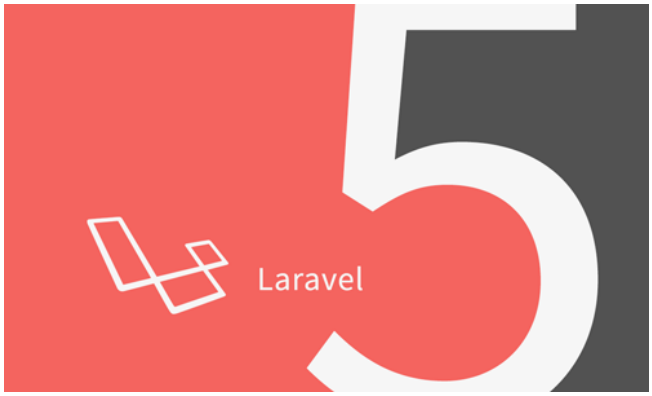


Laravel 5.2 Course

Preparation:

Mohammad Eid Altal

الفهرس

ما هو الـ Laravel ؟ 2

التثبيت والتنصيب 2

تمهيد: 2

الـ Controller: 5

تمرير البيانات: 6

مفهوم الـ master page: 8

ضبط اعدادت الاتصال بالـ Database: 9

ما هو الـ maigration؟ 10

جلب البيانات من قاعدة المعطيات: 14

ما هو الـ Model؟ 15

إضافة بيانات الى الداتابيز: 16

حذف البيانات من الداتابيز: 17

العلاقات بين الجداول: 18

التعديل على البيانات: 22

الـ Validation في الـ Laravel: 25

عملية حذف الـ notes الخاصة بكل page: 28

تسجيل الدخول وتسجيل الخروج والتسجيل في موقع في الـ Laravel: 31

الاذونات (الصلاحيات): 32

إنشاء صفحة المستخدم User Profile: 34

أدوار وصلاحيات المستخدم: 36

إنشاء صفحة الـ Admin: 38

تسجيل الدخول باستخدام الـ Username: 42

رابط مشروع الـ notes على الـ GitHub: 44

رابط مشروع الـ users على الـ GitHub: 44

ما هو الـ Laravel ؟

هو framework مبني على لغة الـ php

التثبيت والتنصيب

1. ندخل الى موقع الـ getcomposer.org
2. نقوم بتحميل برنامج الـ Composer
3. نفتح الـ cmd ونقوم بالدخول الى مجلد المشروع المطلوب
4. ننسخ هذا الأمر

composer create-project --prefer-dist laravel/laravel NAME_PROJECT "5.2.*"

5. ثم ننسخ هذا الأمر php artisan serv
6. نقوم بتشغيل المتصفح ونذهب الى localhost:8000

تمهيد:

ندخل على المجلد الخاص بالمشروع ثم نذهب الى:

app → Http → routes.php

هاد الملف لتوجيه الصفحات

اذا كان عنا هاد الشكل:

```
Route::get('/', function () {
    return view('welcome');
});
```

يعني انا هون ع الصفحة الرئيسية اللي هية: http://localhost:8000/

لو جربنا نغير:

```
Route::get('/home', function () {
    return view('welcome');
});
```

صرت انا بالصفحة هي: http://localhost:8000/home

إعادة شرح للفكرة:

بس يجيني get بالـ url اللي فوق برجعله الصفحة welcome

```
return view('welcome');
```

هي الصفحة welcome وين موجودة؟

resources → views → welcome.blade.php

اللي عرفناه لحد الآن:

من خلال ملف الـ routes.php منحدد الـ route ومنختار الـ link وبنفس الوقت منختار الصفحة اللي لح تنعرض من خلال هاد الـ link

لنفرض عملت أنا route جديد وطلبت صفحة الـ about:

```
Route::get('about', function () {
    return view('about');
});
```

وقت اللي روح على هادا الرابط: http://localhost:8000/about

لح يعطيني error لانو ما عندي صفحة بالـ views → resource اسمها about

منرجع لملف الـ routes.php: لنفرض انا ما بدي ياه يرجعلي view فيني اطلب منه يرجعلي text مثلاً

```
Route::get('about', function () {
    return "this is about page";
});
```

كمان منقدر نرجع متغير او نرجع أي شي حسب حاجتنا.

ملاحظة: وقت انا اطلب انو يرجعلي الصفحة بكفي اكتبله اسم الصفحة يعني مالي مضطر اكتب welcome.blade.php كون نظام الـ Laravel ذكي شوي ومساعد

```
Route::get('home', function () {
    return view('welcome.blade.php');
});
```

ملاحظة: بكفي اكتب اسم الصفحة فقط ولا داعي للكتابة مثل فوق لانو لح يطلعلي error

كيف فيني أعمل صفحة فرعية ؟

مثلاً: عندي انا خمس صفحات اللي هنن:

page1 – page2 – page3 – page4 – page5

بدي حطن بقلب مجلد اسمه pages

كيف بدي استدعيهن؟؟

```
Route::get('page1', function () {
    return view('pages.page1');
});
```

ملاحظة صغيرة: أي كود php بينكتب داخل الـ routes وأي كود HTML بينكتب داخل الـ views

مثال: اذا بدي مرر متغير لصفحة الـ HTML

```
Route::get('home', function () {
    $v1 = "Abo Alsham";
    return view('welcome' , compact('v1'));
});
```

وهيك بستدعيه بصفحة الـ HTML:

```
<div class="content">
    <div class="title">{$v1}}</div>
</div>
```

الـ Controller:

لنفرض انو عندي موقع كبير كتير وعندي صفحات كتير هل من المعقول نكتب كل الأكواد البرمجية ونستدعي ونوجه الصفحات اللي بالموقع كلن بصفحة وحدة اللي هية routes.php؟

من هون اجت فكرة الـ controller وهو مكان مخصص للأكواد البرمجية ومنخلي صفحة الـ routes.php مخصصة فقط لتوجيه الصفحات

هيك بيتقسم الشغل وبيتنظم أكتر وتصبح آلية إصلاح واكتشاف الأخطاء أسهل

في أي مكان توجد صفحة الـ Controller؟

app → Http → Controllers → Controller.php

هاد ملف الـ Controller أساسي بيجي مع النسخة

فيني أنشئ controller خاص فيني انا من خلال الـ cmd عن طريق التعليمة التالية:

```
php artisan make:controller NAME_OF_CONTROLLER
```

وسوف يتم انشاء الـ controller

```
E:\laravel\5.2\blog>php artisan make:controller PageController
Controller created successfully.
```

نلاحظ أنه تم انشاء صفحة:

app → Http → Controllers → PageController.php

كيف منستفاد من الـ controller اللي أنشئناه ؟

منروح على ملف الـ routes.php

```
Route::get('home', 'PageController@p1');
```

هيك انا قلتله انو يستدعي الدالة اللي اسمها p1 الموجودة بالـ PageController

وبالدالة بكتب الأكواد اللي بدي ياها انا

```
public function p1(){
    return view('welcome');
}
```

