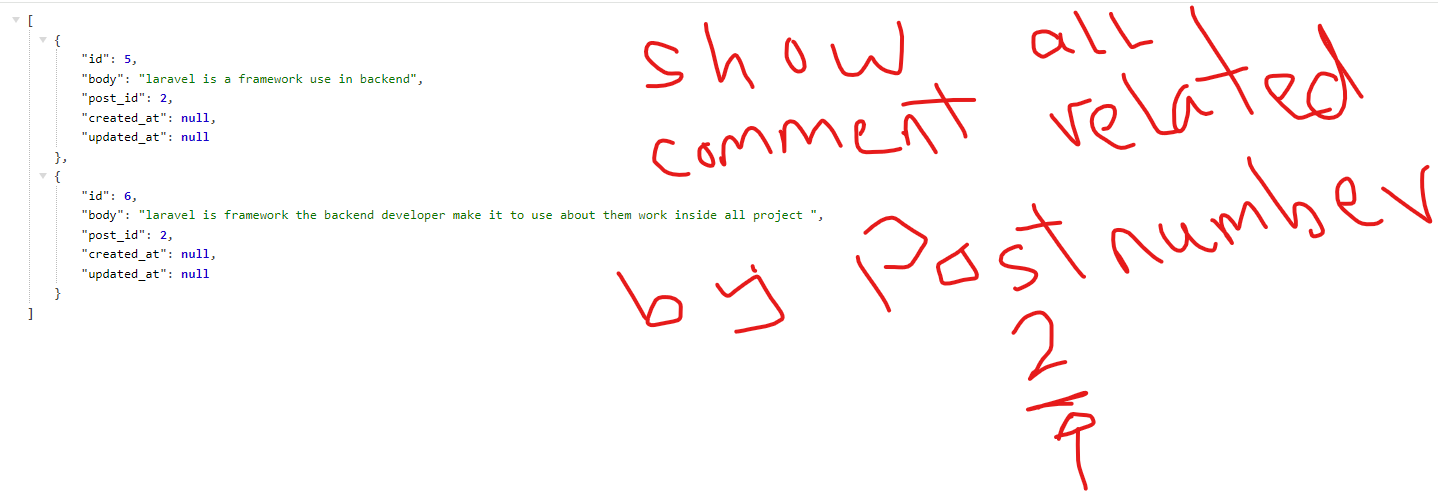
Php artisan make:model Post -m

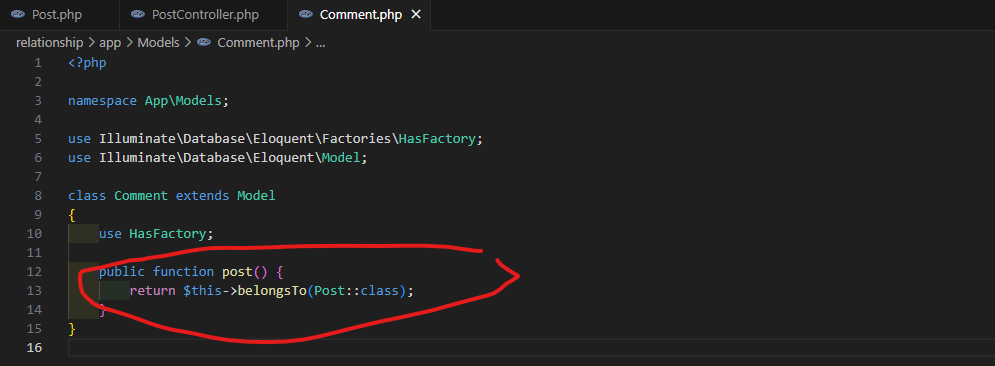
Php artisan make:model Comment -m

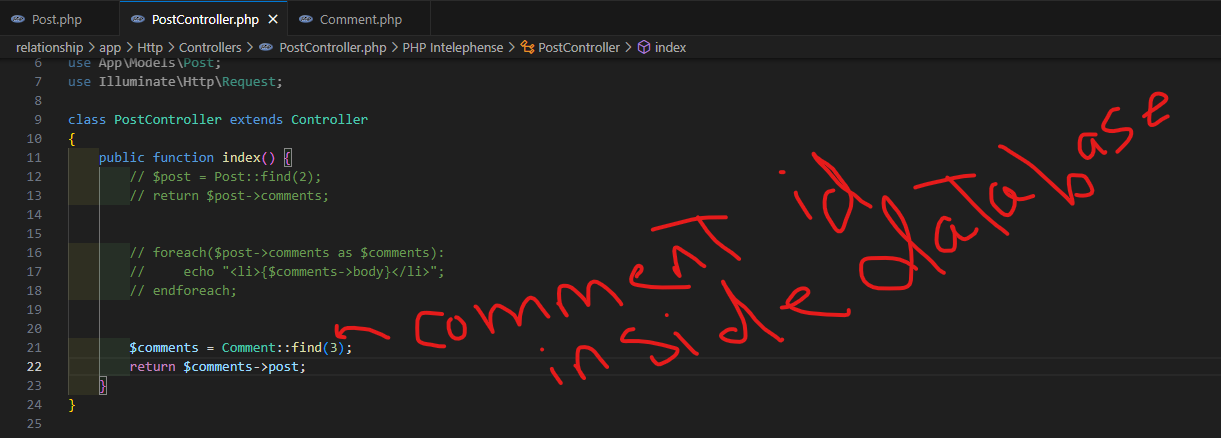
لو عايز اظهر الكومنتات من خلال جدول البوستات

Php artisan make:controller PostController





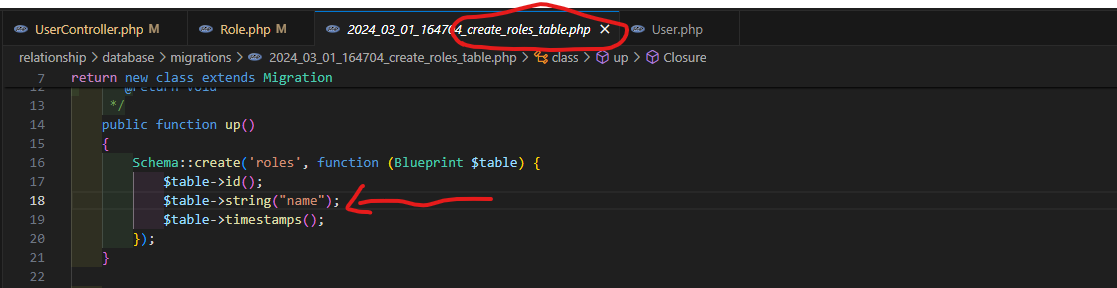
طب لو انا عايز اعرض البوست من خلال الكومنت

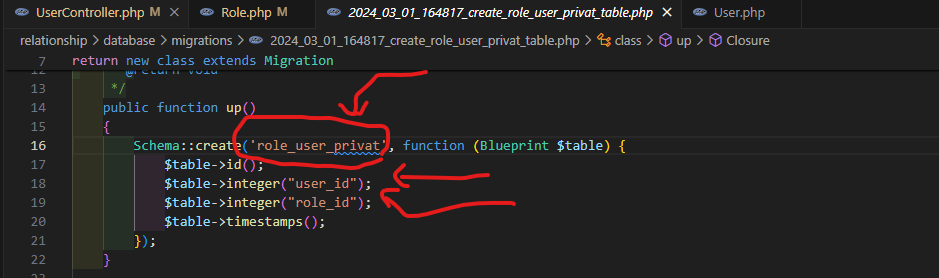


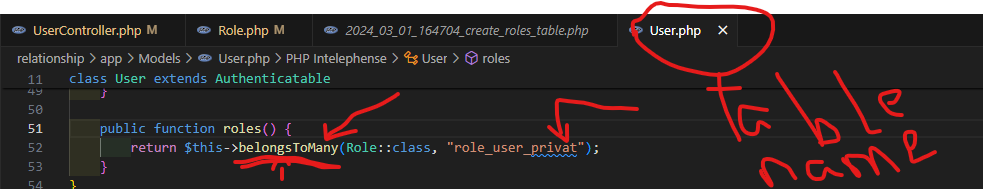
1. **Many to many**

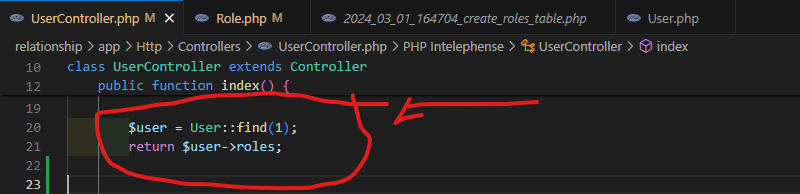
لو هعمل جدولين جدول المستخدم وجدول نوع المستخدم حيث يمكن للمستخدم استخدام اكتر من طريقة للدخول في الحالة دي هتكون Many to many العلاقة عبارة عن علاقة

\*-\* في هذه الحالة لازم نعم جدول تالت بحيث نقدر نربط بينهم بسهولة

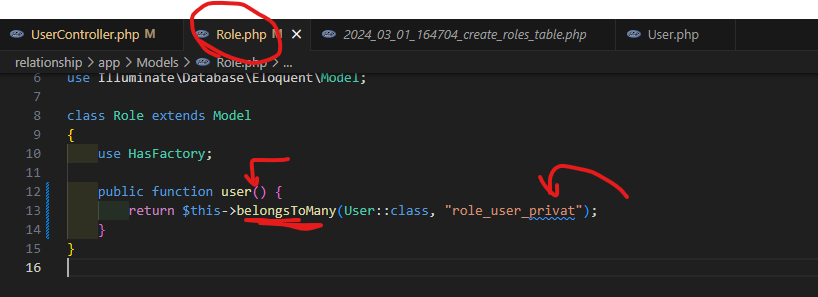
php artisan make:model Role -m

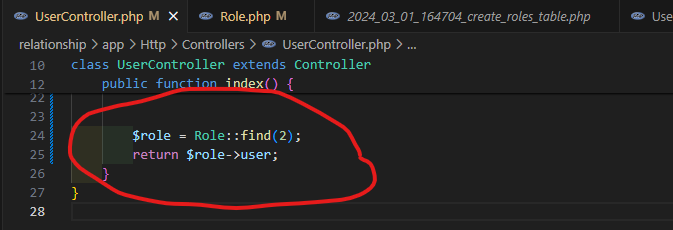
php artisan make:migration create\_user\_role\_privat\_table





