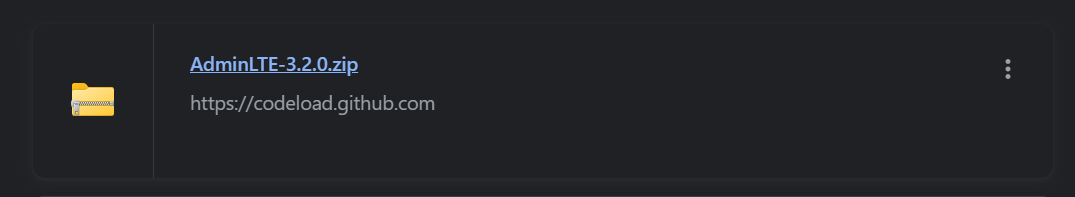
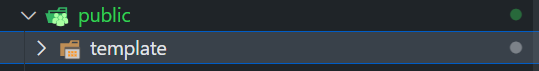
1. Template form
2. Penggunaan template AdminLTE Bootstrap Admin Dashboard Template dengan langkah pertama akses https://adminlte.io/ , lalu klik download pada source code (zip)
3. Lakukan extract All dan ubah nama folder AdminLTE-3.2.0 menjadi template, folder template pindahkan ke laragon/www/PWL\_2024/public. Buka folder public pada VSCode maka akan tampil folder baru bernama template
4. Copy isi public/template/index2.html pada welcome.blade.php lalukan edit agar dapat diload, edit href =”{{ asset (‘template/…..’) }}” dan src= ” {{ asset (‘template/…..’)” sesuai gambar berikut



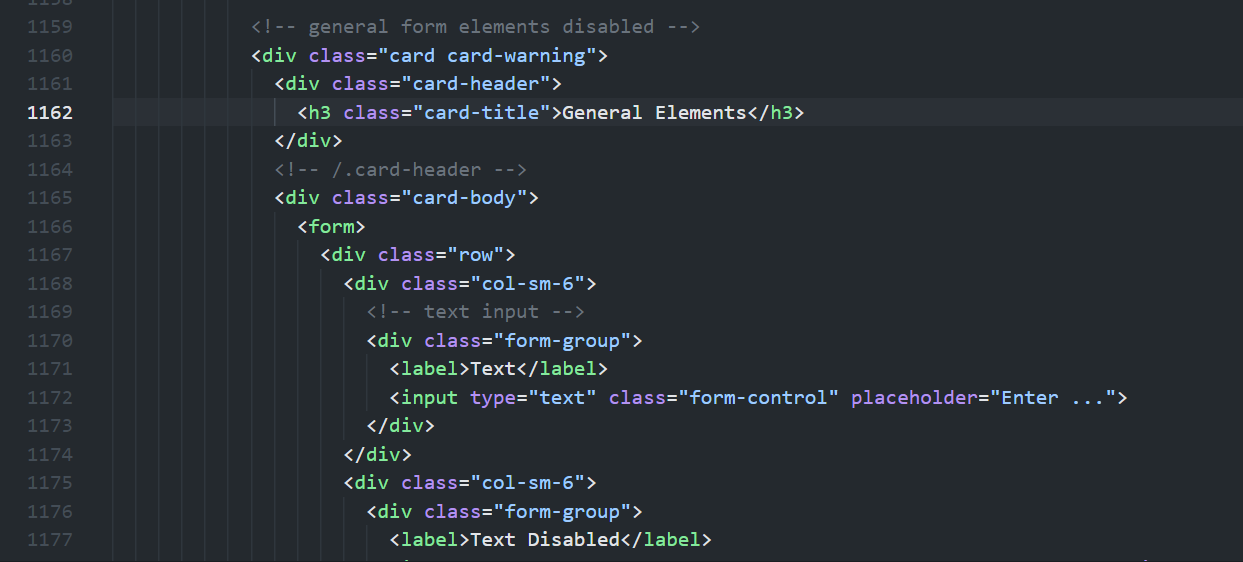
1. Scroll ke bagian paling bawah dan sesuaikan dengan kode berikut



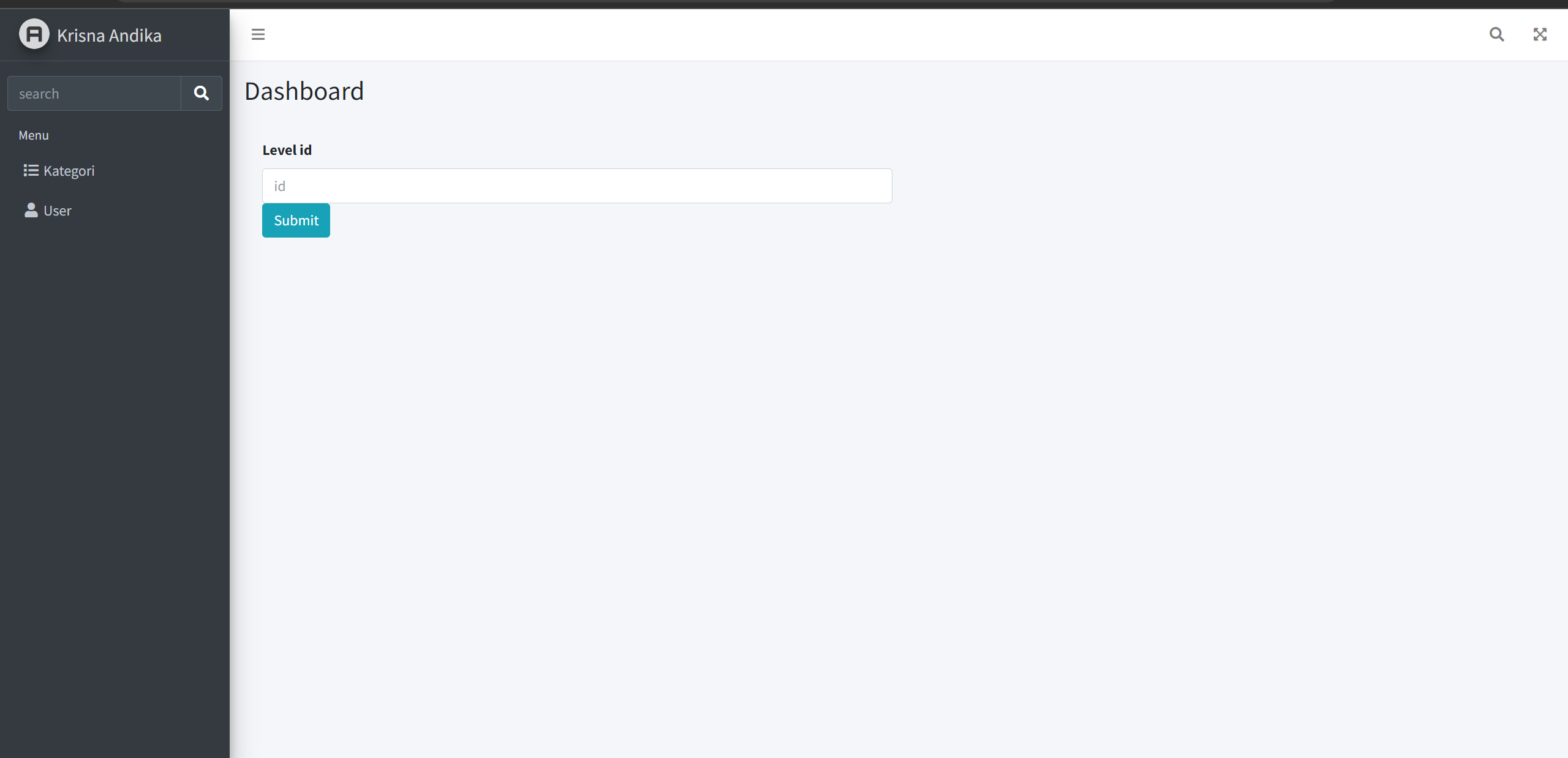
1. Hasil running



1. Pada bagian menu terdapat pilihan Form kita khususkan dulu pada Form yang disediakan oleh Admin LTE • General Elements • Advanced Elements • Editor • Validation Cara menggunakan template bisa dilihat nama formnya contoh judulnya ‘General Elements’, agar dapat digunakan dapat melakukan akses pada folder yang terdapat template adminLTE contoh : C:\laragon\www\PWL\_2024\public\template\pages\forms
2. Buka file pada form> general cari keyword dengan ‘general elements’ dan coba modifikasi pada welcome.blade.php