هيك منكن نظمنا الشغل اكتر وقسمناه وخصصنا صفحة الـ routes.php فقط لتوجيه الصفحات

تمرير البيانات:

منرجع لتمرير البيانات:

متل ما اتفقنا انو بقدر مرر أي شي بدي ياه للـ view متل متغيرات او مصفوفات

لح ناخد مثال صغير: عنا هي المصفوفة وبدنا نمررها للـ view

```
Route::get('about', function () {

    $mobiles = ['apple' , 'samsung' , 'sony'];
    return view('about');
});
```

في عنا طريقتين للتمرير:

الطريقة الأولى: (الطريقة العادية)

```
Route::get('about', function () {

    $mobiles = ['apple' , 'samsung' , 'sony'];
    return view('about' , ['mobiles' => $mobiles]);
});
```

الطريقة الثانية:

```
Route::get('about', function () {

    $mobiles = ['apple' , 'samsung' , 'sony'];
    return view('about' , compact('mobiles'));
});
```

خلينا نشوف كيف بدنا نعرض هي المصفوفة داخل صفحة الـ about:

كمان عنا طريقتين:

الطريقة الأولى: (طريقة php العادية)

```php
<?php
    foreach($mobiles as $mobile){
        echo $mobile ."<br>";
    }
?>
```

الطريقة الثانية: طريقة blade

ملاحظة: Blade هو محرك القوالب الخاص بـ Laravel

```blade
@foreach($mobiles as $mobile)

    {{$mobile}} <br>

@endforeach
```

طريقة الـ blade بتريحنا من الاقواس وبتسهل علينا الشغل كتير

كيفية كتابة باقي الـ statement بنظام الـ blade:

```blade
@if ()

@endif


@for ()

@endfor

@while ()

@endwhile
```

كيفية تمرير أكتر من متحول للـ view:

```
Route::get('about', function () {

    $mobiles = ['apple' , 'samsung' , 'sony'];
    $num = 1000;
    return view('about' , compact('mobiles' , 'num'));
});
```

مفهوم الـ master page:

متل ما منعرف انو صفحات الويب التابعة لموقع واحد بتشترك بنفس الـ header والـfooter والـ sidebar أحياناً

لهيك نحنا وقت يكون عنا كزا صفحة مافي داعي نكتب الـ header والـfooter والـ sidebar لكل صفحة منعملها مو منطق!!

منعمل صفحة فيها الشغلات المشتركة بين الصفحات ومنسميها مثلاً master.blade.php

```
<!DOCTYPE html>
<html>
    <head>
        <title>Laravel</title>
    </head>
    <body>

        <h1>This is headre</h1> <hr>

            @yield('content')

        <hr>
        <h1>This is footer</h1>
    </body>
</html>
```

هلق متل ما اتفقنا انو الكود اللي فوق مشترك بين جميع الصفحات

المقصود بـ `@yield('content')` انو عرّفت قسم جديد لح يحوي شغلات متغيرة بين كل صفحة وصفحة ((يعني بالمشرمحي الشغلات اللي عم تتغير))

لنفرض انا بدي اعمل صفحة جديدة، كيف فيني استفيد من الكود اللي فوق وحط الشغلات الخاصة فيني بقسم الـ content؟؟

```
@extends('master')

@section('content')

    mohammad altal

@stop
```

هاد الكود الخاص بالصفحة الجديدة اللي عملتا

اول الشي استدعيت صفحة الـ master

تاني شي قلتله عند الـ section اللي اسمه content كتبلي mohammad altal

تالت شي سكرت قسم الـ content

<u>ملاحظة: ممكن يكون عندي أكثر من @yield بصفحة الـ master</u>

ضبط اعدادت الاتصال بالـ Database:

config → database.php

يوجد في هذا الملف عدة أنواع لقواعد المعطيات منها:

pgsql – mysql – sqlite

يأخذ mysql افتراضي للداتا بيز

```
'default' => env('DB_CONNECTION', 'mysql'),
```

نقوم بضبط اعدادات الداتابيز من هنا:

نقوم بتحديد اسم الداتابيز واسم المستخدم وكلمة المرور

```
'mysql' => [
    'driver' => 'mysql',
    'host' => env('DB_HOST', 'localhost'),
    'port' => env('DB_PORT', '3306'),
    'database' => env('DB_DATABASE', 'forge'),
    'username' => env('DB_USERNAME', 'forge'),
    'password' => env('DB_PASSWORD', ''),
    'charset' => 'utf8',
    'collation' => 'utf8_unicode_ci',
    'prefix' => '',
    'strict' => false,
    'engine' => null,
],
```

بعد تظبيط الاعدادات هلق بدي أنشئ الجداول الخاصة فيني

بدايةً منروح ع المجلد اللي اسمه

database → migrations

ما هو الـ maigration؟

هو أي عملية منعملها على الداتابيز ، بشكل افتراضي عاملي هو 2 migrations واحد للمستخدمين اذا بدي حط مستخدمين عندي بالموقع وواحد لإعادة تعيين كلمة السر في حال نسي المستخدم كلمة السر الخاصة فيه

كيف فيني أنشئ migration خاص فيني؟

```
C:\note-laravel5.2\blog>php artisan make:migration create_pages_table --create=pages
Created Migration: 2018_08_08_001151_create_pages_table
```

من خلال التعليمة السابقة تم انشاء migration خاص فيني اسمه pages يعني بالمختصر تم انشاء هيكلية لجدول اسمه pages

لكن ملاحظة جداً ضرورية: لا يتم انشاء الجدول في الداتابيز لانو من خلال التعليمة السابقة انشأنا فقط الهيكلية تبع الجدول

منروح ع مجلد الـ migrations منلاقي الملف اللي انشأناه عن طريق الـ cmd

```php
<?php

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreatePagesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('pages', function (Blueprint $table) {
            $table->increments('id');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::drop('pages');
    }
}
```

بتابع الـ up هون منحدد الـ properties تبع الجدول (واصفات الجدول)

بشكل افتراضي عنا حقل ال id وحقل الـ timestamp بيعطي زمن انشاء وزمن اخر تعديل على الجدول

لإنشاء حقل جديد (عمود) بهي الطريقة:

```
public function up()
{
    Schema::create('pages', function (Blueprint $table) {
        $table->increments('id');
        $table->string('title');
        $table->timestamps();
    });
}
```

بهي الطريقة عملنا حقل جديد اسمه title من النمط string.

