


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.

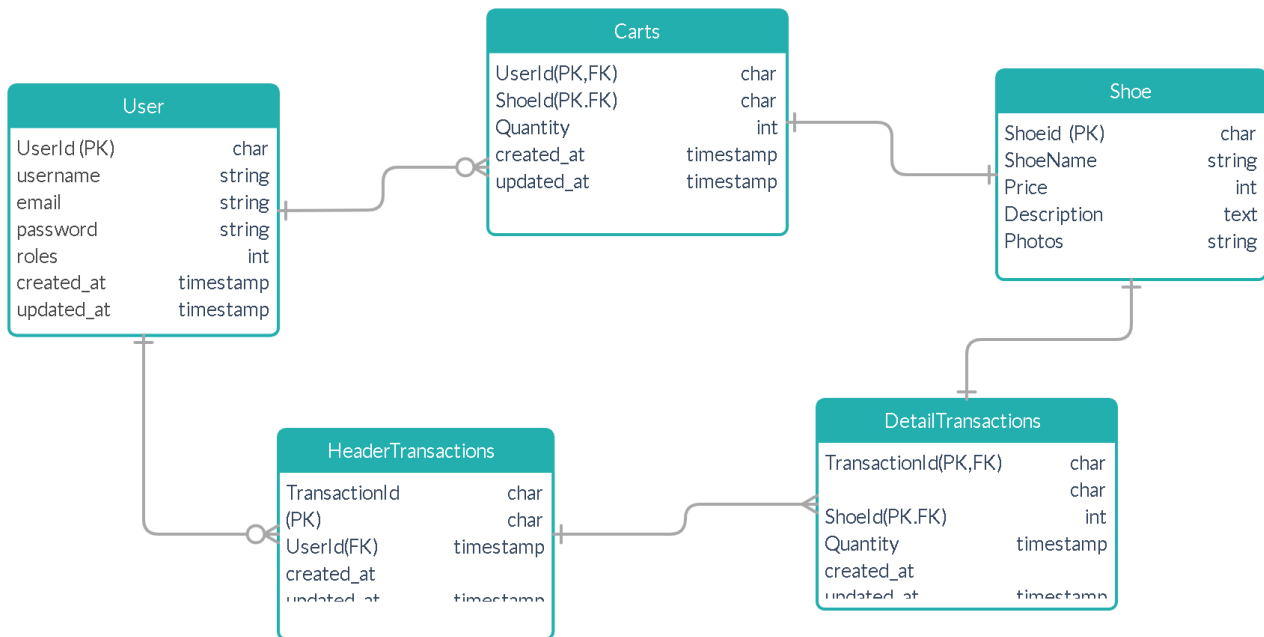
Di sini, user bisa melakukan **CRUD** khusus untuk cart saja, di mana user bisa **Create** untuk memasukkan shoes ke dalam Cart, **Read** untuk melihat isi dari cart, **Update** untuk meng-update isi cart, dan **Delete** untuk menghapus item dalam cart. User juga bisa melakukan **check-out**, di mana 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.

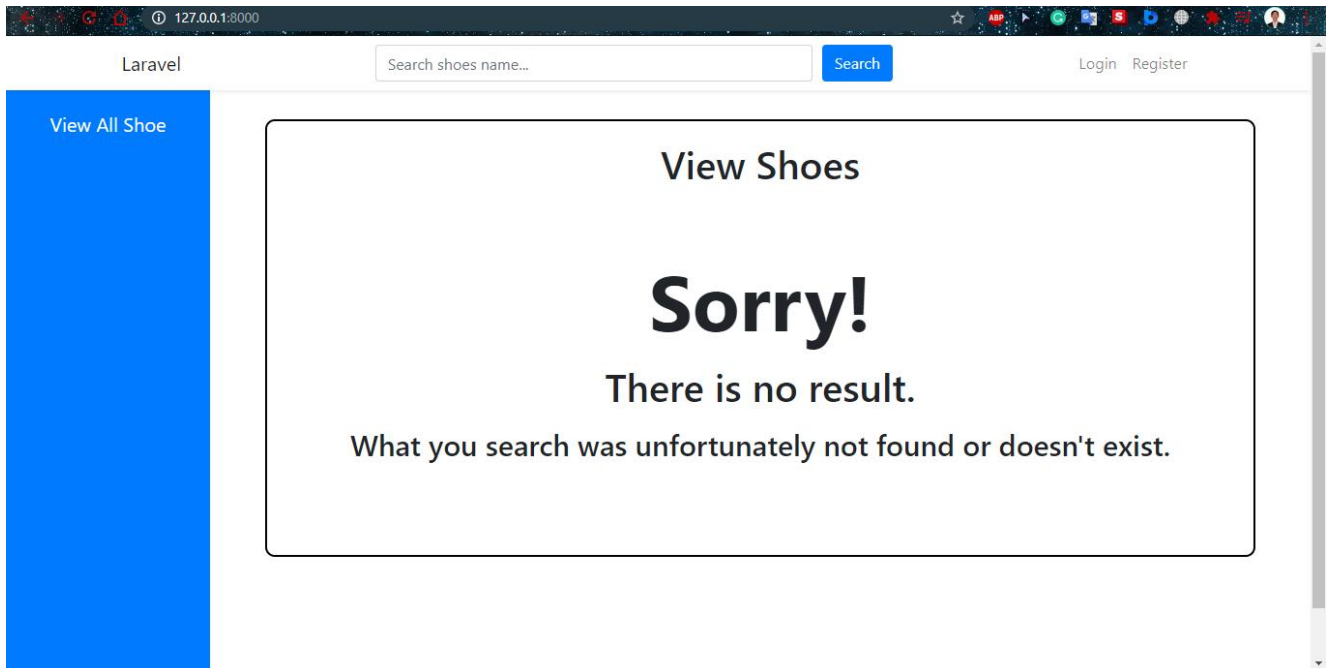
<https://laravel.com/docs/7.x/eloquent-relationships#many-to-many>

Berikut adalah table data pada database kami

Tabel	Tindakan	Beri
<input type="checkbox"/> detail_transactions	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	
<input type="checkbox"/> failed_jobs	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	
<input type="checkbox"/> header_transactions	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	
<input type="checkbox"/> migrations	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	
<input type="checkbox"/> password_resets	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	
<input type="checkbox"/> shoes	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	
<input type="checkbox"/> shoe_user	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	
<input type="checkbox"/> users	★ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	