بتقولة انت كدة اعرضلي نوع اللي داخل بيه المستخدم عن طريق المستخدم \* بمعني اصح اعرف المستخدم دة نوعه ايه \*





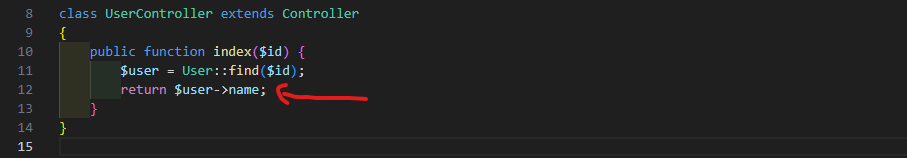


**Accessor function**

Use it if you want edit in returned data from database

بستخدمها لو عايز اعدل علي بيانات معينه راجعه من الداتابيز بتاعتي

* [Accessor function](https://laravel.com/docs/10.x/eloquent-mutators#defining-an-accessor)  =>بتجيب البيانات وبعدين بتعدل عليها
* [Mutator function](https://laravel.com/docs/10.x/eloquent-mutators#defining-a-mutator) =>وهي بتحط البيانات بتعمل التعديل عليها

Accessor function :-

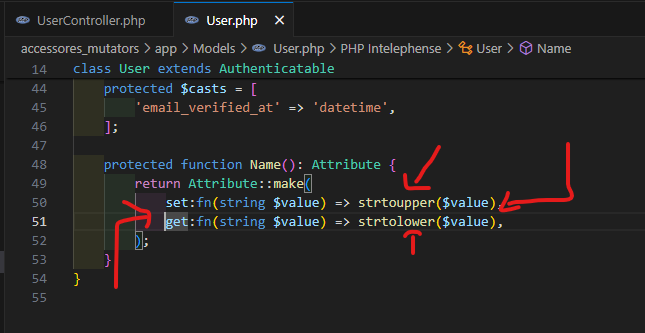
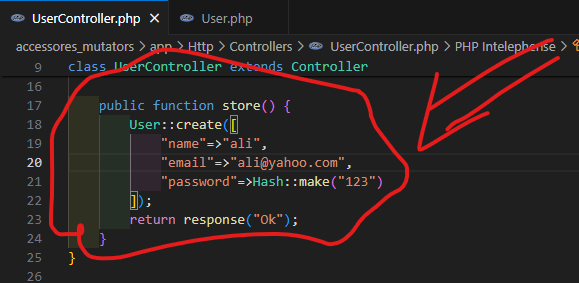
insert data inside database and get this data and customize about

Capital بعدل علي شكل الداتا ذي مانا عايز يعني هنا مثلا انا جبت الاسم من الداتا بيز وغيرت الصيغة بتاعته خليتها كلها



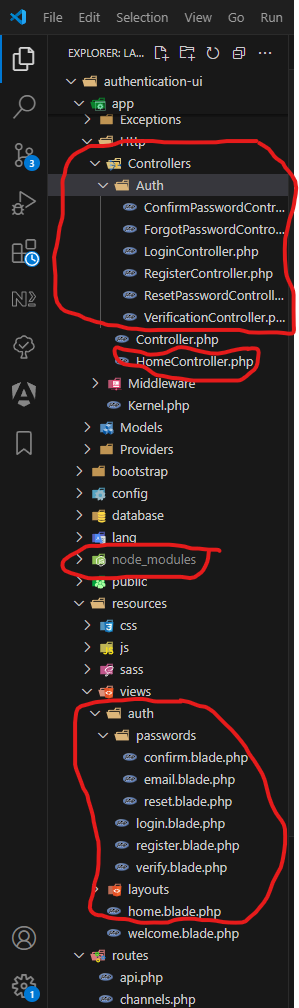
Mutator function :-

وهو داخل الداتا بيزCapitalبتديله شروط معينة ينفذها علي البيانات وهي داخله الداتا بيز يعني مثلا لازم الاسم يبقي

set() عن طريق Capital انت هنا بتعرفه وبتقولة البيانات اللي هتدخل عندك الداتا بيز لازم الاسم يكون

get() عن طريق lower وبتقوله برضو اعرضلي البيانات من الداتا بيز بس غيري ف شكل الاسم رجعهولي

**Authentication-UI**

Laravel عن طريقLogin or register لو انت عايز تعمل

1- Composer require Laravel/ui

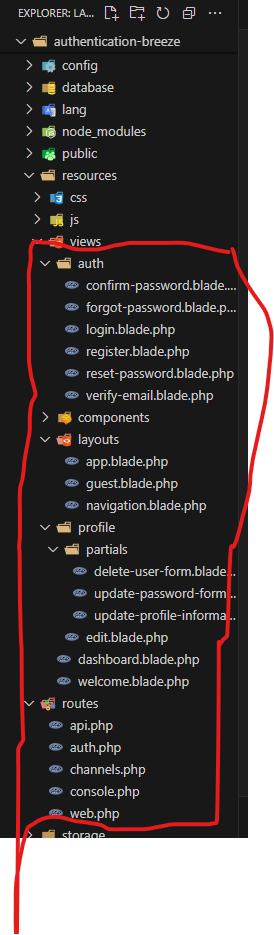
2- Php artisan ui bootstrap –auth

3-Npm install

4-Npm run build

Login and register لازم تكتب الأوامر دي في الترمينال علشان تقدر تعمل

اول ما تنشئ الأوامر دي تلقائي هتلاقي الملفات دي اتعملت من نفسها \*-\*

**Authentication-Breeze**

Laravel عن طريقLogin or register لو انت عايز تعمل

1- Composer require Laravel/breeze --dev

2- Php artisan breeze:install

3- choose blade

Login and register لازم تكتب الأوامر دي في الترمينال علشان تقدر تعمل

اول ما تنشئ الأوامر دي تلقائي هتلاقي الملفات دي اتعملت من نفسها \*-\*

Authentication Ui & Authentication Breeze- طب ايه الفرق بين

- الفرق بيبقي في شكل التصميم

**Authentication-jetstream**

Laravel عن طريقLogin or register لو انت عايز تعمل

1- Composer require Laravel/jetstream

2- Php artisan Jetstream livewire --teams

3- npm install

4- npm run build

Login and register لازم تكتب الأوامر دي في الترمينال علشان تقدر تعمل

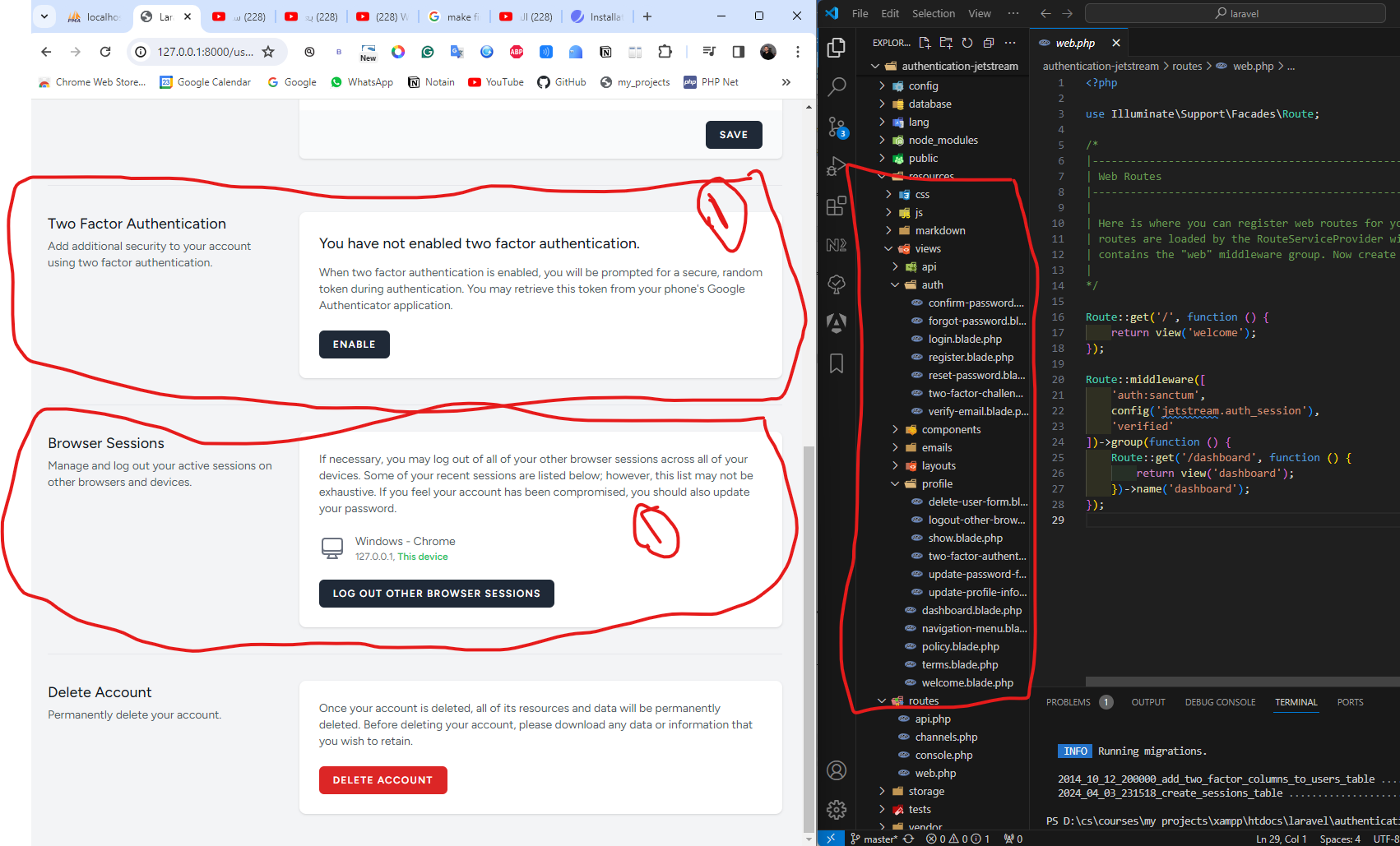
اول ما تنشئ الأوامر دي تلقائي هتلاقي الملفات دي اتعملت من نفسها \*-\*

**؟Authentication jetstream- لية استخدم ال**

1- بيضيف خاصية اننا لما اجي اعمل تسجيل دخول لازم يبعت رساله ( كود تسجيل ) علي برنامج اسمه (ِجوجل اسينتيكتور) علي الموبيل علشان اقدر اسجل دخول .

