|  |  |
| --- | --- |
| **Project Case – External Document** |  |
| COMP6144  Web Programming |
| Odd Semester Year 2020/2021 |

* Project Title

Just Du It!

* Introduction

Project ini ditujukan untuk mata kuliah COMP6144-Web Programming semester 5 kelas **BF-01**. Kelompok kami yang beranggotakan **Jackie Leonardy** dan **Luis Genesius** membuat sebuah website E-commerce berupa online shoes shopping. Untuk **Front-End** kami menggunakan **Bootstrap 4**, dan untuk **Back-End**nya menggunakan **PHP** dan **Laravel** sebagai Frameworknya. Project

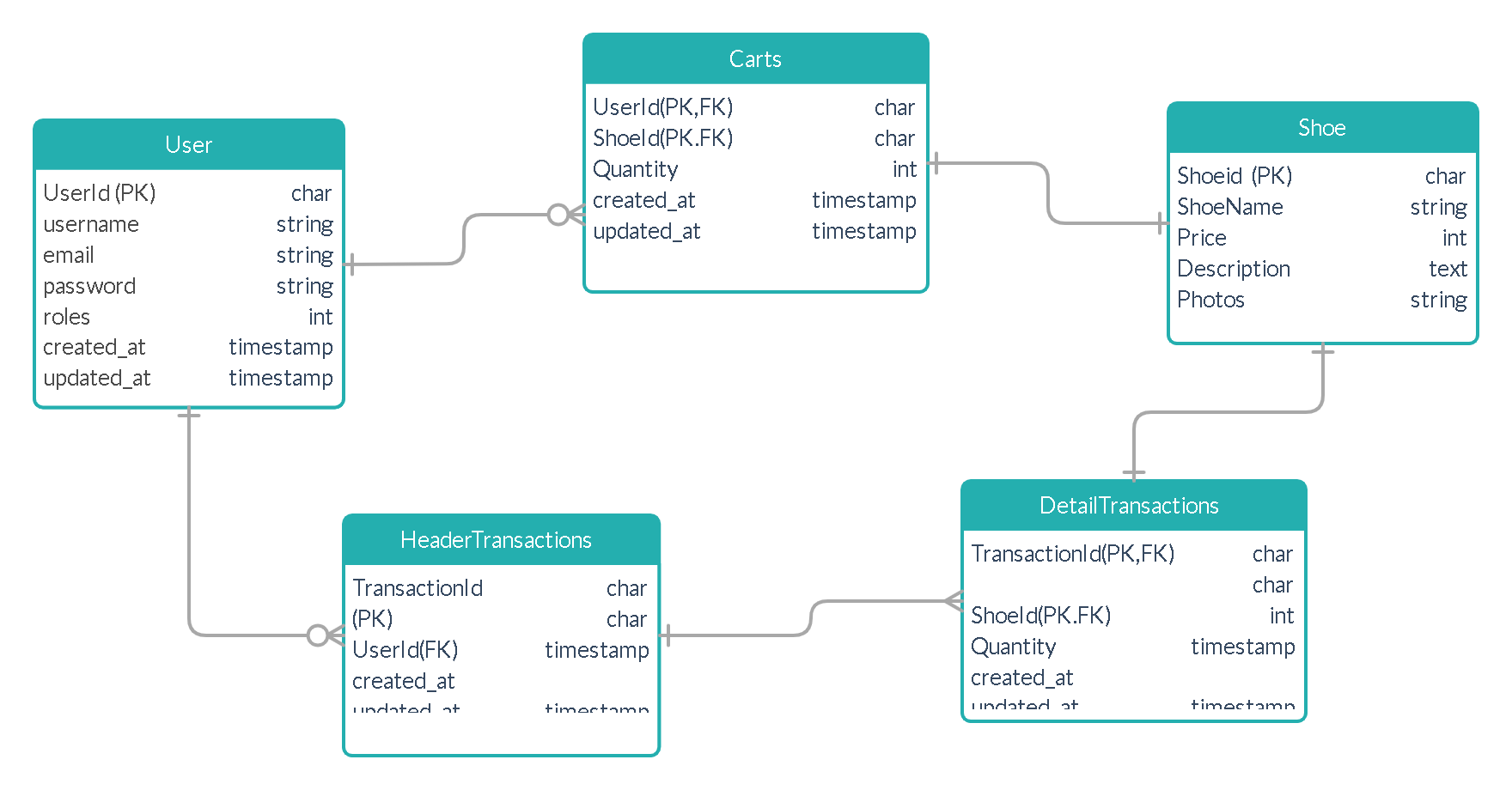
Pada website yang kami buat, user dapat mengunjungi website kami dan melihat-lihat list sepatu yang tersedia. Namun, untuk dapat membeli (memasukkan ke dalam cart) user harus ter-authentikasi terlebih dahulu. Oleh karena itu, user dapat login ataupun melakukan registrasi. Setiap halaman sudah terauthentikasi sehingga user non-login tidak bisa sembarangan mengakses URL pada halaman ini.

Di sini, user bisa melakukan **CRUD** khusus untuk cart saja, di mana user bisa **Create** untuk memasukkan shoes ke dalam Cart, **Read** untuk melihat isi dari cart, **Update** untuk meng-update isi cart, dan **Delete** untuk menghapus item dalam cart. User juga bisa melakukan **check-out**, di mana semua item dalam cart user akan di-checkout dan masuk ke dalam transaction history. Sehingga, semua history transaksi user dapat diakses.

Untuk admin sendiri, bisa melakukan **CRU** juga, namun CRU yang dilakukan adalah **Create** untuk **membuat/memasukkan** data shoes baru ke dalam database, **Read** untuk mengakses **semua** **transaction history** dari semua user, dan **Update** shoes yang sudah dimasukkan ke dalam database, baik dari attribut nama, harga, deskripsi, ataupun foto. Untuk Deletenya, kami membuatnya harus mengakses database secara langsung, sehingga tidak bisa langsung menghapusnya dari website.

Kami juga menambahkan beberapa validasi, seperti ketika user ingin menghapus pesanan pada cart, akan muncul pop-up modal yang akan memvalidasi apakah user yakin ingin menghapus barang tersebut dari cart. Sehingga, ketika user tidak sengaja meng-klik delete button, hal ini bisa dicegah.

* Report / Documentation

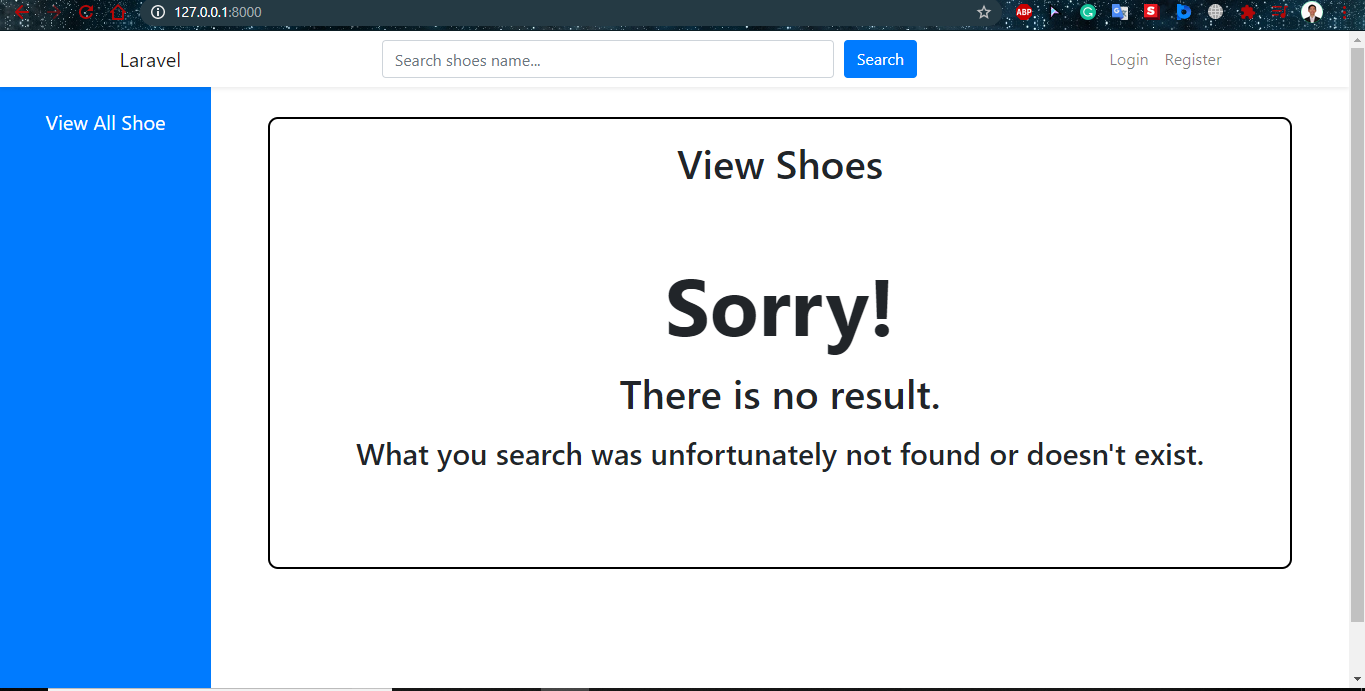
1. **ERD Diagram**

Berikut adalah Database Schema yang kami buat, ada satu tabel yang kami buat berbeda nama dengan ERD yang sudah kami buat. Kami mengganti table carts menjadi shoe\_user, hal ini karena kami mencoba mengikuti naming convention dari Laravel, di mana untuk setiap **pivot table harus menggunakan nama table1\_tabel2** dan terurut **alphabetically** agar Laravel secara otomatis tau bahwa pivot table ini menghubungkan tabel1 dan tabel 2. Sedangkan untuk tabel pivot lain seperti **detail\_transactions**, kami membuat namanya sama sesuai ERD, namun kami harus menspesifikkan nama tersebut pada model dan controller.

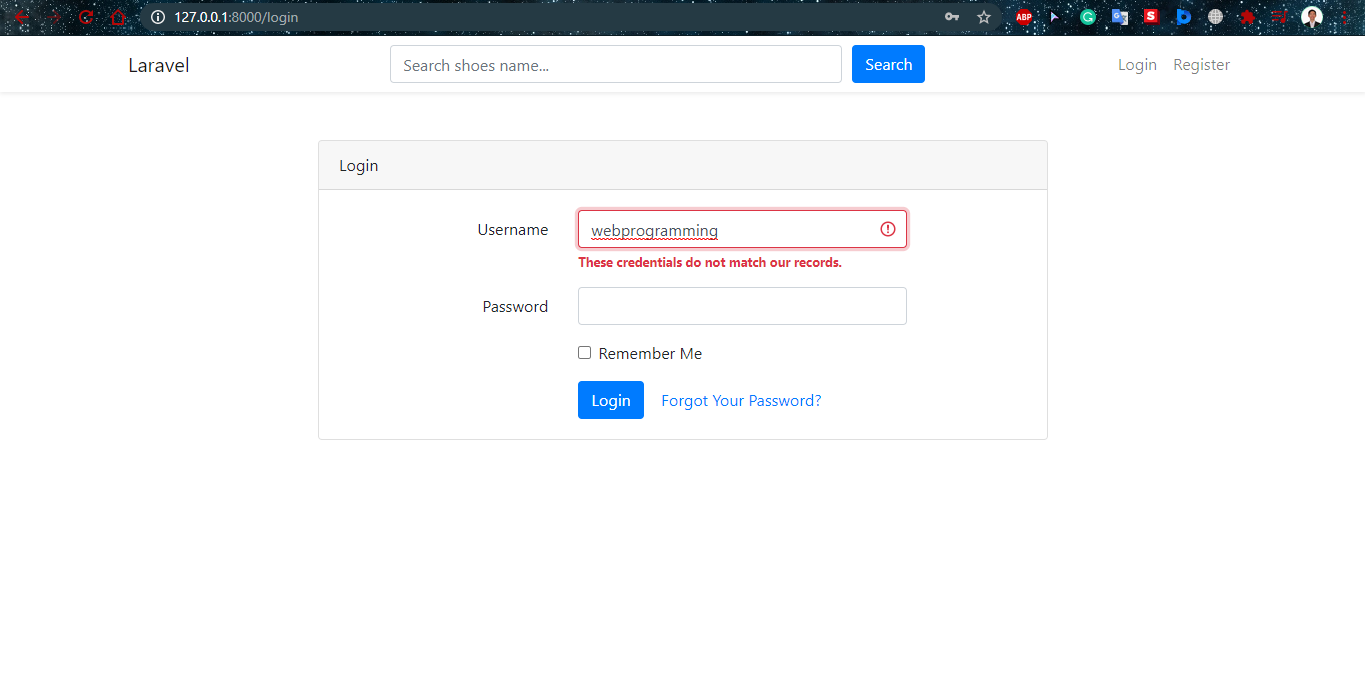
<https://laravel.com/docs/7.x/eloquent-relationships#many-to-many>