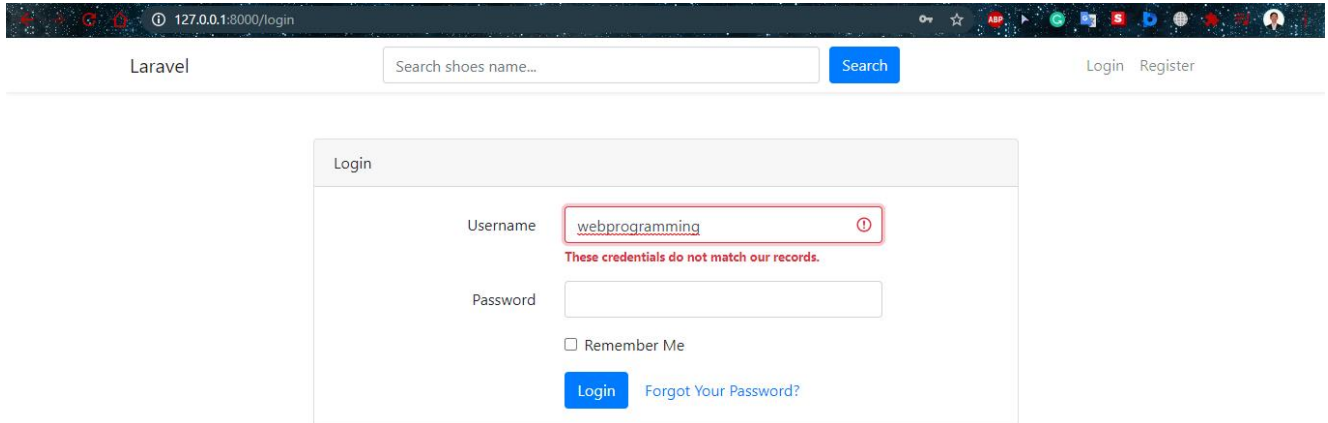
2. Home Page

Tampilan Homepage, jika data kosong

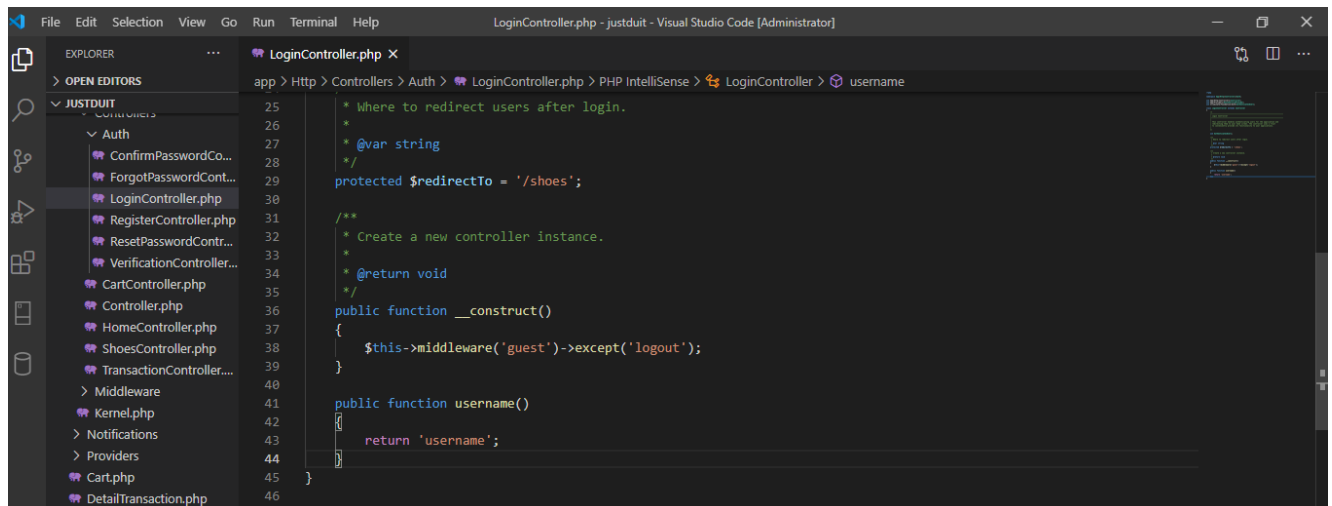


3. Login & Registration

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

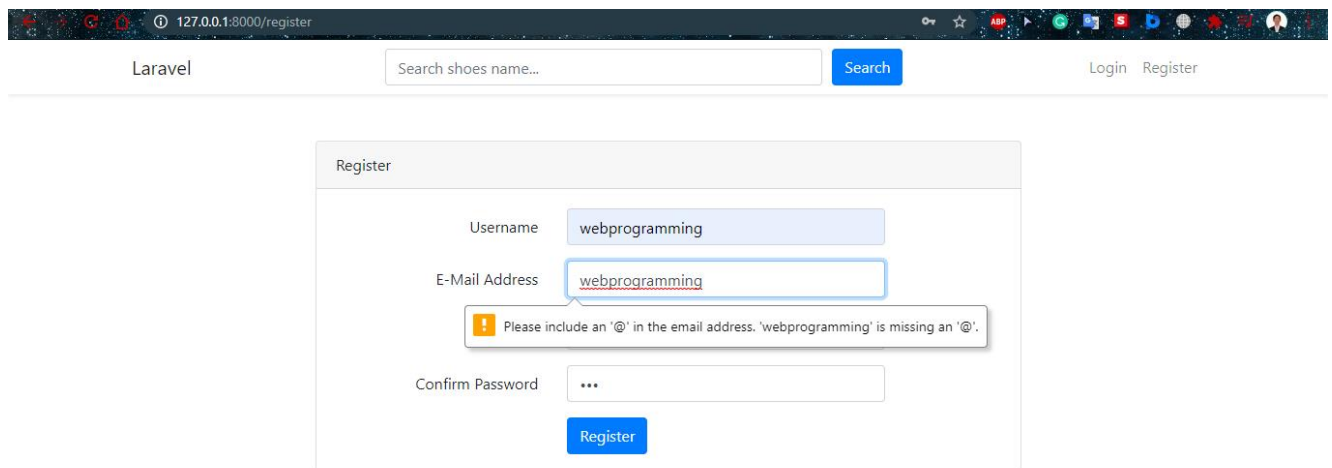


Di sini kami mengatur untuk login menggunakan username, bisa saja pakai email dan kami menggantinya pada login controller



```
25 * Where to redirect users after login.
26 *
27 * @var string
28 */
29 protected $redirectTo = '/shoes';
30
31 /**
32 * Create a new controller instance.
33 *
34 * @return void
35 */
36 public function __construct()
37 {
38     $this->middleware('guest')->except('logout');
39 }
40
41 public function username()
42 {
43     return 'username';
44 }
45
46 }
```

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 terregistrasi.



Laravel

Search shoes name... Search

Login Register

Register

Username webprogramming

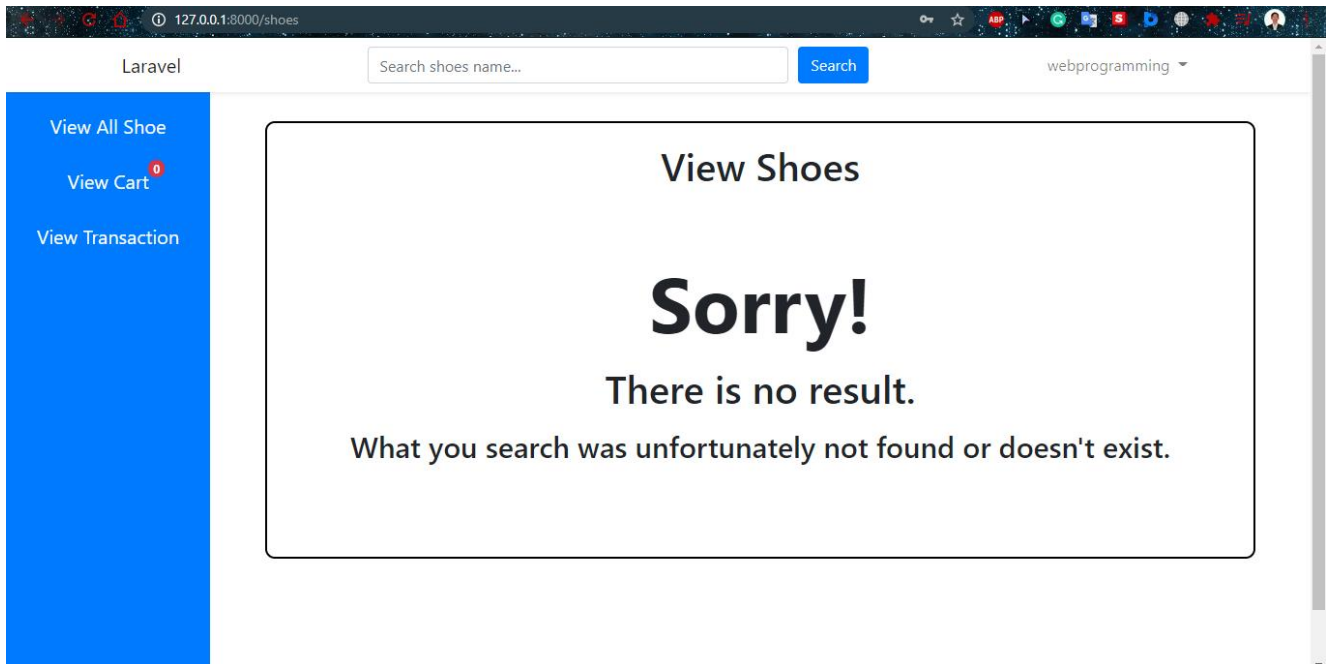
E-Mail Address webprogramming

Please include an '@' in the email address. 'webprogramming' is missing an '@'.

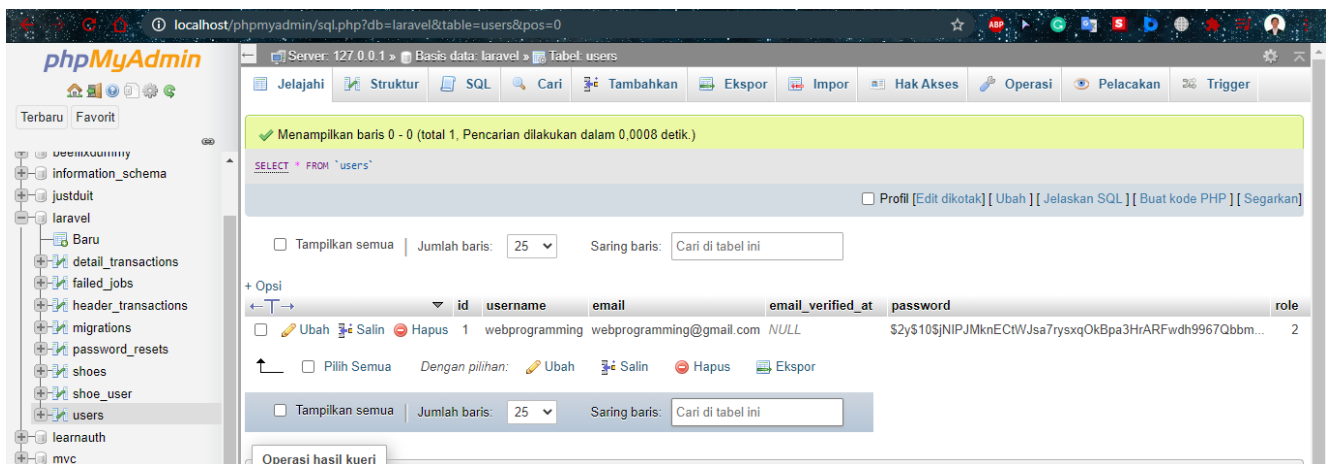
Confirm Password ...

Register

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.

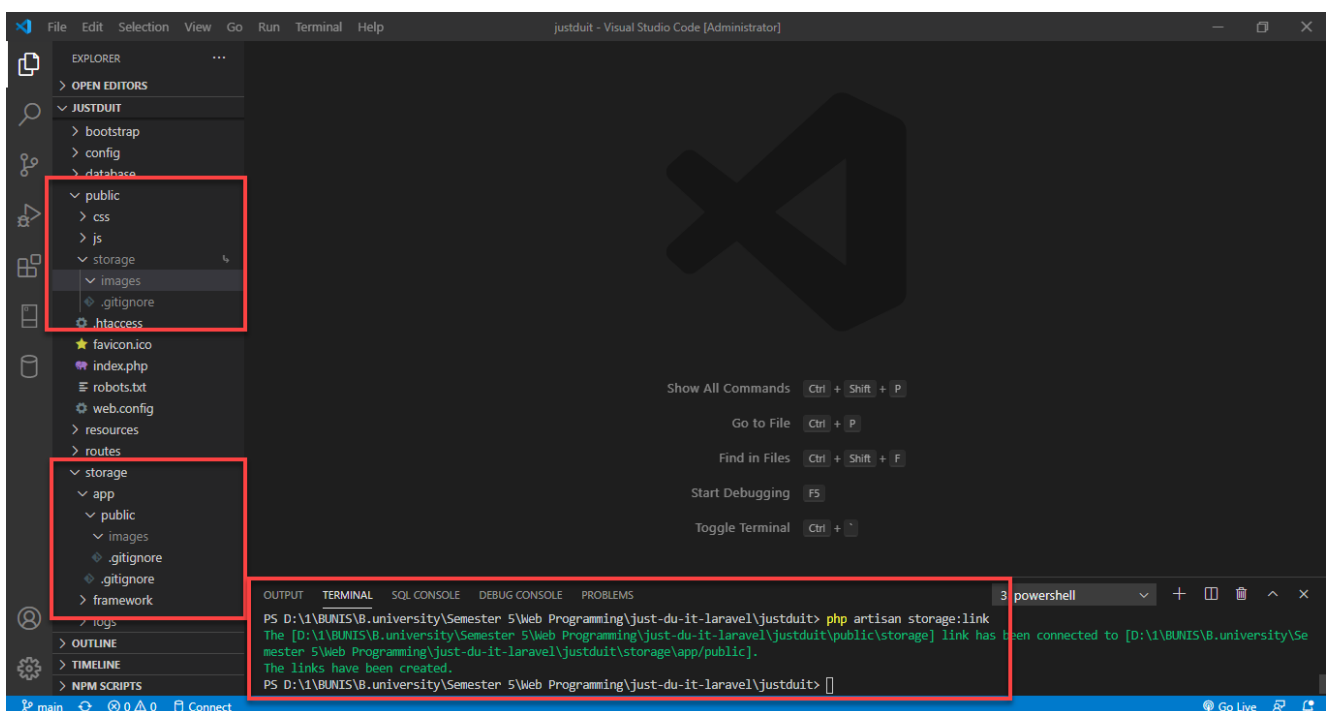
4. Storage Link

Image yang diupload kami tamping pada storage public, hal ini dimaksudkan 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 saja yang bisa diakses oleh internal dan bisa dikendalikan.