2- اقدر من خلالها اعرف انا فاتح من علي كام جهاز واقدر اتحكم في الأجهزة اللي انا فاتح من عليها من خلال اننا اسيبه فاتح او اخليه يسجل خروج تلقائي .

3- اقدر اعمل مجموعة او جروب للتيم الخاص بيا علي المشروع واقدر اتحكم في التيم ده ويظهرلي بيانات التيم دة

****Config->Jetstream ( Features::profilePhotos() 4- اقدر اضيف صورة للمستخدم من خلال ملف (

**Middleware**

بستخدمه علشان امنع ان أي شخص يفتح أي لينك للموقع من غير مايكون عامل تسجيل دخول او أكون محدده انه يكون ادمن في الموقع

**Php artisan make:middleware CheckUser**

Registration to middleware علشان اقدر اعمل كده لازم اعمل عملية

**طيب اعمل كده اذاي ؟**

**App => Http => kernel1 – هروح علي ال**

**علشان اقدر استخدمة Middlewaer 2- اعمل لينك لل**

    protected $routeMiddleware = [

        'CheckUser' => \App\Http\Middleware\CheckUser::class,

    ];

Web.3- هروح علي ملف ال

Route::middleware(['auth:sanctum', config('jetstream.auth\_session'),'verified',])->group(function () {

    Route::get('/dashboard', function () {

        return view('dashboard');

    })->name('dashboard');

Route::get("posts", [PostController::class, "index"])->middleware("CheckUser");// CheckUserالا ان كان في حالة middleware ركز انت هنا قولتله متعملش

});

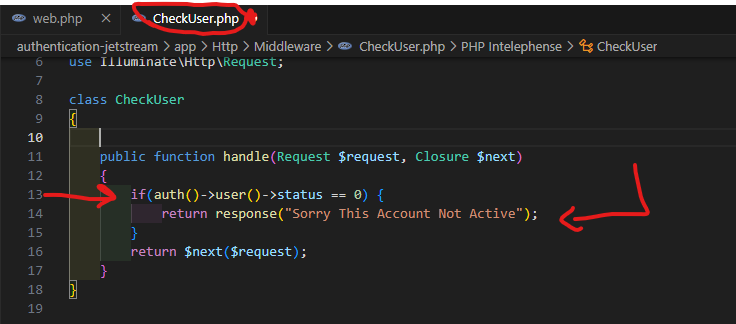
\*-\* بمعني اصح لو انا عامل سيستم مطعم وعايز اعمل صفحة للعناصر اللي اتباعت وسعر اليوم والفلوس اللي دخلت المطعم كل يوم

middleware والصفحة دي مينفعش حد يشوفها غير الادمن فقط هعمل

هتعمل داخل جدول المستخدمين :-

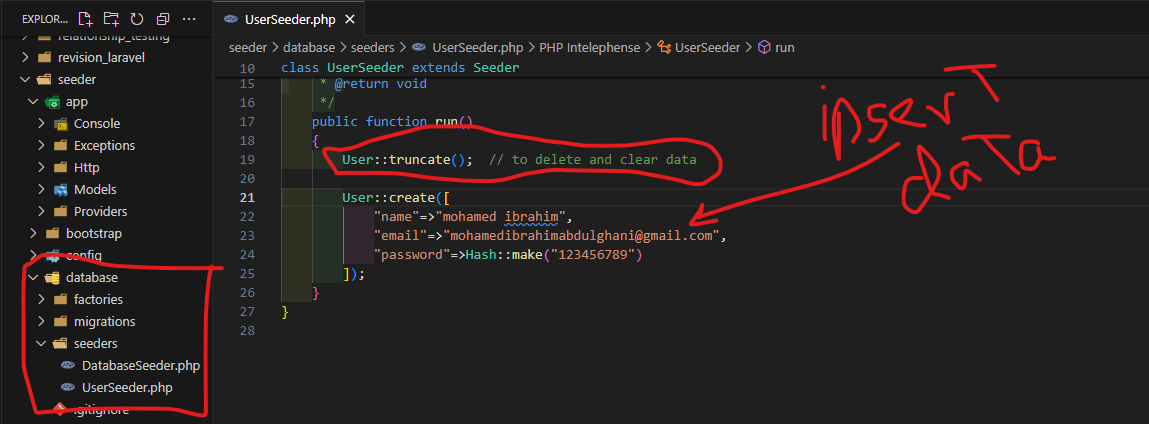
            $table->integer("status")->default(1);

= 0 معني كده انه مستخدم عادي Status = 1 معني كده انه ادمن يقدر يشوف صفحة الأسعار لو ال Status انت كده بتعرفه ان لو

ميقدرش يفتح صفحة الأسعار

**Seeder**

**formبستخدمه في حاله انا عايز اضيف داتا داخل الداتابيز من غير مااضيفها من خلال**

** Php artisan make:seeder UserSeeder seeder انا بعمل الامر ده علشان ينزلي ال**

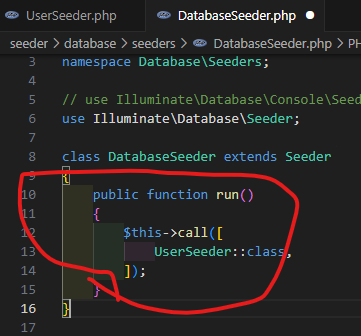
**Php artisan db:seed –class=UserSeeder علشان اقدر انفذ الامر ده بقا والبيانات تنضاف الي الدابيز**

**؟Seeder طب ايه الفايده من كل الكلام ده او انا هستفاد ايه لما استخدم ال**

بيقوم بحذف كل البيانات الموجودة عندي فيPhp artisan migrate:freshانا لما بدخل بيانات واعمل

seederالداتابيز ف لو انا عامل اكونت للادمن هيتمسح هو كمان ف ممكن تعمل اكونت الادمن من خلال ال

بدل ماكل شوية اكتب الامر Seeder \*-\* طب لو انا عايز اعمل اكتر من

Php artisan db:seed –class UserSeeder

DatabaseSeeder.php لا انا افتح الملف اللي اسمه -

واكتب الامر ده في الترمينال

Php artisan db:seed

**Factory**

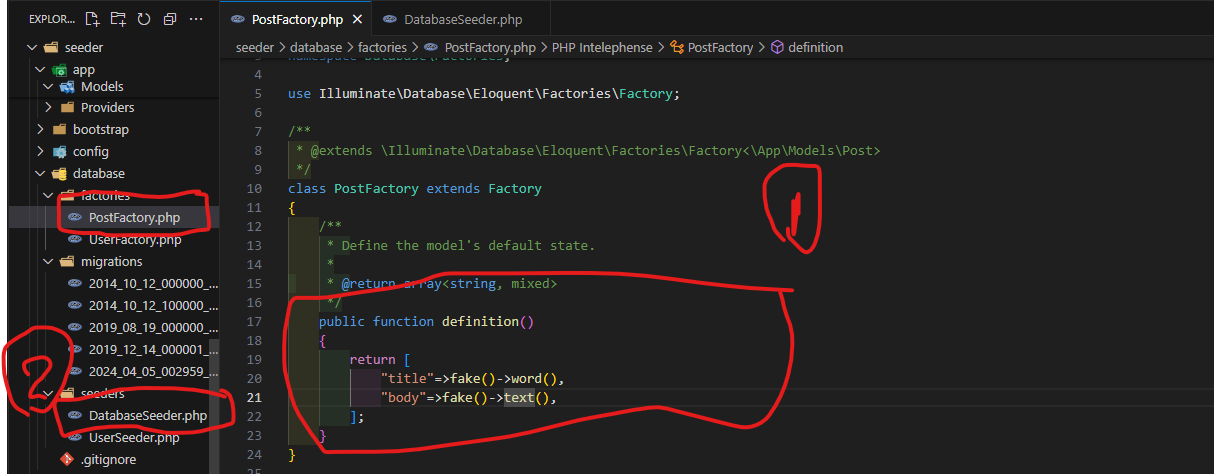
**Use it when you make fake data to check of the project ( Testing )**

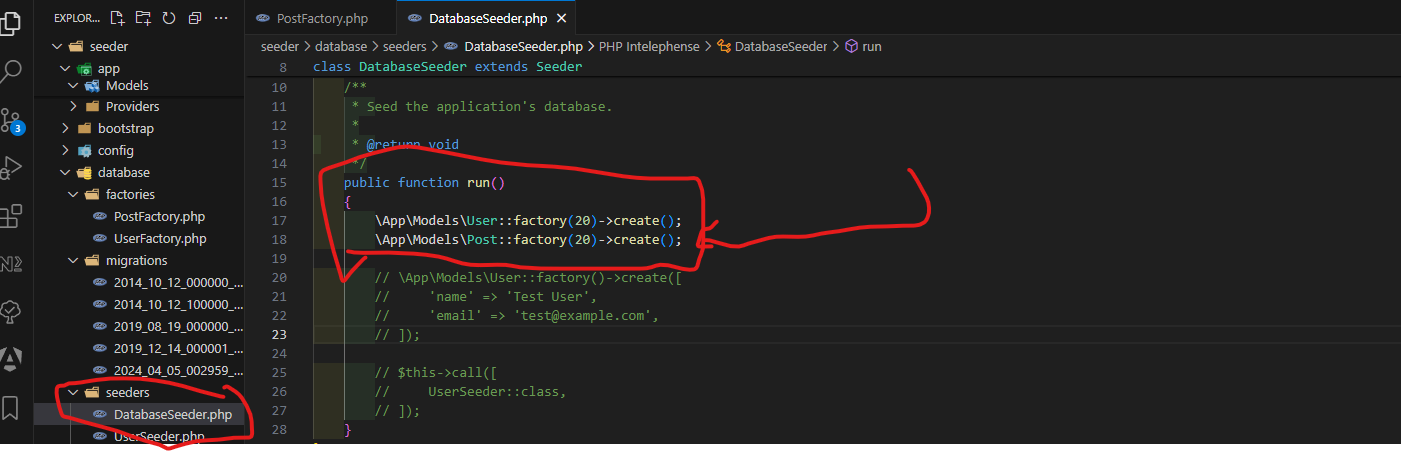
Open database => seeders => DatabaseSeeder.php

