

# Simple Pharmacy System

## Roles :-

### 1- Admin

- Admin will have access to everything in the system ,**he can see any links or make any action Pharmacy and Doctors can do** with these extra functionalities
- We will have **only one admin** in the system and by **default will be seeded through [laravel seeders](#)** , with email ([admin@admin.com](mailto:admin@admin.com)) & pw (123456) **but we can create other admins through console command , for example i can make this**

```
php artisan create:admin --email=admin2@admin.com  
--password=123456
```

- In the left menu we will have these links (**Pharmacies, Doctors , Users , Areas, User Addresses, Medicines, Orders, Revenue**) these links will allow us to do CRUD operations
- **Pharmacy & Doctor** will have (email, name , password , national\_id, avatar\_image )
- national\_id , email must be unique | password must be at least 6 chars | avatar\_image must be an image with extensions jpg,jpeg and **if there's no avatar\_image then we will provide a default one**
- Admin can make CRUD on **Areas** which will have (name, address) .
- When Creating a pharmacy you must assign it an **area\_id & priority**

- When **deleting a pharmacy** make sure that it doesn't have any **order** assigned to it . and also make sure i can restore it [Check Soft Delete](#)

**Note:** there will be no registration form in our app . only login & forget pw screens.

## 2- Pharmacy (or can be named Pharmacy Owner)

- **Pharmacy** can do what **Doctors** Can do except he can create new doctors and also change the pharmacy info except the serving area & priority it can only be changed by admin

### Doctors Tab

- **Pharmacy** will have a menu called **Doctors** where he can make **CRUD on Doctors** and **Ban Or UnBan Doctor**
- In **Doctors** page we will need to show these **columns** (name , created at date only --- check mutator in laravel, avatar\_image, email, national\_id, is\_baned) **but if the current logged in user is Admin** then we will need to **show an extra column called Pharmacy Name** , which is the name of the Pharmacy who created the doctor and **make sure Pharmacy can see it's own doctors only** .
- (Actions column will have **Edit , Delete** ) and **if banned then there will be an action called unban** , and if unbanned then will have an action called **ban** .

## Orders Tab

- **Pharmacy Or Doctor** will have a menu called **Orders** where they can make **CRUD on Orders**
- **Orders index page** . will have a dataTables that show this  
**ID, Ordered User Name, Delivering Address, Creation Date, Doctor Name, Is Insured, Status, Actions**  
**Note Doctor Name can be empty if this order isn't created by a doctor**

And if i am logged in as admin then we need to show extra columns **Creator Type** where it shows wither this order **Created by Doctor Or Pharmacy Or User**

**Assigned Pharmacy** . where it shows the assigned pharmacy name and empty if none.

- When **Creating Order** it will show a form , where pharmacy or doctor chooses a user from a dropdown list , **and inserts the medicine name and quantity and type and it's price** . and then enters the **visa card number through stripe** , **Price Must Be Stored In Database in Cents And To Be Shown In Dollars**  
**Note1: pharmacy or doctor can add multiple medicines .**  
**Note2: medicine names can be autocomplete while typing or and if medicine doesn't exist the doctor or pharmacy can insert them normally in the same typing area so check [Adminlte Select2 & Select2 Tags](#) .**
- Please make sure that orders follow the flow . like i can't edit an order that has been changed to **WaitingForUserConfirmation**

## Revenue Tab

- We need to show the **total revenue as a card** at the top of this page , if i am logged in as **pharmacy**, then will show the **total revenue of my pharmacy** , and if i am logged in as a **admin** it will show the **total revenue of all pharmacies** and also make a **dataTables** like this

PharmacyAvatar	PharmacyName	TotalOrders	TotalRevenue
	Elazaby	50	2,000\$

### 3- Doctor (who works in pharmacy)

- All this role does is just modify the flow of order . so it will have one tab called **Orders** . where doctor can modify the order like making it **WaitingForUserConfirmation** or **Canceled ... etc.**

### 4- User (it will be an api only)

- User will have an **endpoint to register** , he needs to enter his (name ,email , gender ,password , password confirmation, date of birth , profile\_image, mobile number, national\_id) all of these fields are required , after he register [you must send an email verification link to his email](#)

- After User is **verified** we will need to greet him by sending an **email using laravel notifications** and it **must be Queued again Must Be Queued - Must Be Queued - Must Be Queued**

**So read this page very very very carefully**

<https://laravel.com/docs/master/notifications#queueing-notifications>

- User will have an **endpoint to login** , using email & password, response will be object that contains the user info with access token (**same response that we did in labs like sanctum package**)
- User will have an **endpoint to update his profile info except email**
- User will have **CRUD endpoints on his addresses** . and address info will be : { area\_id, street\_name , building\_number, floor\_number , falt\_number , is\_main }
- User will have an **endpoint to see his orders and end point to see order details** for example of /orders json object {id:1 , medicines: ['name': 'panadol', 'type': 'capsule' , 'quantity':2 , 'total\_price':10] , order\_total\_price: 10, ordered\_at : '2 hours ago', status : 'delivered' , 'assigned\_pharmacy' : {id:1 , name: 'elazaby' , address: 'mahtetmasr', avatar\_image\_url: 'http://someimage.jpg'}}
- User will have **endpoints to make an order** /orders POST and /orders/{id} PUT request to update order data like uploaded prescriptions or cancel it . **make sure the user can't modify a delivered, processing, canceled, confirmed order he only can modify new orders .**