Untuk Storage link, kita bisa menggunakan command

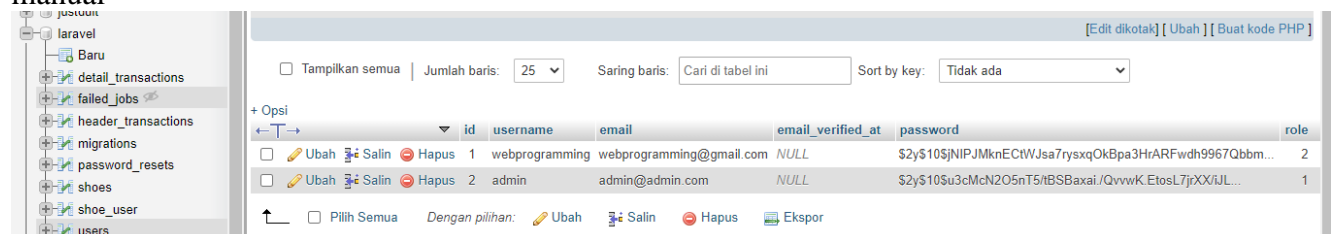
```
php artisan storage:link
```

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

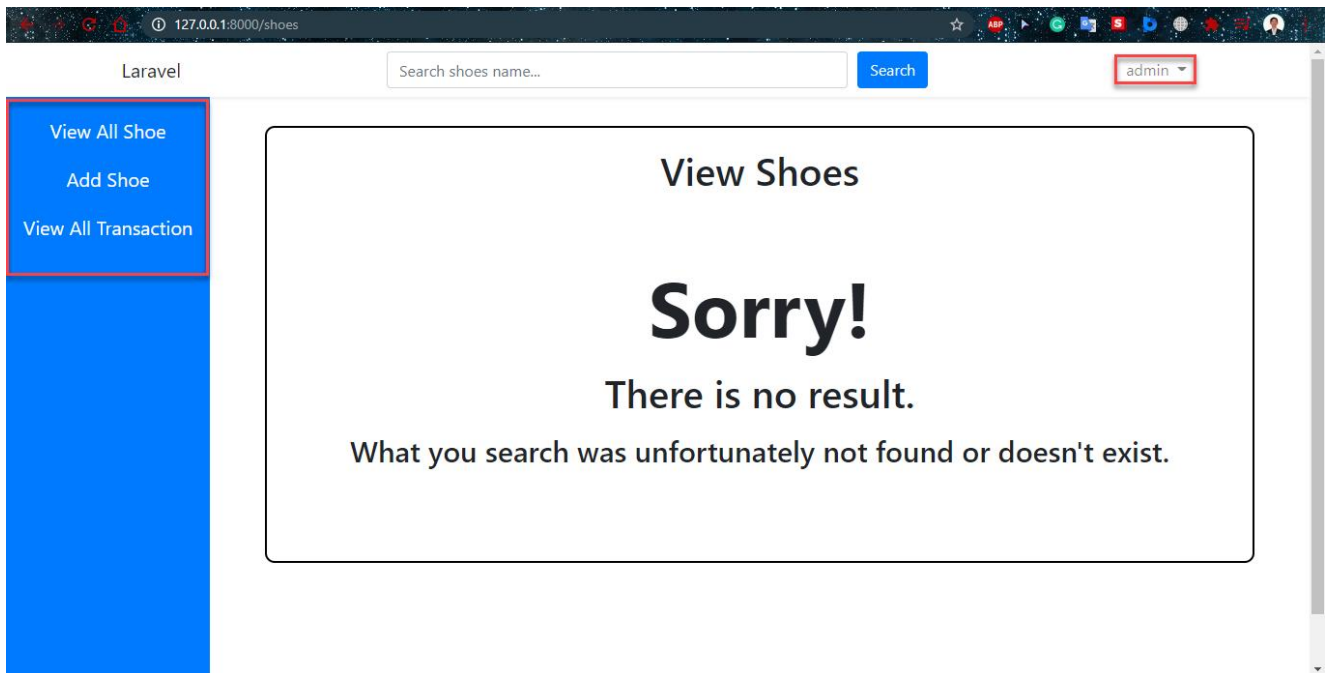


5. Adding Shoes

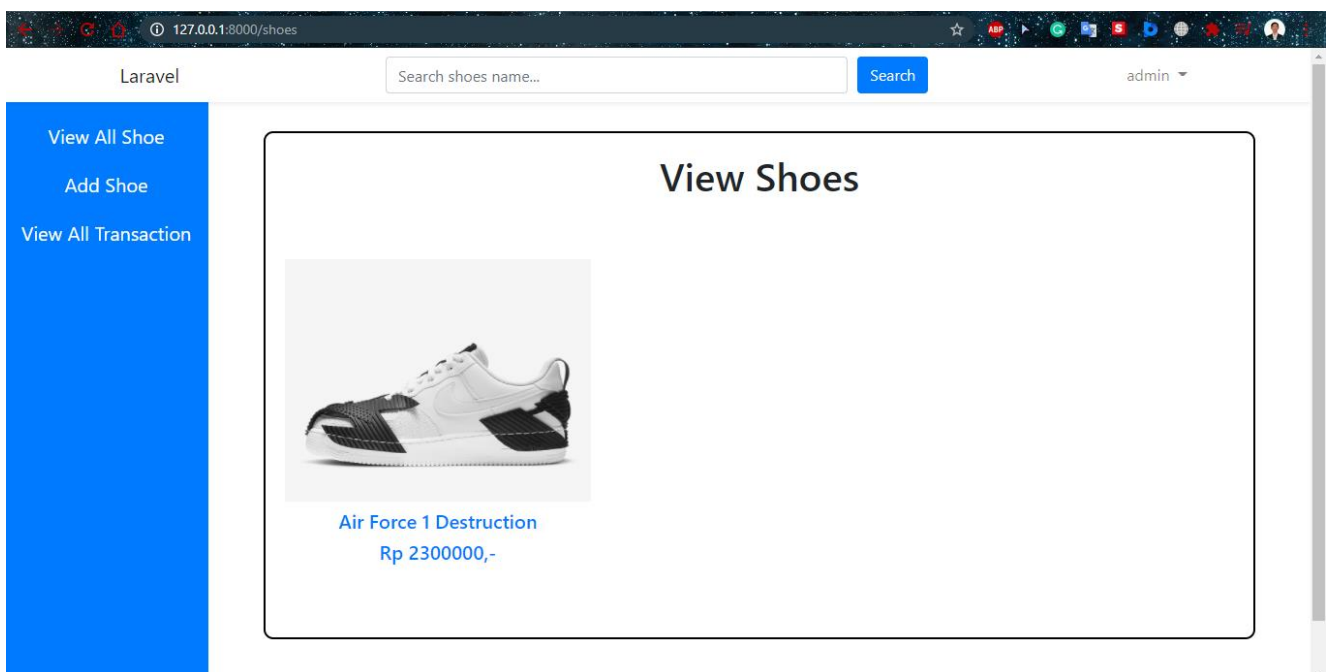
Untuk admin, seperti yang sudah diterangkan di atas, kita akan mengatur rolenya secara manual



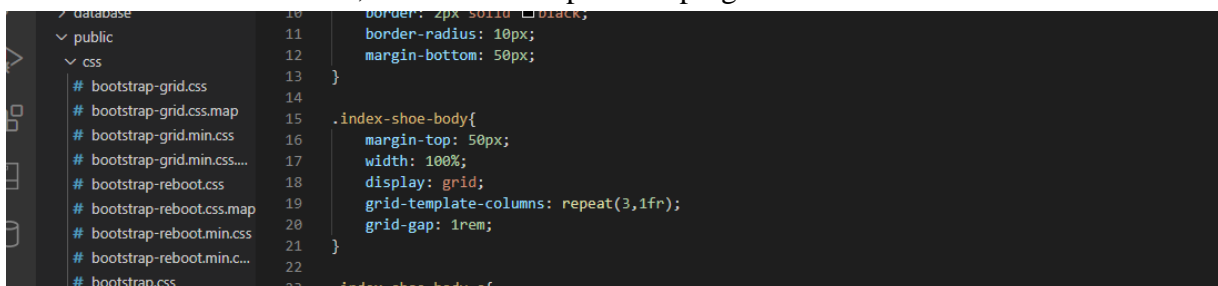
Tampilan side-bar untuk admin juga berbeda, terdapat Add Shoe dan View All Transaction.

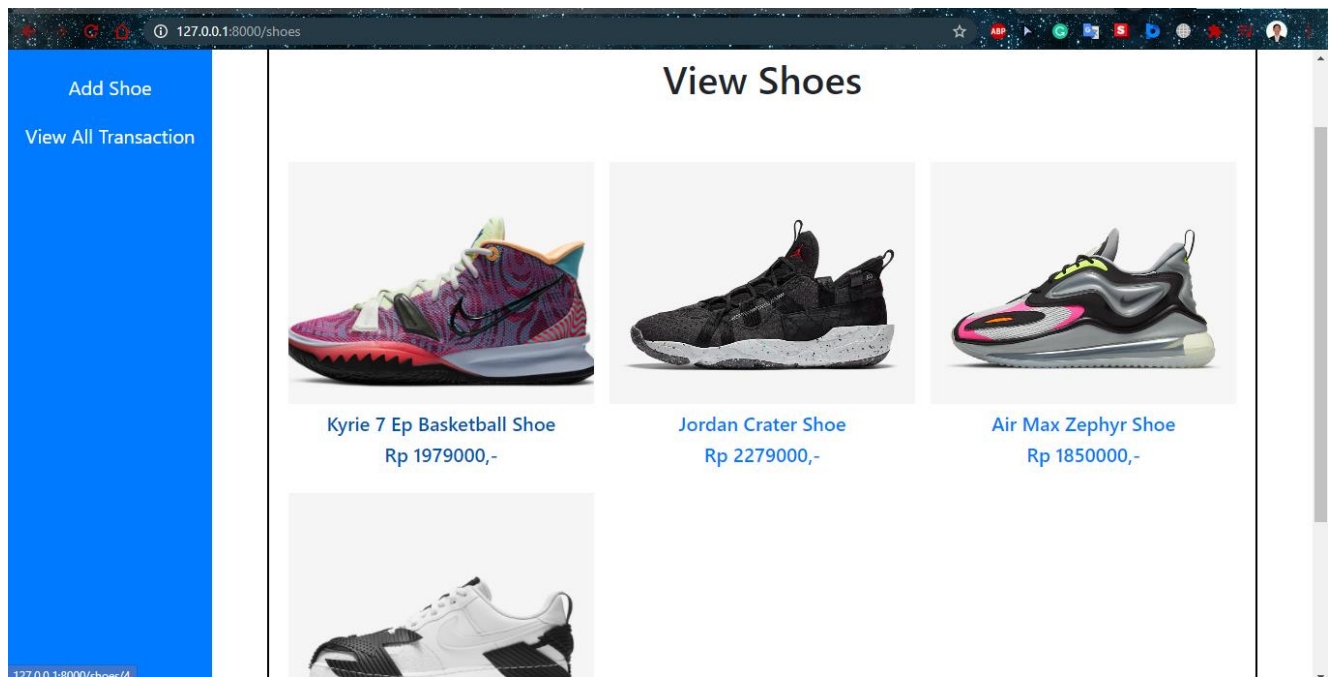


Setelah kita input, hasilnya menjadi seperti ini.