\App\Models\Users::factory(20)->create(); // this is line to make 20 user fake in database

Then : write this in terminal ( php artisan db:seed) to created users in database

**To Create New Factroy:-**

Php artisan make:factory PostFactory



**Mail**

**To install mailgun:-**

1. composer require symfony/mailgun-mailer symfony/http-client
2. open config => services.php

'mailgun' => [

'domain' => env('MAILGUN\_DOMAIN'),

'secret' => env('MAILGUN\_SECRET'),

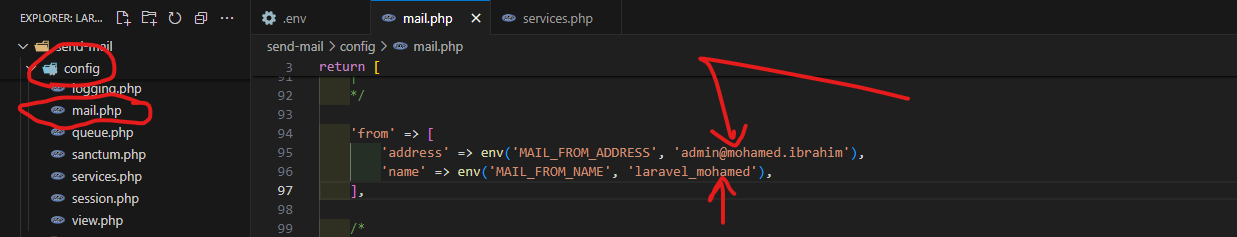
'endpoint' => env('MAILGUN\_ENDPOINT', 'api.mailgun.net'),

'scheme' => 'https',

],

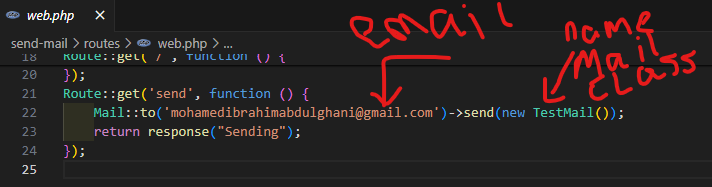
1. open file .env and write this inside the file, because the mailgun write from .env
   1. MAILGUN\_DOMAIN={ Open Mailgun Send Website And Take Doman Name } => the domain name is ( sandbox20f6775b26634cf1a6c33054b4f3bf76.mailgun.org )
   2. MAILGUN\_SECRET={ Open Mailgun Send Website And Take Secret Api Key }

=> the secret api key is(0676fd0097455fb3b4dbcc63f85f96e7-4c205c86-903f895a )

If you want change the information for mail open config => mail.php

To create mail:-

Php artisan make:mail TestingMail

Go to web app to make route

**Markdown in mail**

لانها بتقوم بتغير شكل الرسالة فقط في الMarkdown بنستخدم ال

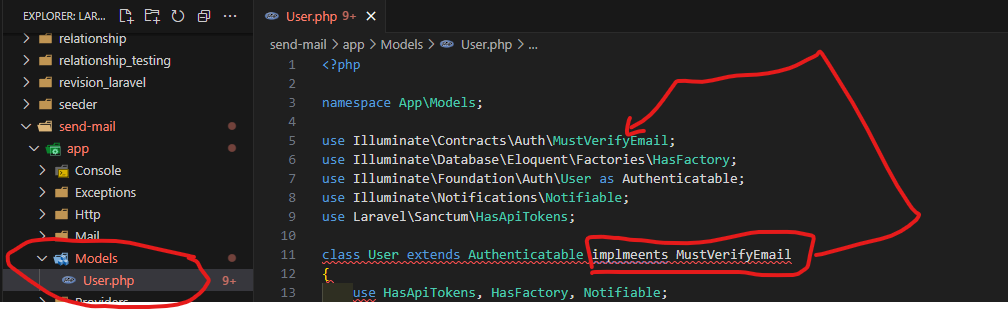
**php artisan make:mail TestMail --markdown=emails.test**

test وبداخل الفولدر ده أنشئ ملف emails وانشئلي ملف فولدر اسمه mail في الامر ده انت بتقوله اعملي

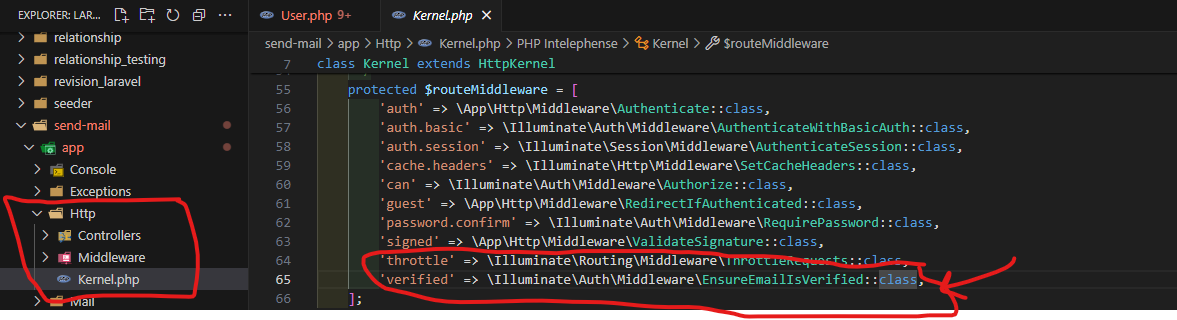
**Verification Email**

To make verification must make it:-

1. Open file model and make implement class from verifyemail



1. Open kernel.php from Http Folder



1. web وهتحطه في صفحة ال Verified هتاخد اسم ال

Route::get('/dashboard', function() {

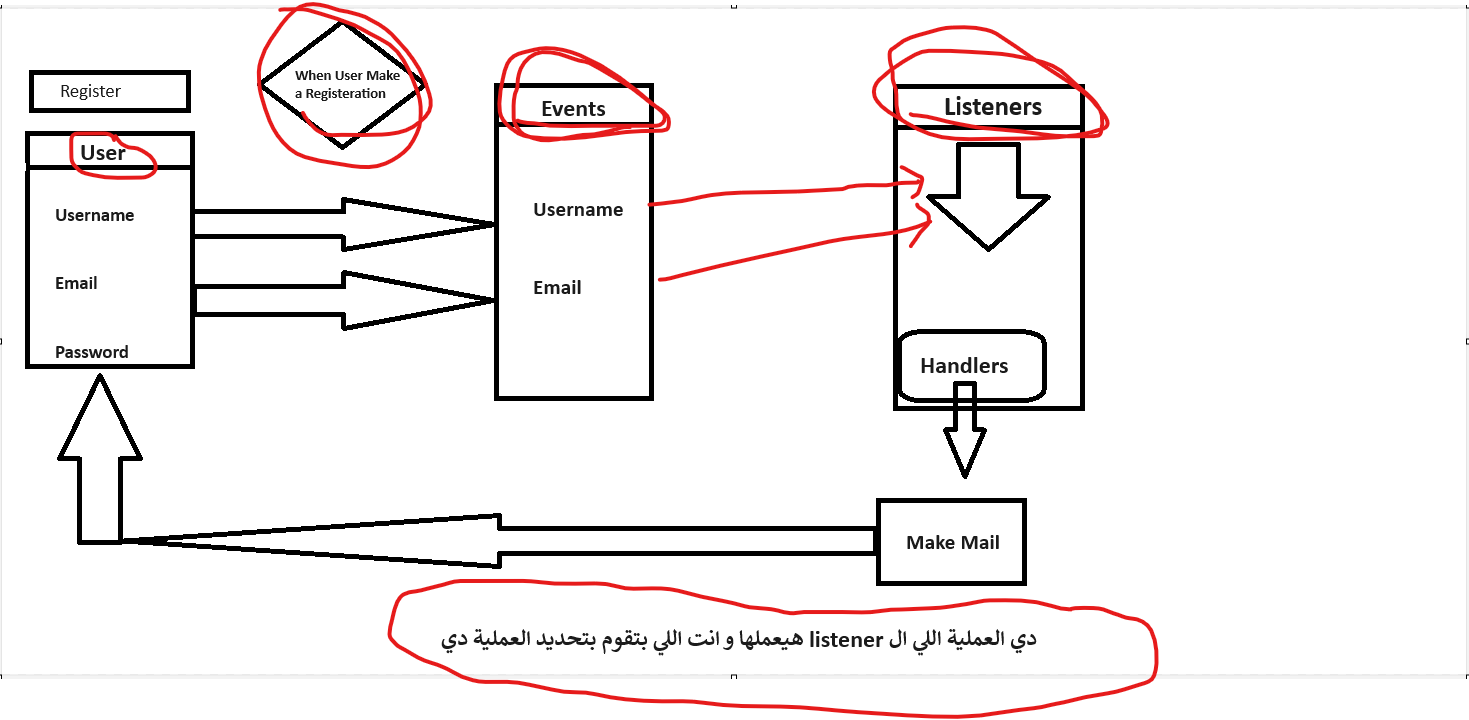
return view('dashboard');

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

\*-\* باللي انت عملته ده كده انت خليتك أي حد مينفعش يسجل ف الموقع غير لما يعمل تأكيد من حسابه من علي جوجل