لحد هي اللحظة متل ما قلنا انا ماعندي شي موجود بالداتابيز كل هاد هيكلية فقط لا غير لانو فعلياً نحنا انشأنا الـ migration الخاص بالجدول فقط يعني ما انشأنا الجدول

طيب كيف فيني فعّل الـ migration ويطلع عندي الجدول بالداتابيز؟؟

عن طريق الـ command line مرة تانية

```
C:\note-laravel5.2\blog>php artisan migrate

 [PDOException]
 SQLSTATE[HY000] [1045] Access denied for user 'homestead'@'localhost' (using password: YES)
```

طلعلي error كيف فيني حل الـ error ؟؟؟

مندخل على ملف الـ env. ومنظبط الاعدادات كمان

```
APP_ENV=local
APP_DEBUG=true
APP_KEY=base64:YRU8zBQv8JTzRA3dEkmCecn15NHgqoOHbHAvsvn6MmA=
APP_URL=http://localhost

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=homestead
DB_USERNAME=homestead
DB_PASSWORD=secret

CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

منغير اسم الداتابيز واسم المستخدم وكلمة السر ومنرجع منفذ التعليمة بالـ cmd

```
C:\note-laravel5.2\blog>php artisan migrate
Migration table created successfully.
Migrated: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrated: 2018_08_08_001151_create_pages_table
```

وهيك بيكونوا صاروا الجداول عندي بالداتابيز

جلب البيانات من قاعدة المعطيات:

ان عملية جلب البيانات من الداتابيز بسيطة جداً

```
use DB;
class PageController extends Controller
{
    //
    public function show(){
        $pages = DB::table('pages')->get();
        return view('pages.show' , compact('pages'));
    }
}
```

استدعينا كلاس DB ومن خلاله منقدر نستخدم الـ methods الموجودة فيه

في المثال السابق: عرفنا متحول اسمه $pages واسندنا له بيانات الجدول اللي اسمه pages

ومررنا المتحول للصفحة اللي اسمها show الموجودة بقلب مجلد الـ pages

هلق هيك منروح على صفحة show ومنعرض البيانات كالتالي:

```
@extends('master')

@section('content')

@foreach($pages as $page)

 {{ $page->title }}
 <br>

@endforeach

@stop
```

عرضنا حقل الـ title الموجود في الجدول pages

ما هو الـ Model؟

الـ model هو عبارة عن class من خلاله منقدر نربط الجداول مع بعض ، نعمل علاقات بين الجداول ، وايضاً نعطي صفات معينة لحقول أو متغيرات

كما يفيد الـ model في موضوع الحماية والأمن

في الوضع الطبيعي يشير الـ model الى الـ table بمعنى أنه كل model يقابل table في الداتابيز

مثلاً عندي جدول الـ pages بالداتابيز يقابله model اسمه page

عادةً: لما يكون عندي الجدول اسمه pages بيكون عندي الـ model اسمه page يعني مناخد الاسم المفرد منه - هيك العادة بس مانو شرط أساسي -

كيفية إنشاء model عن طريق الـ command line؟

```
C:\note-laravel5.2\blog>php artisan make:model Page
Model created successfully.
```

نلاحظ أنه تم انشاء ملف اسمه Page.php بداخل مجلد الـ app

بهاد المشروع لح نتعلم نساوي تطبيق للملاحظات:

1. إضافة ملاحظات
2. حذف ملاحظات
3. تعديل ملاحظات
4. كل ملاحظة تابعة لصفحة معينة
5. الصفحة تملك title
6. يوجد في الصفحة عدة ملاحظات

إضافة بيانات الى الداتابيز:

ملاحظة صغيرة: غالباً لما بدي ابعت بيانات عن طريق الـ form لازم تكون الدالة بملف الـ route هية post مو get

```
Route::post('pagesstore', 'PageController@store');
```

عند عملية الإضافة انا لح استدعي الـ page اللي اسمها pagesstore من خلال الـ function اللي اسمها store الموجود في الـ PageController

داخل الـ function اللي اسمه store لح استقبل البيانات من الـ form وخزنها بالداتابيز

```
public function store(Request $request)
{
    $page = new Page;

    $page->title = $request->title;

    $page->save();

    return back();
}
```

- البارمتر $request هو متحول يحوي البيانات اللي لح اخدها من الـ form
- نعرف object من نمط Page ((اللي هو الـ model اللي عملناه قبل شوي))
- نخزن في الحقل اللي اسمه title قيمة الـ input اللي اسمه title اللي جاية من الـ form
- الدالة save() لحفظ البيانات في الداتابيز
- بعد ما انتهي من عملية التخزين نرجع للصفحة اللي كنا فيها
- طبعاً هون نحنا عرفنا كائن من النمط Page منستخدم use Page كالتالي:

```
use App\Page;
```

منروح على الـ form:

```
<div class="row list-group-item">
    <form method="POST" action="pagesstore">
    {{ csrf_field() }}
        <div class="input-group">
          <input type="text" name="title" class="form-control" placeholder="Add Page . . .">
          <span class="input-group-btn">
          <button class="btn btn-default" type="submit">Add</button>
          </span>
        </div>
    </form>
</div>
```

- الـ method من نمط post
- الـ action رايح لصفحة الـ pagesstore
- اجباري لازم نحط {{ csrf_field() }} بقلب كل form مشان الحماية والأمان وما بيشتغل الـ form اذا ما حطيناها

حذف البيانات من الداتابيز:

بدايةً منعمل route خاص بالـ delete

```
Route::get('pages/{page}/delete', 'PageController@delete');
```

طبعاً لحتى نعمل delete لازم نحدد الـ page اللي بدنا نحذفا مو نحذف كل الـ pages اللي عنا منحدده عن طريق {page} وهلق منشوف كيف مناخده من الـ view

```
<div><a href="pages/{{ $page->id }}/delete" class="btn btn-danger pull-right">Delete</a> </div>
```

هلق ضل انو نعمل دالة الـ delete:

```
    public function delete(Page $page)
  {

    $page->delete();
    return back();
}
```

منمرر بارامتر من نمط الكلاس Page وبيكون اسم الكائن $page حصراً نفس الاسم اللي مررناه بالـ route

العلاقات بين الجداول:

يوجد عنا بالتطبيق جدول واحد فقط ولحتى نعمل علاقة لازم ع الأقل يكون عنا جدولين

بدايةً لح نساوي جدول خاص بالـ notes

```
C:\note-laravel5.2\blog>php artisan make:migration create_notes_table --create=notes
Created Migration: 2018_08_09_011643_create_notes_table
```

لح نضيف الحقول المناسبة:

```
public function up()
{
    Schema::create('notes', function (Blueprint $table) {
        $table->increments('id');
        $table->integer('page_id')->unsigned();
        $table->text('text');
        $table->timestamps();
    });
}
```

حقل اسمه text من النمط text يحتوي على المحتوى الخاص بالـ note

حقل اسمه page_id من النمط integer وأيضاً لازم يكون unsigned يعني الرقم غير سلبي وهاد الحقل لربط الجدولين مع بعض

هلق لازم نعمل migrate للداتابيز:

```
C:\note-laravel5.2\blog>php artisan migrate
Migrated: 2018_08_09_011643_create_notes_table
```

هلق فعلياً عنا جدول بالداتابيز اسمه notes

الخطوة التالية ← لازم نعمل model خاص بجدول الـ notes متل ما اتفقنا

```
C:\note-laravel5.2\blog>php artisan make:model Note
Model created successfully.
```

لحتى نكوّن العلاقة منعمل التالي:

بدايةً منروح عالـ model الخاص بالـ Page ومنكتب التالي:

```
namespace App;

use Illuminate\Database\Eloquent\Model;

class Page extends Model
{
    public function notes(){
        return $this->hasMany(Note::class);
    }
}
```

نحنا عنا بكل صفحة في عدة ملاحظات

هيك نحنا فهّمناه هاد الشي من خلال الـ hasMany

هلق لازم نروح نكوّن العلاقة من الطرف التاني (من طرف الـ notes)

```
class Note extends Model
{
    public function pages(){
        return $this->belongsTo(Page::class);
    }
}
```

معنى هاد الحكي انو كل note تابعة لـ page وحدة وفهّمناه هاد الحكي من خلال belongsTo

هيك انتهينا من الربط بين الجدولين

<u>خلينا نطبق هالحكي عملياً:</u>

بدنا نعمل صفحة تعرضلنا كل الـ note الخاصين بـ Page معينة

منروح على ملف الـ routes.php

```
Route::get('pages/{page}', 'PageController@onepage');
```

هلق منروح على الـ PageController ومنضيف الـ method اللي اسمها onepage

```
public function onepage(Page $page)
{
    return view('pages.onepage' , compact('page'));
}
```

بقلب الـ method منمرر بارامتر اسمه من نفس الاسم اللي مرقناه بالـ route هاد البارامتر بيحوي كلشي بيانات خاصة بالـ page بالإضافة لكل الـ notes التابعين لنفس الـ page

طبعاً نحنا عاملين صفحة خاصة بالـ view اسمها onepage هي خاصة بعرض الـ notes الخاصة بكل page

الكود الخاص بصفحة الـ onepage.blade.php:

```
1  @extends('master')
2
3  @section('content')
4
5  <span><b><a href="#">Back to Pages List ></a></b></span>
6
7  <div class="row list-group-item-info page-title">
8      <div class="col-xs-12">
9       {{ $page->title }}
10     </div>
11 </div>
12
13
14 @foreach($page->notes as $note)
15 <div class="row list-group-item">
16     <div class="col-xs-8">
17      {{ $note->text }}
18     </div>
19     <div class="col-xs-4">
20         <button type="button" class="btn btn-danger pull-right">Delete</button>
21         <button type="button" class="btn btn-default pull-right">Edit</button>
22     </div>
23 </div>
24 @endforeach
```

<u>شرح بسيط:</u>

- في السطر 9 عرضنا الـ title الخاص بالصفحة
- في السطر 14 عملنا حلقة ومرقنا على كل الـ notes الخاصة بالصفحة وطبعنا الـ text الخاصة بكل note

الآن سوف نقوم بعملية اضافة الـ notes متل ما قمنا بعملية اضافة الـ Pages

نقوم بعمل Controller جديد خاص بالـ notes مشان نفصل الدوال الخاصة بالـ pages عن الدوال الخاصة بالـ notes

```
E:\laravel\note-laravel5.2\blog>php artisan make:controller NoteController
Controller created successfully.
```

نقوم بإنشاء route جديدة خاصة بتخزين الـ notes:

```
Route::post('pages/{page}/notesstore', 'NoteController@store');
```

نلاحظ انو مررنا متحول page بالرابط

الآن نقوم بإنشاء دالة الـ store الخاصة بالـ notes:

```
public function stroe(Request $request, Page $page)
{
    $note = new Note;
    $noe->text = $request->text;
    $page->notes()->save($note);
    return back();
}
```

بما انه استخدمنا كلاس الـ Page وكلاس الـ Note نقوم باستدعائهم:

```
use App\Page;

use App\Note;
```

لازم نتأكد من انو الـ Form ظابط 100%:

```
<div class="row list-group-item">
    <form method="POST" action="../pages/{{ $page->id }}/notesstore">
      {{ csrf_field() }}
        <div class="input-group">
          <input type="text" name="text" class="form-control" placeholder="Add
          Note . . .">
          <span class="input-group-btn">
            <button class="btn btn-default" type="submit">Add</button>
          </span>
        </div>
    </form>
</div>
```

التعديل على البيانات:

متل ما منعرف نحنا عنا ملاحظات وعنا زر التعديل Edit لح نقوم بتفعيل هادا الزر هلق..

بدنا ننتبه لشغلة كتير مهمة واللي هية: اذا بدنا نعمل route للـ Edit لازم نعمل 2 Routes

1. الـ Route الأول هو اللي بينقلنا لصفحة الـ Edit اللي بتحتوي على Form
2. الـ Route التاني هو اللي لح ينفذ عملية الـ Edit ويعمل Update ع الداتابيز

```
Route::get('note/{note}/edit', 'NoteController@edit');

Route::post('note/{note}/update', 'NoteController@update');
```

منلاحظ انو الـ route الأول من نمط get لانو يقوم بعرض صفحة تحتوي على form فقط

بينما الـ route الثاني post لانو لح نستقبل بيانات ونخزن بالداتابيز

هلق لح نقوم بانشاء الصفحة الخاصة بالتعديل:

```
@extends('master')

@section('content')

<div class="row list-group-item-info page-title">
    <div class="col-xs-12">
      Edit Note..
    </div>
</div>

  <div class="row list-group-item">
      <form method="POST" action="#">
      {{ csrf_field() }}
          <div class="input-group">
            <input type="text" name="text" class="form-control" placeholder="Add
            Page . . .">
            <span class="input-group-btn">
            <button class="btn btn-default" type="submit">Edit</button>
            <a href="#" class="btn btn-danger">Cancel</a>
            </span>
          </div>
      </form>
  </div>

@stop
```

هلق منروح علـ Controller ومنكتب الـ Function اللي لح تعرضلنا الصفحة الخاصة بالـ Edit

```
public function edit(Note $note)
{
    return view('notes.edit' , compact('note'));
}
```

كمان منكتب الـ Function الخاصة بالـ Update:

```
public function update(Request $request, Note $note)
{
    $note->update($request->all());
    return redirect('pages/'. $note->page_id);
}
```

بقي شغلة وحدة واللي هية:

منروح ع كلاس الـ Note ومنضيف شي اسمه

وظيفته حماية المدخلات من التعديل بهي الحالة مالح يكون في خطر كبيرة اذا تعدلت Note

مثال آخر: لنفرض في عندي جدول لحسابات الموظفين وفي عندي حقل خاص بالرصيد بهي الحالة لح يكون في عندي خطر اذا عدل هاد الحقل بسهولة