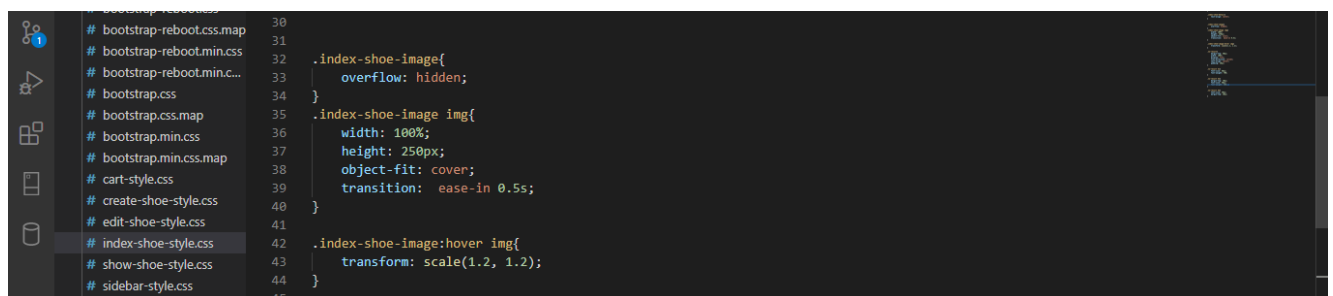


Mari kita coba input lebih banyak data, di sini kami menggunakan Grid sehingga ketika data ditambahkan lebih dari 3, maka akan wrap ke samping.

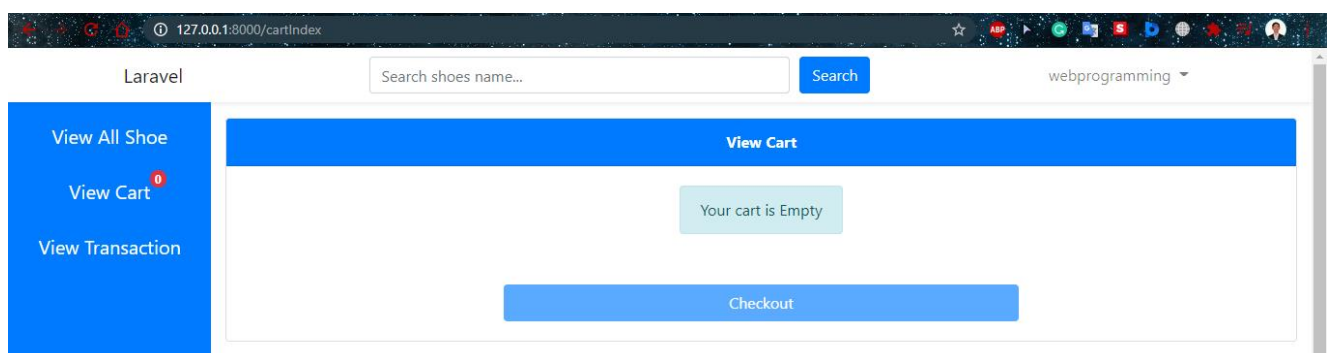




Ketika foto dihover, maka akan ada animasi zoom menggunakan transform dan scale dari CSS.



6. 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**


```

33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
@elseif ($carts as $cart)
<tbody>
<tr>
<td>
<a href="/shoes/{{ $cart->shoe_id }}" target=" blank">

</a>
</td>
<td class="align-middle"><p class="card-text">{{ $cart->name }}</p></td>
<td class="align-middle"><p class="card-text">{{ $cart->quantity }}</p></td>
<td class="align-middle"><p class="card-text">Rp {{ $cart->price * $cart->quantity }}</p></td>
<td class="align-middle"><a href="/cartUpdate/{{ $cart->shoe_id }}/edit" class="btn btn-primary">Edit</a></td>
</tr>
</tbody>
@empty
<div class="d-flex justify-content-center">
<div class="alert alert-info">
Your cart is Empty
</div>
</div>
</tbody>
</endif>

```

Di sini kamu menggunakan **forelse**, bukan **foreach**. Kenapa? Hal ini karena dengan adanya fforelse, 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

```

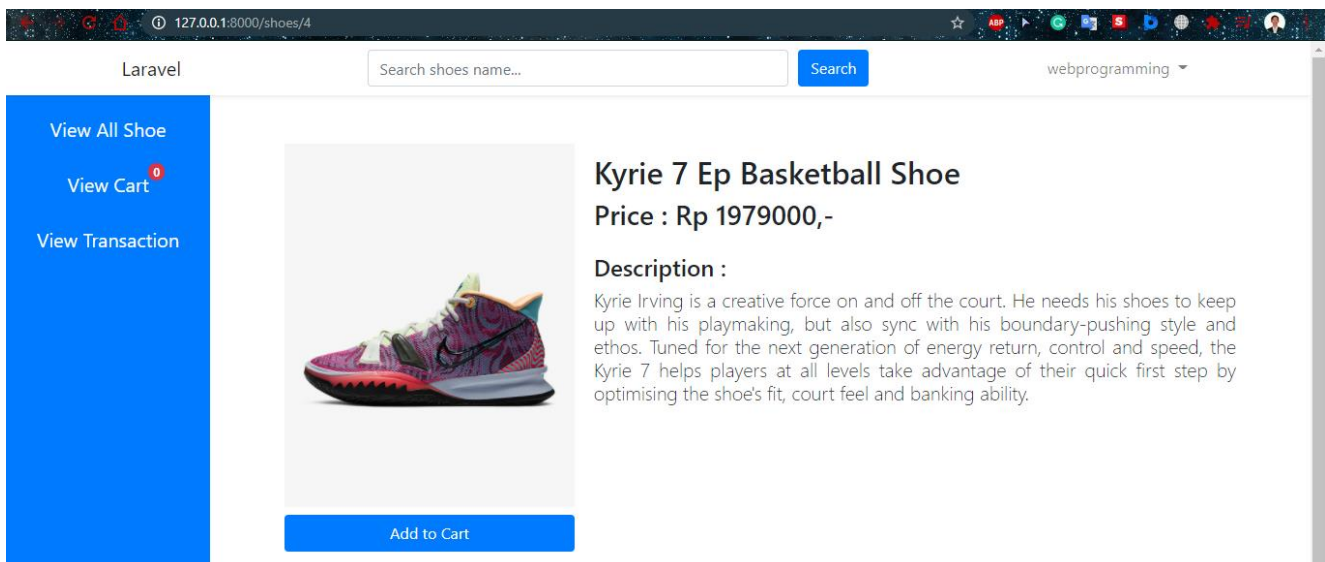
8
9
<a href="/transaction" data-toggle="modal" data-target="#exampleModal" class="btn btn-primary {
{{ $carts->isEmpty() ? 'disabled' : '' }} checkout">Checkout</a>

```

7. Adding item to Cart

Ketika foto diklik, maka berikut tampilannya pada beberapa user

User yang sudah login



User yang tidak Login

The screenshot shows a web browser window with the URL `127.0.0.1:8000/shoes/4`. The page has a header with the word "Laravel" on the left, a search bar with the placeholder text "Search shoes name..." and a blue "Search" button, and links for "Login" and "Register" on the right. A blue sidebar on the left contains the link "View All Shoe". The main content area displays a product card for the "Kyrie 7 Ep Basketball Shoe". The card includes a side-view image of a purple and pink sneaker, the product name, the price "Price : Rp 1979000,-", and a description: "Kyrie Irving is a creative force on and off the court. He needs his shoes to keep up with his playmaking, but also sync with his boundary-pushing style and ethos. Tuned for the next generation of energy return, control and speed, the Kyrie 7 helps players at all levels take advantage of their quick first step by optimising the shoe's fit, court feel and banking ability."

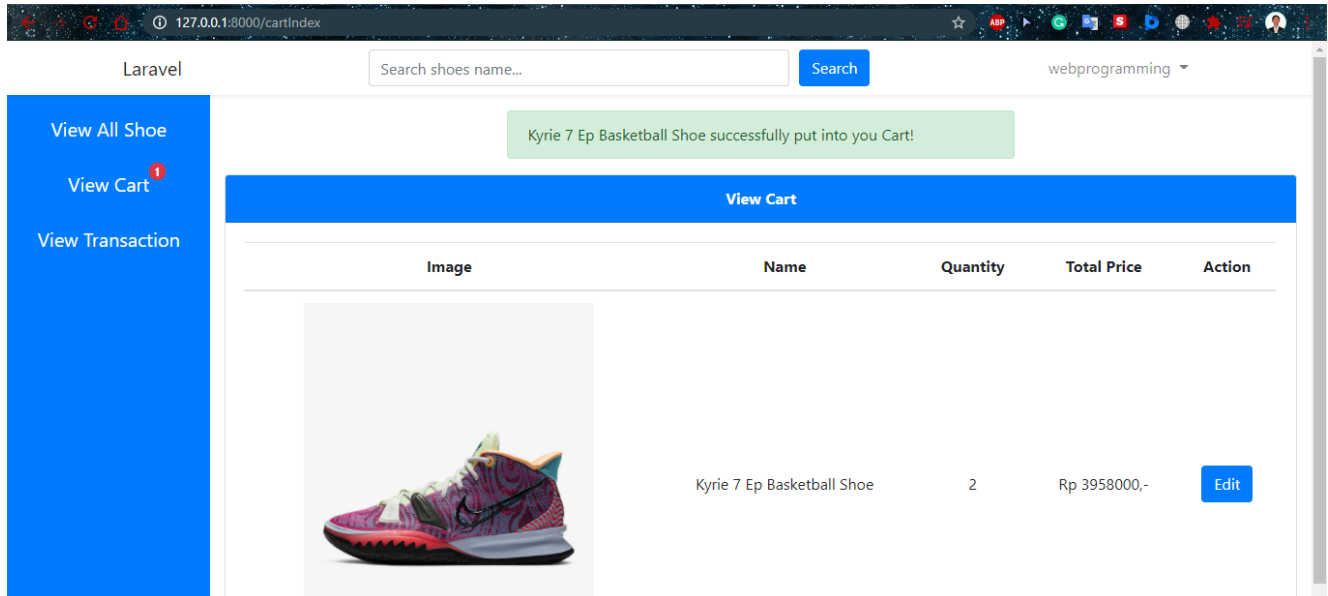
Admin

The screenshot shows the same web browser window with the URL `127.0.0.1:8000/shoes/4`. The header is identical to the previous view, but the user is now logged in as "admin", indicated by a dropdown menu next to the "Search" button. The blue sidebar on the left now contains three links: "View All Shoe", "Add Shoe", and "View All Transaction". The main content area displays the same product card for the "Kyrie 7 Ep Basketball Shoe". Below the product image, there is a blue button labeled "Update Shoe".