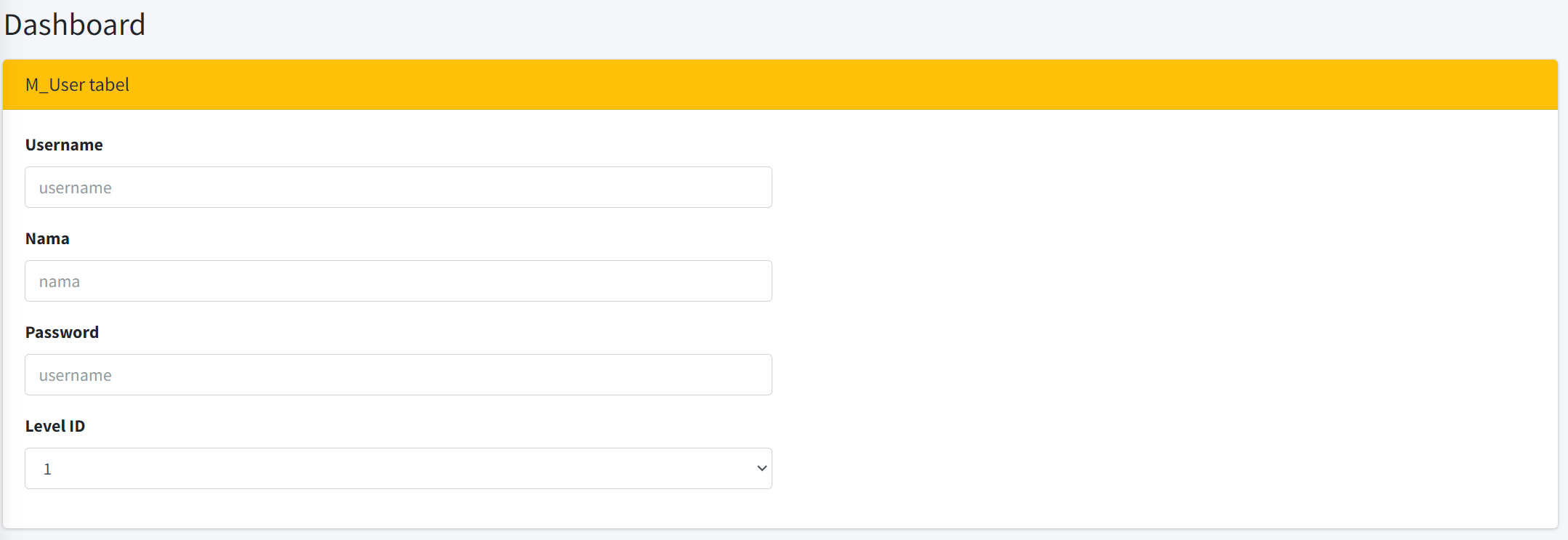


1. Apa yang tampil?

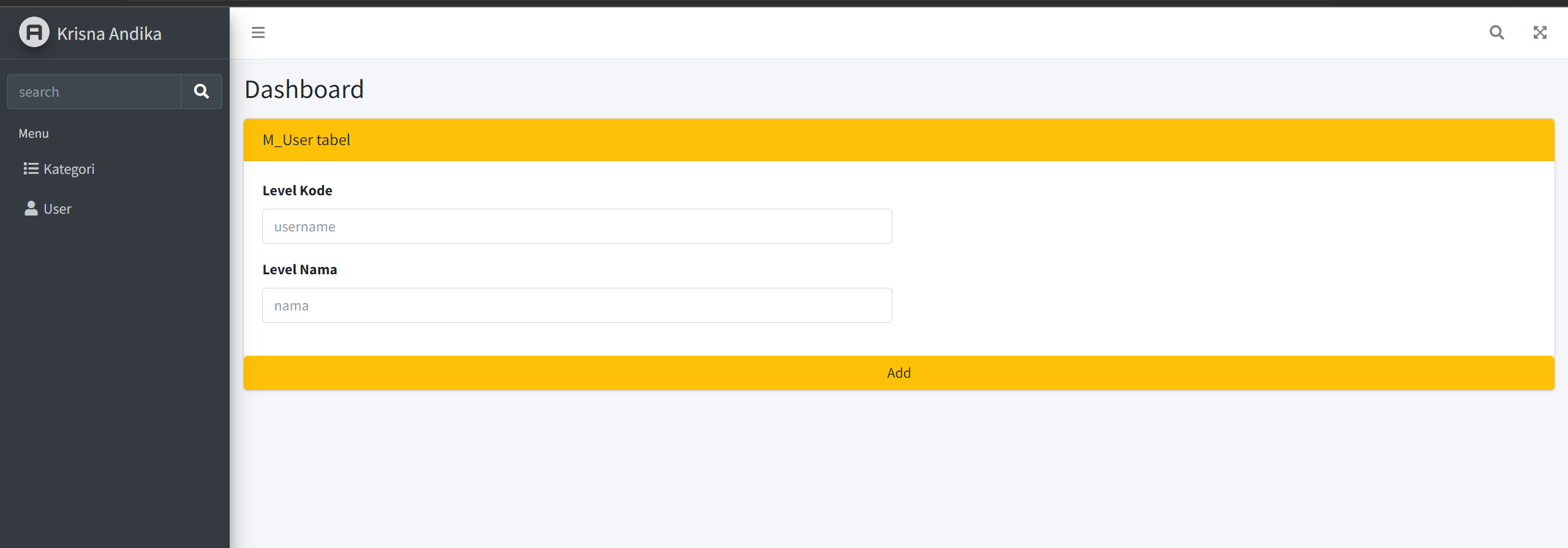


1. . Eksplorasi jenis form pada adminLTE dan coba terapkan yang sesuai untuk studi kasus POS PWL, buatlah form untuk tabel m\_user dan m\_level

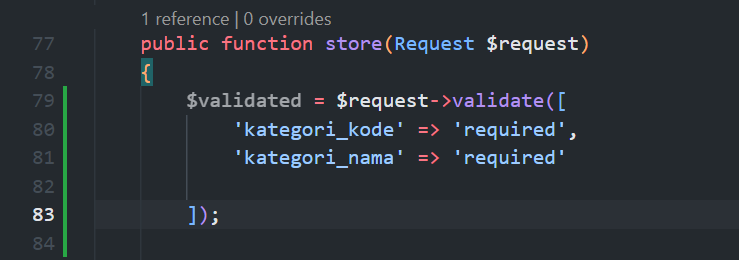
M\_User



M\_level



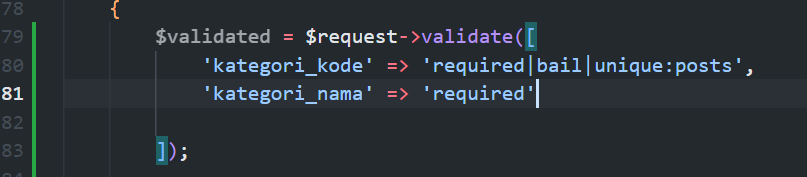
1. Validasi pada server
2. Edit pada KategoriKontroller dengan kode:



1. Tulis perbedaan penggunaan validate dengan validateWithBag!

Validatewithbag digunakan untuk memvalidasi dengan pesan store error dengan named error bag

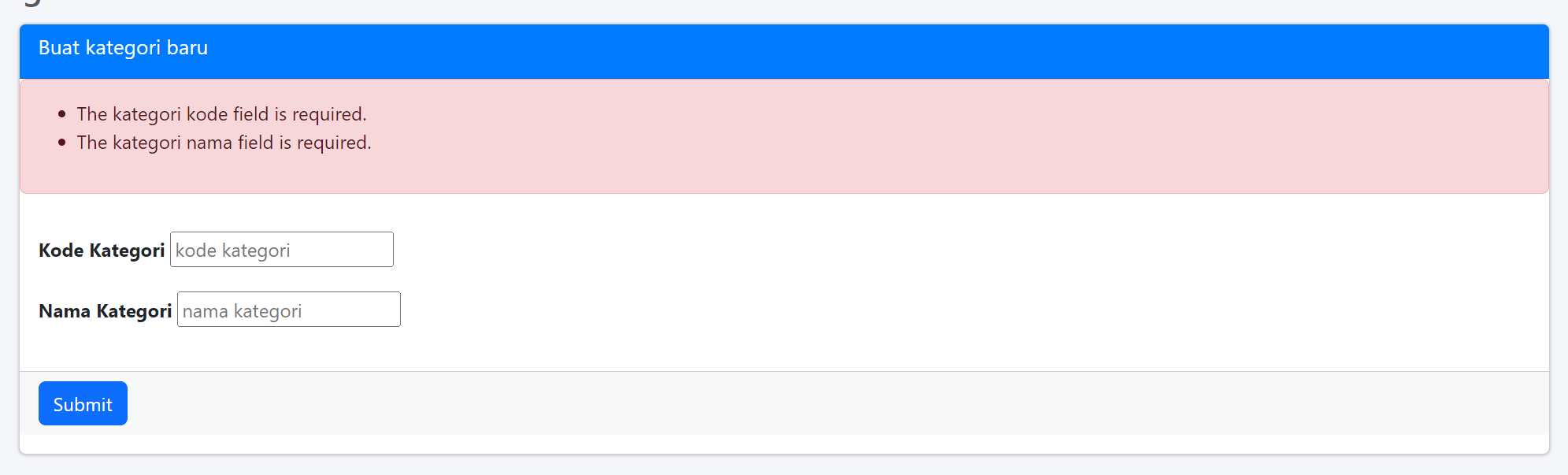
1. Terkadang Anda mungkin ingin berhenti menjalankan aturan validasi pada suatu atribut setelah kegagalan validasi pertama. Untuk melakukannya, tetapkan bail aturan ke atribut: coba sesuaikan dengan field pada m\_kategori, yuk bisa.. apa yang terjadi?