```
class Note extends Model
{
    protected $fillable = ['text'];
```

عبارة عن Array منكتب فيها كل المتغيرات اللي بدنا نحميها وحالتنا هي مافي غير حقل الـ text اللي عم نعدل عليه

طبعاً ما لازم ننسى نظبط الـ Form

هلق منعمل زر الـ Delete الخاص بالـ Note متل ما عملناه قبل بمرة للـ Page بنفس الطريقة

منتجه لملف الـ routes.php ومنعمل route جديد خاص بالـ delete

```
Route::get('note/{note}/delete', 'NoteController@delete');
```

منروح على الـ NoteController ومنعمل الـ Function الخاصة بالـ Delete

```
public function delete(Note $note)
{
    $note->delete();
    return back();
}
```

الـ Validation في الـ Laravel:

في Function خاصة بالـ Validation اسمها validate منستدعيها ومنعرف الشغلات اللي بتحتاج لـ validate ومنحط قواعد الـ validate الخاصة فينا واللي هية موجودة جاهزة

مثلا: عند عملية الإضافة للـ note مابدي ياه يدخلي note فاضية

منروح علـ function الخاصة بالإضافة ومنكتب التالي:

```
public function store(Request $request, Page $page)
{
    $this->validate($request , [
        'text' => 'required'
    ]);
```

طيب لنفرض انو الـ note مابدي ياها تتكون من حرف واحد فقط بدي ياها أقل شي من خمس حروف ، كمان منروح علـ validate ومنكتب التالي:

```
public function store(Request $request, Page $page)
{
    $this->validate($request , [
        'text' => 'required|min:5'
    ]);
```

هلق فعلياً مابقدر ضيف note فاضية وأقل من خمس حروف

طيب فعلياً ندي مشكلة صغيرة واللي هية بدي اعرض الرسالة الخاصة بكل error

الـ Laravel بتخزن كل الأخطاء بمصفوفة اسمها errors

```
@if(count($errors))

  <ul>
    @foreach($errors->all() as $error)
      <li>{{ $error }}</li>
    @endforeach
  </ul>

@endif
```

هيك منكون عرضنا كل رسائل الـ error اللي بتطلع معنا نتيجة الإضافة

لنفرض انا بدي خزّن بدل الملاحظة Email ، بروح على الـ validate وبكتب Email كالتالي:

```
public function store(Request $request, Page $page)
{
    $this->validate($request , [
        'text' => 'required|min:5|email'
    ]);
```

طيب لنفرض اني كتبت انا الملاحظة وكانت أقل من خمس حروف وما انضافت للداتابيز ، معقول وقت اللي بدي ارجع اكتبها بدي اكتبها من أول وجديد ؟؟

في عنا شي بالـ Laravel بقدر اني خزّن القيمة القديمة المكتوبة كالتالي:

```
<input type="text" name="text" value="{{ old('text') }}" class="
form-control" placeholder="Add Note . . .">
```

منلاحظ انو بالـ value قلتله خزنلي لقيمة القديمة المكتوبة دائماً

طيب انا وقت اعرض رسائل الخطأ عم يعرضلي رسائل الخطأ الخاصة بالـ Laravel

السؤال هون هو كيف فيني انا اعمل رسائل error خاصة فيني؟؟

```
public function store(Request $request, Page $page)
{
    $this->validate($request , [
        'text' => 'required|min:5'
    ] ,

    [
        'text.required' => 'Add somthing' ,
        'text.min' => 'the note is short'
    ]

    );
```

هيك لما اكتب note أقل من 5 حروف لح تطلعلي الرسالة التالية:

the note is short

طبعاً الرسائل ممكن تكون بأي لغة ممكن تكون باللغة العربية مافي مشكلة

طبعاً بنهاية قسم الـ Validation لازم نعرف انو في قائمة طويلة للـ validation موجودة على موقع الـ Laravel

عملية حذف الـ notes الخاصة بكل page:

عملياً نحنا وقت نحذف الـ page من عندي الـ note الخاصة فيها عم تضل بالدتابيز ماعم تنحذف

انا هلق بدي اعمل شغلة واللي هية:

- اذا الـ page اللي عندي تملك note ينقلني لصفحة تأكيد الحذف واذا تمت الموافقة على الحذف من قبل المستخدم منحذفها ومنحذف الـ notes الخاصين فيها
- اذا الـ page اللي عندي لا تملك note يحذفها فور

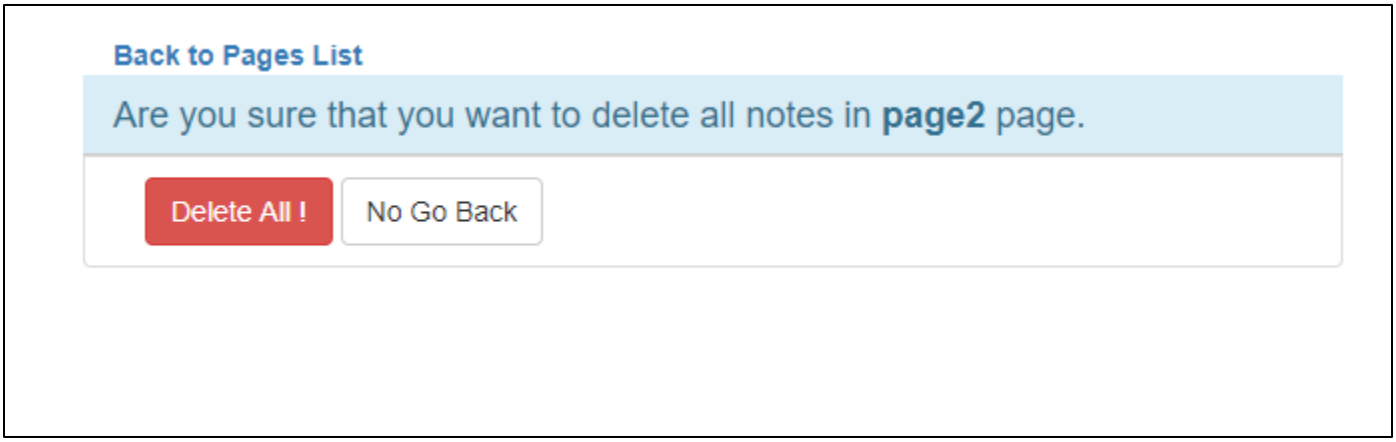
هلق منروح على الـ PageController منعدّل الـ Function الخاصة بحذف الـ Page كالتالي:

```
public function delete(Page $page)
{
    if(count($page->notes))
    {
        return view('pages.deleteall' , compact('page'));
    }
    else
    {
        $page->delete();
        return back();
    }
}
```

صفحة الـ delelteall فيها رسالة تأكيد الحذف:

```
@extends('master')

@section('content')

<span><b><a href="../../pages">Back to Pages List</a></b></span>

<div class="row list-group-item-info page-title">
  <div class="col-xs-12">
    Are you sure that you want to delete all notes in <b>{{ $page->title }}</b>
    page.
  </div>
</div>

<div class="row list-group-item">
  <div class="col-xs-12">
    <a href="../../pages/{{ $page->id }}/deleteall" class="btn btn-danger">Delete
    All !</a>
    <a href="../../pages" class="btn btn-default">No Go Back</a>
  </div>
</div>

@stop
```

الـ view الخاص بالصفحة:



هلق نقوم ببرمجة زر الـ Delete All ! نتجه لملف الـ routes ونقوم بعمل route جديدة خاصة بحذف الـ page مع جميع الـ notes الخاصين فيها:

```
Route::get('pages/{page}/deleteall', 'PageController@deleteall');
```

هلق منروح على الـ PageController ومنكتب الـ Function اللي اسمه deleteall:

```php
public function deleteall(Page $page)
{

        $page->delete();

        $page->notes()->delete();

        return redirect('../../pages');

}
```

- حذفنا الـ page
- حذفنا الـ notes الخاصين فيها
- رجعنا بعد الحذف لصفحة الـ pages

تسجيل الدخول وتسجيل الخروج والتسجيل في موقع في الـ Laravel:

من خلال تعليمة وحدة في الـ Laravel لح ننشئ الداتابيز الخاصة بالـ user area ولح ننشئ الصفحات الخاصة بالـ login والـ logout والـ register كل هاااااد بتعليمة وحدة

لك يا بلاااااش 😂

بدايةً لح ننشئ مشروع جديد ونسميه users-project ونطبق عليه هاد الكلام عملياً

كيف بدنا ننشئ الـ user area ؟

```
E:\laravel\users-project>php artisan make:auth
Created View: E:\laravel\users-project\resources/views/auth/login.blade.php
Created View: E:\laravel\users-project\resources/views/auth/register.blade.php
Created View: E:\laravel\users-project\resources/views/auth/passwords/email.blade.php
Created View: E:\laravel\users-project\resources/views/auth/passwords/reset.blade.php
Created View: E:\laravel\users-project\resources/views/auth/emails/password.blade.php
Created View: E:\laravel\users-project\resources/views/layouts/app.blade.php
Created View: E:\laravel\users-project\resources/views/home.blade.php
Created View: E:\laravel\users-project\resources/views/welcome.blade.php
Installed HomeController.
Updated Routes File.
Authentication scaffolding generated successfully!

E:\laravel\users-project>
```

صار عندي صفحات خاصة الـ login والـ logout والـ register

طبعاً ذا عملنا user جديد بالموقع وسجلنا لح يعطينا خطأ ليششششش؟

لانو متل ما اتفقنا لازم نضبط اعدادات المشروع مع الداتابيز اللي عنا ونربطن مع بعض عن طريق صفحتين اللي هنن .env و database.php

- اول خطوة منروح على الـ phpMyAdmin ومننشئ الداتابيز اللي بدي ياها
- تاني خطوة منضبط الاعدادات
- تالت خطوة منرجع منطفي السيرفر ومنرجع منشغله ومنعمل php artisan serv
- رابع خطوة منعمل migrate للـ tables اللي عندي من خلال تعليمة الـ php artisan migrate

هلق صار فيني سجل بالموقع بشكل نظامي وسجل دخور وخروج وكل الامور تمام عندي

الاذونات (الصلاحيات):

كيف فيني اعطي اذن لل user يدخل ع صفحة معينة اذا كان مسجل دخول وكيف فيني امنعه اذا مالو مسجل دخول؟؟؟

منروح على المشروع منشوف شو انشألنا Laravel

```
Route::get('/', function () {
    return view('welcome');
});

Route::auth();

Route::get('/home', 'HomeController@index');
```

من خلال سطر الـ Route::auth(); استدعالنا الصفحات الخاصة بالتسجيل وتسجيل الدخول والخروج و .... الخ

في عنا صفحة اسمها Home ما بتفتح الا اذا كنا مسجلين دخول

طيب انا كيف فهمت Laravel انو هي الصفحة ما تفتح الا في حال كان المستخدم مسجل دخول؟

منروح على الـ HomeController:

```
8  class HomeController extends Controller
9  {
10     /**
11      * Create a new controller instance.
12      *
13      * @return void
14      */
15     public function __construct()
16     {
17         $this->middleware('auth');
18     }
19
20     /**
21      * Show the application dashboard.
22      *
23      * @return \Illuminate\Http\Response
24      */
25     public function index()
26     {
27         return view('home');
28     }
29 }
30
```

من خلال السطر 17عرفنا اذن خاص بالـ user ممنوع اي حدا تاني يفوت عليه

يعني بمجرد ما استدعي الكلاس اللي اسمه HomeController لح يتنفذ الباني الافتراضي تبعه ويبني اذن خاص بالـ user

معنى هاد الكلام انو اذا بدي اعمل اي صفحة خاصة بالـ user لازم حطا بالـ HomeController او اي Controller تاني يحتوي على السطر 17 هيك بكون منعت باقي الزوار يكون الهن access ع هي الصفحة

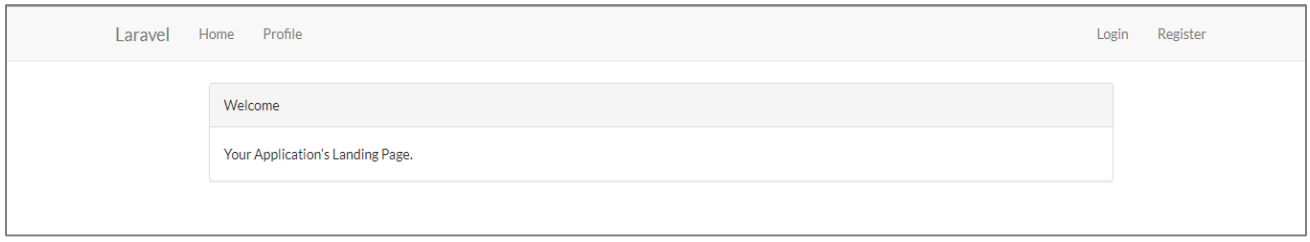
إنشاء صفحة المستخدم User Profile:

لح نتعلم ننشأ صفحة خاصة تحتوي على معلومات الـ user مثل الاسم والايميل و.... الخ

لح نعمل زر بالناف بار نسميه prfile ولإضافة هاد الزر منروح على مجلد الـ layout على صفحة الـ app.blade.php