Berikut adalah table data pada database kami

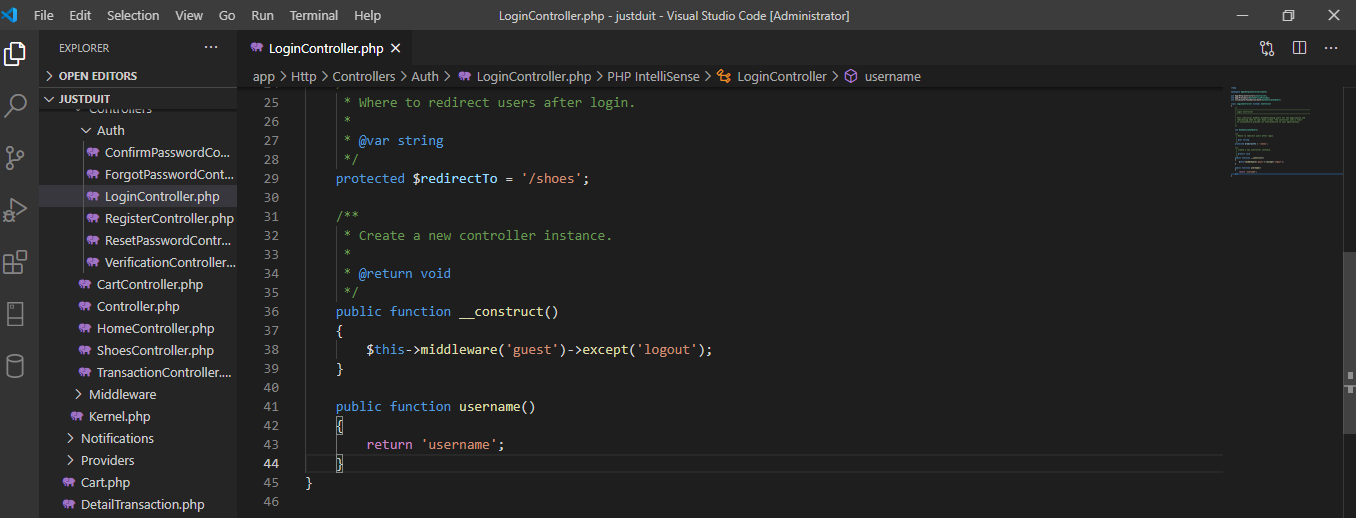
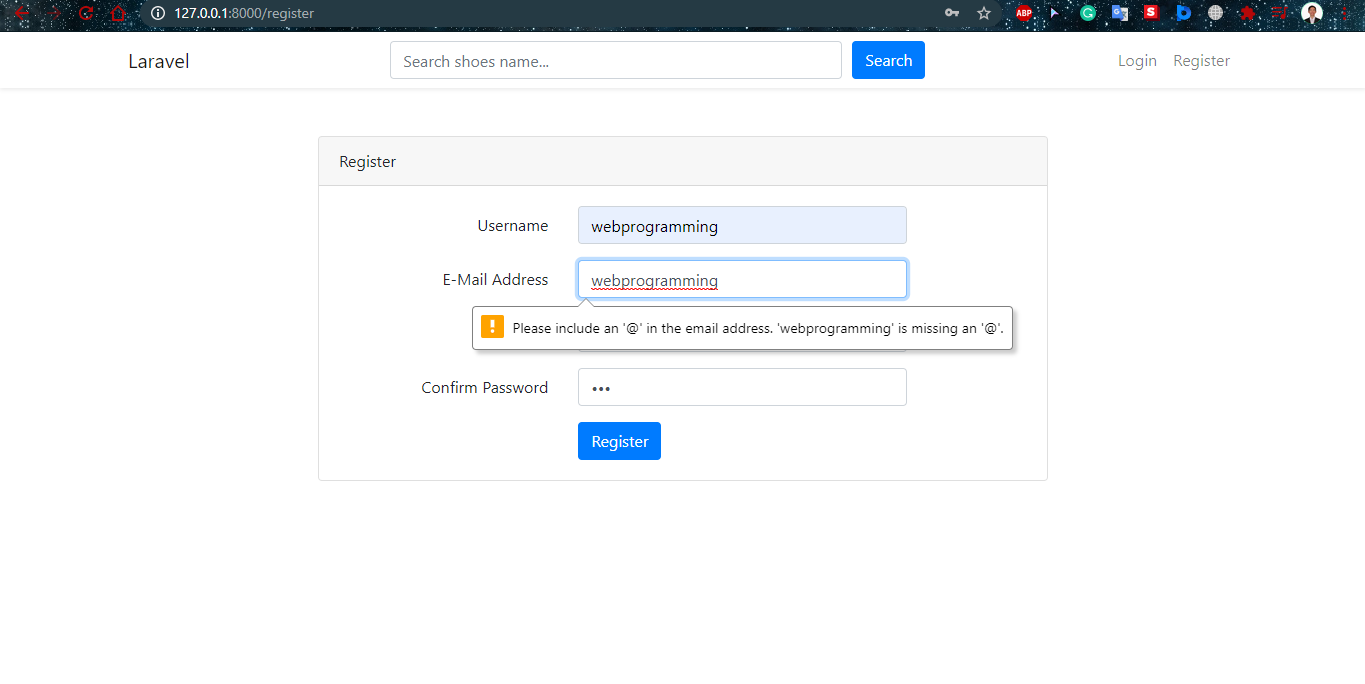
1. **Home Page**

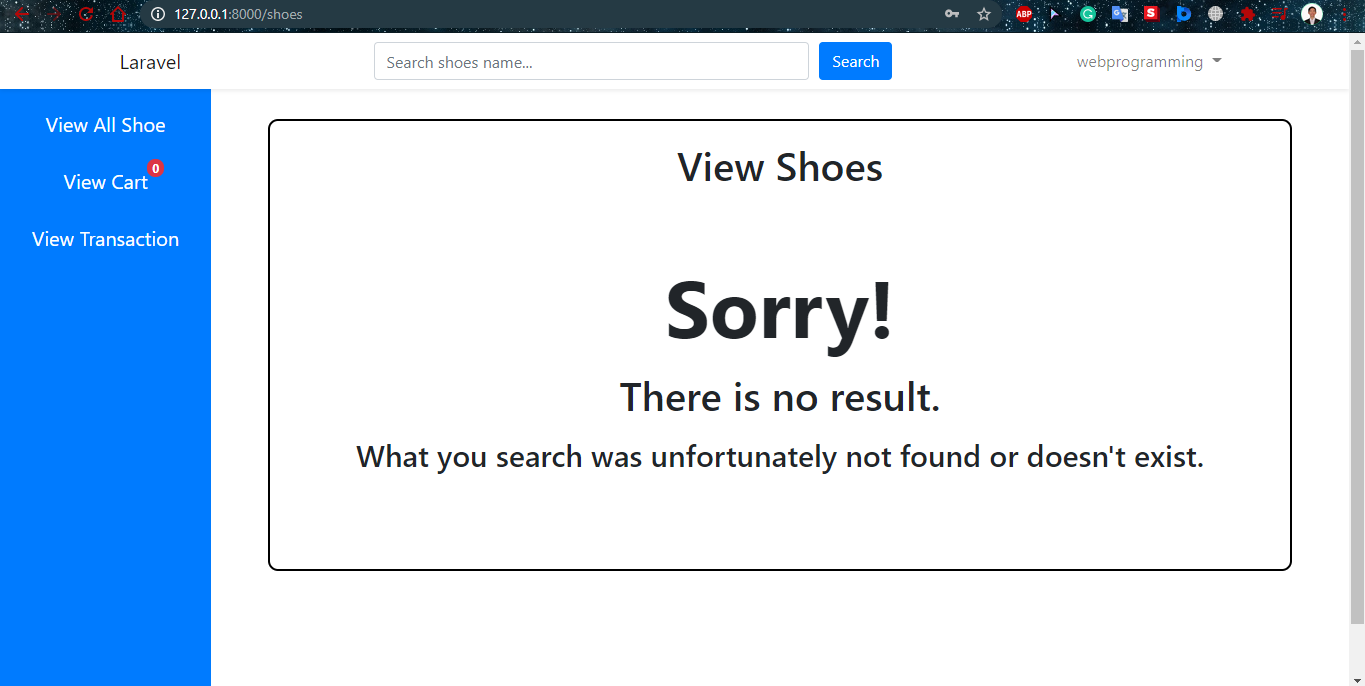
Tampilan Homepage, jika data kosong

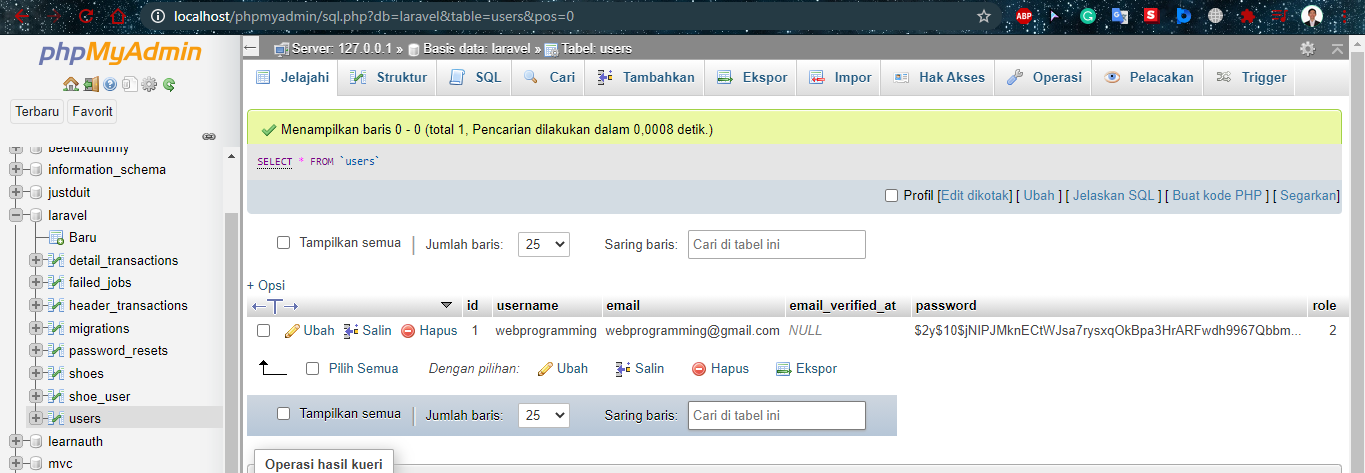
1. **Login & Registration**

Jika kita mencoba login, dan data tidak ada maka akan muncul error message sebagai berikut

Di sini kami mengatur untuk login menggunakan username, bisa saja pakai email dan kami menggantinya pada login controller

Kita akan registrasi data diri terlebih dahulu, terdapat juga validasi untuk email. Kemudian, username yang dimasukkan juga harus unik, sehingga jika sudah ada pada database maka username tersebut harus berbeda dengan yang sudah tergistrasi.

Berikut tampilan home user, bisa dilihat side bar **View Cart** dan **View Transaction** bisa diakses hanya kalau user tersebut ter-authentikasi

Secara default data akan masuk ke dalam database dengan role 2 (menandakan user biasa)

**PENTING DIBACA**

**Untuk registrasi admin, kita harus manual mengganti rolenya pada database menjadi 1.**

Hal ini untuk mencegah hal yang tidak diinginkan, sehingga akan lebih baik jika data diubah manual saja pada database, selain juga database hanya bisa diakses pihak internal saja.