**Events And Listeners**

Event => if condition is correct => make action



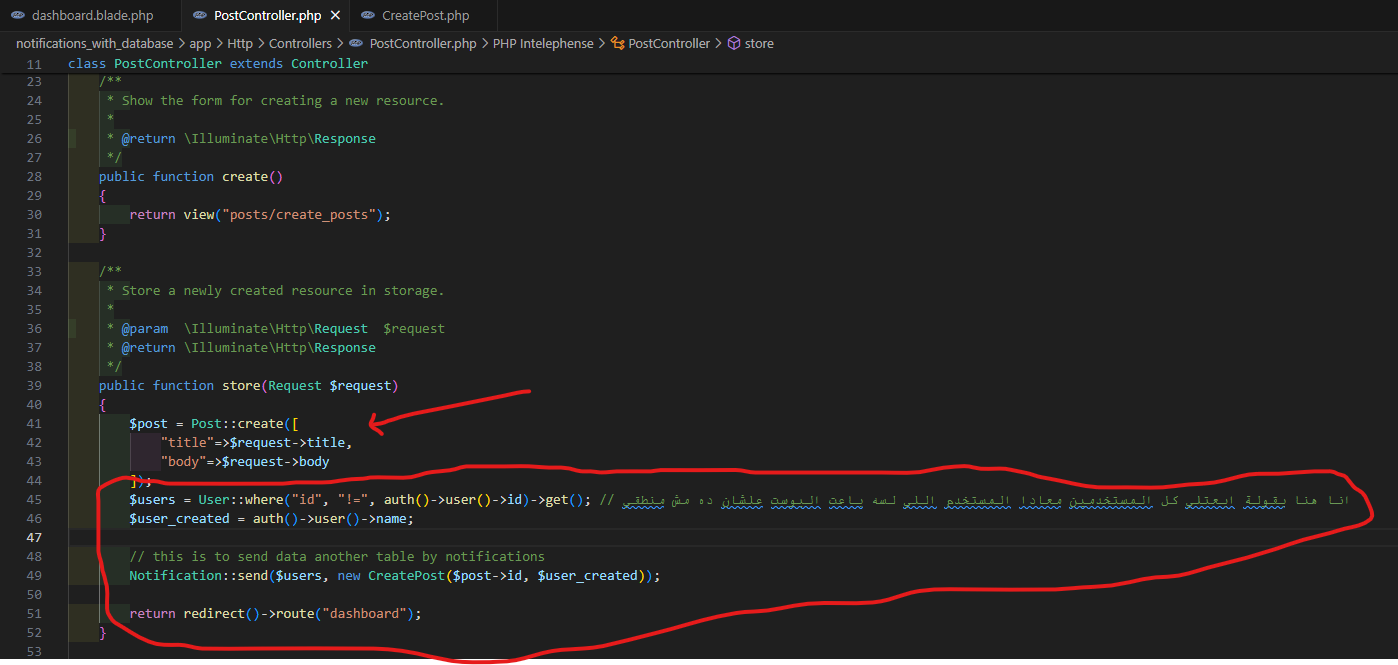
**\*-\* There are events build in by Laravel**

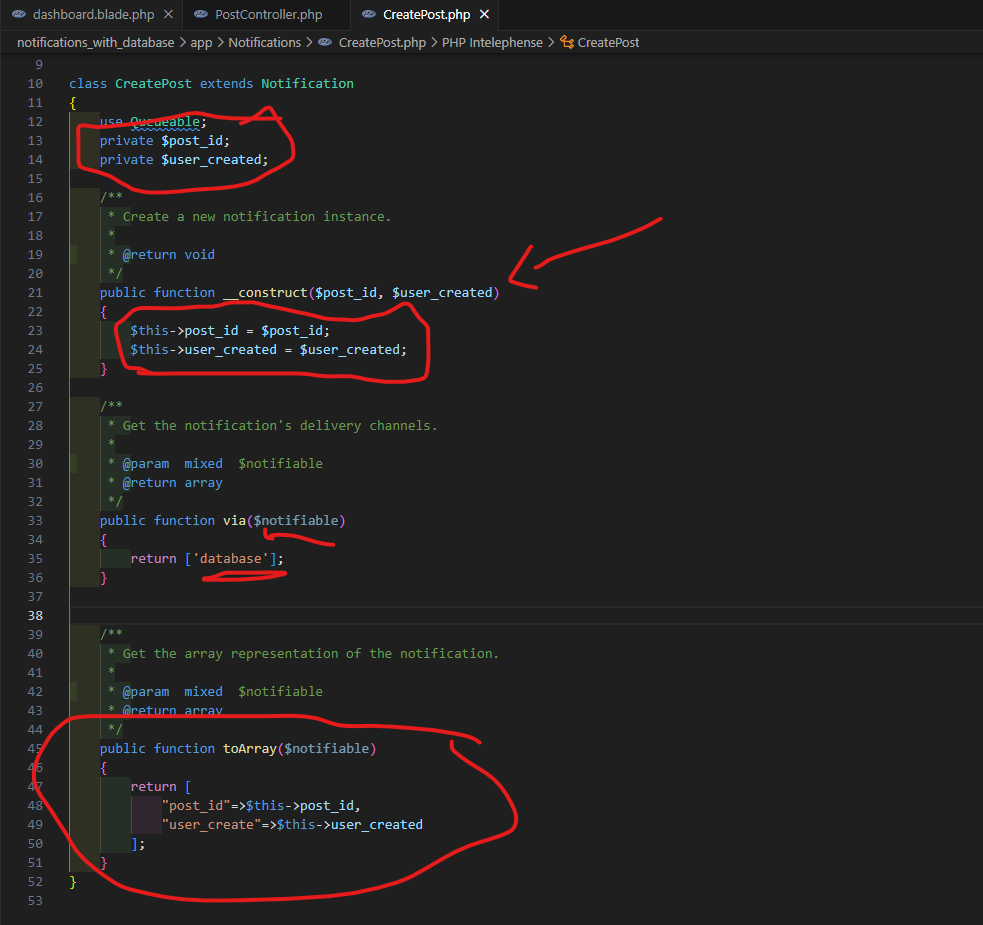
**Notifications With Database**

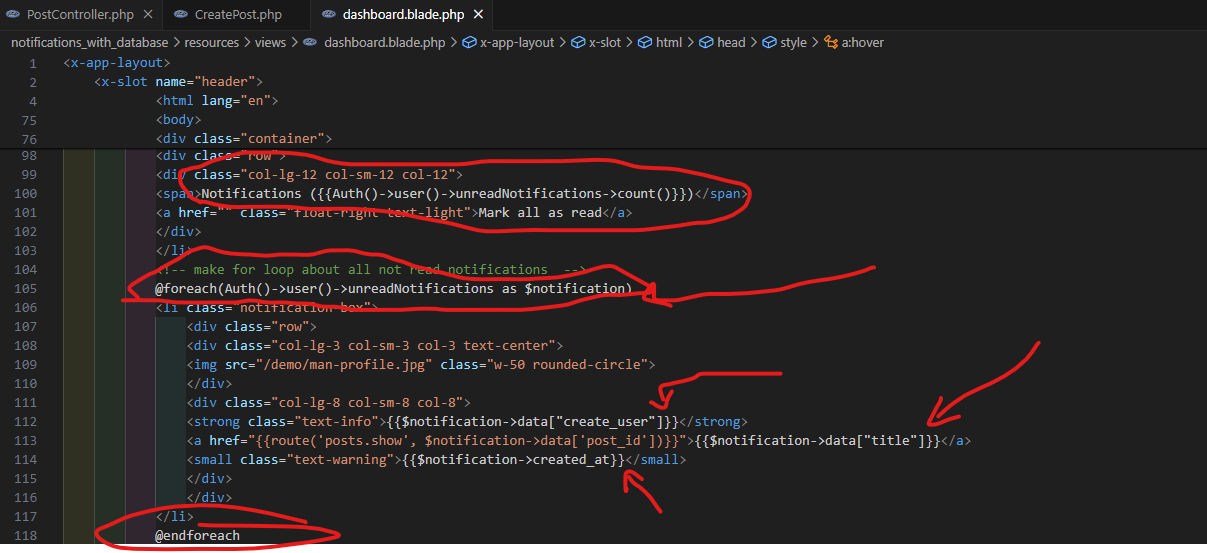
To install notification with database:

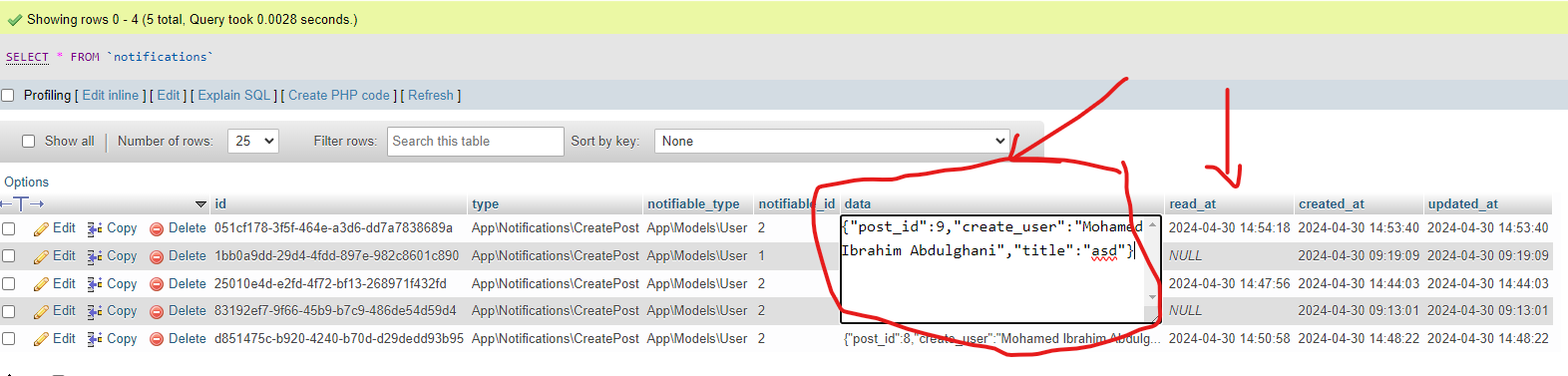
1. Php artisan notifications:table // to make create notificaions table in database

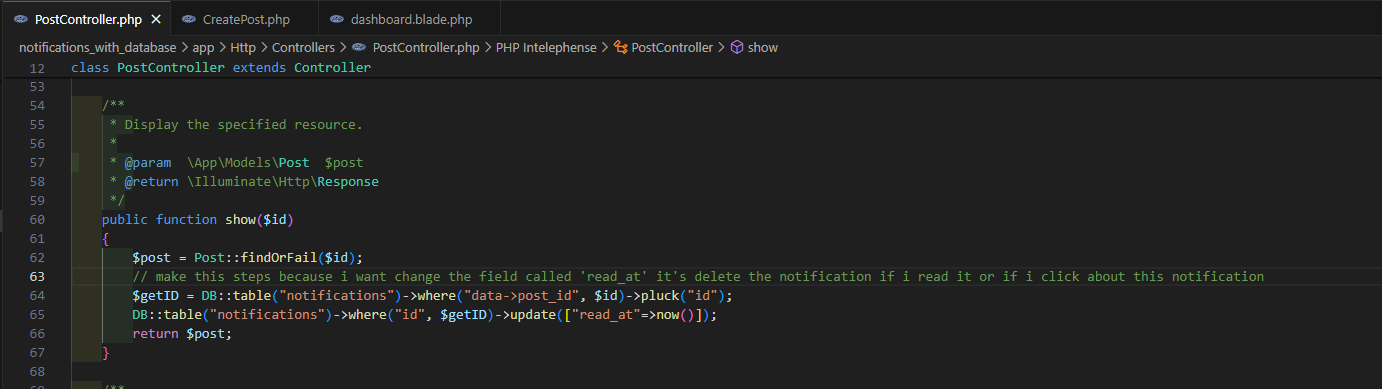
**Php artisan make:notification CreatePost** // to create notifications