1. Pada view/create.blade.php tambahkan code berikut agar ketika validasi gagal, kita dapat menampilkan pesan kesalahan dalam tampilan:

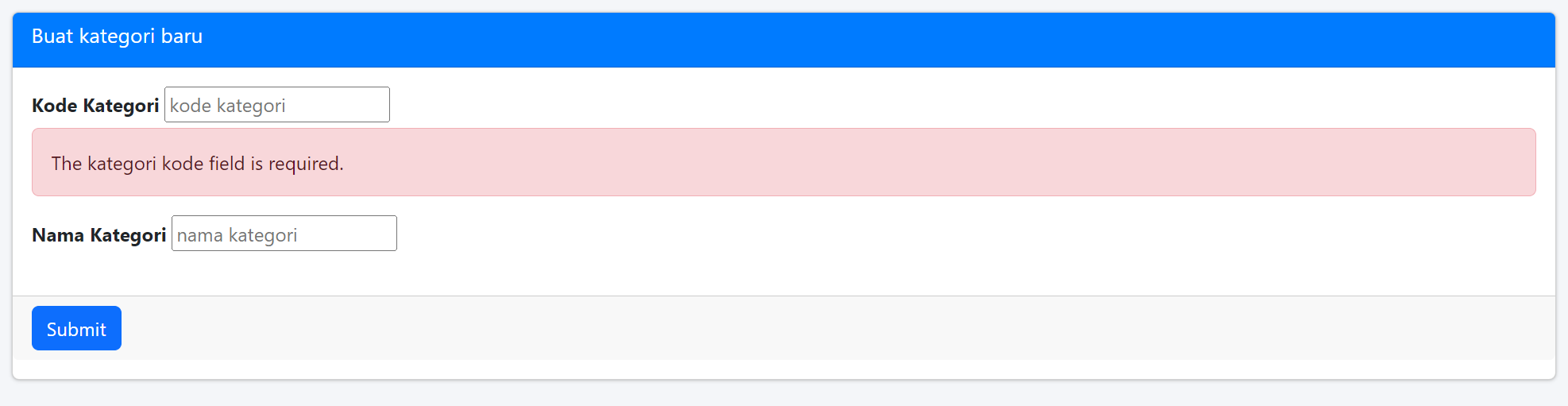


1. Run

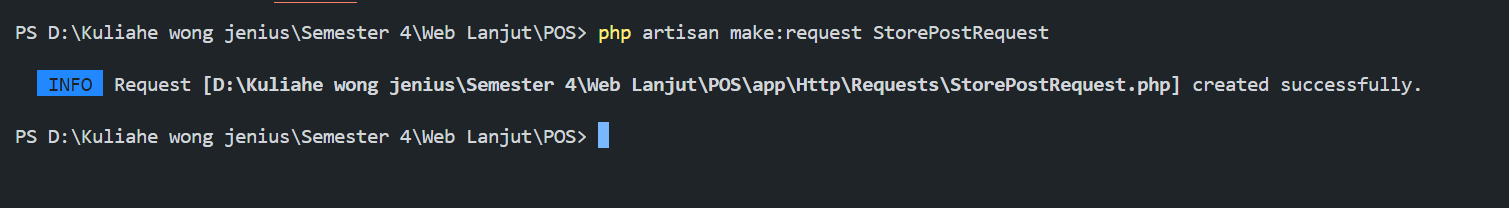


1. Pada view/create.blade.php tambahkan dan coba running code berikut :

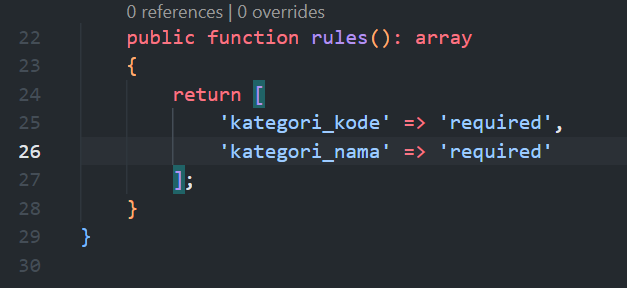




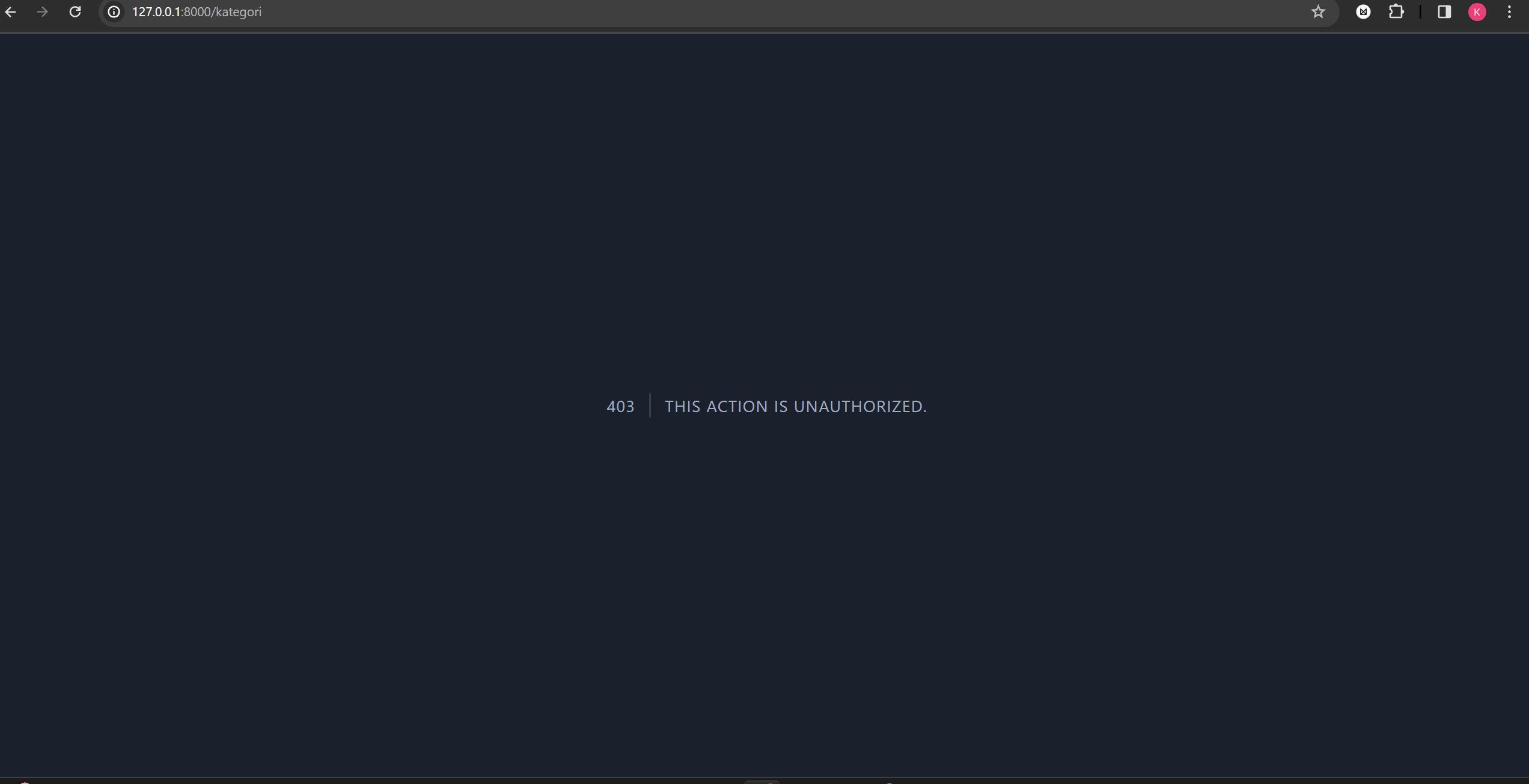
1. Form request Validation
2. Membuat permintaan form dengan menuliskan pada terminal



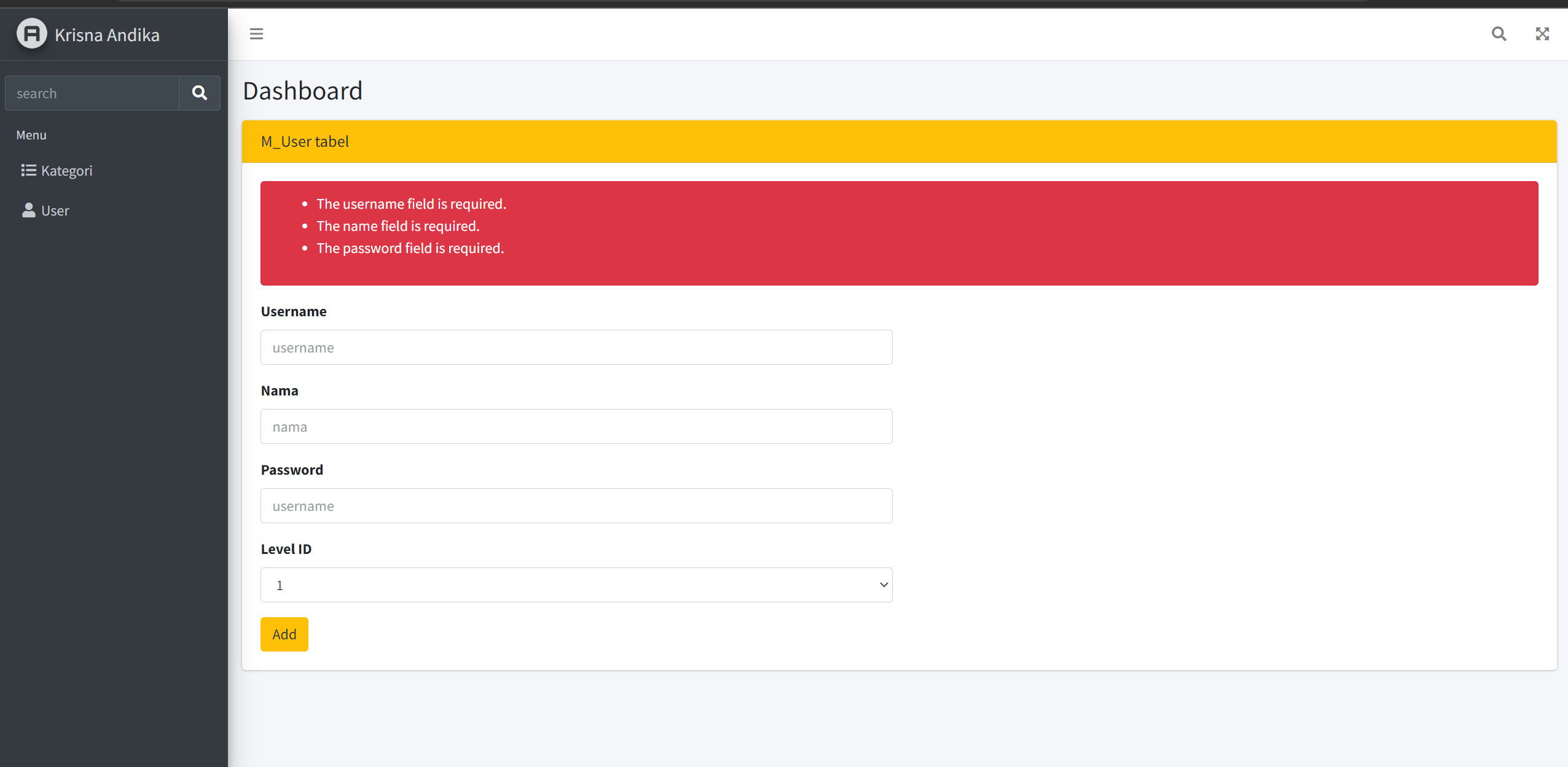
1. Ketik kode berikut pada Http/request/StorePostRequest