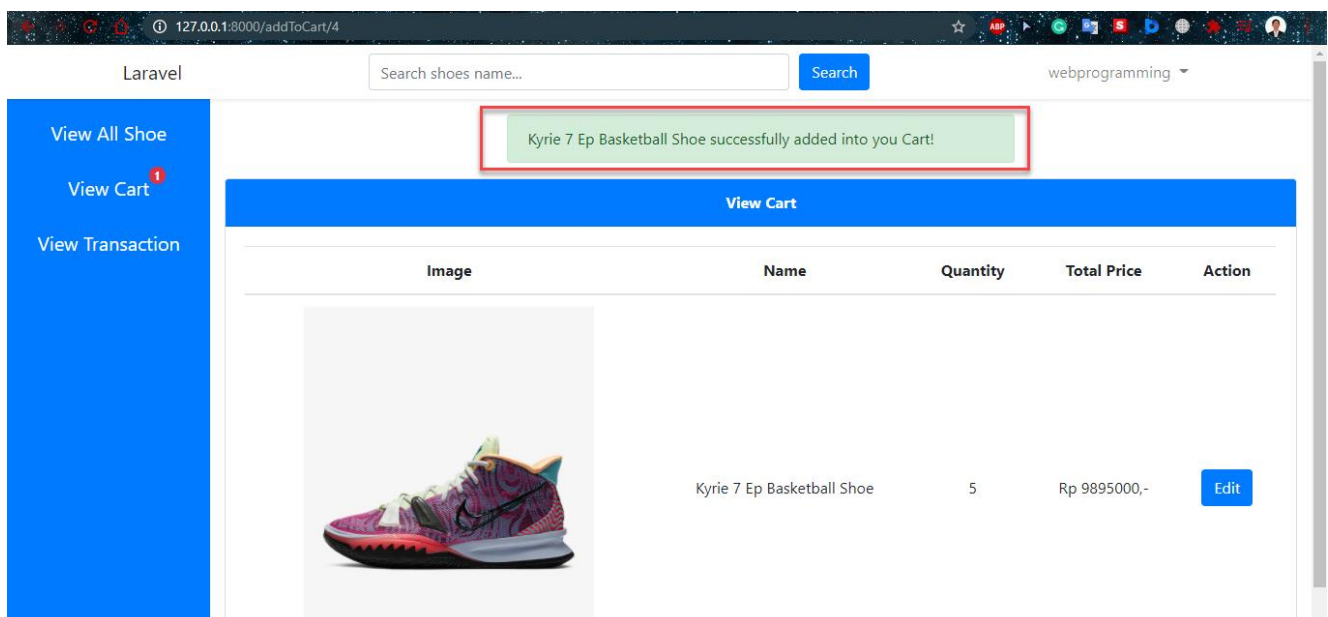
Mari kita tambahkan shoe ke dalam cart

The screenshot shows the web browser window with the URL `127.0.0.1:8000/addToCart/4`. The header is identical to the previous views, but the user is now logged in as "webprogramming", indicated by a dropdown menu next to the "Search" button. The blue sidebar on the left contains three links: "View All Shoe", "View Cart" (which has a red badge with the number "0"), and "View Transaction". The main content area displays the product card for the "Kyrie 7 Ep Basketball Shoe". Below the product image, there is a "Quantity" input field with the value "2" and a blue "Add to Cart" button.

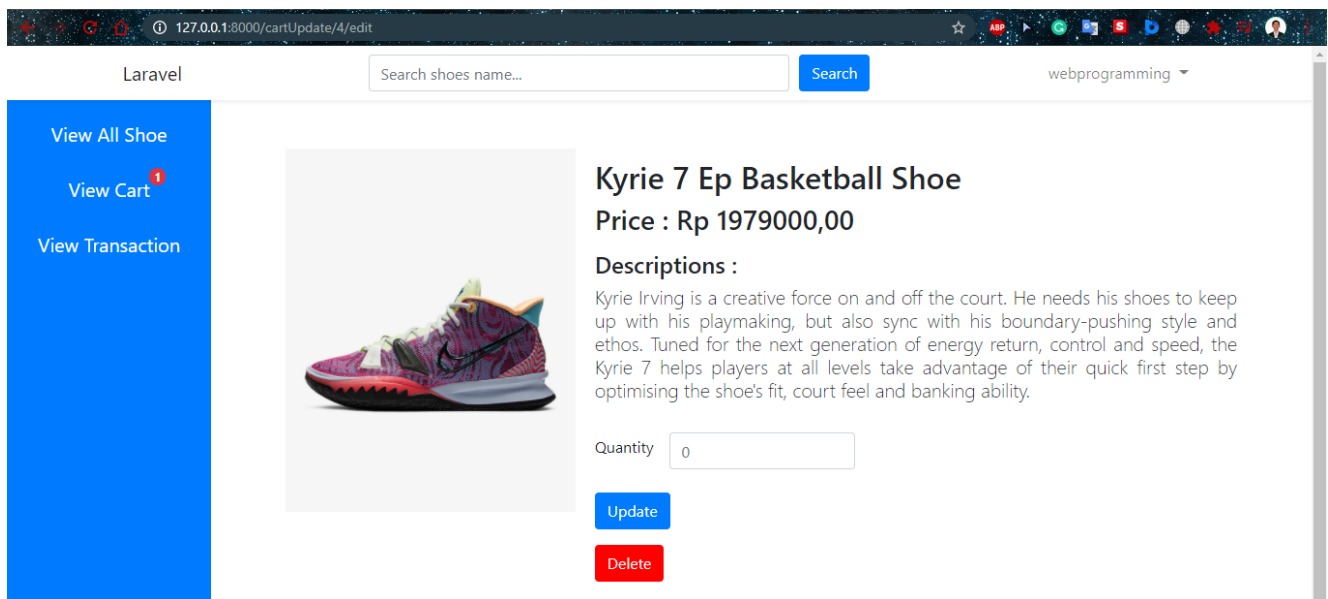
Maka akan ada alert Message yang dikirim melalui session flash, dan Notifikasi pada View Cart akan menyesuaikan data pada cart



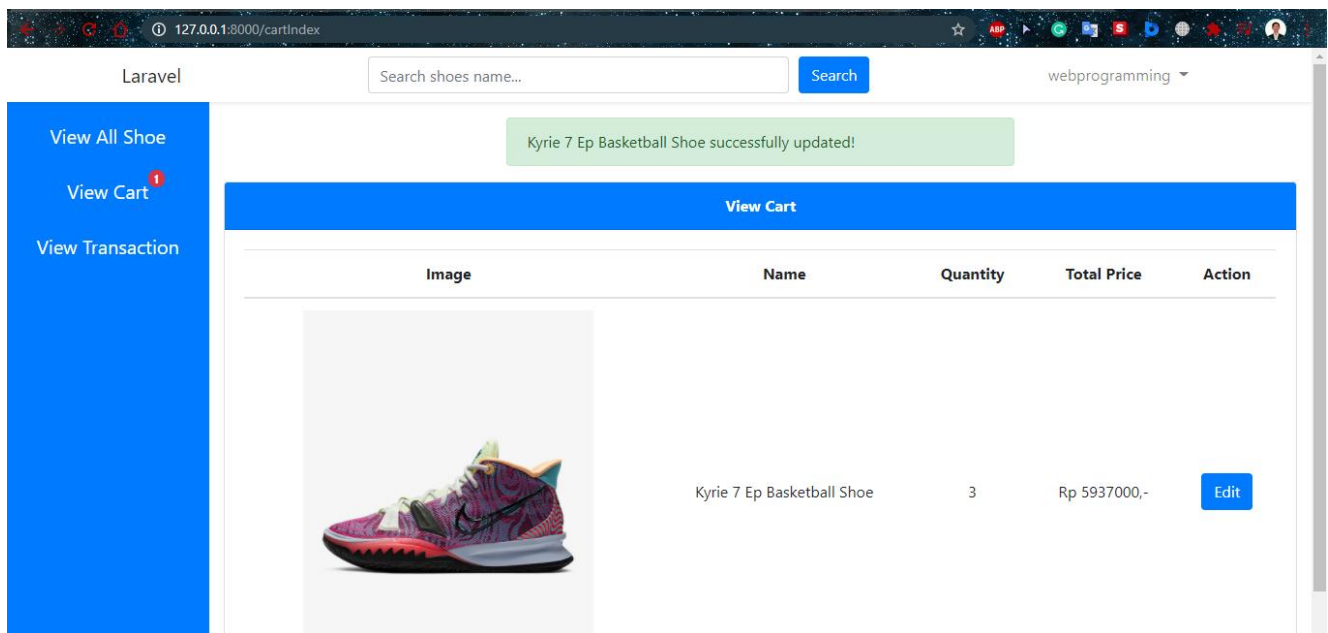
Mari kita coba tambahkan quantity 3 pada shoe yang sama, bisa kita lihat yang bertambah adalah quantity dari $2 + 3 = 5$, dan hal ini dibuktikan dari alert message yang berbeda, kalau misal datanya sudah ada maka akan tertulis “**added**”, kalau belum ada akan tertulis “**put**”