****



 Notifications لو انا عايز اعرض البيانات من جدول ال



طب لو انا عايز لما ادوس علي الاشعار افتحه وأول ماافتحه يتحذف تلقائي من قائمة الاشعارات اللي لسه موجودة ومتفتحتش

*Idهتحدد الاشعار او البوست عن طريق ال - 1*

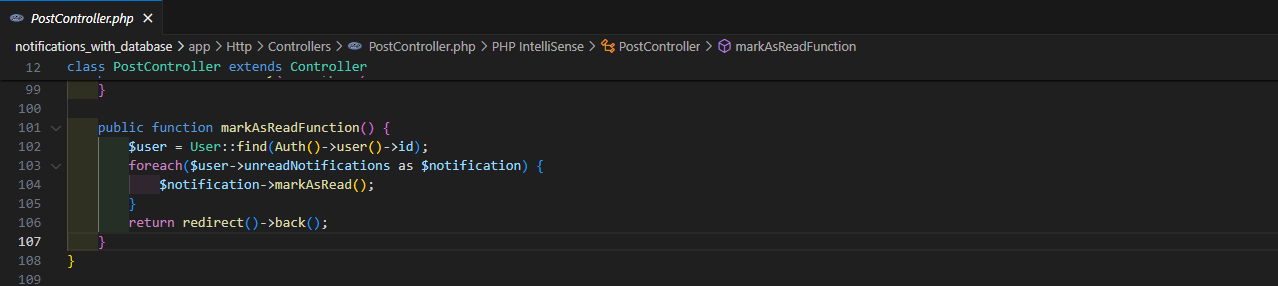
*اللي موجودid الخاص بالبوست اللي اتبعت = ال id2- هتختار جدول الاشعارات وهتحطله شرط لما يكون ال*

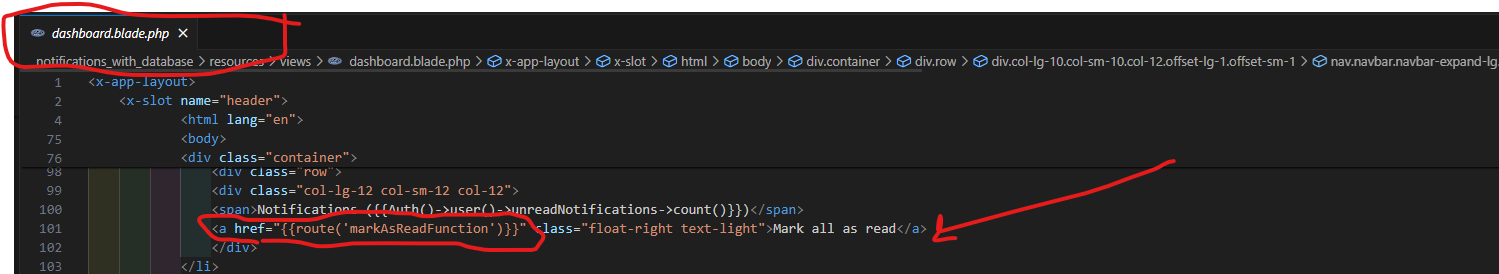
*في جدول الاشعارات*

*الخاص بالبوست اللي اتبعت لو الشرط اتحقق id الخاص بجدول الاشعارات = ال Id 3- هتعمل شرط لو ال*

*غير الوقت اللي فيها بالوقت الحالي وبالتالي انت كده خليته يغير قيمتهاRead\_at وغيرلي ال Update هتقوله اعمل*

*فبالتالي هيبقي انت كده قرأت الاشعار ف هيختفي تلاقي من قائمة الاشعارات*

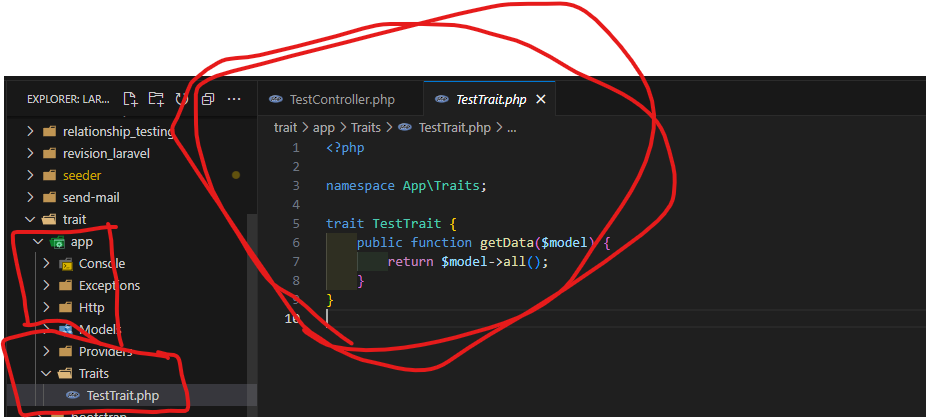
 Notifications او اننا شوفت كل ال Read all notificationsطب لو عايز اعمل



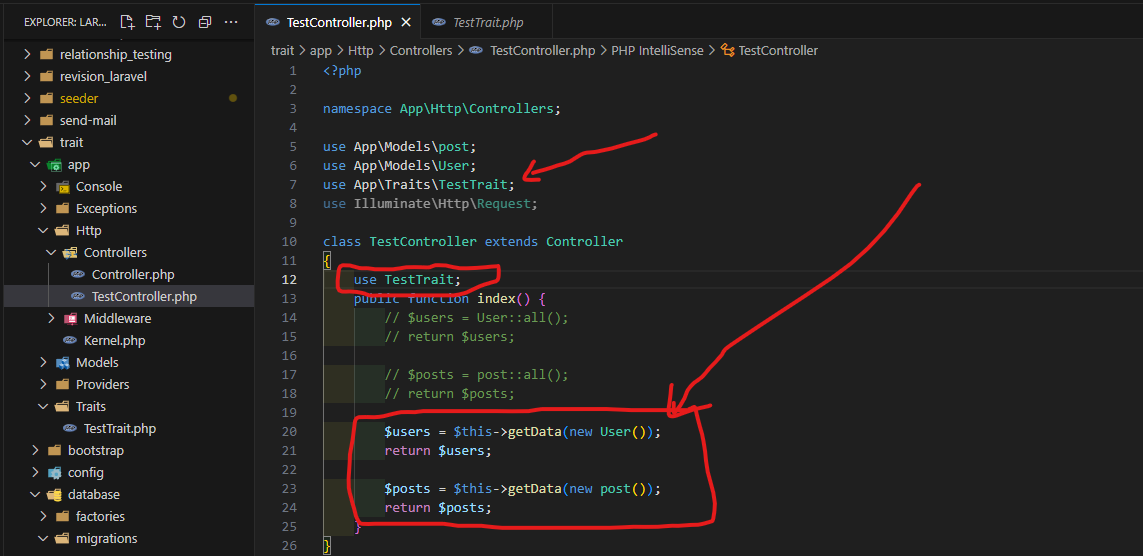
**Trait**

ده امتي ولية ؟ traitاحنا بنستخدم ال

- احنا بنستخدمه لما يكون عندي امر معين هكرره اكتر من مره مثلا داله رفع الصور مش كل traitشوية هكتبها ف أي مكان هعوز ارفع فيه الصورة لا انا هعملها مره واحده داخل ال

 ده ف أي مكان انا عايزه هنادي عليه بسTrait واستخدم ال

وكتبت جواه أي حاجه TestTrait.php وعملت جواه ملف سميته Traits انا عملت فولدر سميته

انا عارف اننا هستخدمها اكتر من مره ف أي صفحه تانيه

اخرClass داخل أي Use TestTraitاللي انا عملته لازم اكتر الامر Trait علشان اقدر استعدي ال

**Helpers**

بقدر استخدمها في اكتر من مكان واكتر من مره داخل المشروعFunctionsعبارة عن

App داخل فولدر ال helpers.php1- بعمل ملف اسمه

composer.json2- ادخل علي ملف

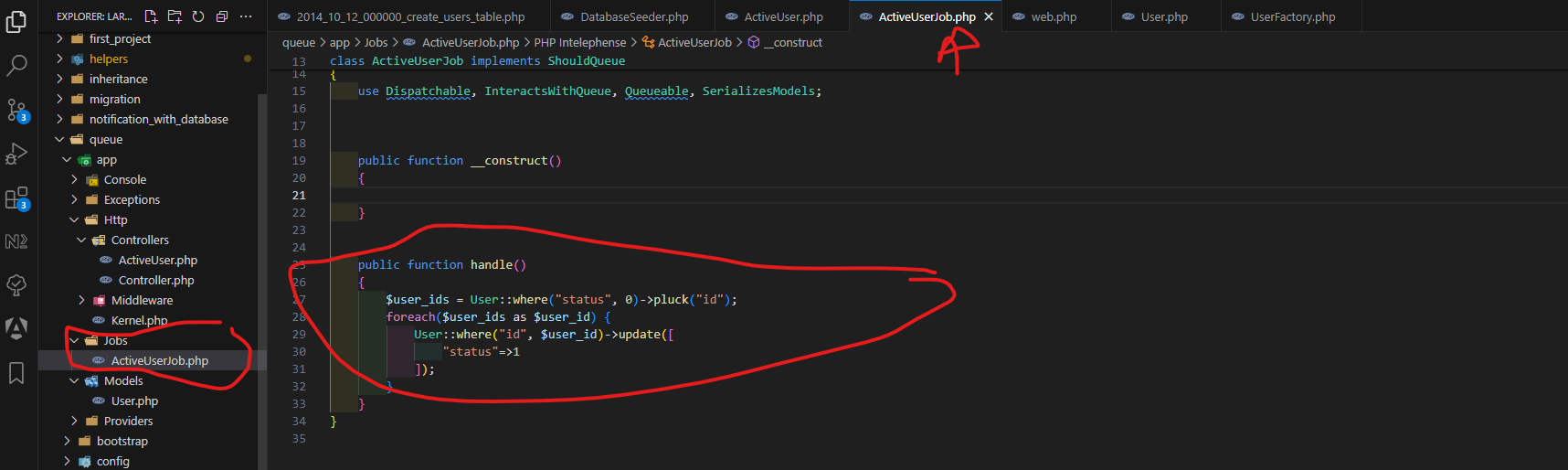
لازم تكتب الامر دهhelpers.phpعلشان المشروع يشوف الملف بتااعي اللي اسمه