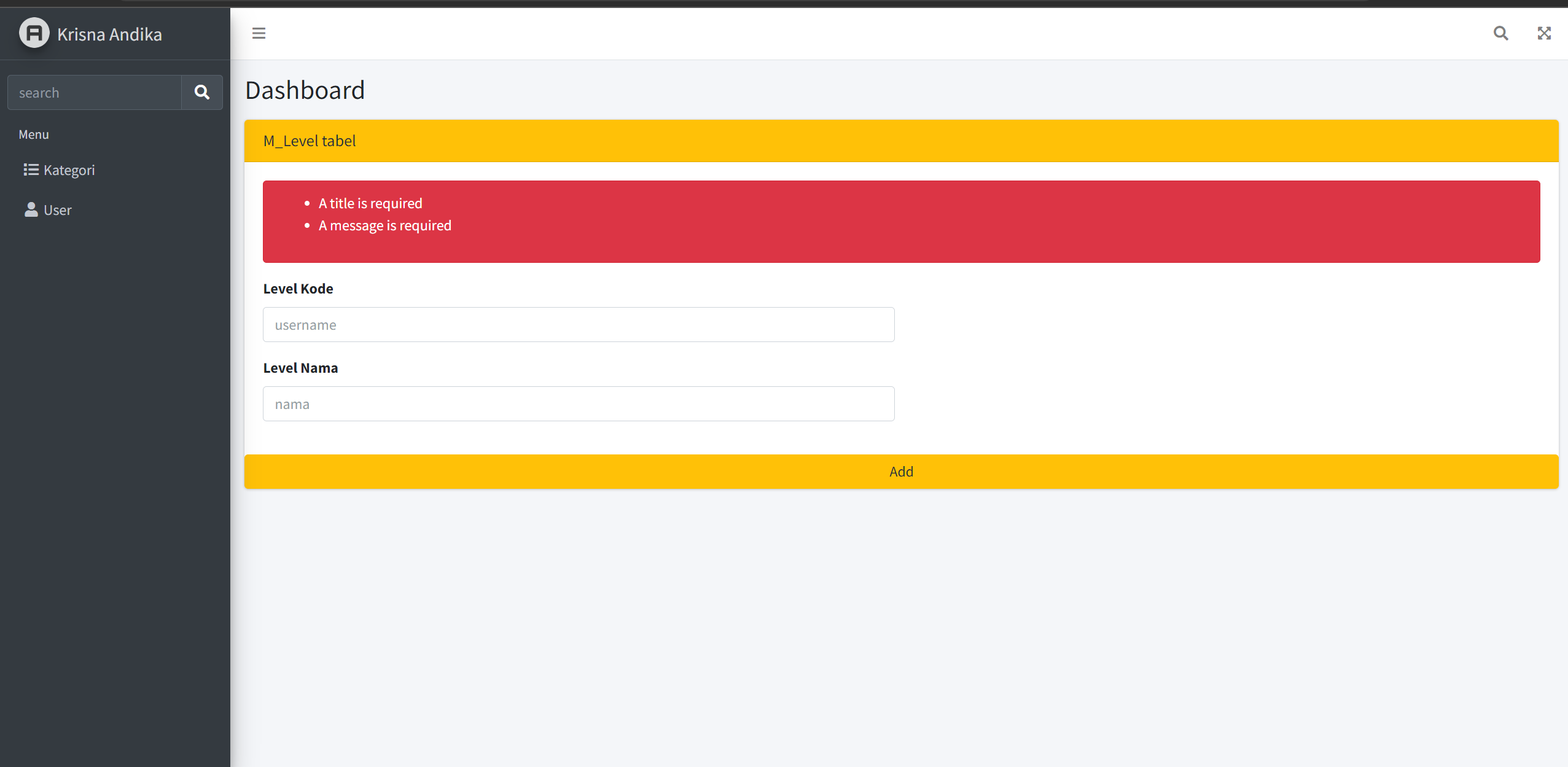




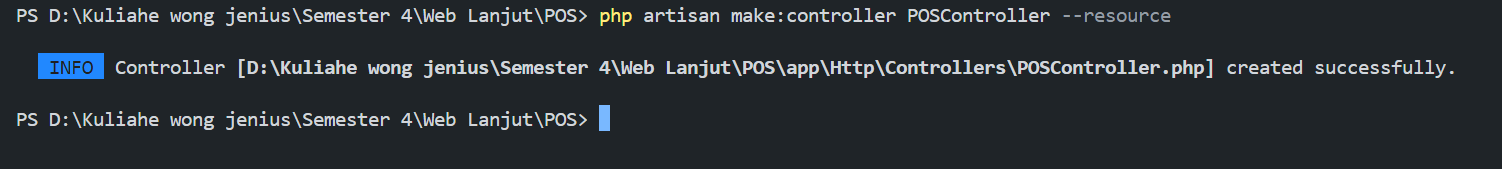


1. Terapkan validasi juga pada tabel m\_user dan m\_level

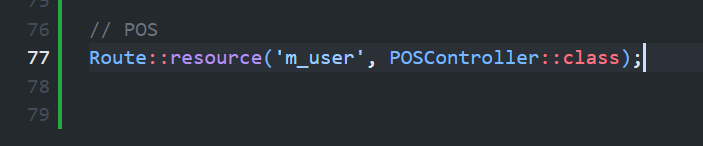




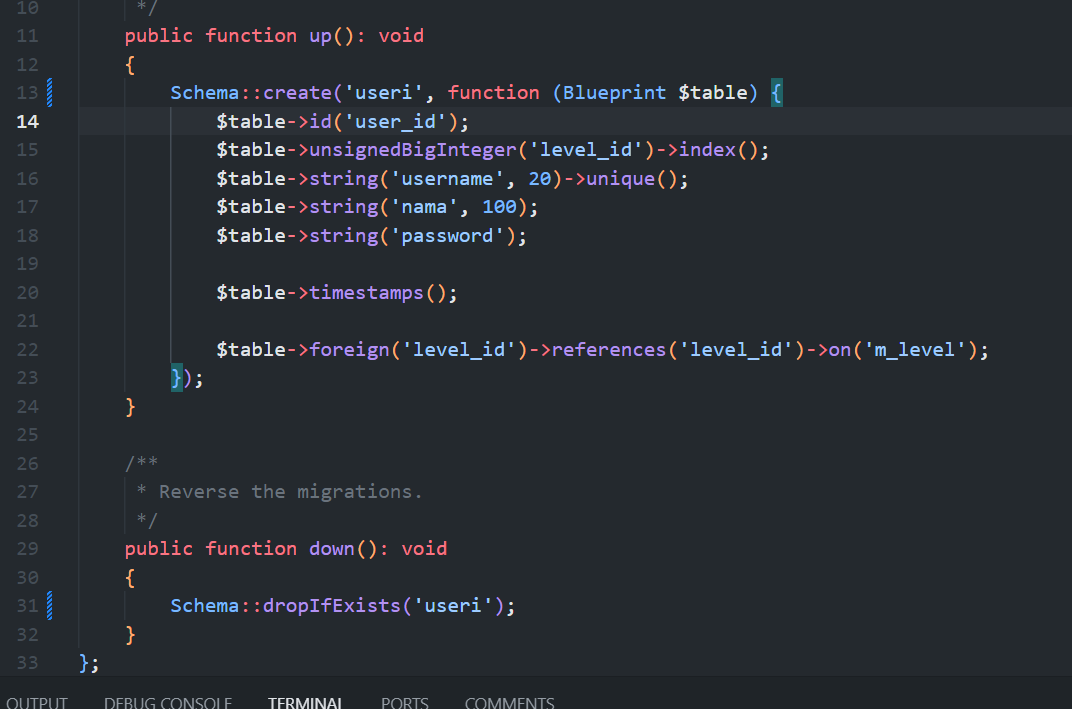
1. CRUD
2. Buat POSController lengkap dengan resourcenya, Membuat Resource Controller dan Route yang berfungsi untuk route CRUD sehingga tidak perlu repot-repot membuat masing-masing route seperti post, get, delete dan update.