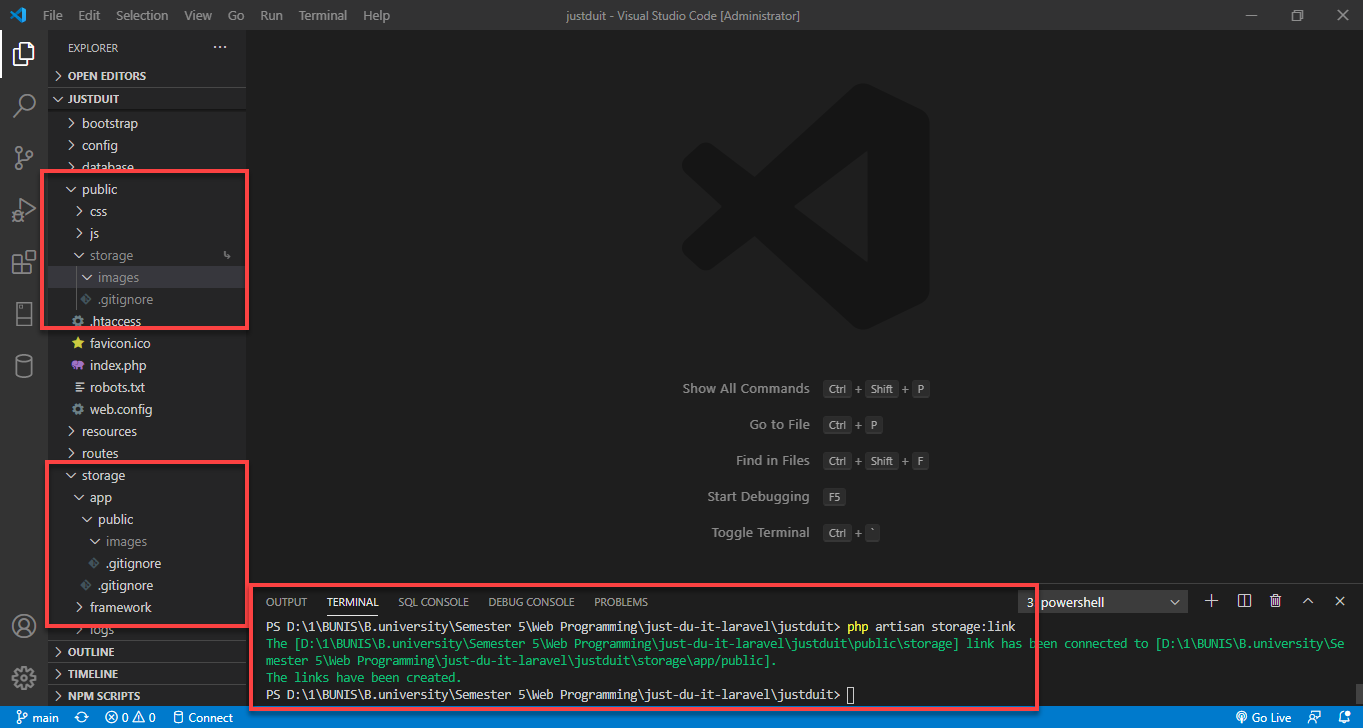
1. **Storage Link**

Image yang diupload kami tamping pada storage public, hal ini dimaksukan bahwa ketika kita memasukkan data ke dalam database, data tersebut akan masuk ke dalam storage internal. Hal ini akan berbahaya jika kita memberikan akses data dari dalam storage internal, oleh karena itu, akan lebih aman jika menggunakan storage link di mana SL ini akan mengambil data dari internal, dan memasukkannya ke dalam data eksternal. Sehingga, kita bisa mengatur data-data apa saa yang bisa diakses oleh internal dan bisa dikendalikan.

Untuk Storage link, kita bisa menggunakan command

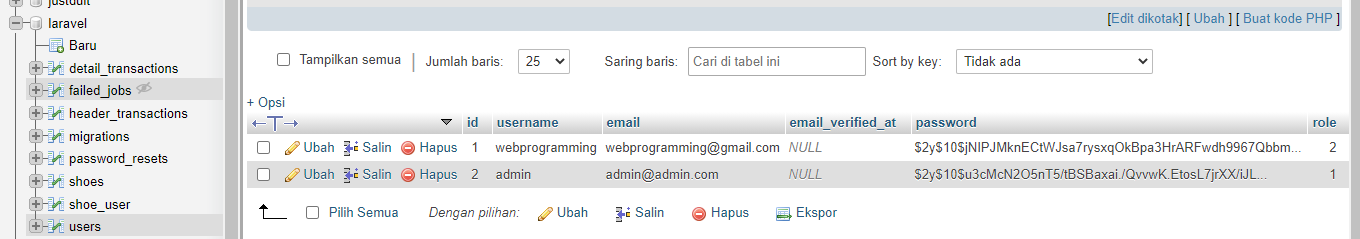
php artisan storage:link

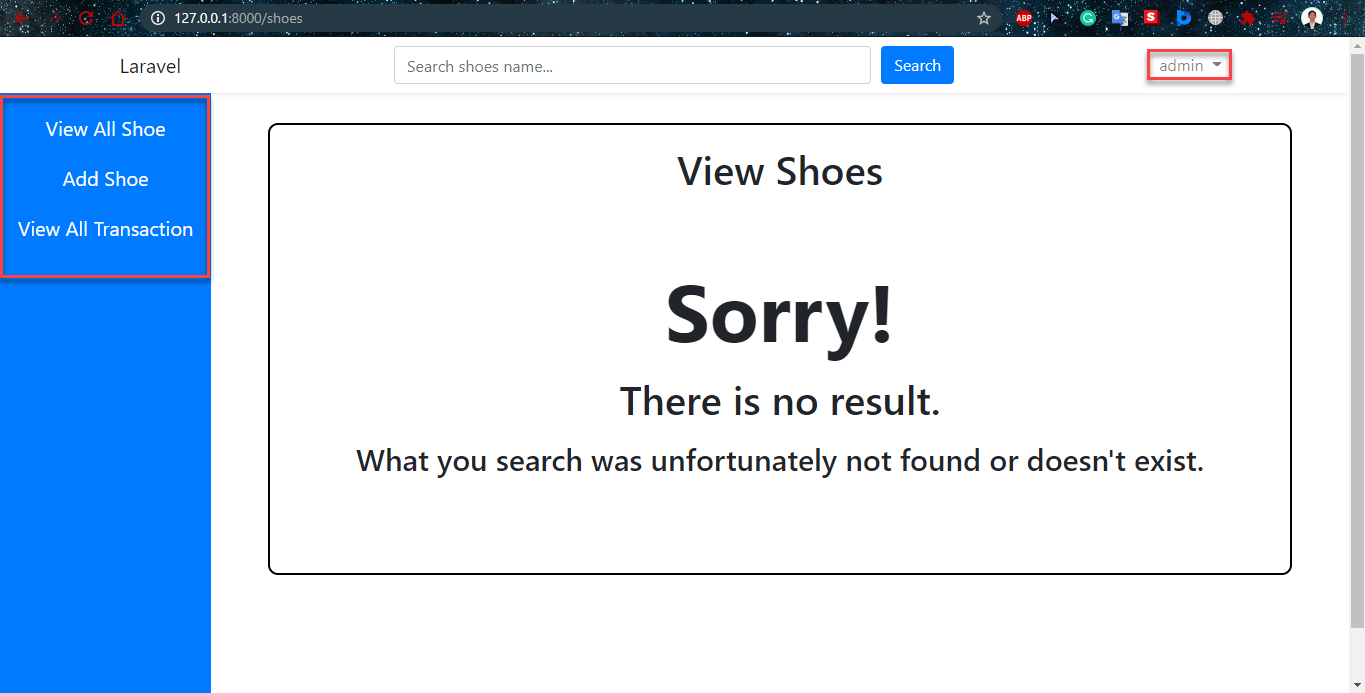
Hal ini agar data yang kita masukkan ke dalam internal, bisa ditarik ke data eksternal/public.

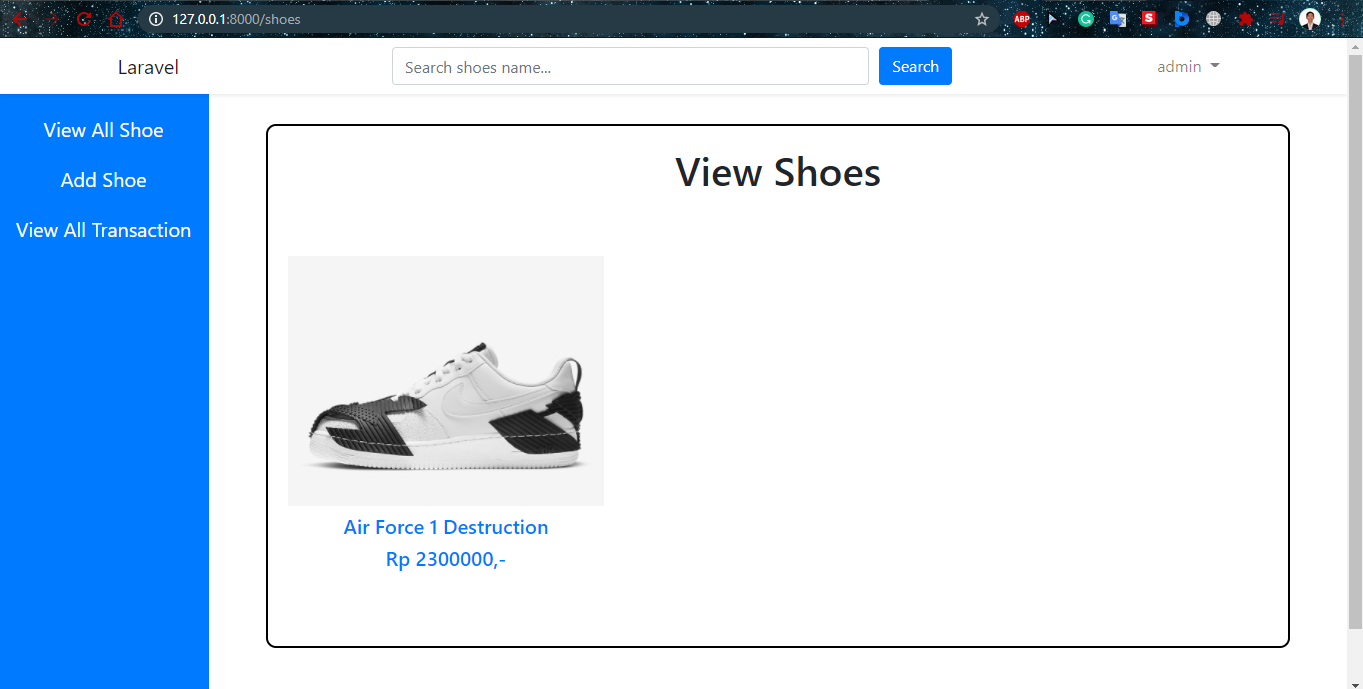


1. **Adding Shoes**

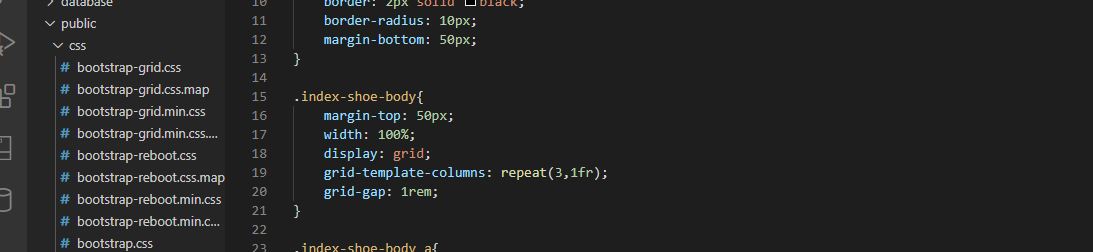
Untuk admin, seperti yang sudah diterangkan di atas, kita akan mengatur rolenya secara manual

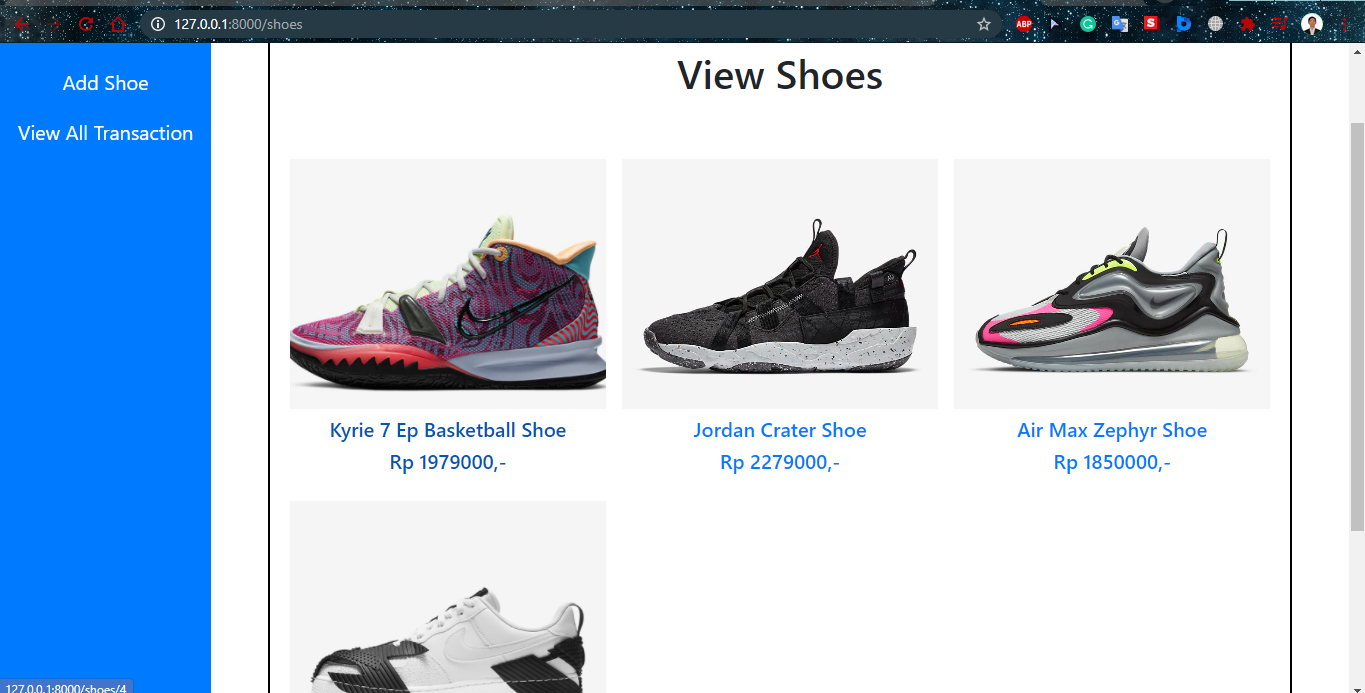


Tampilan side-bar untuk admin juga berbeda, terdapat Add Shoe dan View All Transaction.