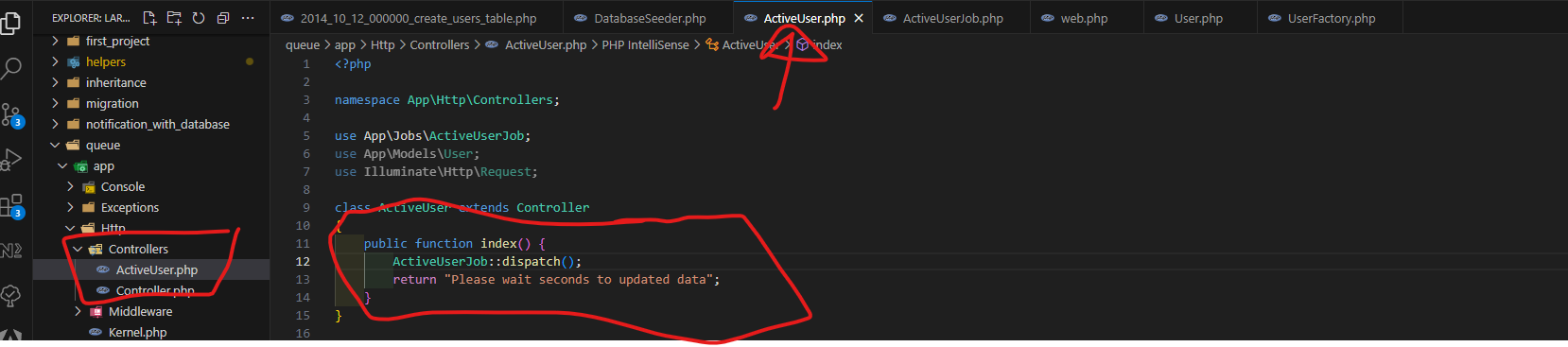
من جديد Autoload علشان تخلي المشروع يتعمله ( Composer dump-autoload )\*-\*

**Job With Queue**

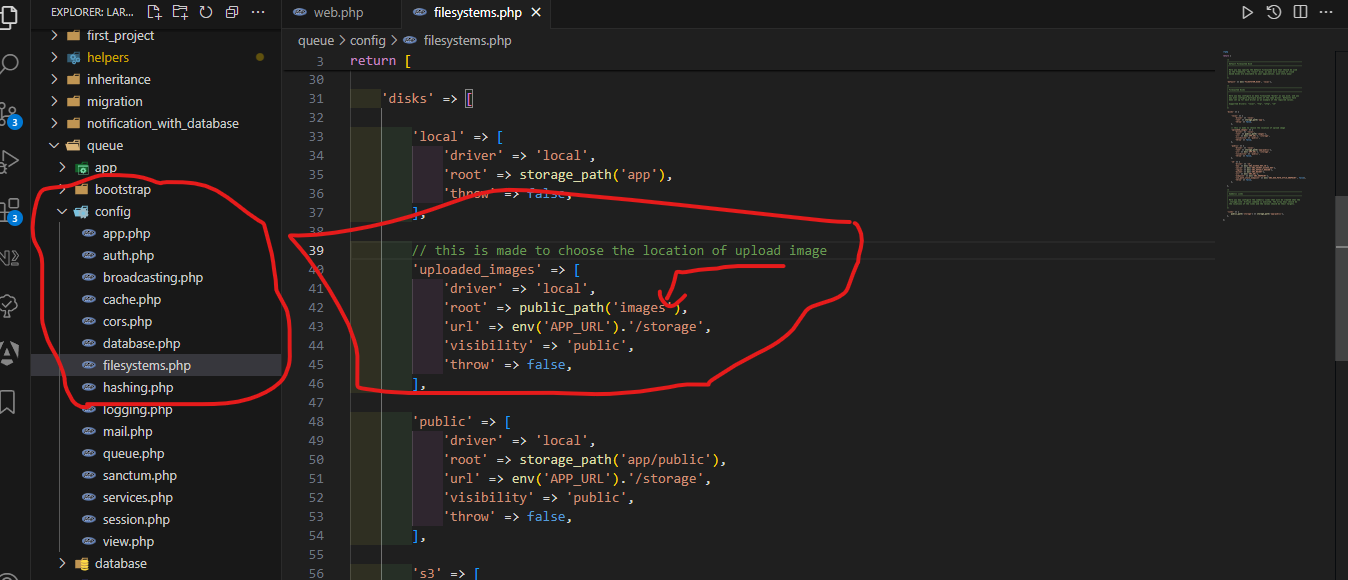
انا بستخدم الحالة دي لما اضيف اكتر من مستخدم او أكون بعمل عمليه هتاخد وقت كبير لحد ماتخلص وهتوقف البرنامج لحد ما العمليه دي تتعمل ذي مثلا بضيف اكتر من 2000 مستخدم ف الموقع بتاعي مش طبيعي ان الموقع يقف ميشتغلش غير لما عمليه الاضافه دي تخلص واننك Jobتضيف 2000 مستخدم ده هياخد وقت كبير ف الحاله دي بنستخدم ال

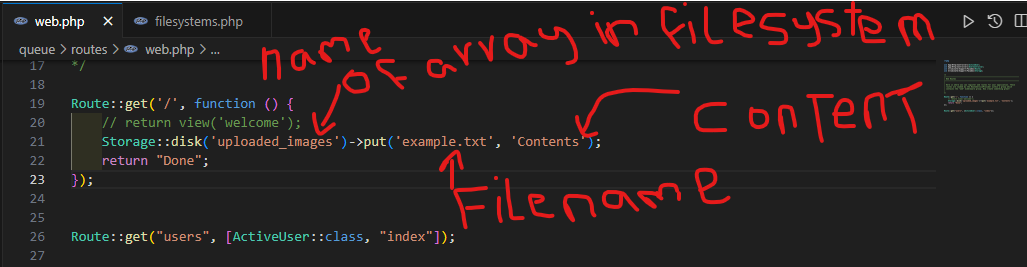
1. .env => change QUEUE\_CONNECTION from ‘ sync ’ to ‘ database ‘
2. Php artisan queue:table => to make Jobs table in database
3. Php artisan migrate
4. Php artisan make:job JobName => to make job
5. Php artisan queue:work => to check the job work or not

