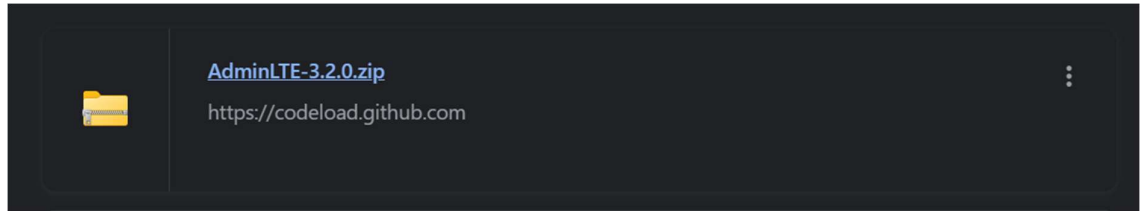
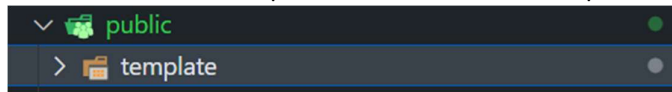


A. Template form

1. Penggunaan template AdminLTE Bootstrap Admin Dashboard Template dengan langkah pertama akses <https://adminlte.io/> , lalu klik download pada source code (zip)



2. 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



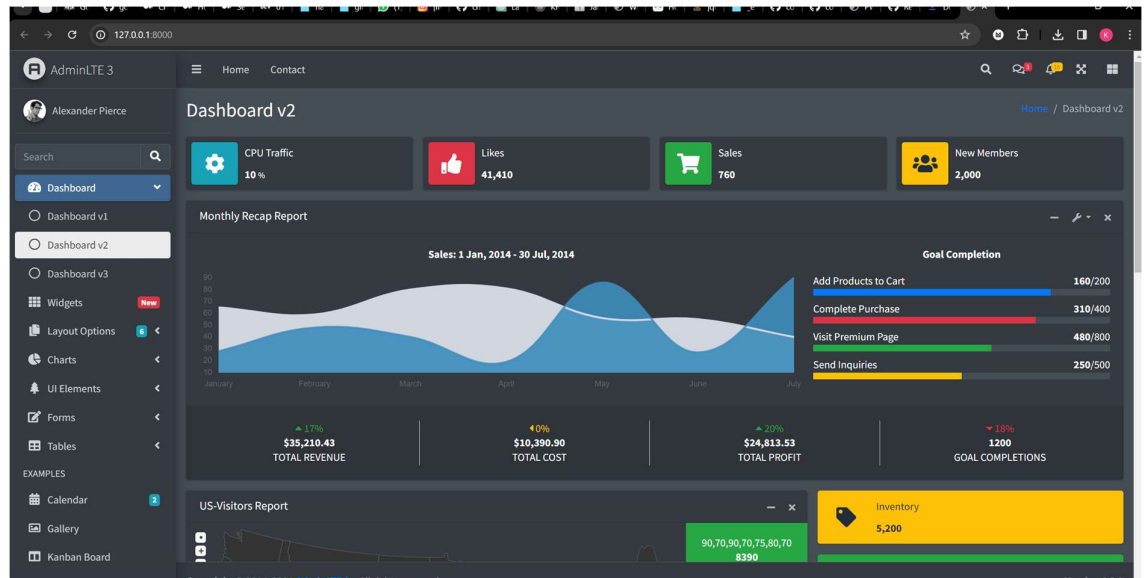
3. Copy isi public/template/index2.html pada welcome.blade.php lakukan edit agar dapat di load, edit href = "{{ asset('template/.....') }}" dan src = "{{ asset('template/.....') }}" sesuai gambar berikut



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



5. Hasil running



6. Pada bagian menu terdapat pilihan Form kita khususnya dulu pada Form yang disediakan oleh AdminLTE • 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
7. Buka file pada form> general cari keyword dengan 'general elements' dan coba modifikasi pada welcome.blade.php

```

1159      <!-- general form elements disabled -->
1160      <div class="card card-warning">
1161        <div class="card-header">
1162          <h3 class="card-title">General Elements</h3>
1163        </div>
1164        <!-- /.card-header -->
1165        <div class="card-body">
1166          <form>
1167            <div class="row">
1168              <div class="col-sm-6">
1169                <!-- text input -->
1170                <div class="form-group">
1171                  <label>Text</label>
1172                  <input type="text" class="form-control" placeholder="Enter ...">
1173                </div>
1174              </div>
1175              <div class="col-sm-6">
1176                <div class="form-group">
1177                  <label>Text Disabled</label>

```

8. Apa yang tampil?

The screenshot shows a web application interface with a dark sidebar on the left. The sidebar contains a logo 'Krisna Andika', a search bar, and a menu with 'Menu', 'Kategori', and 'User'. The main content area is titled 'Dashboard' and contains a form labeled 'Level id'. The form has a text input field with the placeholder 'id' and a blue 'Submit' button below it.