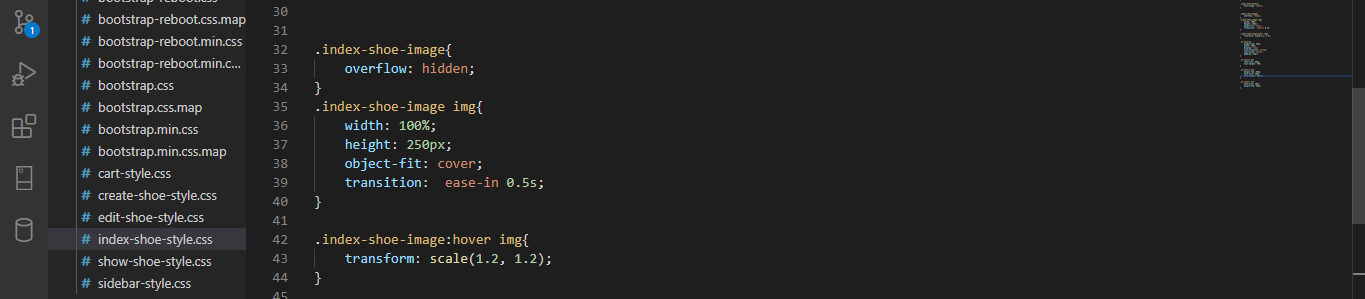
Setelah kita input, hasilnya menjadi seperti ini.

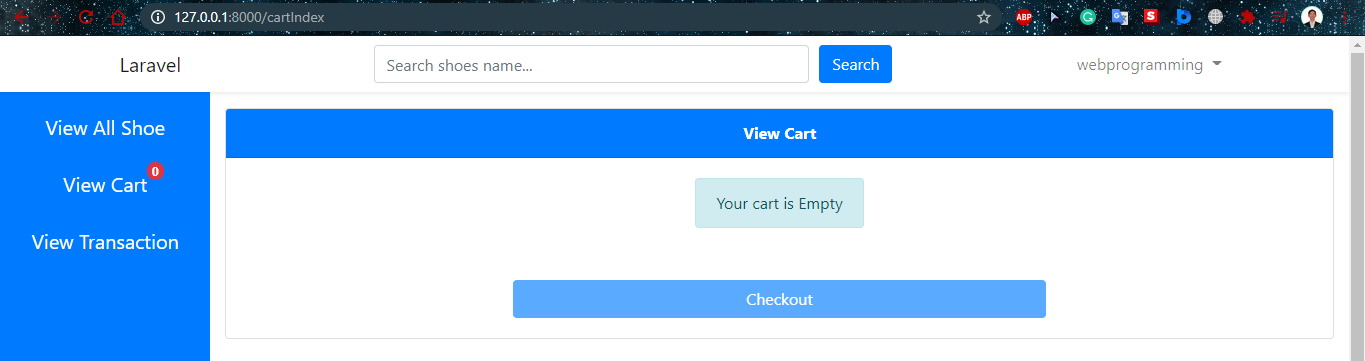
Mari kita coba input lebih banyak data, di sini kami menggunakan Grid sehingga ketika data ditambahkan lebih dari 3, maka akan wrap ke samping.



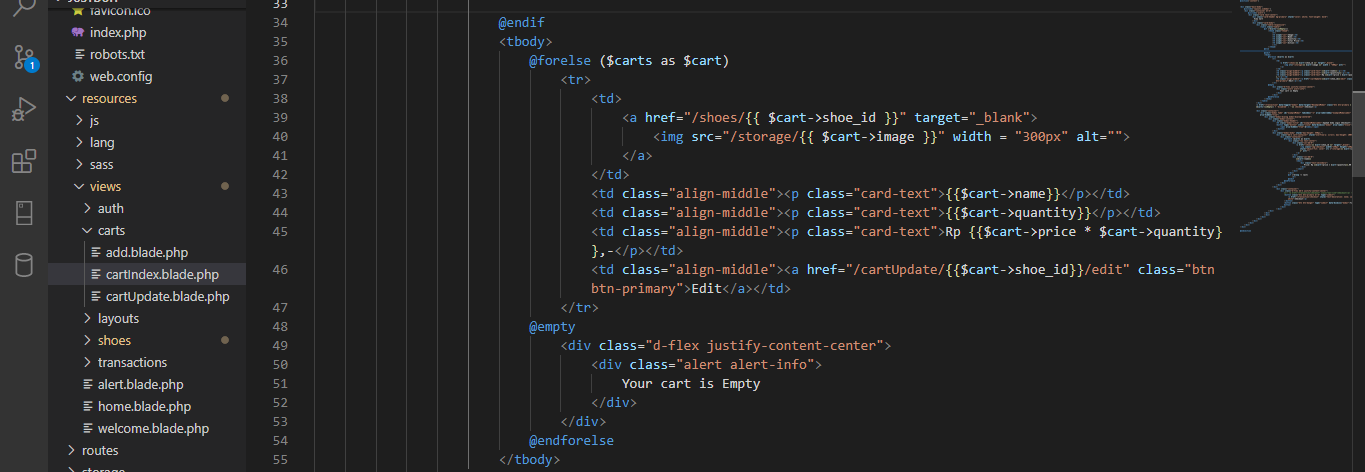


Ketika foto dihover, maka akan ada animasi zoom menggunakan transform dan scale dari CSS.



1. View Cart

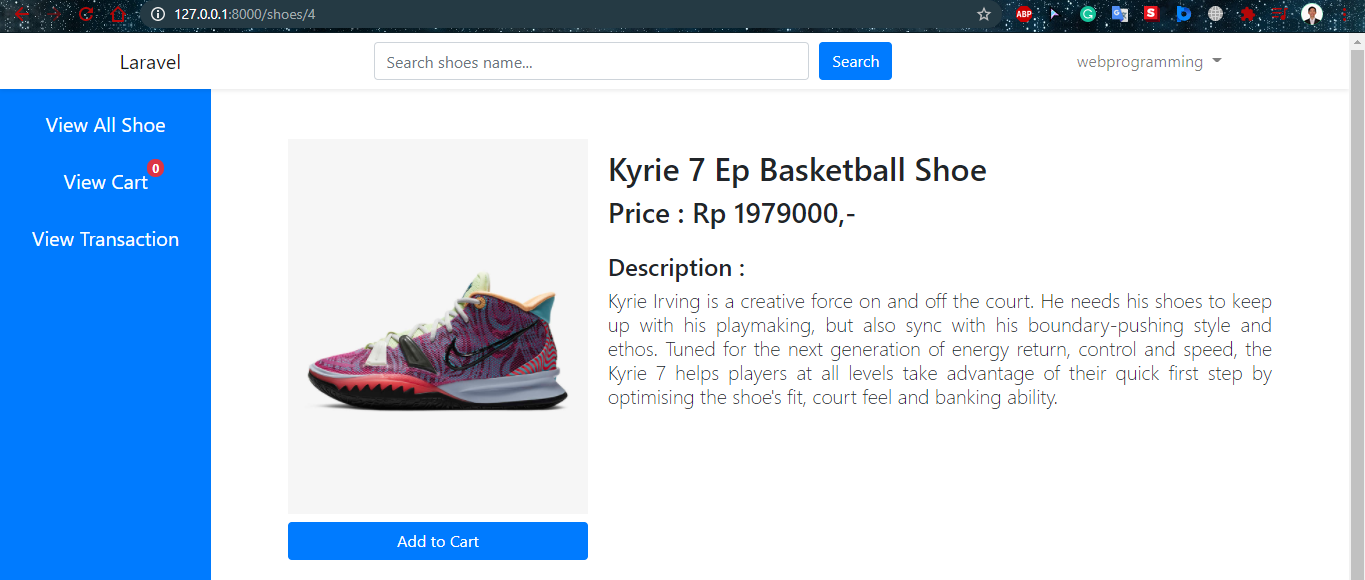
View Cart menampilkan tulisan Your Cart is Empty, dan button Check-out tidak akan bisa diklik ketika cart pada user yang sedang login belum memiliki isi. Hal ini bisa dilihat pada CartIndex.blade.php

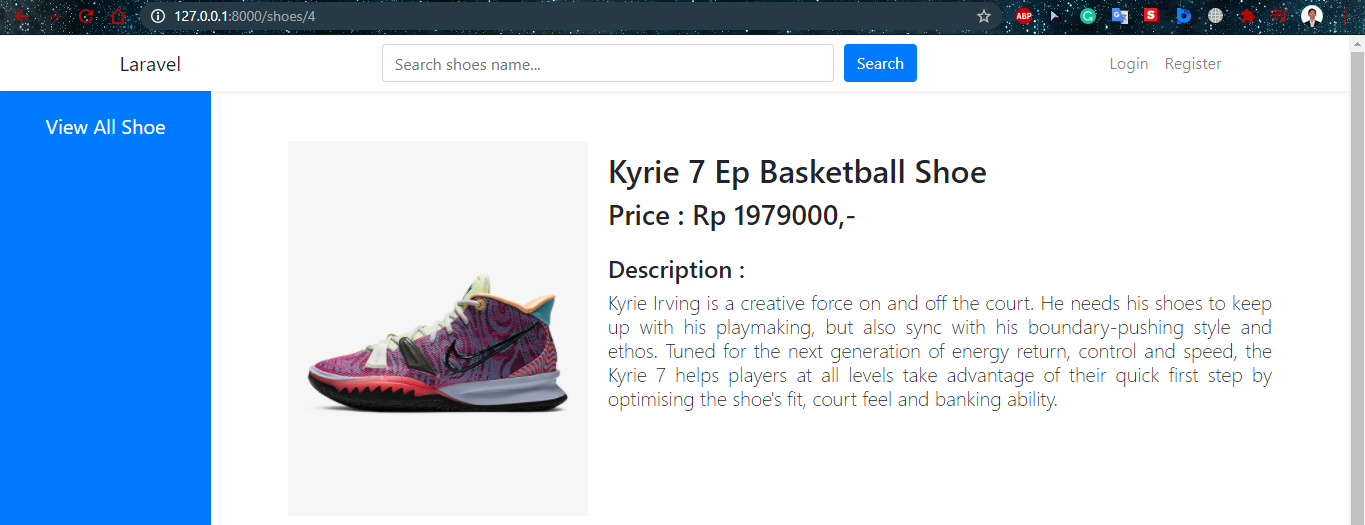
Di sini kamu menggunakan forelse, bukan foreach. Kenapa? Hal ini karena dengan adanya forelse, kita bisa lebih mudah memvalidasi data kalau data tersebut kosong atau @empty. Singkatnya, foreach akan dijalankan hanya kalau data tersebut berisi atau $carts tidak null, namun jika null, maka @empty akan dijalankan, yaitu menampilkan info “Your Cart is Empty”