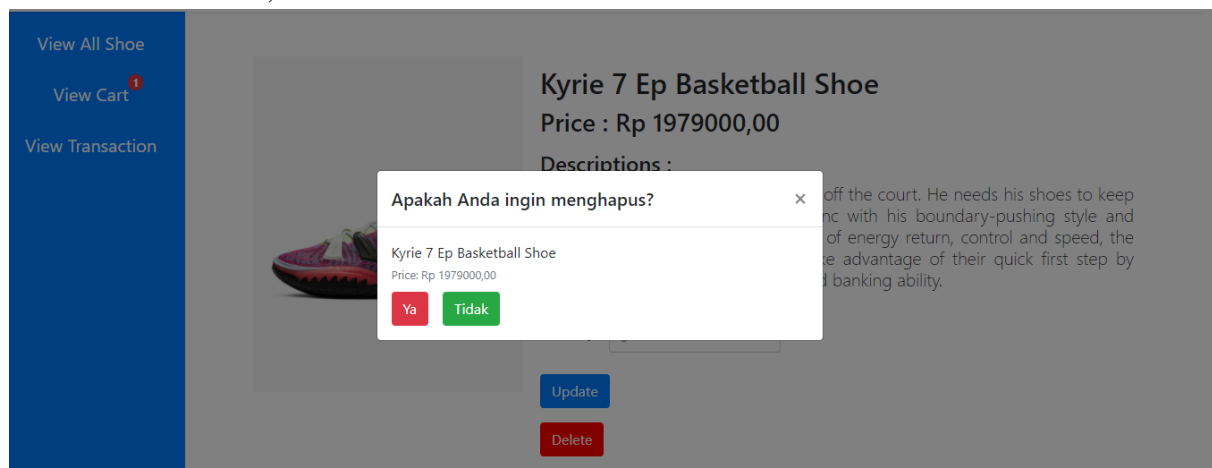
8. Edit Cart's items



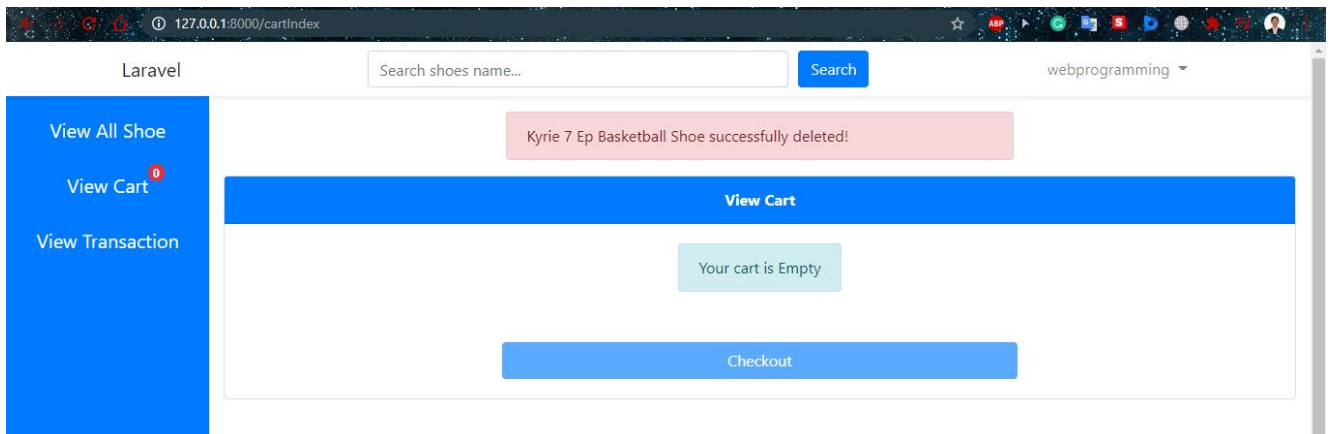
Kita Update jadi 3, akan muncul message yang berbeda



Kita coba delete, akan muncul konfirmasi

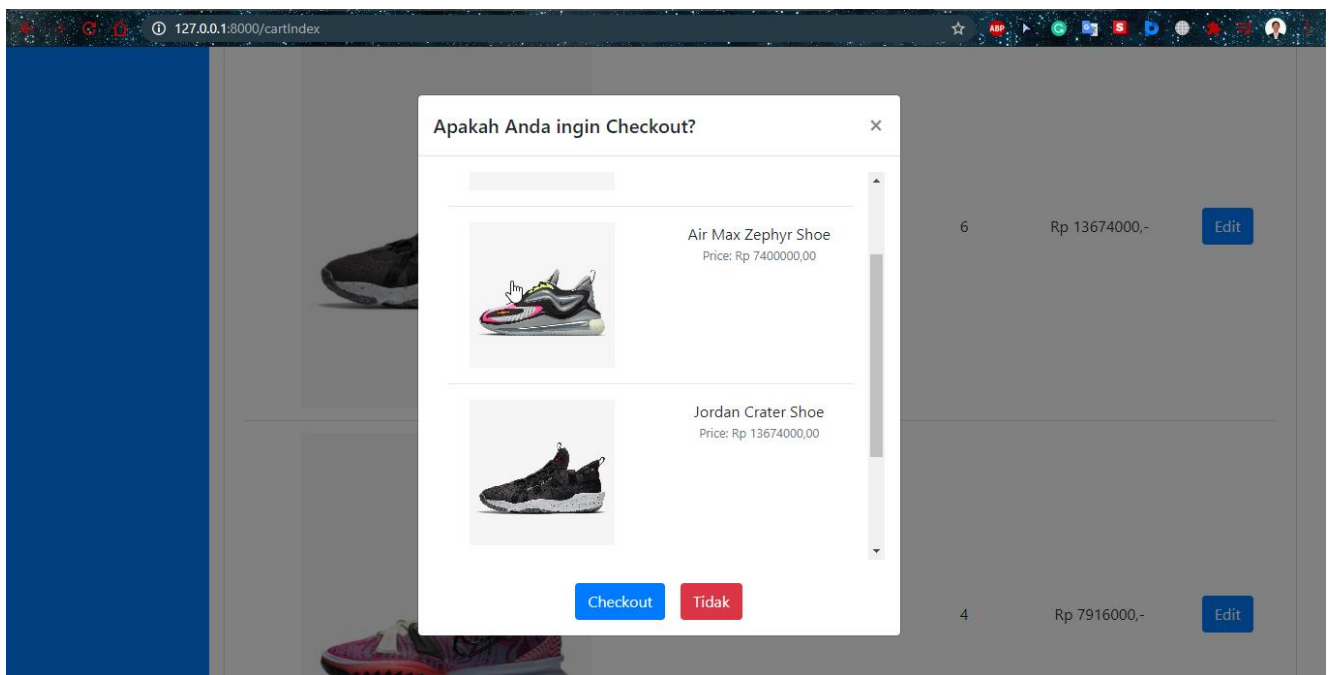


Setelah kita menghapusnya, maka cart akan kosong, notif jadi 0, dan muncul alert message merah.



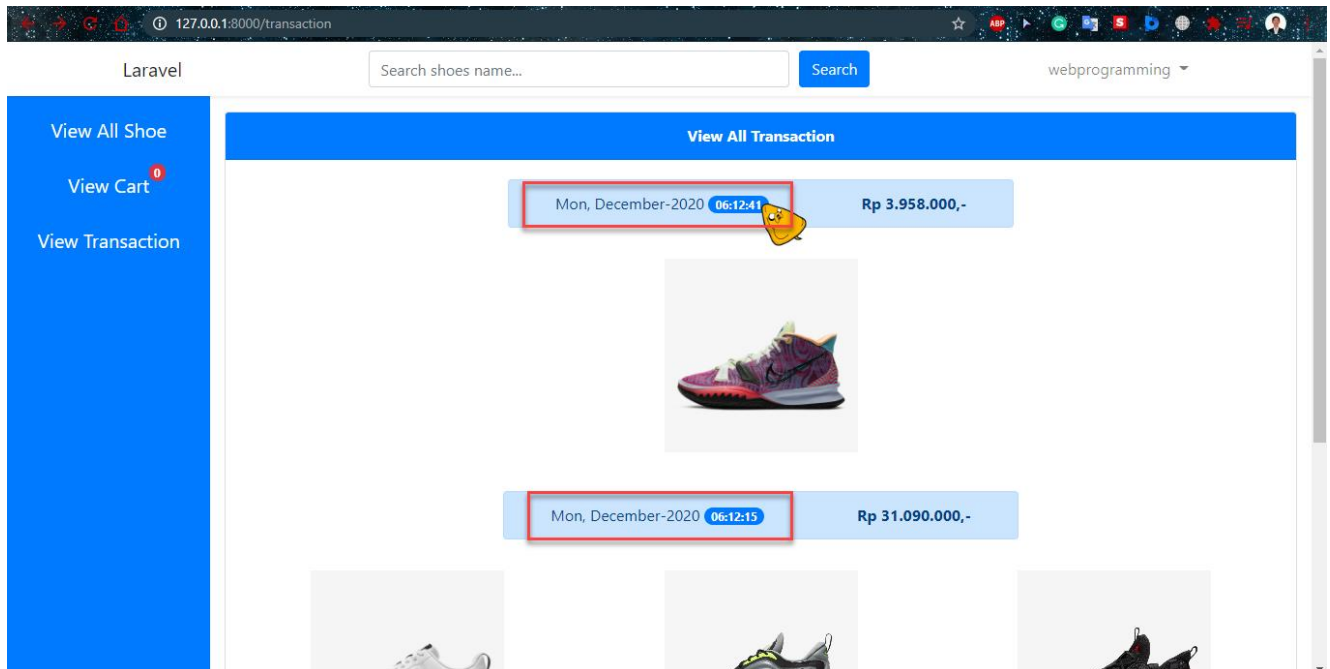
9. Checkout

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.



10.Transaction History

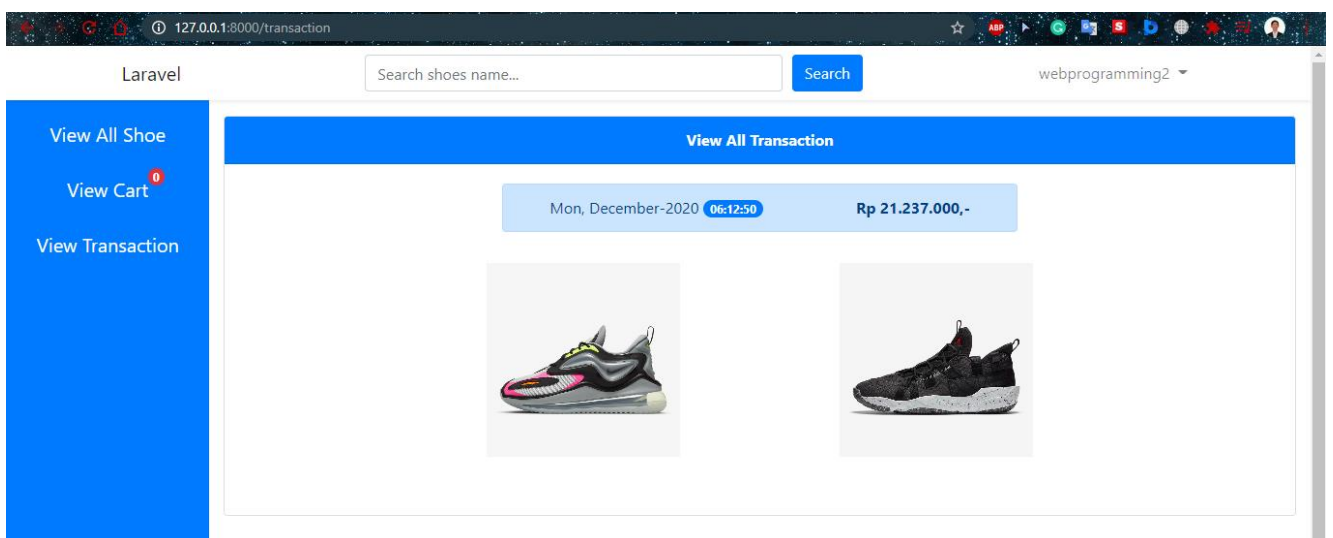
Hasil dari checkout kita akan masuk ke dalam **View Transaction** dan data pada cart akan hilang, ditandai notif yang menjadi 0. Jika kita menambah data baru maka hasilnya historynya akan bertambah sesuai dengan jam/created_at yang paling baru