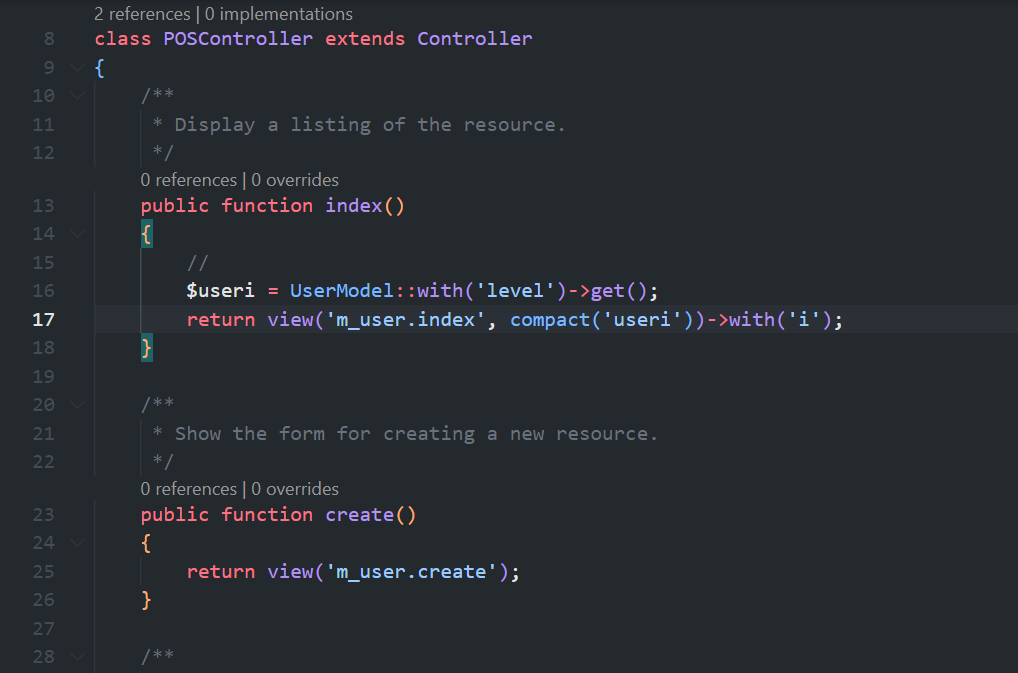
1. Berikut perintahnya: sekarang tambahkan kode pada route/web.php

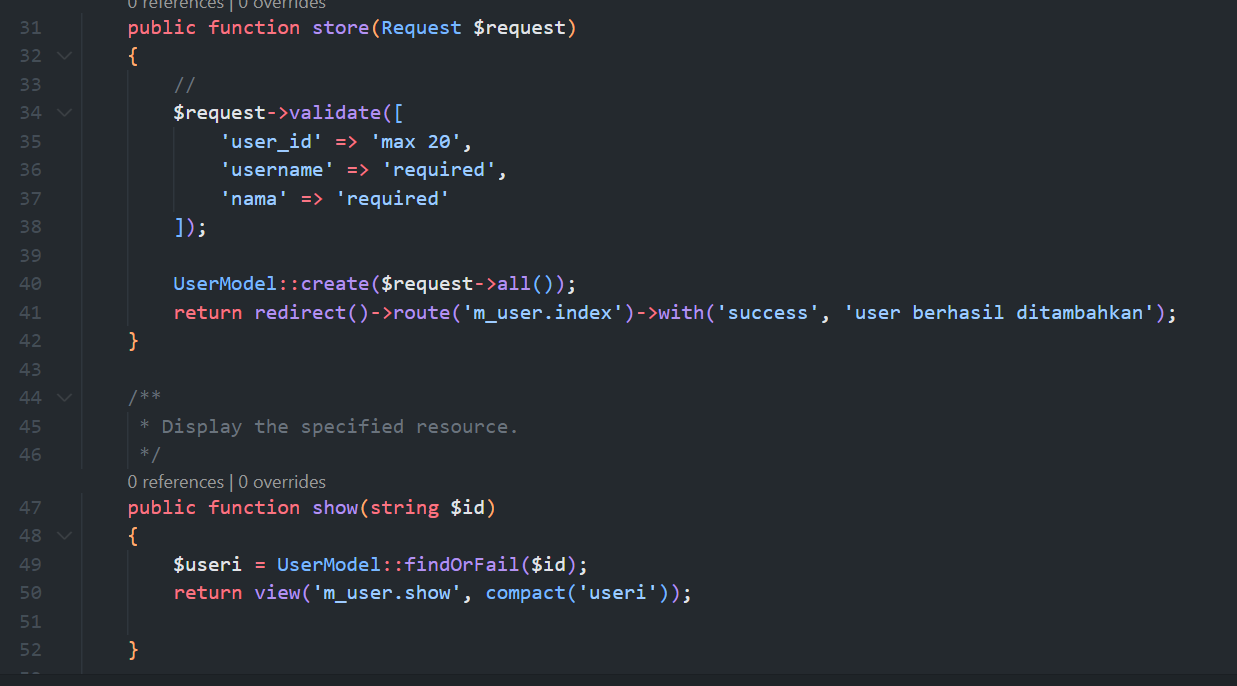


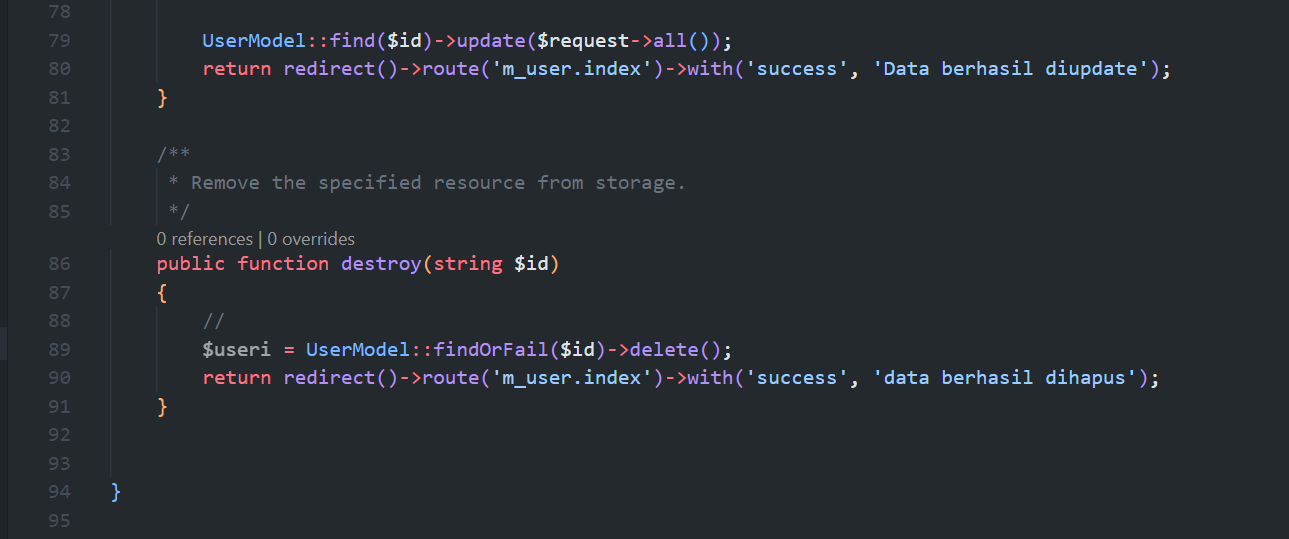
1. Atur pada Models/m\_user



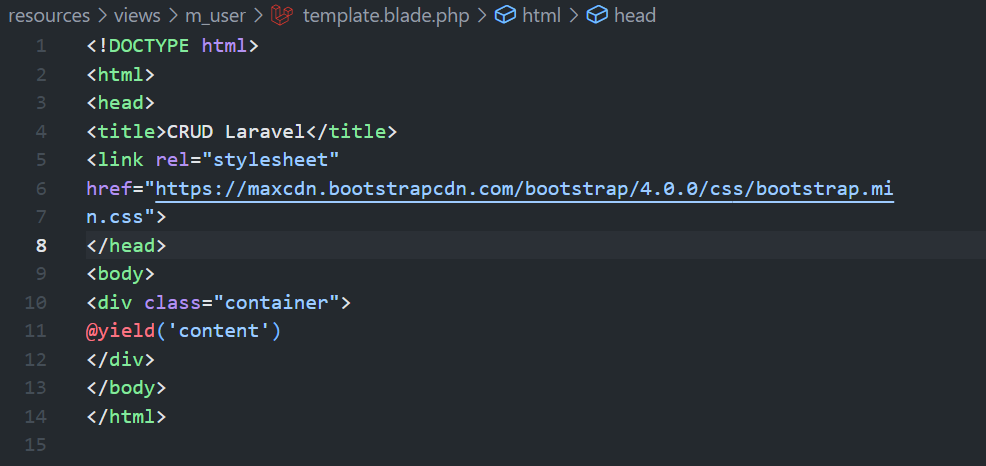
1. Selanjutnya bukalah app/Http/Controllers/POSController.php kemudian ketikkan, kodenya seperti berikut ini:





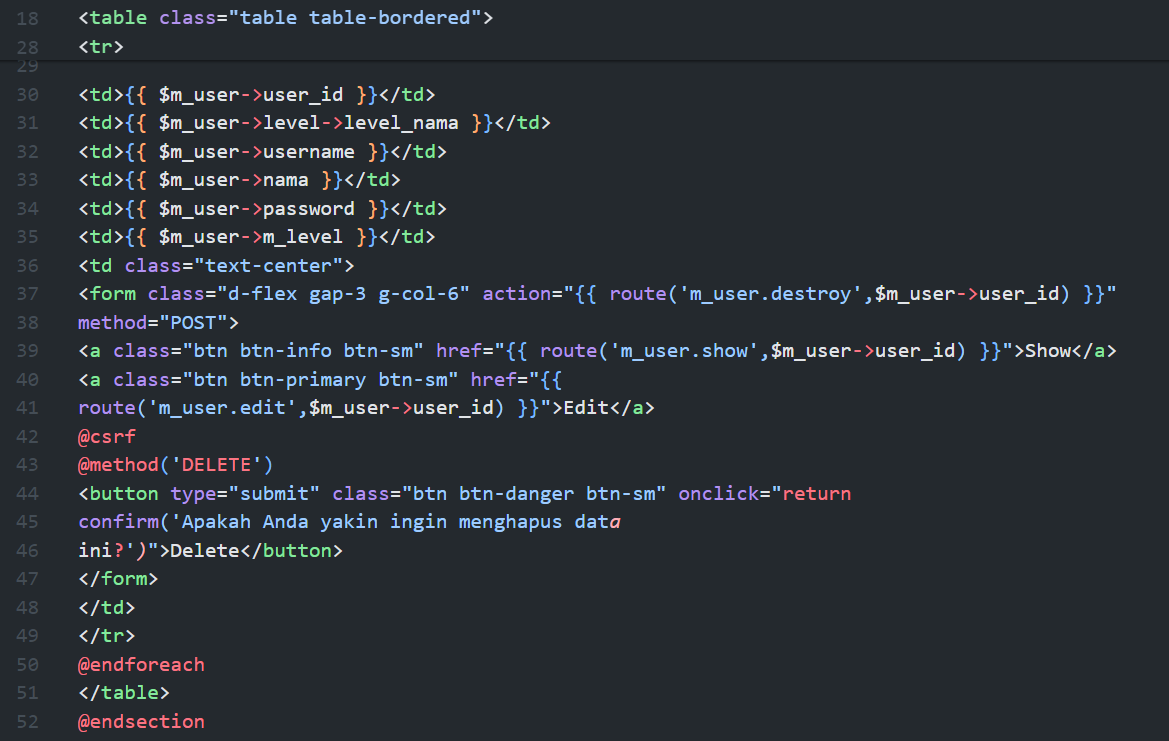


1. Buatlah folder di Resources/Views/m\_user dengan beberapa blade dan isian kode berikut:
2. template.blade.php

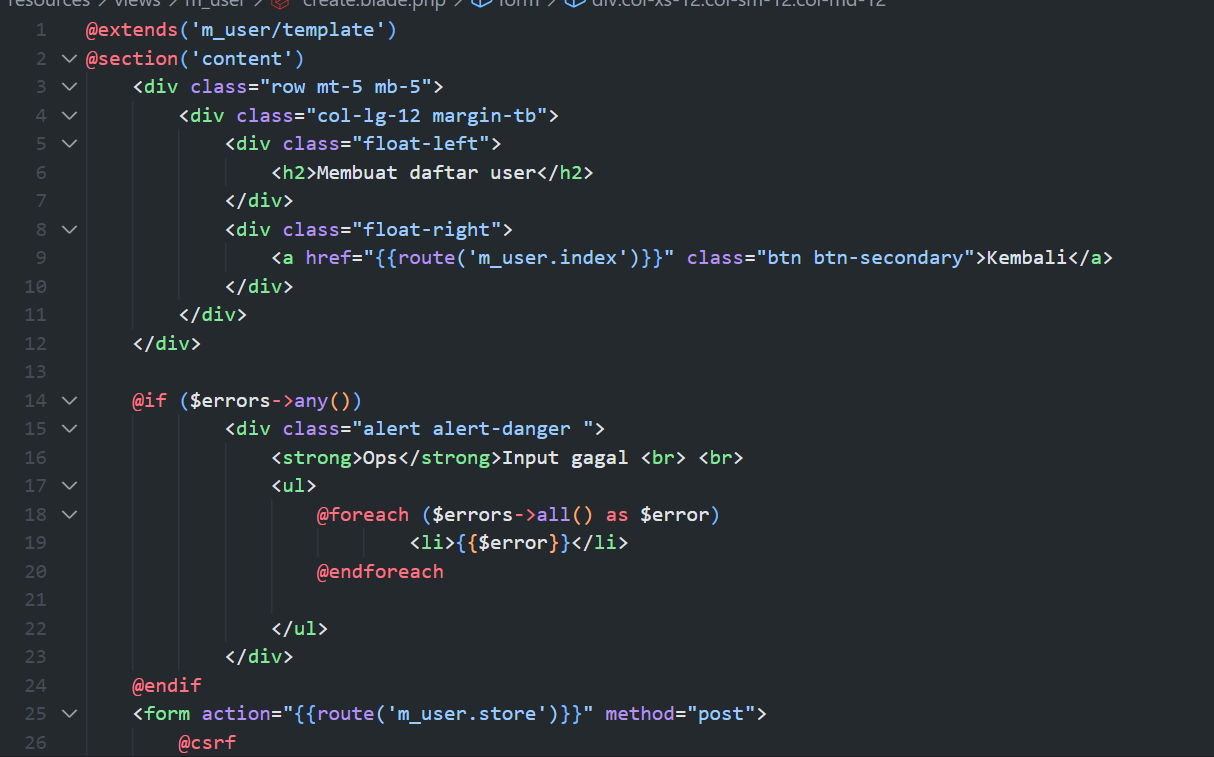


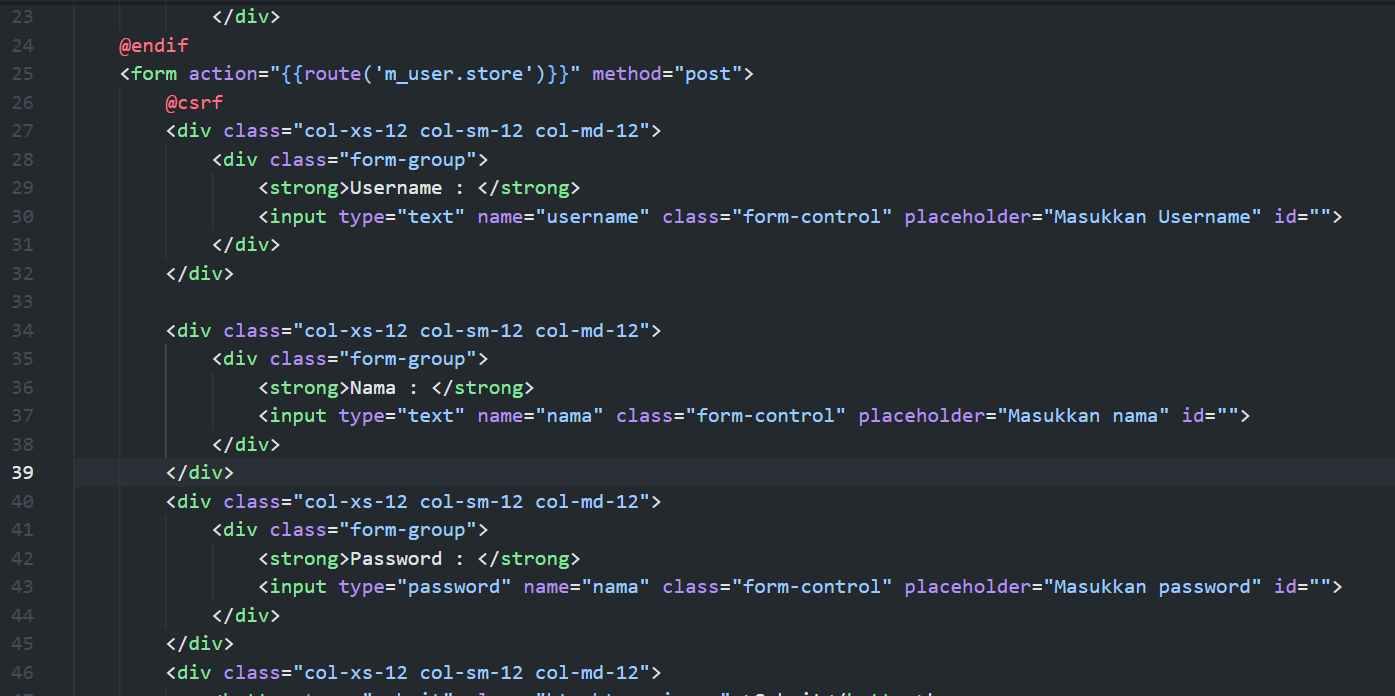
1. index.blade.php



