

فهاد الجزء السابع من site ecommerce ب laravel غادي نزيدو ل order model ول controller من بعد غادي نشوفوا كيفاش نزيدو login و logout خاص بالأدمن باش يكونوا عندنا جوج ديال les systèmes واحد خاص بالمستخدم العادي والثاني بالأدمن.

### 1- إضافة ل Model Order

---

غادي تزييد model Order فيه غادي يكونوا les champs الخاصين بالمبيعات الكود ديال الملف هو :

```
<?php
```

```
namespace App;

use Illuminate\Database\Eloquent\Model;

class Order extends Model
{
    //
    protected $fillable = [
        "user_id", "product_name", "qty",
        "price", "total",
        "paid", "delivered"
    ];
}
```

### 2- إضافة ل OrderController

---

غادي تزييد controller سمييه OrderController فيه جوج ديال les fonctions وحدة كتعدل ال order وكتحدد واش توصل الشاري بل commande ديالو والثانية كتحذف order من قاعدة البيانات.

الكود ديال الملف هو :

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use App\Order;
use Illuminate\Http\Request;

class OrderController extends Controller
{
    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param \App\Order $order
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, Order $order)
    {
        //
        $order->update([
            "delivered" => 1
        ]);
        return redirect()->back()->withSuccess("Order updated");
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param \App\Order $order
     * @return \Illuminate\Http\Response
     */
    public function destroy(Order $order)
    {
        //
        $order->delete();
        return redirect()->back()->withSuccess("Order deleted");
    }
}

```

### 3- تعديل الملف auth.php

باش نزيدو login و logout خاصين بالأدمين خصنا نزيدو تعديلات على الملف auth.php لي كاين ف dossier config ولي كنزيدو فيه admin guard ولي كتخلي حتى ل admin تكون عندو إمكانية باش يدير la connexion من table admins.

الكود ديال الملف بعد التعديل هو :

<?php

```

return [

    /*
    |-----
    --
    | Authentication Defaults
    |-----
    --
    |
    | This option controls the default authentication "guard" and password
    | reset options for your application. You may change these defaults
    | as required, but they're a perfect start for most applications.
    |
    */

    'defaults' => [
        'guard' => 'web',
        'passwords' => 'users',
    ],

    /*
    |-----
    --
    | Authentication Guards
    |-----
    --
    |
    | Next, you may define every authentication guard for your application.
    | Of course, a great default configuration has been defined for you
    | here which uses session storage and the Eloquent user provider.
    |
    | All authentication drivers have a user provider. This defines how the
    | users are actually retrieved out of your database or other storage
    | mechanisms used by this application to persist your user's data.
    |
    | Supported: "session", "token"
    |
    */

    'guards' => [
        'web' => [
            'driver' => 'session',
            'provider' => 'users',
        ],
        'admin' => [
            'driver' => 'session',
            'provider' => 'admins',
        ],
        'api' => [
            'driver' => 'token',
            'provider' => 'users',
        ],
    ],
];

```

```

        'hash' => false,
    ],
],

/*
-----
-- | User Providers
-----
-- |
| All authentication drivers have a user provider. This defines how the
| users are actually retrieved out of your database or other storage
| mechanisms used by this application to persist your user's data.
|
| If you have multiple user tables or models you may configure multiple
| sources which represent each model / table. These sources may then
| be assigned to any extra authentication guards you have defined.
|
| Supported: "database", "eloquent"
|
*/

'providers' => [
    'users' => [
        'driver' => 'eloquent',
        'model' => App\User::class,
    ],
    'admins' => [
        'driver' => 'eloquent',
        'model' => App\Admin::class,
    ],

    // 'users' => [
    //     'driver' => 'database',
    //     'table' => 'users',
    // ],
],

/*
-----
-- | Resetting Passwords
-----
-- |
| You may specify multiple password reset configurations if you have more
| than one user table or model in the application and you want to have
| separate password reset settings based on the specific user types.
|
| The expire time is the number of minutes that the reset token should be

```

```

| considered valid. This security feature keeps tokens short-lived so
| they have less time to be guessed. You may change this as needed.
|
*/

'passwords' => [
  'users' => [
    'provider' => 'users',
    'table' => 'password_resets',
    'expire' => 60,
    'throttle' => 60,
  ],
  'admins' => [
    'provider' => 'admins',
    'table' => 'password_resets',
    'expire' => 60,
    'throttle' => 60,
  ],
],

/*
-----
-- | Password Confirmation Timeout
-----
--
|
| Here you may define the amount of seconds before a password
confirmation
| times out and the user is prompted to re-enter their password via the
| confirmation screen. By default, the timeout lasts for three hours.
|
*/

'password_timeout' => 10800,

];

```

#### 4- إضافة ل Model Admin

غادي تزيد Model Admin فيه غادي يكون نفس الكود لي كاين فل Model User ما عدا ل guard لي كنعطيها admin.

الكود ديال الملف هو :

<?php

```

namespace App;

use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class Admin extends Authenticatable
{
    use Notifiable;

    protected $guard = "admin";
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password'
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}

```

## AdminController ل إضافة ل -5

---

غادي تزيد controller سمييه AdminController فيه les fonctions ديال login & logout وديال استرجاع ل orders و les produits الكود ديال الملف هو :

<?php

```

namespace App\Http\Controllers;

use App\Order;
use App\Product;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;

class AdminController extends Controller
{
    //
    public function __construct()
    {
        $this->middleware('auth:admin')
            ->except(["showAdminLoginForm", "adminLogin"]);
    }

    public function index()
    {
        return view("admin.index")->with([
            "products" => Product::all(),
            "orders" => Order::all()
        ]);
    }

    public function showAdminLoginForm()
    {
        if (auth()->guard("admin")->check()) {
            return redirect("/admin");
        }
        return view("admin.auth.login");
    }

    public function adminLogin(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'email' => 'required|email',
            'password' => 'required|min:4'
        ]);

        if ($validator->fails()) {
            return redirect('admin/login')
                ->withErrors($validator)
                ->withInput();
        }

        if (auth()->guard("admin")->attempt([
            'email' => $request->email,
            'password' => $request->password
        ], $request->get("remember"))) {
            return redirect("/admin");
        }
    }
}

```

```
    } else {
        return redirect()->route("admin.login")->with([
            "errorLink" => "Email ou mot de passe est incorrect"
        ]);
    }
}

public function adminLogout()
{
    auth()->guard("admin")->logout();
    return redirect()->route("admin.login");
}

public function getProducts()
{
    return view("admin.products.index")->with([
        "products" => Product::latest()->paginate(5)
    ]);
}

public function getOrders()
{
    return view("admin.orders.index")->with([
        "orders" => Order::latest()->paginate(5)
    ]);
}
}
```