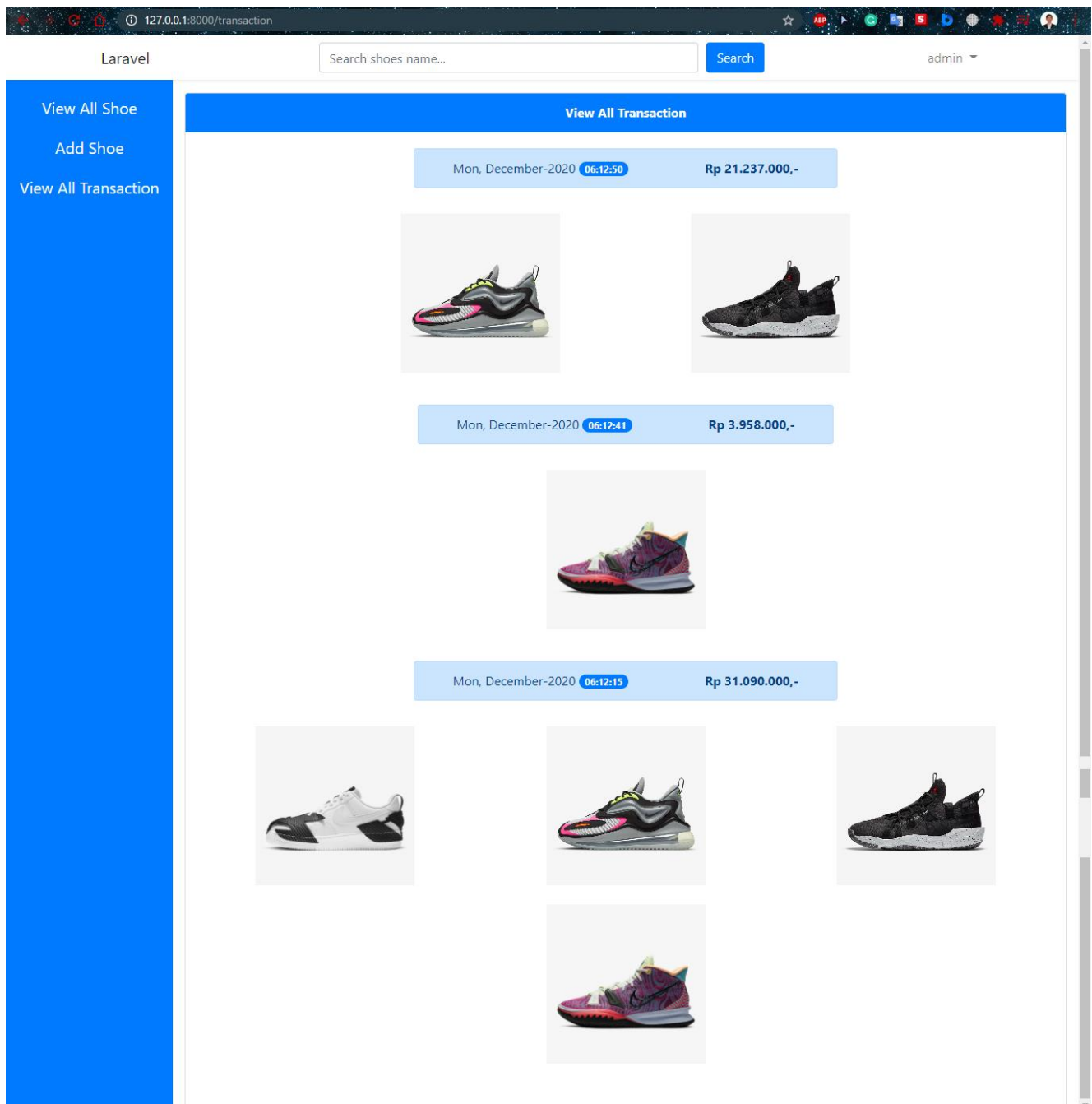
Untuk format angkanya, kami menggunakan fungsi bawaan dari **PHP** yaitu **number_format**, sedangkan untuk tanggalnya kami menggunakan fungsi **format:date_time** dari **Laravel**.

```
20 <div class="alert alert-primary px-3">
21 <div class="d-flex justify-content-between">
22 <div class="font-weight-normal bd-highlight mr-5">{{ $transaction->created_at->format("D, F-Y") }}
23 <span style="color: white;" class="badge badge-pill badge-primary">{{ $transaction->created_at->format("H:m:s") }}
24 </div>
25 <div class="font-weight-bolder bd-highlight ml-5">Rp {{ number_format($transaction->total,0,"",".") }},-</div>
26 </div>
```

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



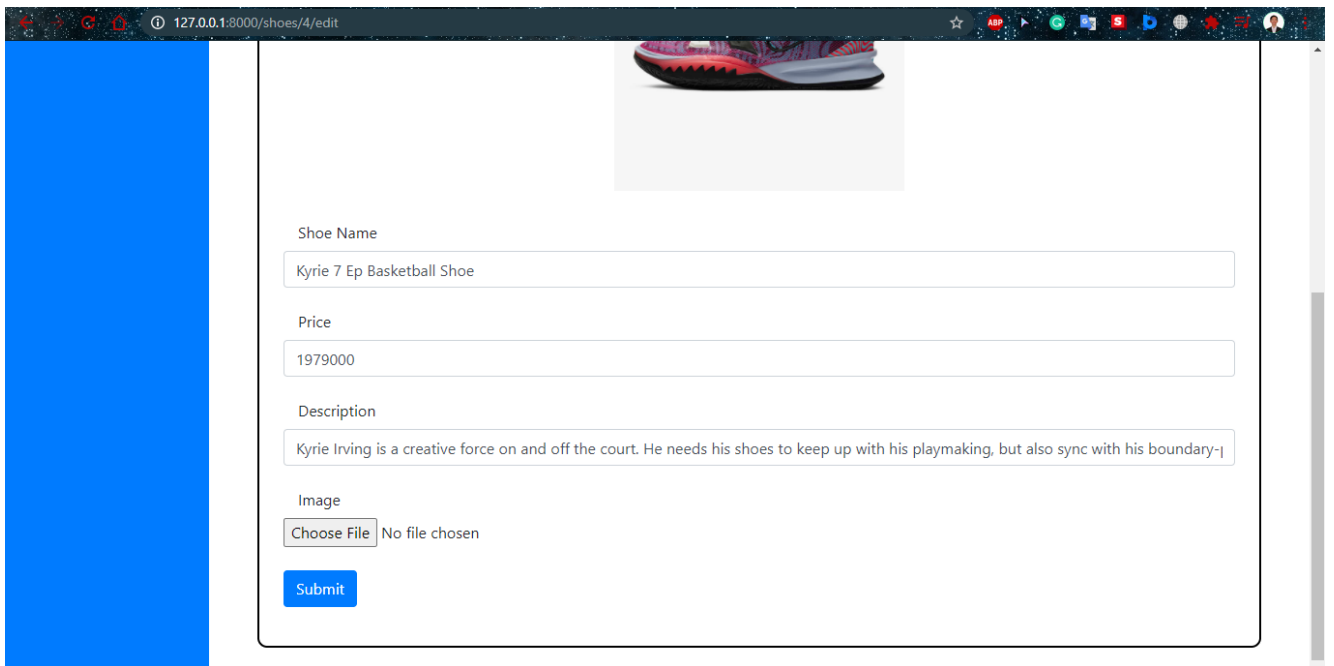
Lalu kita akan menggunakan akun admin untuk melihat **semua transaction history**



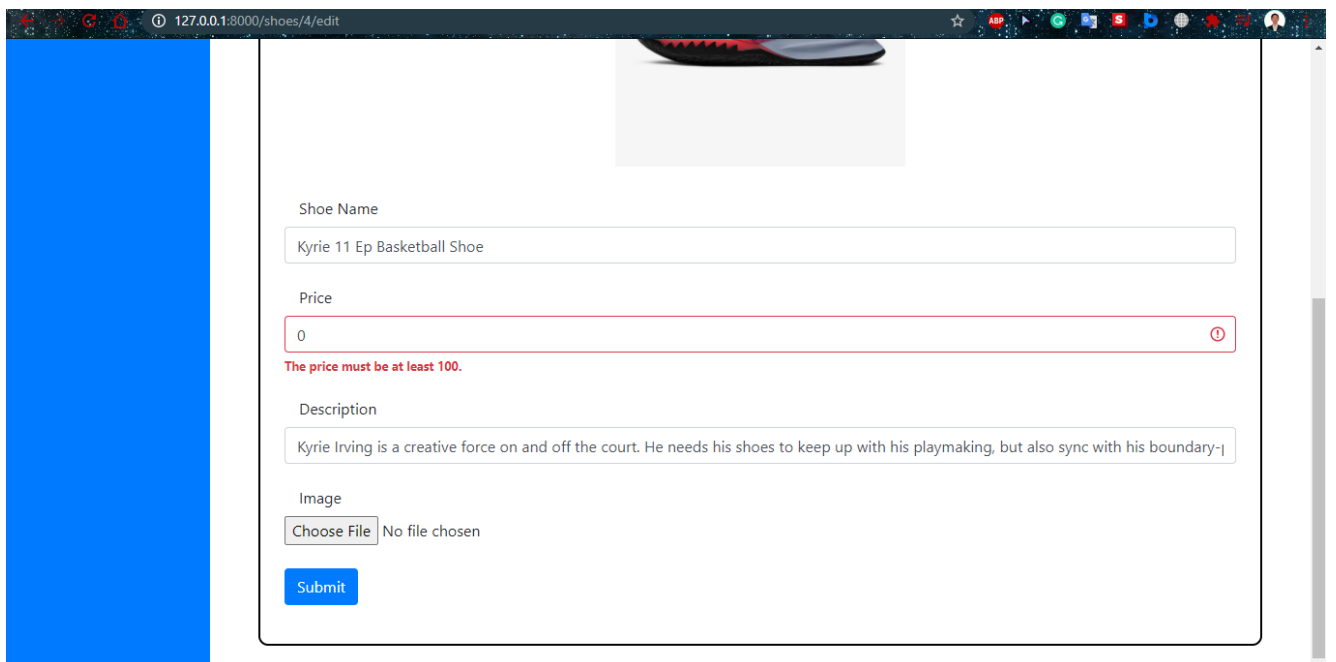
Di sini bisa dilihat, admin bisa melihat semua data transaksi dari yang paling baru sampai ke yang paling lama.