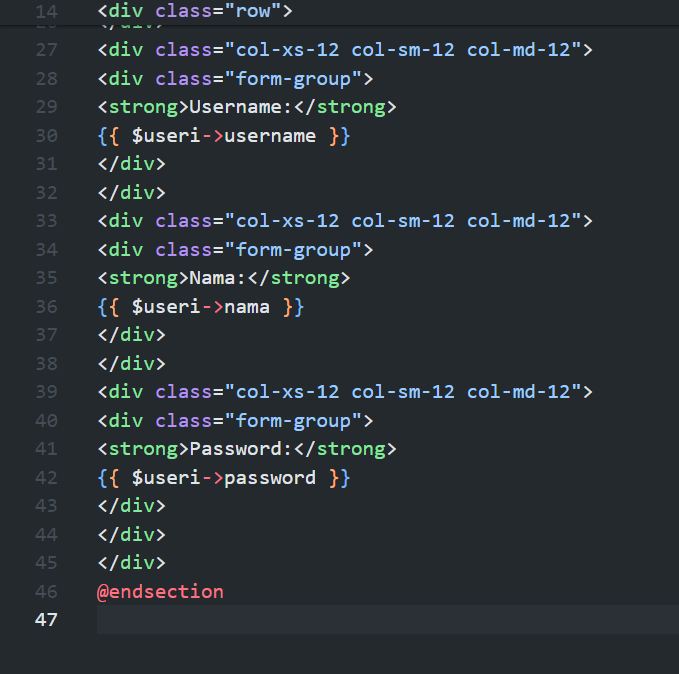
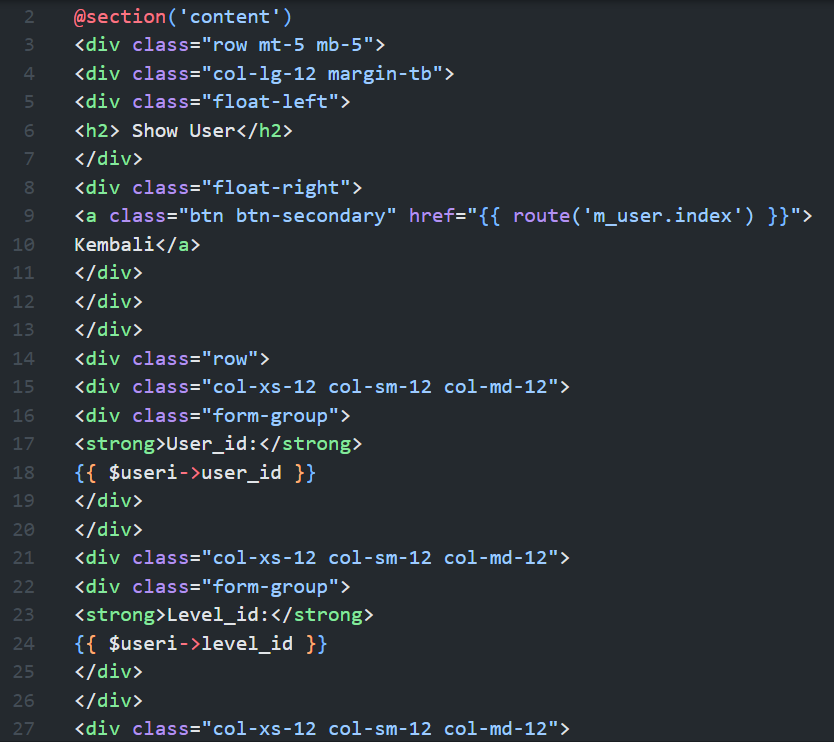


1. create.blade.php

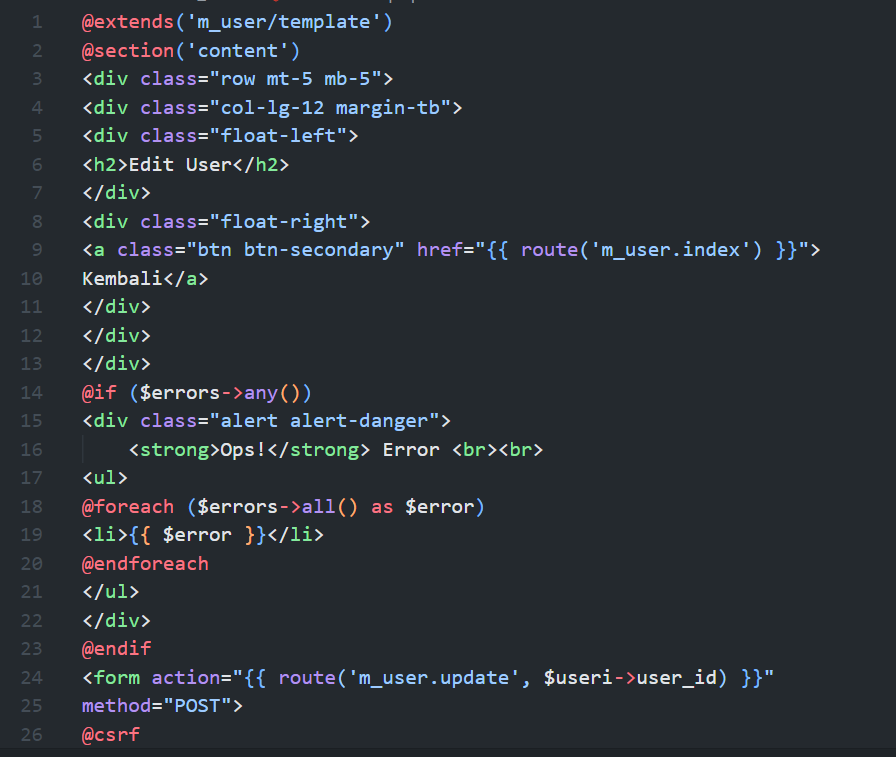


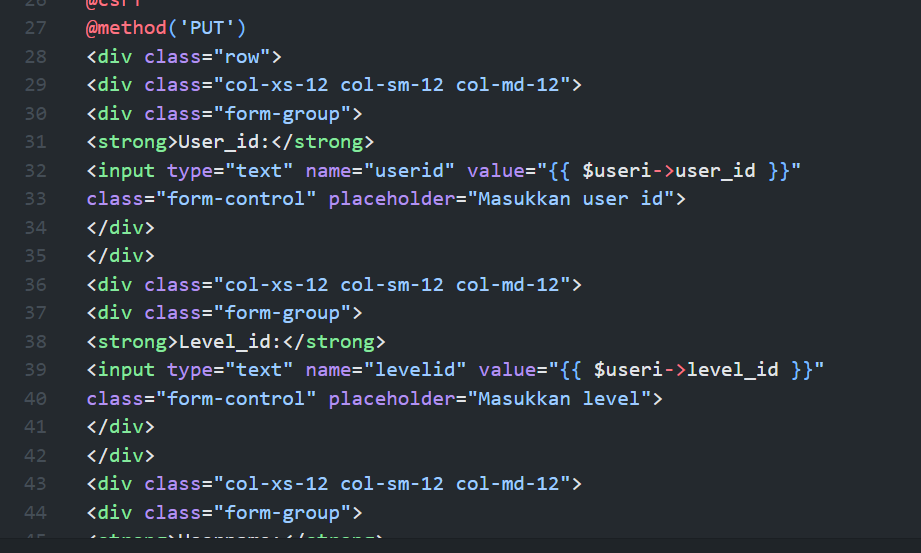


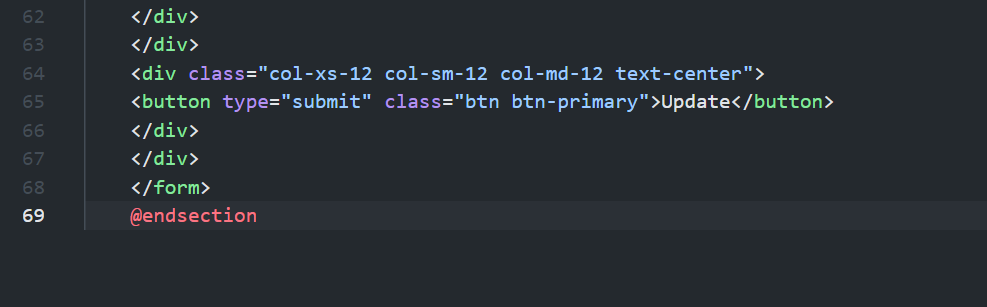
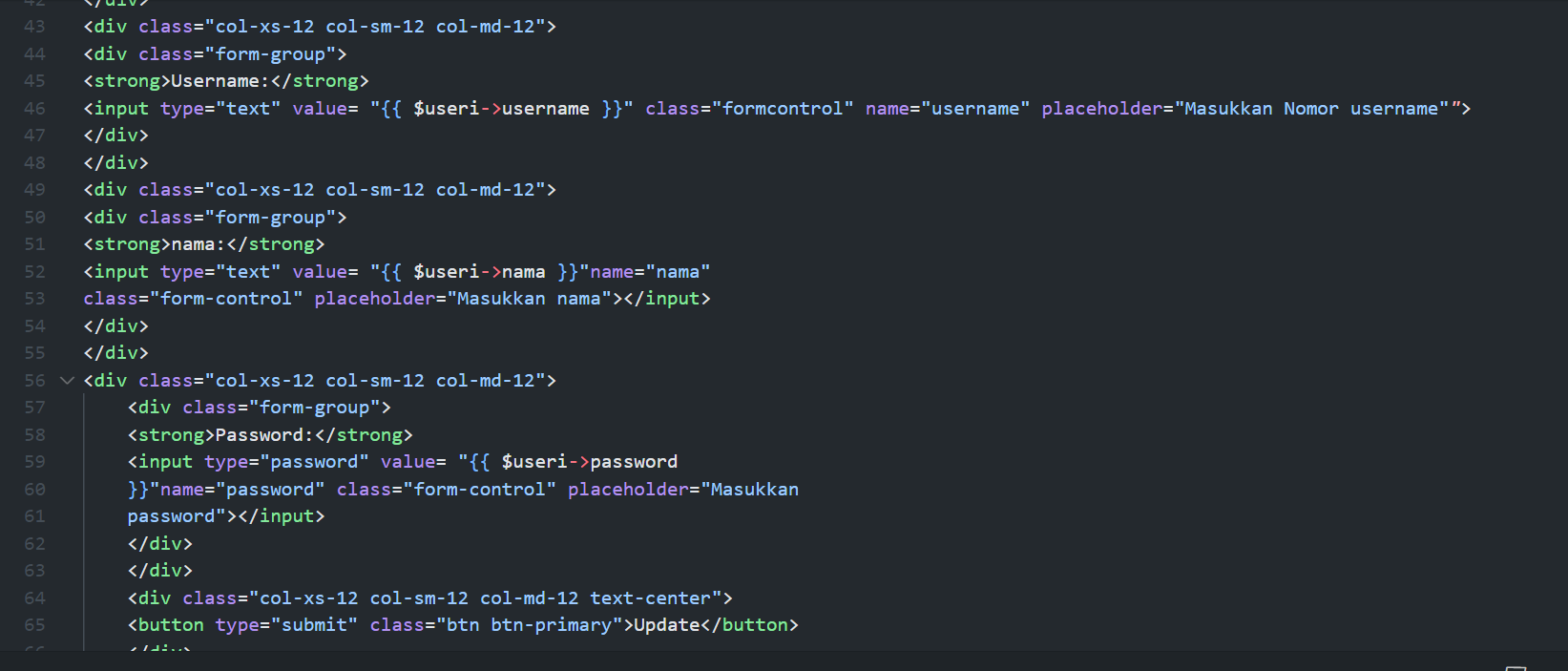
1. show.blade.php