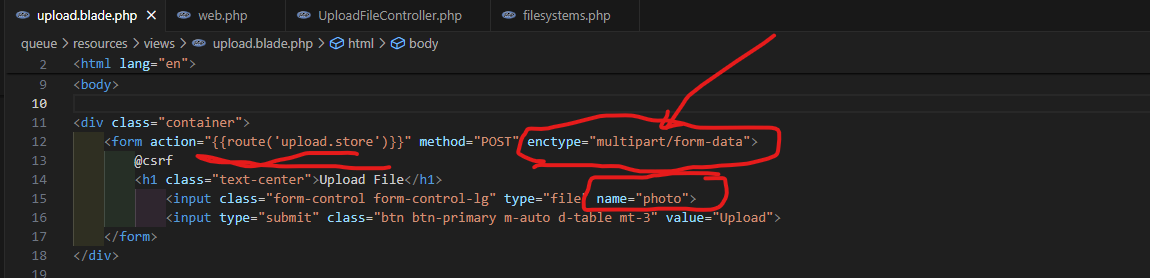


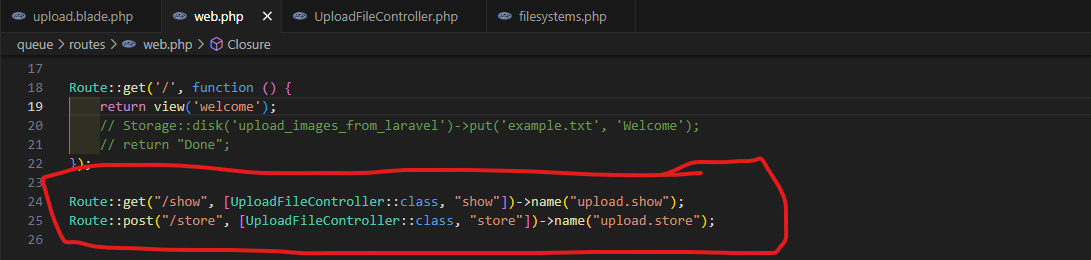
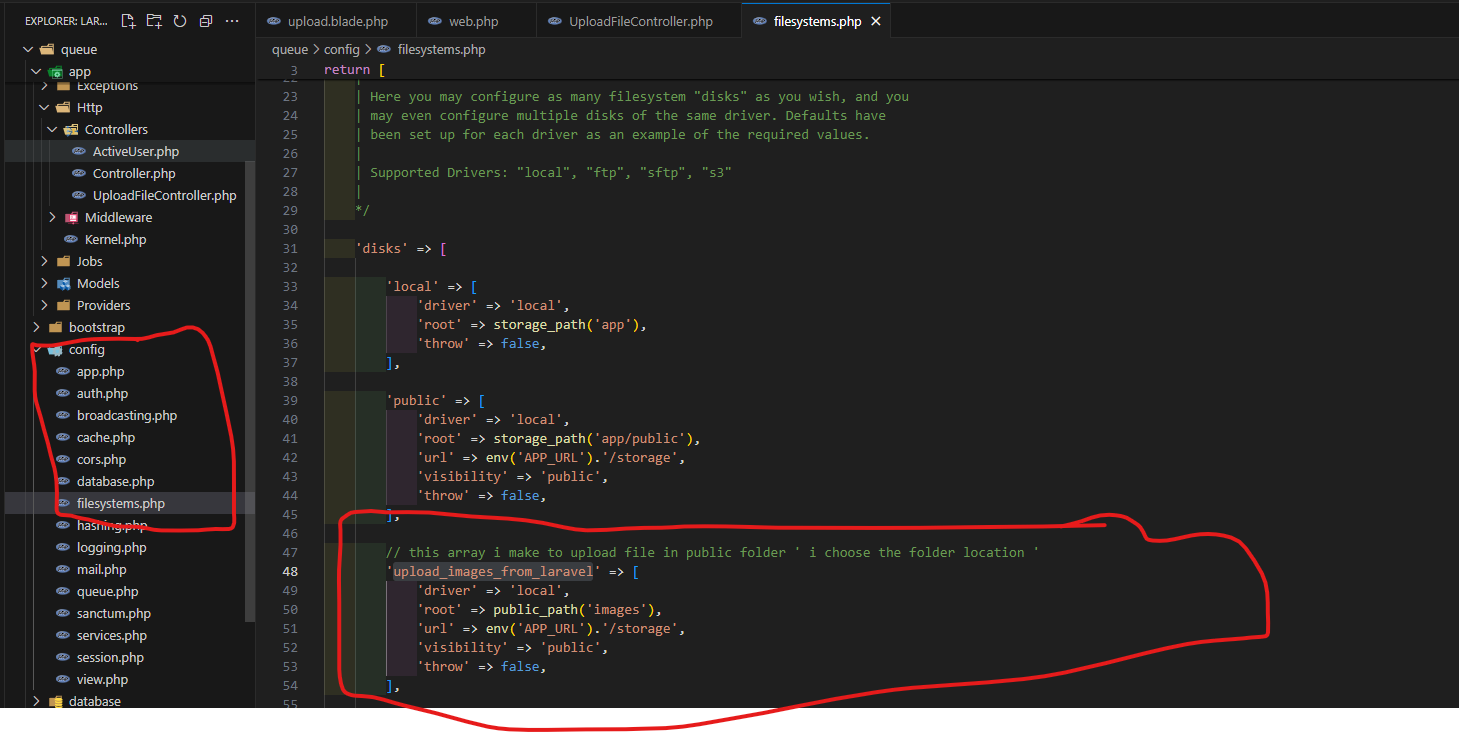
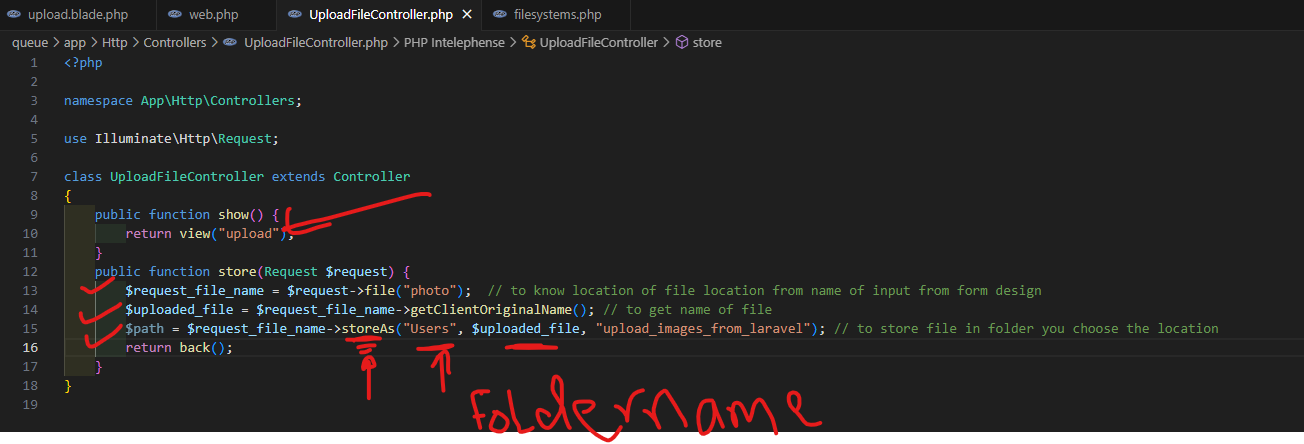
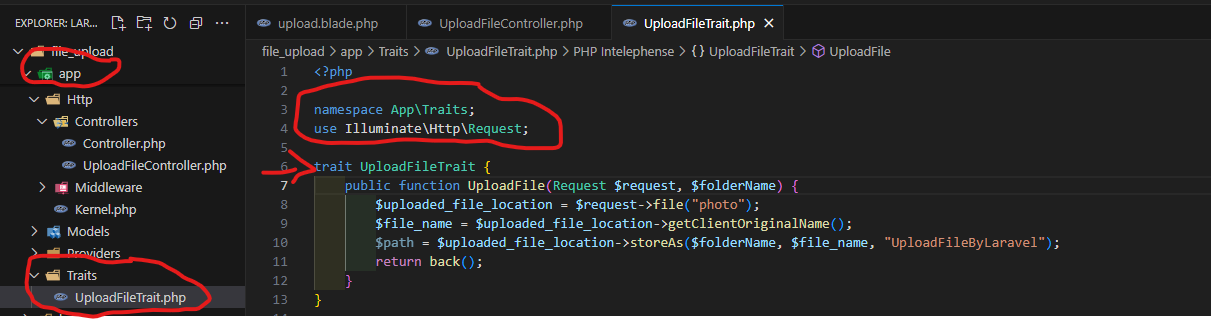


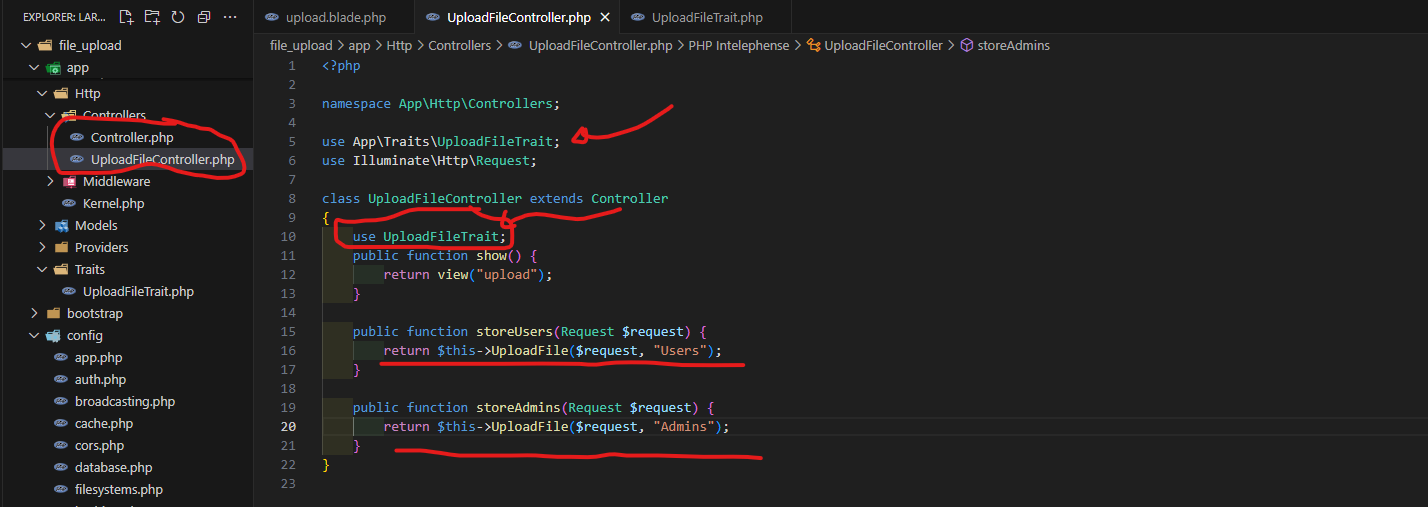
**File storage**

\*-\* علشان اقدر ارفع صورة او ارفع ملف عندي ف المشروع واقدر احدد المكان اللي هيترفع فيه الملفات



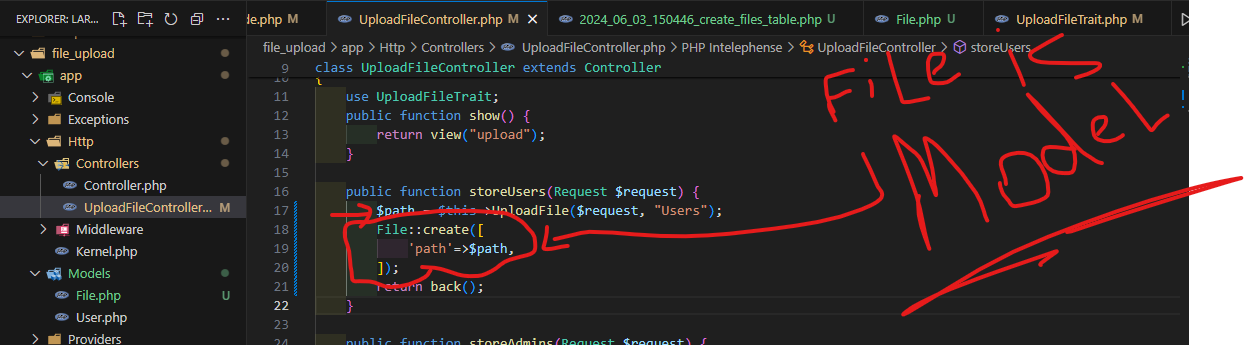
طيب لو عايز ارفع صور للمستخدمينن واحدد انا مكان الصور اللي هتترفع دي تبيق فين ف المشروع \*-\*

  
 في أي مكان في المشروعcall ونعملهم functions ونضيف فيها كل ال trait\*-\* طب ممكن نعمل



\*-\* طب لو انا عايز ارفع الملفات دي علي الداتابيز :-

1- لازم اعمل جدول للبيانات في الداتابيز علشان ارفع البيانات عليه

1. Php artisan make:model Files -m
2. $table=>string(“path”);
3. Protected $fillable[“path”] => model file
4. Php artisan migrate
5. 

\*-\* طب لو عايز اعرض الملفات دي من الداتابيز للمشروع بتاعي اعمل أيه؟