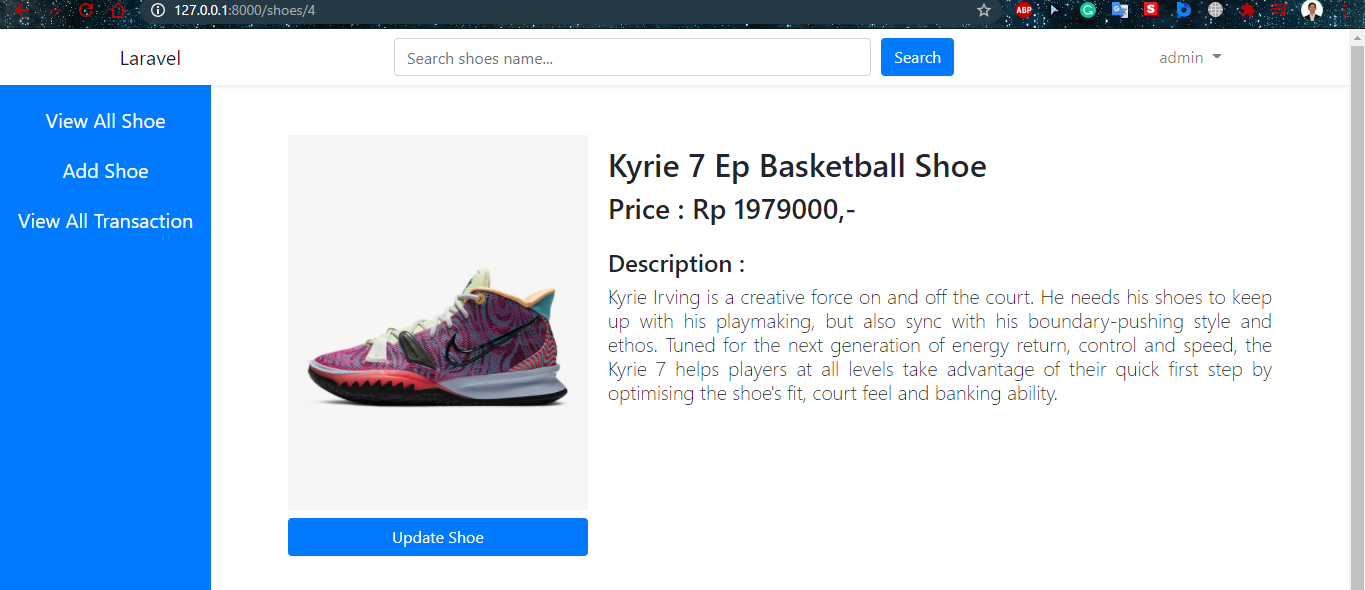
Untuk disable buttonnya kami menggunakan sintaks berikut

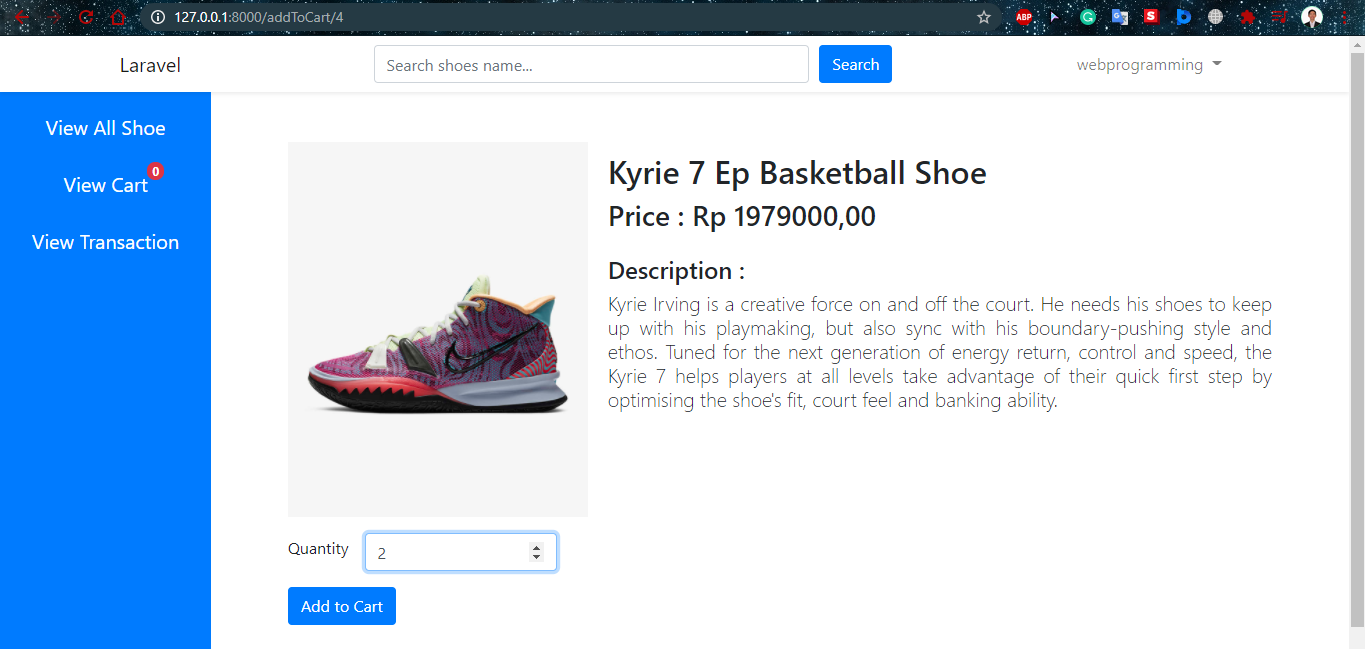
1. Adding item to Cart

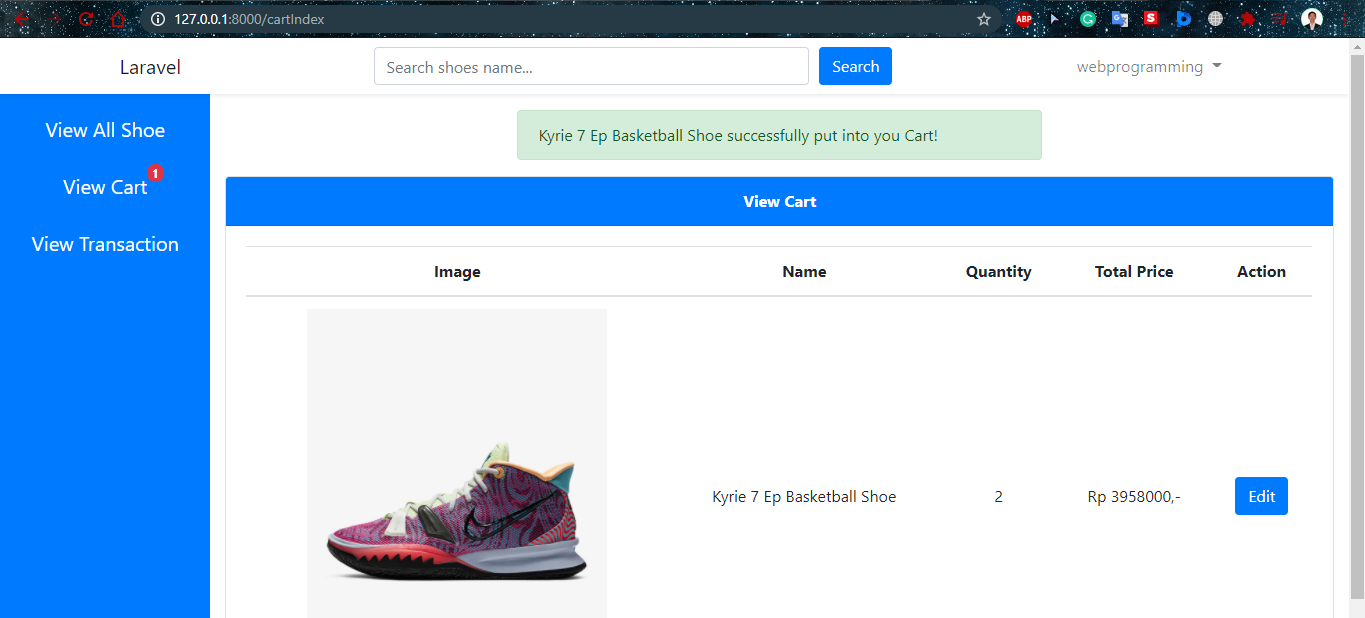
Ketika foto diklik, maka berikut tampilannya pada beberapa user