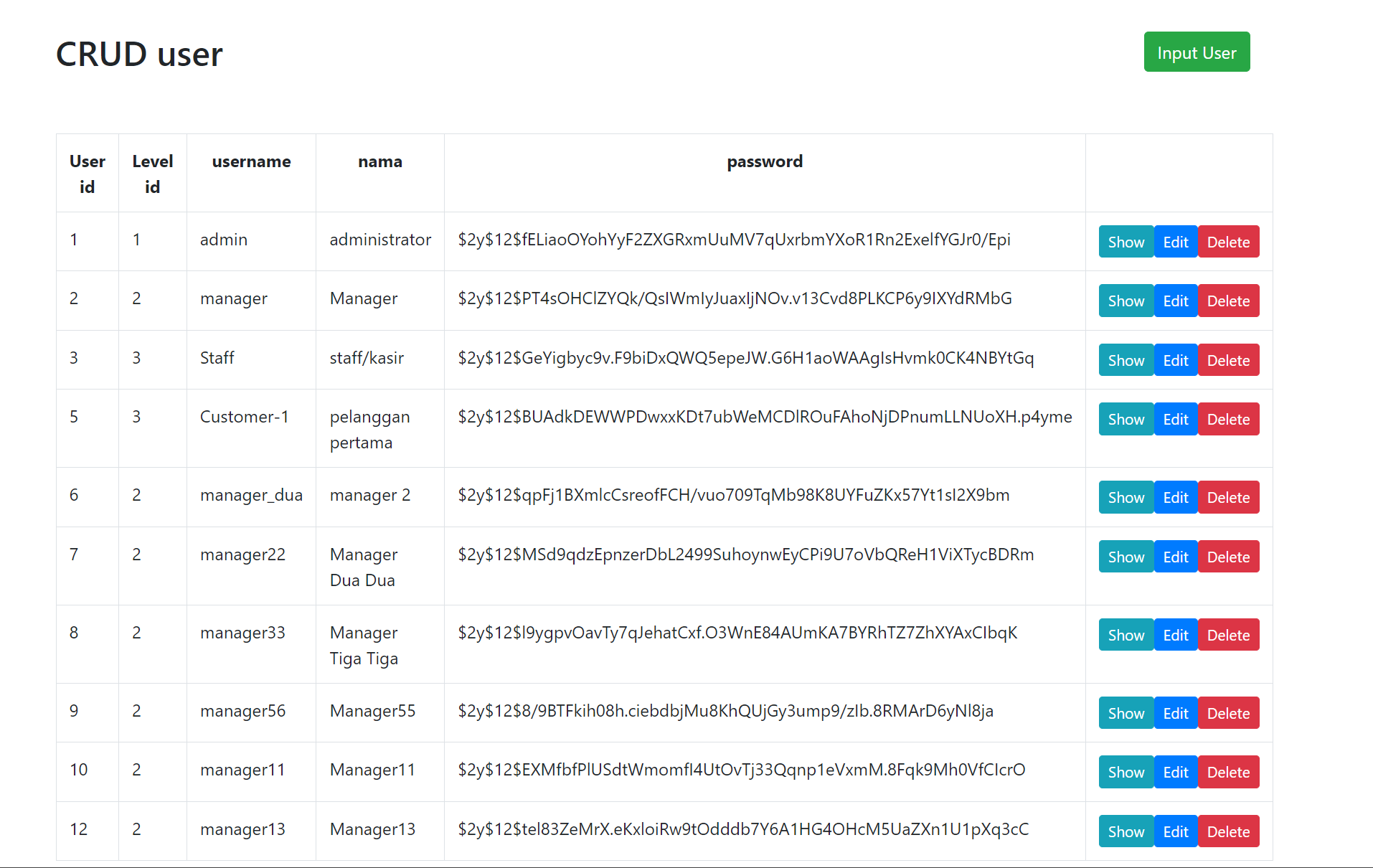
1. edit.blade.php







Hasil

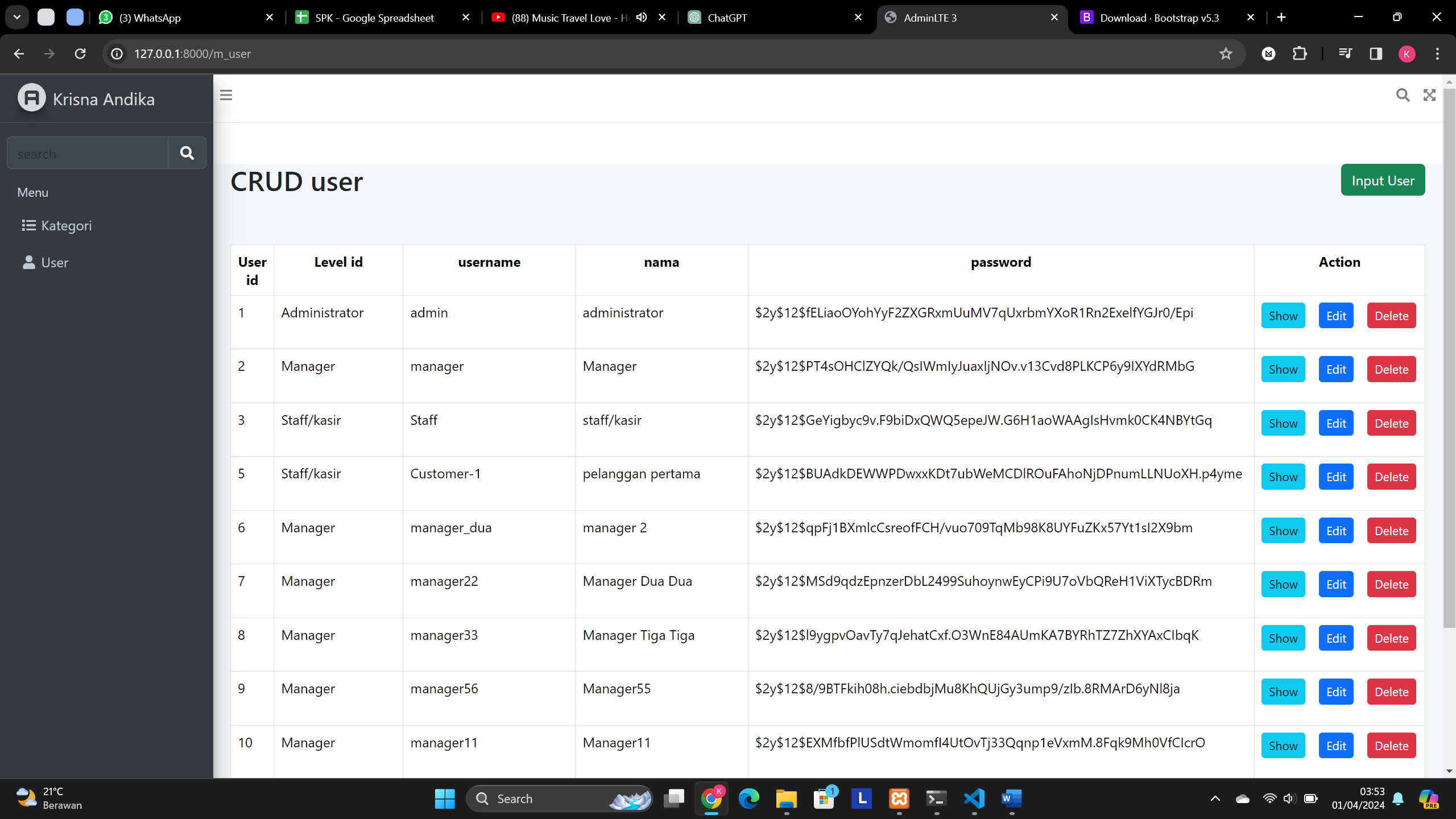


Soal

1. Coba tampilkan level\_id pada halaman web tersebut dimana field ini merupakan foreign key



1. Modifikasi dengan tema/ template kesukaan Anda



1. Apa fungsi $request->validate, $error dan alert yang ada pada halaman CRUD tersebut

Fungsi validate yaitu untuk memastikan bahwa inputan sesuai dengan field yang harus diisikan.

Untuk error apabila terdapat error maka akan tersimpan dalam variable tersebut. Dan untuk menampilkannnya digunakan alert