```
<div class="collapse navbar-collapse" id="app-navbar-collapse">
    <!-- Left Side Of Navbar -->
    <ul class="nav navbar-nav">
        <li><a href="{{ url('/home') }}">Home</a></li>
        <li><a href="{{ url('/profile') }}">Profile</a></li>
    </ul>
```

النتيجة:



منروح على ملف الـ routes.php ومننشئ route جديدة وبما انه هي الصفحة لح تكون خاصة بالمستخدم بقا منعملها دالة بقلب كلاس الـ HomeController اللي بيحتوي بالباني الافتراضي تبعه كود بيمنع اي زئر عادي يفوت ع صفحات محددة

```
Route::get('/profile', 'HomeController@profile');
```

هلق منروح على الـ HomeController ومننشئ الدالة اللي اسمها profile:

```
public function profile()
{
    return view('profile');
}
```

منلاحظ انو عم نقله عرضلي الصفحة اللي اسمها profile.blade.php

هلق لازم ننشأ هي الصفحة اللس اسمها profile:

```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-10 col-md-offset-1">
            <div class="panel panel-default">
                <div class="panel-heading">User Profile</div>

                <div class="panel-body">
                    Username: {{ Auth::user()->name }}
                    <hr>
                    Email:    {{ Auth::user()->email }}
                    <hr>
                    ID:       {{ Auth::user()->id }}
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
```

هلق انا بدي اعمل شغلة احلى واللي هية انا بدي يطلع ر الـ profile بس وقت الواحد يسجل دخول يعني بمعنى اخر مابدي يطلع هاد الزر لما يكون عندي بالموقع زائر عادي

بعمل هيك:

```
47          <div class="collapse navbar-collapse" id="app-navbar-collapse">
48              <!-- Left Side Of Navbar -->
49              <ul class="nav navbar-nav">
50                  <li><a href="{{ url('/home') }}">Home</a></li>
51                  @if (Auth::check())
52                  <li><a href="{{ url('/profile') }}">Profile</a></li>
53                  @endif
54              </ul>
55
56              <!-- Right Side Of Navbar -->
57              <ul class="nav navbar-nav navbar-right">
58                  <!-- Authentication Links -->
59                  @if (Auth::guest())
60                      <li><a href="{{ url('/login') }}">Login</a></li>
61                      <li><a href="{{ url('/register') }}">Register</a></li>
62                  @else
63                      <li class="dropdown">
64                          <a href="#" class="dropdown-toggle" data-toggle="
                            dropdown" role="button" aria-expanded="false">
65                              {{ Auth::user()->name }} <span class="caret"></
                                span>
```

بالسطر 51 الشرط بيدل على انه المستخدم مسجل بالموقع وعامل login

والسطر 59 الشرط بيدل على انه المستخدم زائر عادي غير مسجل او مانو عامل login

نصيحة آخر الفقرة:

يفترض اذا بدي اعمل منطقة المستخدمين لازم اعمل هالشي من البداية لانو في حال كنت بنص مشروعي وقررت انو يكون في بمشروعي مستخدمين وعملت تعليمة اللي هية

php artisan make:auth لح يصير عندي كتير مشاكل لهيك تجنباً للمشاكل بعمل هالشي من البداية

أدوار وصلاحيات المستخدم:

لح يكون عنا بالمشروع 3 مستويات واللي هنن:

- Admin ورقمه 1
- Manager ورقمه 2
- User ورقمه 3

ملاحظة صغيرة: الطريقة اللي لح نستخدمها هلق مو هية الطريقة الرسمية ولا الطريقة المتعارف عليها بشكل عام بس هية طريقة فعّالة وشغالة تمام

كيف بدنا نبلش؟؟

أول الشي لازم نضيف عمود جديد على جدول الـ user نسميه role ومن خلال قمته منقدر نعرف اذا المستخدم Admin أو Manager أو User

لحتى نضيف عمود لازم نعمل Migration بهالشي كالتالي:

```
E:\laravel\users-project>php artisan make:migration add_role_to_users --table="users"
Created Migration: 2018_08_23_004758_add_role_to_users
```

منروح على الـ Migration اللي أنشاناها ومنكتب اسم العمود الجديد اللي بدنا نضيفه

```php
public function up()
{
    Schema::table('users', function (Blueprint $table) {
        $table->integer('role');
    });
}
```

هلق المفروض نعمل migrate للمشروع تبعنا كالتالي:

```
E:\laravel\users-project>php artisan migrate
Migrated: 2018_08_23_004758_add_role_to_users
```

هلق لازم ننشأ زر للـ Admin و الـ Manager ومنسميه Control ومنخليه يتحكم من خلال هاد الزر بالموقع:

- منروح على الـ master page ومنضيف هاد الزر مع شرط انو يكون الـ user هو Admin أو Manager كالتالي:

```blade
<div class="collapse navbar-collapse" id="app-navbar-collapse">
    <!-- Left Side Of Navbar -->
    <ul class="nav navbar-nav">
        <li><a href="{{ url('/home') }}">Home</a></li>
        @if (Auth::check())
        <li><a href="{{ url('/profile') }}">Profile</a></li>
            @if (Auth::user()->role <= 2)
            <li><a href="{{ url('/Control') }}">Control</a></li>
            @endif
        @endif
    </ul>
```

- منعمل الـ route الخاص بالـ Control

```php
Route::get('/control', 'HomeController@control');
```

- منروح على الـ HomeController ومنساوي الـ Method اللي اسمه Control

```php
public function control()
{
    if(Auth::user()->role == 3)
    {
        return redirect('/');
    }
    return view('control');
}
```

المقصود بالشرط انو وقت يكون المستخدم عادي حوّلي ياه ع الصفحة الرئيسية لا تخليلي ياه يفوت ع صفحة الـ control من خلال الرابط http://localhost:8000/control

- منروح منعمل صفحة الـ control.blade.php

إنشاء صفحة الـ Admin:

بالنسبة للـ Admin لح نعرض بالصفحة جميع الـ users ونخليه يقدر يغير صلاحياتهم ويعدّل اللي بده ياه

بالنسبة للـ Manager لح نخليه بس يشوف الـ users بدون ما يقدر يعدّل على الصلاحيات

هلق منروح على صفحة الـ control ومنبلش شغل:

طبعاً منصمم table بسيط بالـ HTML كل row فيه بيحتوي على الـ ID و Name و Email و Rule

هلق منروح على الـ method الخاصة بالـ control:

بقلب هي الـ method منجيب كل الـ users من الداتابيز ومنمررهن لصفحة الـ control كالتالي:

```
public function control()
{
    if(Auth::user()->role == 3)
    {
        return redirect('/');
    }

    $users = DB::table('users')->get();
    return view('control' , compact('users'));
}
```

طبعاً لازم استدعي الـ DB بملف الـ HomeController:

```
use DB;
```

بعدين منروح على صفحة الـ control.blade.php ومنعمل حلقة مشان نطبع كل الداتا اللي جايينتا عن طريق المتغير اللي اسمه users

هلق بالنسبة لتغيير الصلاحيات عنا form بهاد الشكل بدي كل ما غير الـ rule تنعمل refresh للصفحة وتاخد قيمة جديدة:

```
<form method="post">
  <select class="form-control" name="role" on
  change="this.form.submit()">
    <option value="1" {{ ($user->role == 1) ?
    'selected' : null }}>Admin</option>
    <option value="2" {{ ($user->role == 2) ?
    'selected' : null }}>Manager</option>
    <option value="3" {{ ($user->role == 3) ?
    'selected' : null }}>User</option>
  </select>
</form>
```

هلق منعمل route خاصة بالتعديل مع تمرير id الـ user المراد تعديله كالتالي:

```
Route::post('/update-role/{user}', 'HomeController@updateRole');
```

منروح على الـ HomeController منعمل الـ method اللي اسمها updateRole:

```
public function updateRole(Request $request , User $user)
{
    $user->update($request->all());

    return redirect('control');
}
```

وما منسى نستدعي الـ user فوق:

```
use App\User;
```

كمان لازم نحمي الحقل rule كالعادة:

```
protected $fillable = [
    'name', 'email', 'password', 'role' ,
];
```

هلق منظبط الـ form:

```
<form method="post" action="/update-role/{{ $user->id }}">
  {{ csrf_field() }}
  <select class="form-control" name="role" onchange="this.form.submit()">
    <option value="1" {{ ($user->role == 1) ? 'selected' : null }}>Admin</
    option>
    <option value="2" {{ ($user->role == 2) ? 'selected' : null }}>Manager
    </option>
    <option value="3" {{ ($user->role == 3) ? 'selected' : null }}>User</
    option>
  </select>
</form>
```

هلق متل ما اتفقنا انو الـ admin والـ manager بيقدرو يوصلوا لصفحة الـ control بس انا بدي الـ manager ما يقدر يعدّل على الصلاحيات بدي بس الـ admin يعدّل على الصلاحيات فقط بقا منضيف هاد الشرط البسيط:

```
@if(Auth::user()->role == 2)
<b>Disabled</b>
@else
    <div class="form-group" style="margin-bottom: 0px;">
        <form method="post" action="/update-role/{{ $user->id }}">
            {{ csrf_field() }}
            <select class="form-control" name="role" onchange="this.form.submit()
            ">
              <option value="1" {{ ($user->role == 1) ? 'selected' : null
              }}>Admin</option>
              <option value="2" {{ ($user->role == 2) ? 'selected' : null
              }}>Manager</option>
              <option value="3" {{ ($user->role == 3) ? 'selected' : null }}>User
              </option>
            </select>
        </form>
    </div>
@endif
```

كمان لازم الغي تعديل صلاحية الـ user اللي الـ id تبعه 1 لانه اذا بدي عدّل صلاحية اول user عندي وساويه غير الـ Admin ماعاد يضل عندي حدا بالموقع Admin وماعاد اقدر عدّل على اي user بدي ياه لهيك بلغي تعديل صلاحية أول user

```
@if(Auth::user()->role == 2 || $user->id == 1)
```

هلق المشكلة التالتة والاخيرة هية كيف فيني أعطي role لـ user مسجل جديد عندي بالموقع:

app→Http→Controllers→Auth→AuthController.php

ومننزل لتحت منضيف حقل الـ role منحط قيمته الافتراضية 3 كالتالي:

```
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => bcrypt($data['password']),
        'role' => '3',
    ]);
}
```

هيك بيكون كل user بيسجل جديد بياخد الـ role الخاص فيه رقم 3

تسجيل الدخول باستخدام الـ Username:

كيف فينا نعمل login باستخدام الـ username بدلاً من الـ Email؟

- أول الشي لازم نعمل عمود بجدول الـ users ونسميه username بدايةً منعمل migration:

```
E:\laravel\users-project>php artisan make:migration add_username_to_users --table="users"
Created Migration: 2018_08_23_201932_add_username_to_users
```

- بعدين مننشأ العمود في الـ migration كالتالي:

```
public function up()
{
    Schema::table('users', function (Blueprint $table) {
        $table->string('username')->unique();
    });
}
```

- بعدين منعمل migrate للداتابيز:

```
E:\laravel\users-project>php artisan migrate
```

- منروح على صفحة الـ view الخاصة بالـ login ومنعدّل منشيل كل email ومنحط بداله username كالتالي:

```
<div class="form-group{{ $errors->has('username') ? ' has-error' : '' }}">
    <label for="username" class="col-md-4 control-label">Username</label>

    <div class="col-md-6">
        <input id="username" type="text" class="form-control" name="username"
        value="{{ old('username') }}">

        @if ($errors->has('username'))
            <span class="help-block">
                <strong>{{ $errors->first('username') }}</strong>
            </span>
        @endif
    </div>
</div>
```

- منروح على صفحة الـ view الخاصة بالـ register ومنضيف مكان خاص لإضافة الـ username كالتالي:

```
<div class="form-group{{ $errors->has('username') ? '
has-error' : '' }}">
    <label for="username" class="col-md-4 control-label">
    Username</label>

    <div class="col-md-6">
        <input id="username" type="text" class="
        form-control" name="username" value="{{
        old('username') }}">

        @if ($errors->has('username'))
            <span class="help-block">
                <strong>{{ $errors->first('username') }}</
                strong>
            </span>
        @endif
    </div>
</div>
```

- هلق منروح على الـ AuthController ومنظبط:

```
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => 'required|max:255',
        'username' => 'required|max:50|unique:users',
        'email' => 'required|email|max:255|unique:users',
        'password' => 'required|min:6|confirmed',
    ]);
}
```

```
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'username' => $data['username'],
        'email' => $data['email'],
        'password' => bcrypt($data['password']),
        'role' => '3',
    ]);
}
```

- منروح على الـ model اللي اسمه User ومنحمي الـ username:

```
protected $fillable = [
    'name', 'email', 'password', 'role' , 'username' ,
];
```

- هلق كيف بدي فهم Laravel انو تسجيل الدخول لح يتم عن طريق الـ username موعن طريق الـ email منروح على الـ AuthController ومنضيف السطر التالي:

```
protected $username = 'username';
```

هيك صار فيني سجل دخول باستخدام الـ username بدلاً من الـ email والامور كها عندي تمام

رابط مشروع الـ notes على الـ GitHub:

https://github.com/MohammadAltal/note-laravel5.2

رابط مشروع الـ users على الـ GitHub:

https://github.com/MohammadAltal/UsersProject-laravel5.2

تم بحمد الله

Mohammad Eid Altal