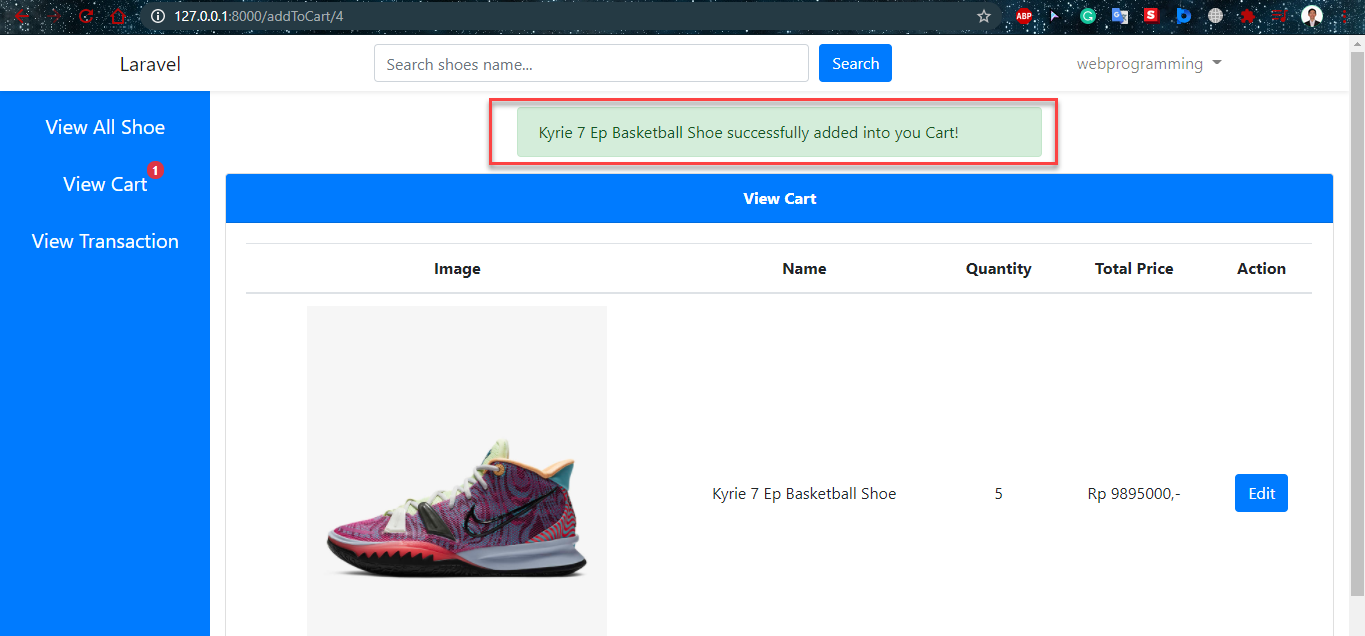
User yang sudah login

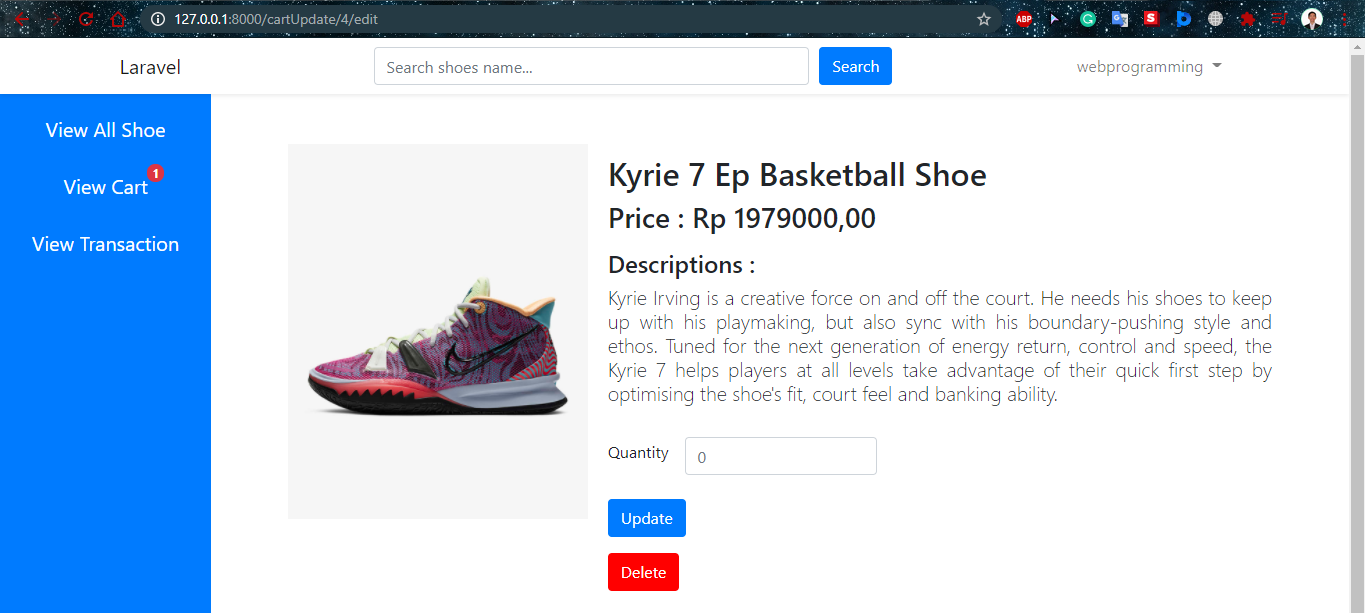
User yang tidak Login

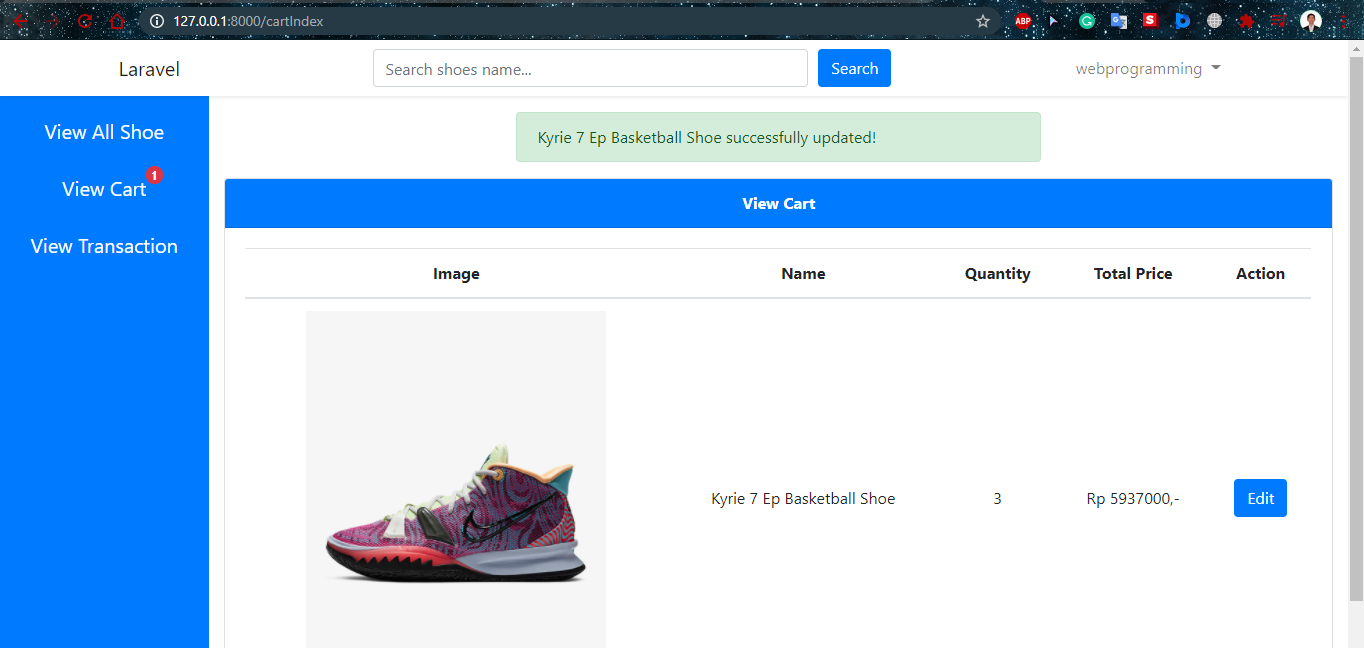
Admin

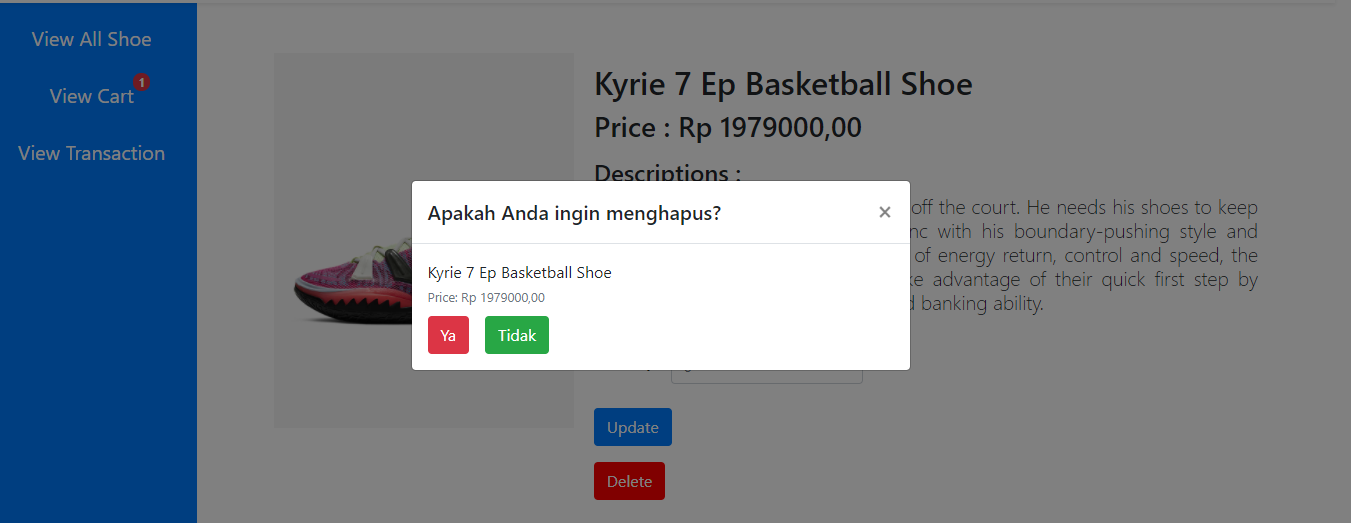
 Mari kita tambahkan shoe ke dalam cart

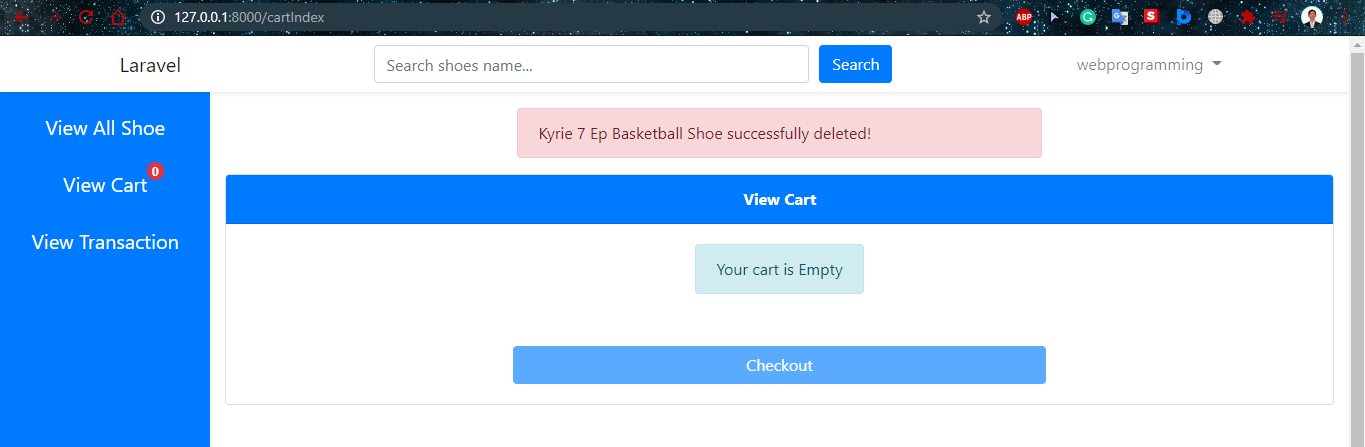
Maka akan ada alert Message yang dikirim melalui session flash, dan Notifikasi pada View Cart akan menyesuaikan data pada cart

Mari kita coba tambahkan quantity 3 pada shoe yang sama, bisa kita lihat yang bertambah adalah quantity dari 2 + 3 = 5, dan hal ini dibuktikan dari alert message yang berbeda, kalau misal datanya sudah ada maka akan tertulis “added”, kalau belum ada akan tertulis “put”

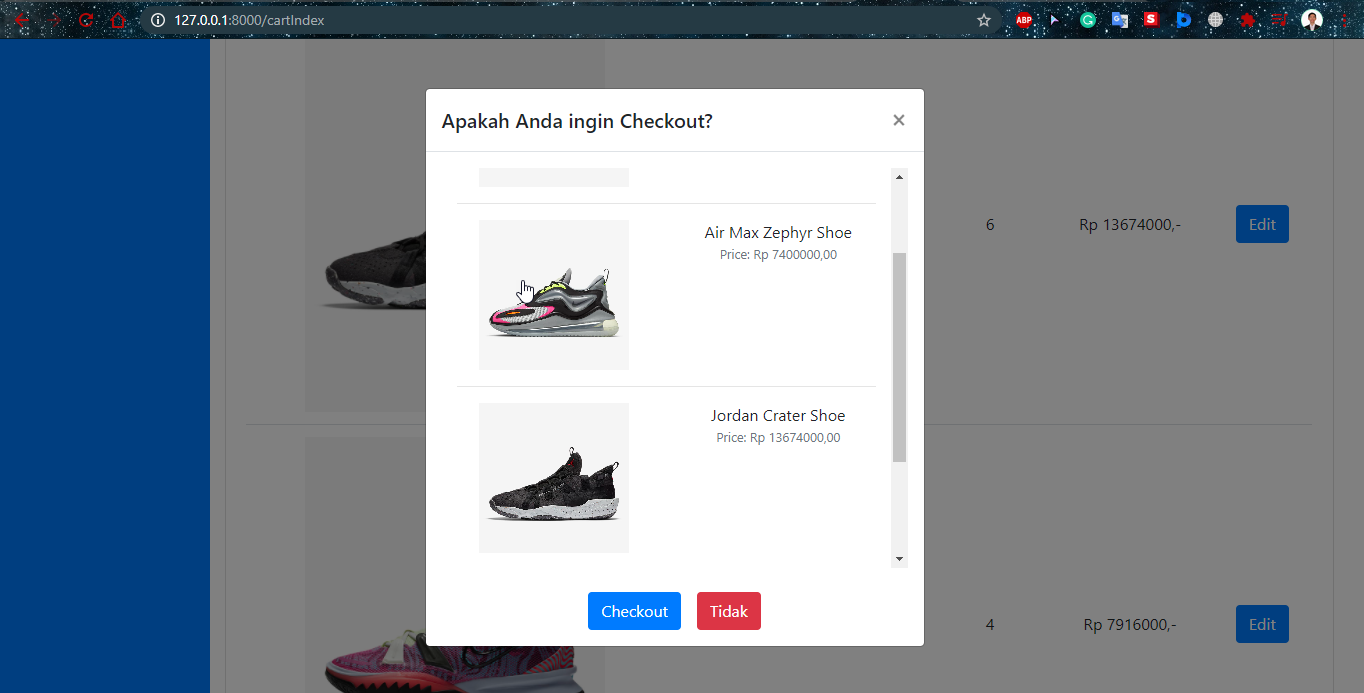
1. Edit Cart’s items

Kita Update jadi 3, akan muncul message yang berbeda

Kita coba delete, akan muncul konfirmasi

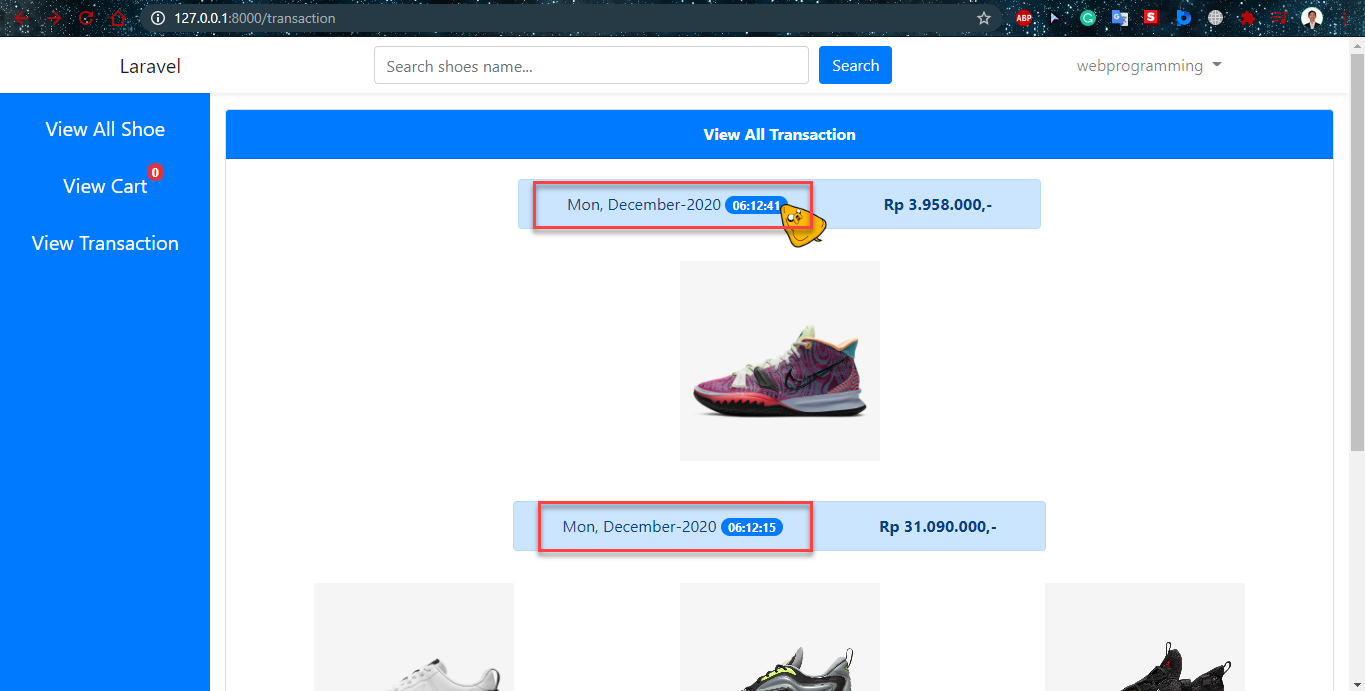
Setelah kita menghapusnya, maka cart akan kosong, notif jadi 0, dan muncul alert message merah.