11. 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.

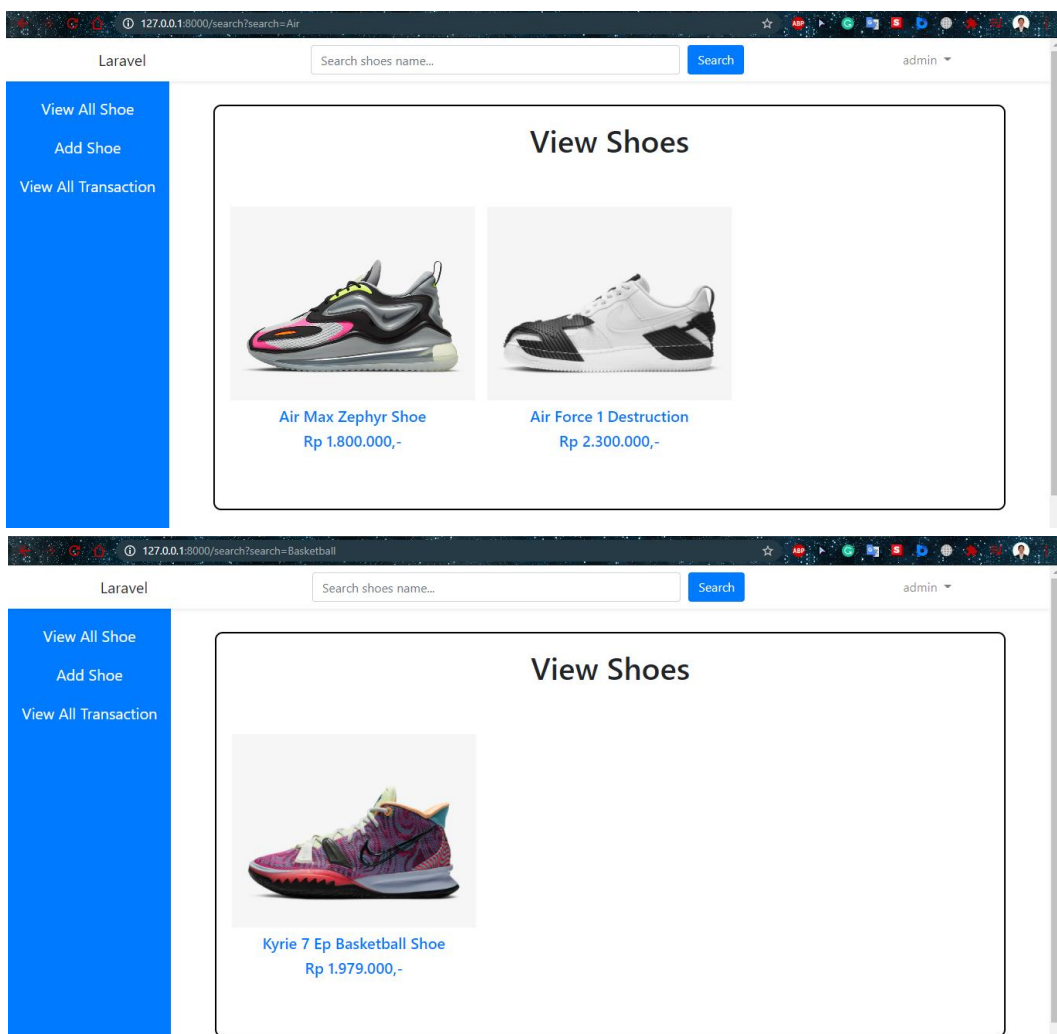


Kami menggunakan fungsi **old** untuk mengambil data lama yang diinput user

```
Run Terminal Help edit.blade.php - justduit - Visual Studio Code [Administrator]
# index-shoe-style.css cartIndex.blade.php web.php TransactionController.php edit.blade.php X
resources > views > shoes > edit.blade.php > ...
19
20 <div class="row edit-shoe-image mt-4">
21 | 
22 </div>
23
24 <div class="form-group row mt-4">
25 | <label for="name" class="col-md-4 col-form-label">Shoe Name</label>
26
27 | <input type="text" id="name" class="form-control{{ $errors->has('name') ? ' is-invalid' : '' }}"
28 | name="name" value="{{ old('name') ?? $shoe->name }}">
29
30 | @if ($errors->has('name'))
31 | | <span class="invalid-feedback" role="alert">
32 | | <strong>{{ $errors->first('name') }}</strong>
33 | | </span>
34 | @endif
35 </div>
```

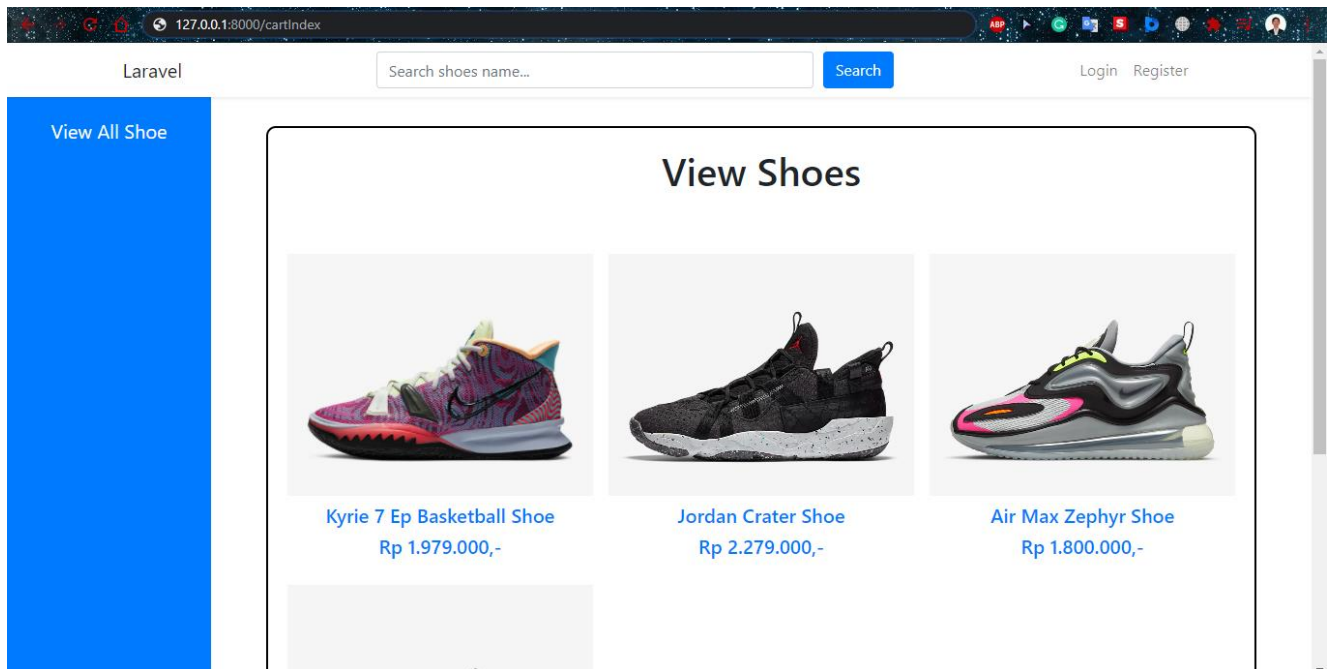
12. Searching

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

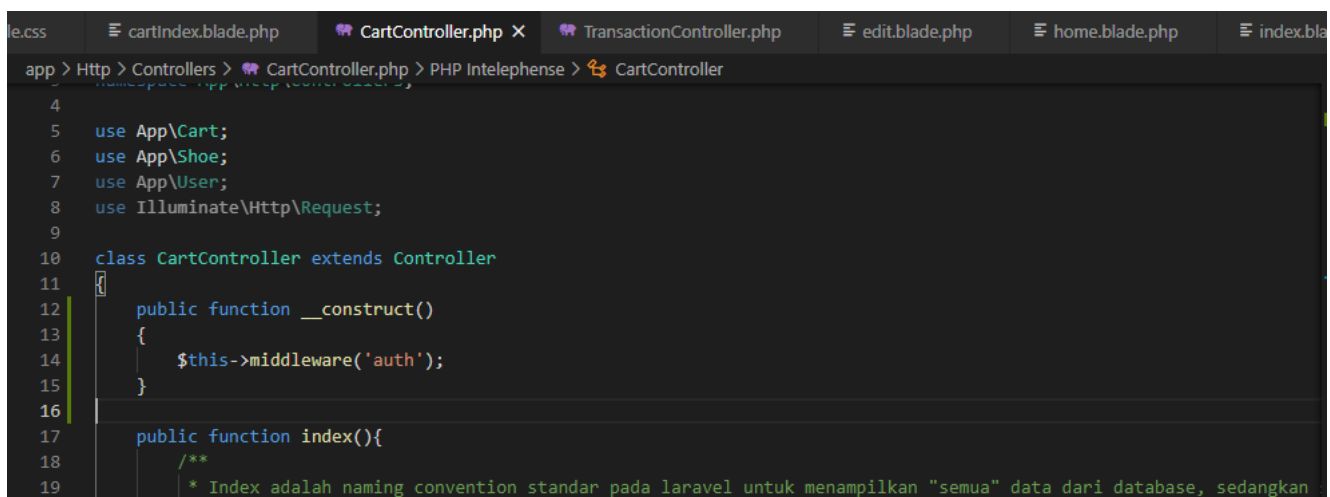


13. Authentication and Authorization

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

```
30     }
31
32     public function show(Shoe $shoe){
33
34         /**
35          * Menampilkan cart sesuai dengan user_id
36          * Join digunakan agar dari pivot table bisa mengakses data-data pada tabel shoes, seperti title, harga,
37          * ataupun imagenya
38          */
39
40         if(auth()->user()){
41             $count = Cart::where('user_id', auth()->user()->id)->join('shoes', 'shoe_user.shoe_id', '=', 'shoes.id')
42                 ->count();
43             return view('carts.add', ['shoe' => $shoe, 'count' => $count]);
44         }
45         abort(401);
46     }
47 }
```

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 Genesisus