- Order information that the user will enter : **list of prescription images** and **whether this order is insured or not** and **delivering\_address\_id** from his addresses . a json example  

```
{
    is_insured : true,
    prescription : [image1.jpg, image2.jpg],
    delivering_address_id : 1 //make sure it belongs to this user
}
```
- We need a **schedule command that runs daily** using laravel , that will send an email notification to users **who didn't login from the past month** , and telling them that we miss you

A Code Example that will be in **console/Kernel.php**:-

```
$schedule->command('notify:users-not-logged-in-for-month')
->daily();
```

**So read this link Very Veryyyyyyyyy Carefully**

- We need a **schedule command that runs every minute** to scan the **new** orders and assign them to the highest priority pharmacy that serves same delivering address
- **Authentication** will be using [Laravel Sanctum](#) , **all routes are allowed only for authenticated** users except for register & login , also any other routes like attend or edit profile [must be verified](#)

## Summary Of Workflow

- 1- User makes an order
- 2- Order gets assigned to the highest priority pharmacy that serves the same area of order delivering address.
- 3- pharmacy starts to put the medicines and there prices and then clicks notify user to pay the price
- 4- then user can see the total price and choose whether to pay that price or cancel the order
- 5- after that the pharmacy can change the order to be delivered

## Order Statuses

**New:** means the order is just created and must be assigned to pharmacy

**Processing :** means the pharmacy is ready to put medicines and prices

**WaitingForUserConfirmation :** this happens when the pharmacy submit a form with medicine names and prices and clicks notify the user to pay the order button . then we send email notification for the user to pay the order

**Canceled :** happens when user cancels his order

**Confirmed:** happens when user confirm the order and pay the price

**Delivered:** happens when pharmacy delivers the order

## System General Rules

- All tables in the system must use **DataTables Plugin** and all logical **Fields Must Be Searchable And Orderable** like (name , price , email , etc )
- All charts in the system must use **Chart Js** and **Data Retrieved By Ajax**
- All **Delete Buttons** must show a **warning** before deleting and also must be done through **ajax requests** , and if the ajax request succeeded you can then **refresh the Datatables**
- **Price Must Be Stored In Database in Cents And To Be Shown In Dollars**
- All of the users **can change their profile info** like name or email
- Use default laravel auth , and make **sure the forgot password functionality works**



Bonus (**Bonus Is Approved If The Old Code Is Clean**):-

- **Try To Reduce Or Eliminate The Duplication In Code , And Make Your Code More Readable And Pleasing For Eye , If The Code Isn't Clean Then Any Bonus Won't Be Counted As Finished**

**So Please Clean Your Code First Before Implementing The Bonus**

#### **Statistics**

- We will have a menu Tab called **Statistics** , it will open a page which will have four charts

**(Male - Female Attendance) pie chart** that shows how much percent males to females attendance

**(Revenue) basic line chart** which will show how much money we got in 12 months for this year, this chart **by default will show current year**

**(Top Ordered Users) pie chart** to show the **top 10 users** with there bough orders number , for example will show (ahmed : 20 , mohamed : 3 , ali :4 , ... )

- Deploy Your Project On Heroku  
<https://mikateach.com/setting-up-laravel-5-6-on-heroku/>
- Use laravel cache to temporary cache drop down list of user when creating an order

- **All Charts support filtering by year** , so i can choose a specific year on each chart and it will show it's statistics
- **Pharmacy Can Export** all the orders as **excel** sheet

## Project General Rules

- All classes must follow **StudyCase** (pascal case) naming for example ( PostsController , UsersController , Post , StorePostRequest)
- All routes must follow resource naming convention
- All **views must be divided into sub directories** like ( posts/index.blade.php - users/index.blade.php - clients/index.blade.php)
- All **Middlewares should be applied on routes** (check Route::group) and not to be applied on controller
- All **Models** must be Singular like (Post , Comment , User , Country , Floor , Room , Reservation)
- Don't Push Any **Deadly Comments on the Git Repo** like :-  
//dd(\$request); // \$user = User::first();

## Project Resources

- For Admin LTE ( **Mandatory** ) after installing open starter.html file and remove unused bower\_components  
<https://adminlte.io/>
- DataTables have two plugins one will be used on the frontend , and another will be used on backend ( **Mandatory** ) , also all the filtering and searching and ordering **must be done from backend**  
(<https://datatables.net/manual/server-side>)  
<https://datatables.net/> (Frontend)  
<https://github.com/yajra/laravel-datatables> (Backend)
- For roles & permissions ( **Mandatory** )  
<https://github.com/spatie/laravel-permission>
- For Ban Or Unban you can use  
<https://github.com/cybercog/laravel-ban>
- For Api Authentication ( **Mandatory** ) we will use [laravel sanctum](#)
- For charts use chart js( **Mandatory If You Will Work On Bonus** )  
<https://www.chartjs.org/>
- For Stripe , follow the quick start guide ( **Mandatory** )  
<https://stripe.com/docs/quickstart>
- For countries you can use ( **Mandatory** )  
<https://github.com/webpatser/laravel-countries>
- For Excel you can use  
<https://github.com/Maatwebsite/Laravel-Excel>