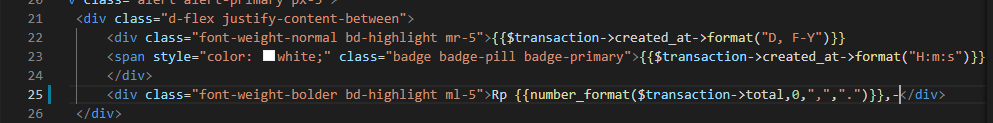
1. Checkout

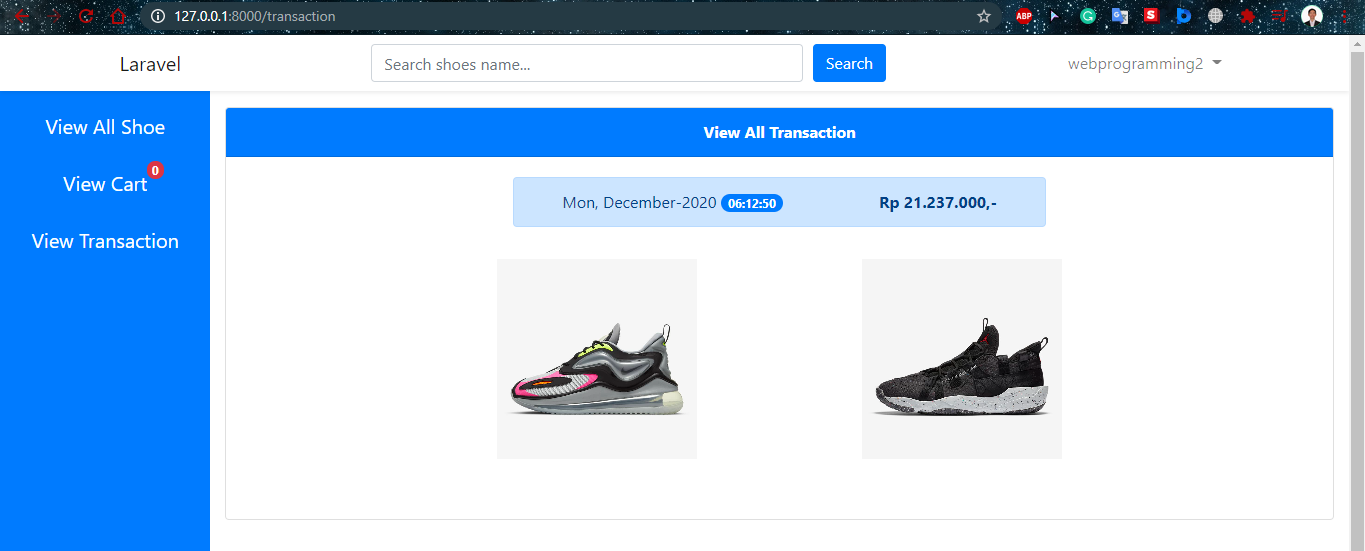
Ketika kita checkout, semua data akan ditampilkan, dan bisa discroll. Foto juga bisa diklik dan akan membawa user ke link foto tersebut, namun membukanya pada halaman baru.

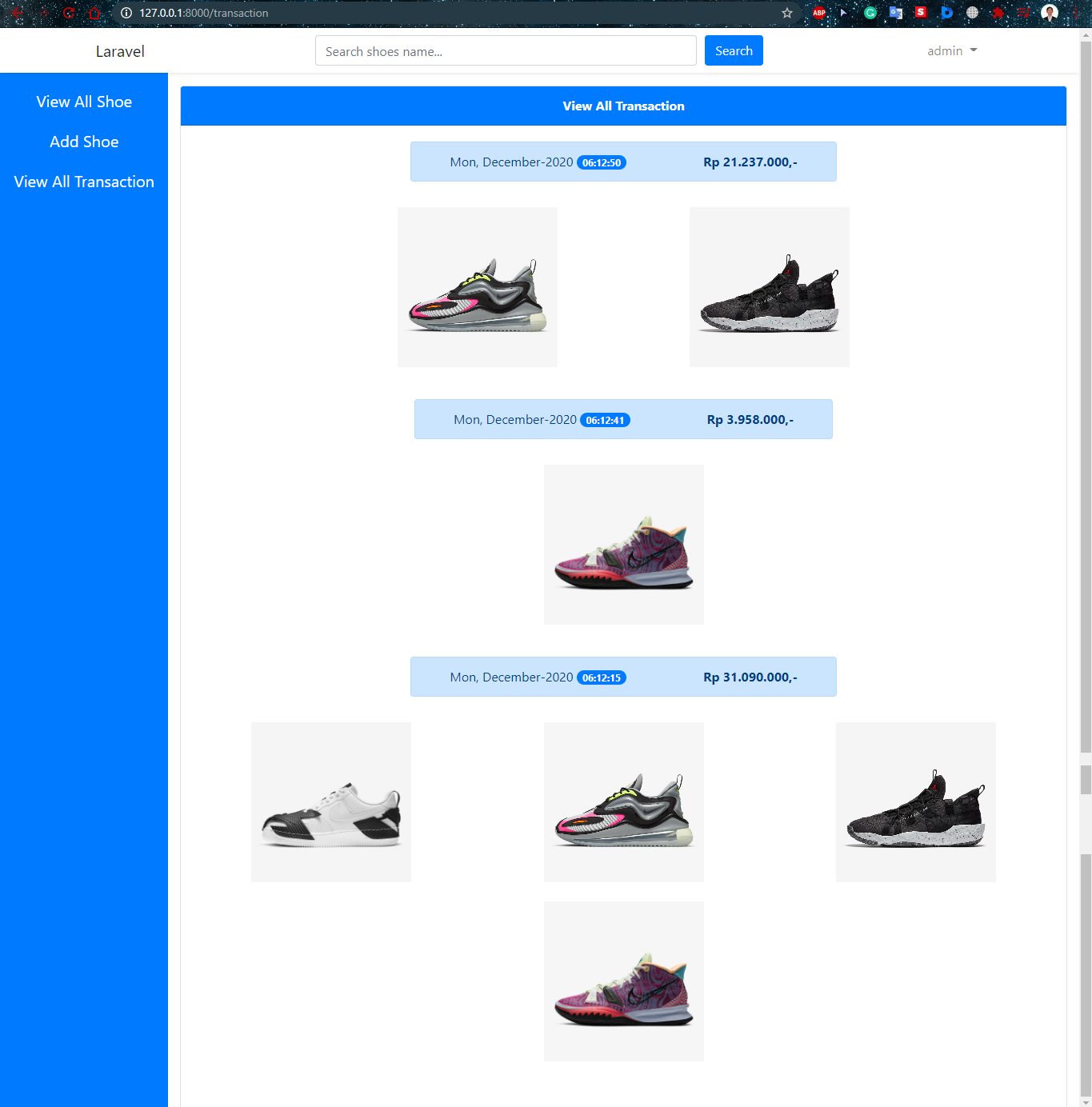
1. Transaction History

Hasil dari checkout kita akan masuk ke dalam View Transaction dan data pada cart akan hilang, ditandai notif yang menjadi 0. Jika kita menambah data baru maka hasilnya historynya akan bertambah sesuai dengan jam/created\_at yang paling baru



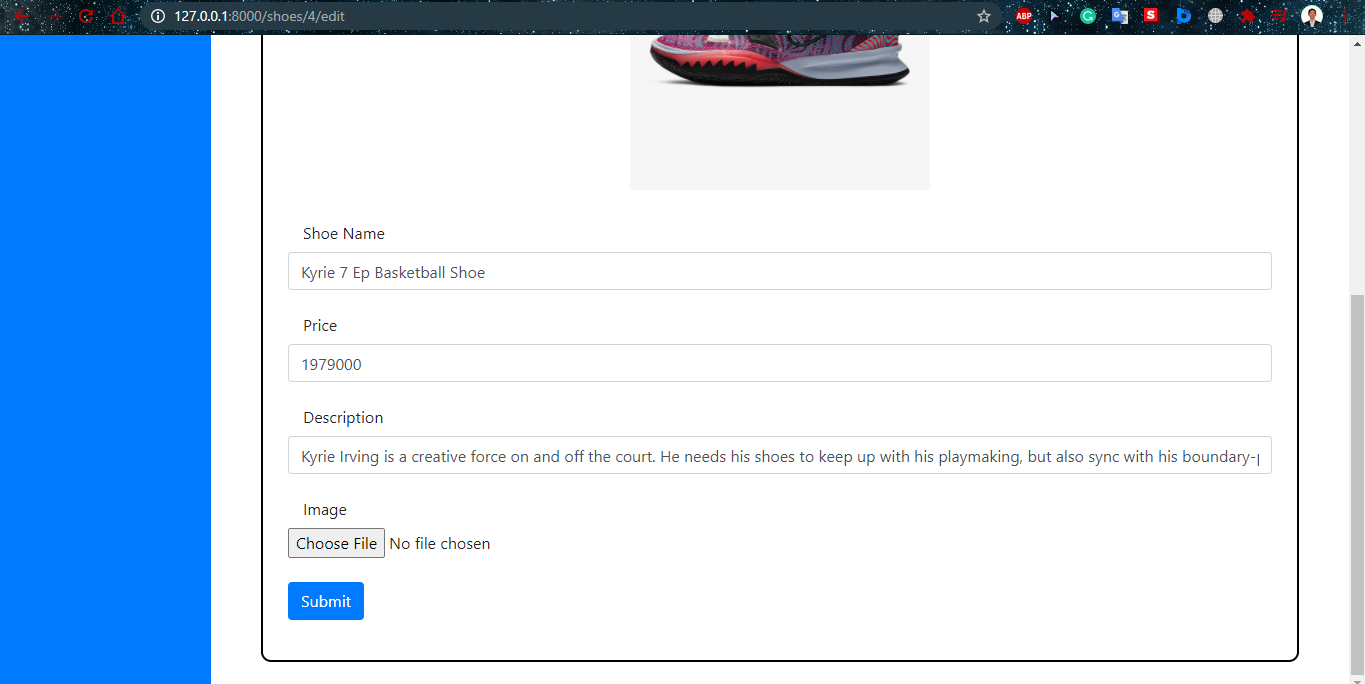
Untuk format angkanya, kami menggunakan fungsi bawaan dari PHP yaitu number\_format, sedangkan untuk tanggalnya kami menggunakan fungsi format:date\_time dari Laravel.

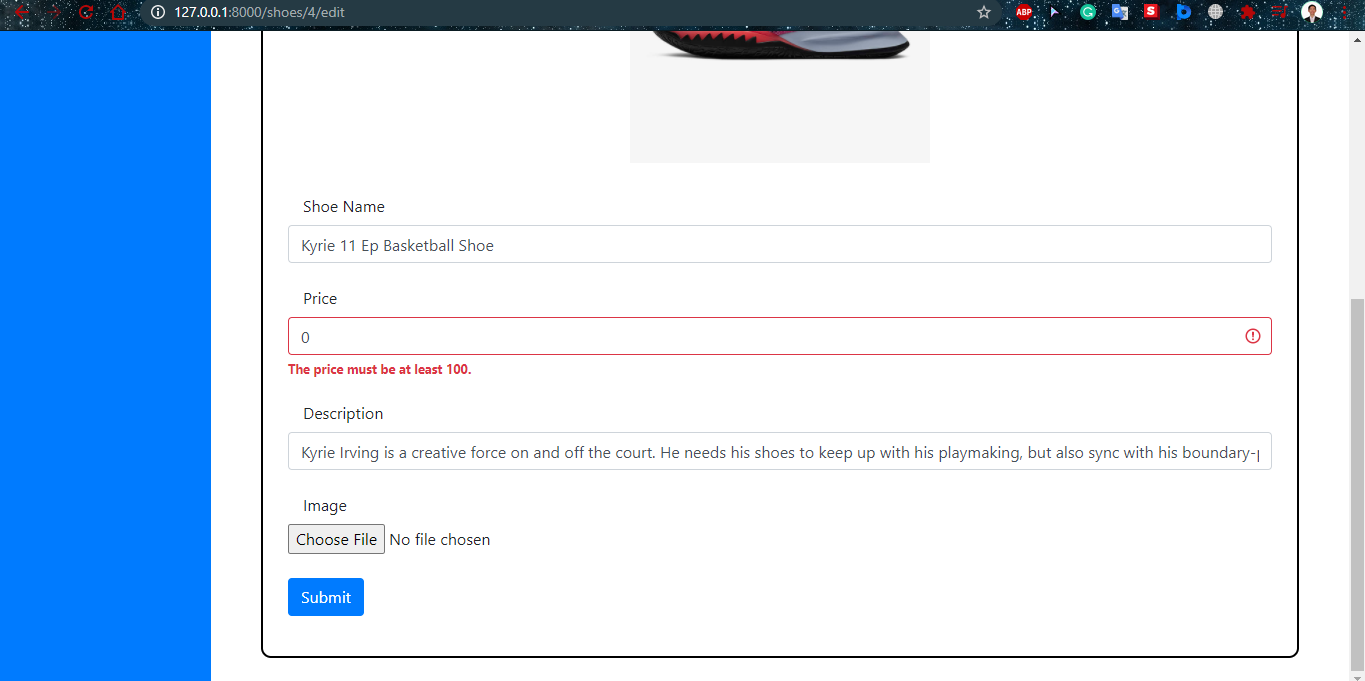
Kemudian, kita coba membuat akun baru dengan lalu lakukan checkout seperti berikut

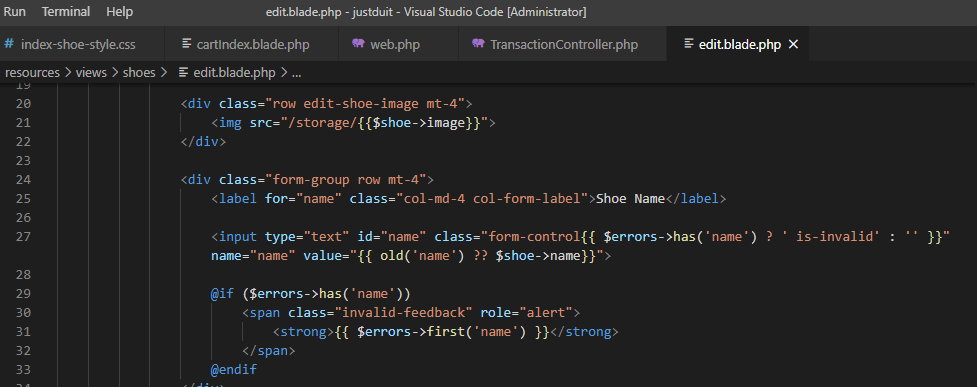
Lalu kita akan menggunakan aku admin untuk melihat semua transaction history

Di sini bisa dilihat, admin bisa melihat semua data transaksi dari yang paling paling baru sampai ke yang paling lama.

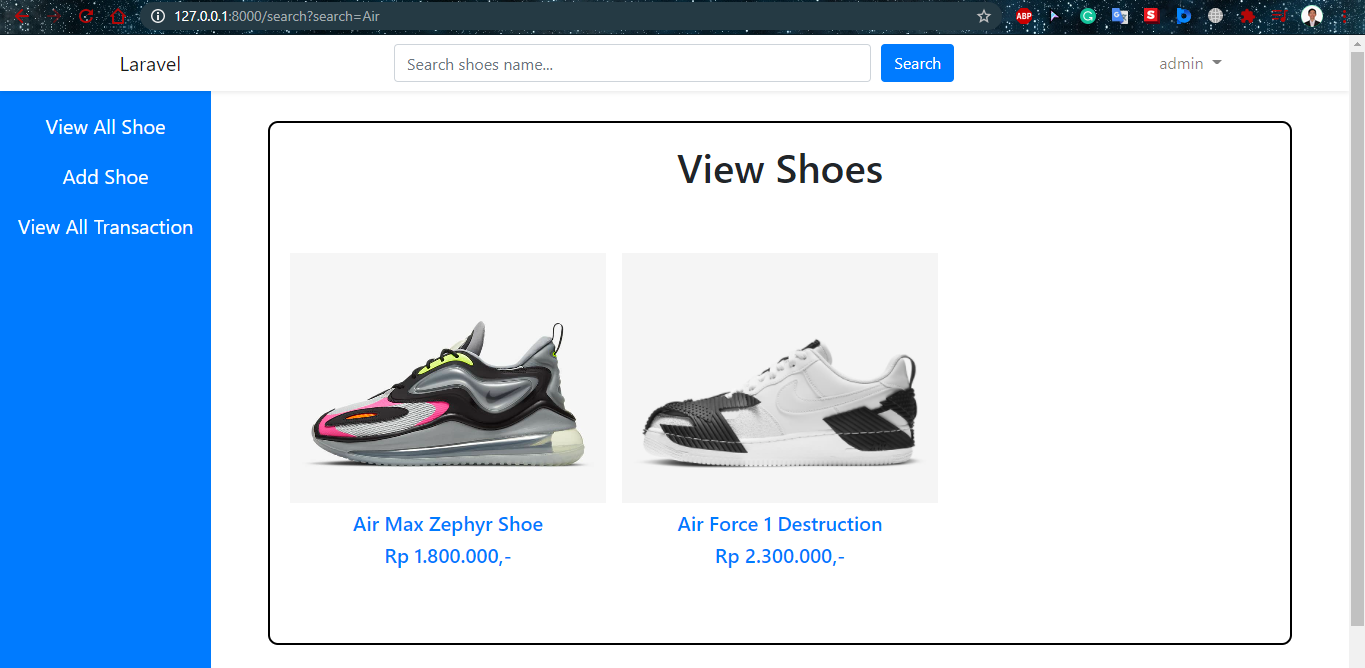
1. Update Shoes

Update shoes hanya muncul dan bisa diakses oleh admin, berikut adalah tampilannya

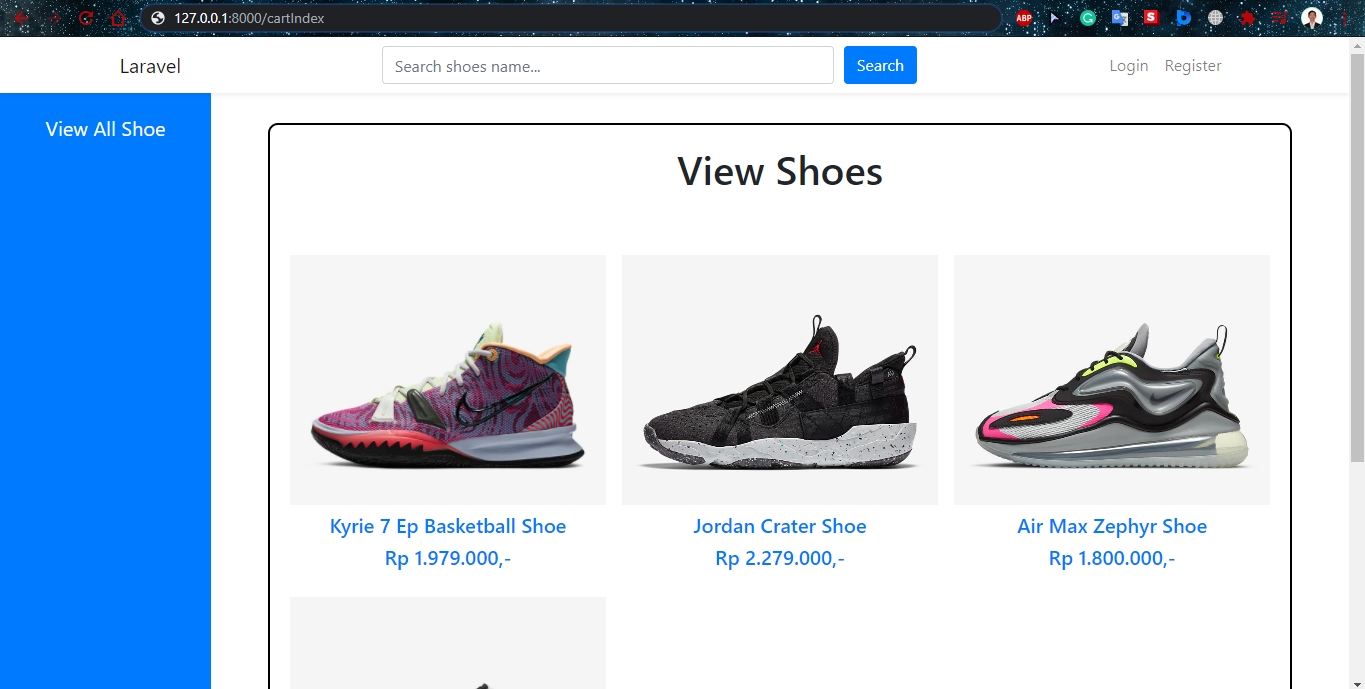
Ketika user memasukkan data, dan ada yang error maka field lain yang sudah diisi tidak akan menjadi kosong, namun mengambil data sebelumnya yang sudah dimasukkan. Misal kita mengganti nama sepatu menjadi Kyrie 11, tapi price tidak sengaja kita set ke 0 dan kesubmit, maka data tidak akan terupdate, namun akan menampilkan data yang sebelumnya user masukkan.

Kami menggunakan fungsi old untuk mengambil data lama yang diinput user

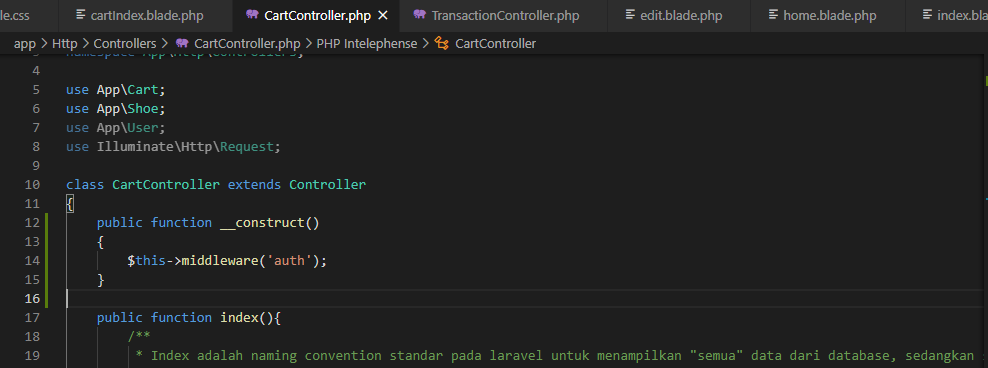
1. Searching

Misal kita searching sepatu dengan nama Air, maka akan muncul semua sepatu yang mengandung Air di dalamnya. Begitu pula misalnya kita searching Basketball.

1. Authentication and Authorization

Tidak semua halaman bisa diakses secara sembarangan, misalnya ketika user yang belum login ingin mengakses halaman /cartIndex yang di mana harusnya user yang sudah terdaftar dan yang sedang login saja yang bisa mengaksesnya, maka akan langsung balik ke menu login

Hal ini karena kami menggunakan middleware(‘auth’) dari Laravel untuk membatasi akses halaman dan fungsi pada controller



Selain itu kita juga bisa menggunakan pengecekan secara manual, dengan mengecek auth user yang sekarang sedang login

Di sini, ketika user tidak login, maka akan masuk ke halaman abort 401, di mana 401 adalah HTTP code untuk memberitahu bahwa user unauthorized dan tidak bisa memasuki halaman tersebut (<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>)

* Reference
* <https://static.nike.com/a/images/t_PDP_864_v1/f_auto,b_rgb:f5f5f5/f1195fb3-5b9b-464b-8169-c318e0e11e61/air-force-1-ndestrukt-shoe-brsx1M.jpg>
* <https://static.nike.com/a/images/t_PDP_864_v1/f_auto,b_rgb:f5f5f5/1663ad51-c357-4818-93ad-b6a07ed1d560/air-max-zephyr-shoe-SDSC8j.jpg>
* <https://static.nike.com/a/images/t_PDP_864_v1/f_auto,b_rgb:f5f5f5/4b7758ea-56e9-450d-945d-32635a1e7131/jordan-crater-shoe-bqx5hT.jpg>
* <https://static.nike.com/a/images/t_PDP_864_v1/f_auto,b_rgb:f5f5f5/b2cee0c4-ffa6-4c28-a7b1-e84d8da7426f/kyrie-7-ep-basketball-shoe-kMwm27.jpg>
* <https://laracasts.com/discuss/channels/laravel/show-integer-price-separate-with-comma-or-dot?page=1>
* <https://youtu.be/ucV7ynY4M8A>
* <https://youtu.be/7daBdm2xgm8>
* <https://laravel.com/docs/7.x/>
* Group Member
* 2201757093 – Jackie Leonardy
* 2201758303 – Luis Genesius