9. . 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

The screenshot shows a web application interface with a dark sidebar on the left. The sidebar contains a logo 'Krisna Andika', a search bar, and a menu with 'Menu', 'Kategori', and 'User'. The main content area is titled 'Dashboard' and contains a form labeled 'M_User tabel'. The form has four fields: 'Username' (text input), 'Nama' (text input), 'Password' (text input), and 'Level ID' (dropdown menu with '1' selected).

M_level

The screenshot shows a web application interface with a dark sidebar on the left. The sidebar contains a logo 'Krisna Andika', a search bar, and a menu with 'Menu', 'Kategori', and 'User'. The main content area is titled 'Dashboard' and contains a form labeled 'M_level tabel'. The form has two fields: 'Level Kode' (text input) and 'Level Nama' (text input). Below the form is a yellow bar with the text 'Add'.

B. Validasi pada server

1. Edit pada KategoriKontroller dengan kode:

```

1 reference | 0 overrides
77 public function store(Request $request)
78 {
79     $validated = $request->validate([
80         'kategori_kode' => 'required',
81         'kategori_nama' => 'required'
82     ]);
83
84

```

2. Tulis perbedaan penggunaan validate dengan validateWithBag!
Validatewithbag digunakan untuk memvalidasi dengan pesan store error dengan named error bag
3. 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?

```

78 {
79     $validated = $request->validate([
80         'kategori_kode' => 'required|bail|unique:posts',
81         'kategori_nama' => 'required'
82     ]);
83
84

```

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

```

15 @method('POST')
16 @if ($errors->any())
17     <div class="alert alert-danger">
18         <ul>
19             @foreach ($errors->all() as $error)
20                 <li>{{ $error }}</li>
21             @endforeach
22         </ul>
23     </div>
24 @endif

```

5. Run

Buat kategori baru

- The kategori kode field is required.
- The kategori nama field is required.

Kode Kategori

Nama Kategori

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

```

25         <div class="card-body">
26             <div class="form-group">
27                 <label for="kodeKategori">Kode Kategori</label>
28                 <input type="text" name="kodeKategori" placeholder="kode kategori" id="" class="@error('kategori_kode') is-invalid @enderror">
29             </div>
30             @error
31             <div class="alert alert-danger">{{ $message }}</div>
32         @enderror
33     </div>

```

Buat kategori baru

Kode Kategori

The kategori kode field is required.

Nama Kategori

C. Form request Validation

1. Membuat permintaan form dengan menuliskan pada terminal

```

PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS> php artisan make:request StorePostRequest

INFO Request [D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS\app\Http\Requests\StorePostRequest.php] created successfully.

PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS>

```

2. Ketik kode berikut pada Http/request/StorePostRequest

```

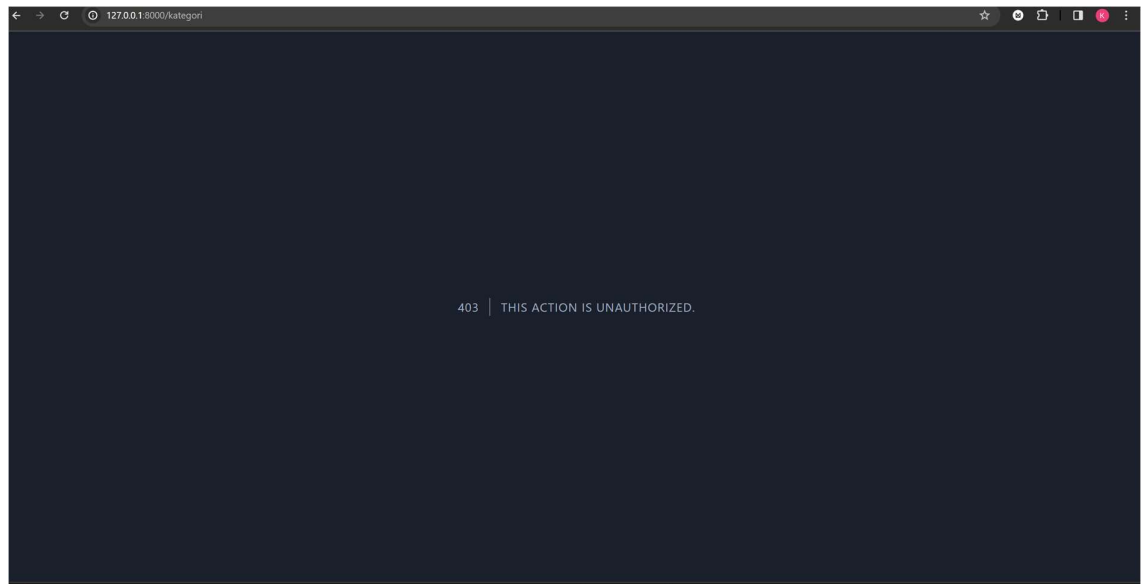
0 references | 0 overrides
22     public function rules(): array
23     {
24         return [
25             'kategori_kode' => 'required',
26             'kategori_nama' => 'required'
27         ];
28     }
29 }
30

```

```

78     public function store(StorePostRequest $request)
79     {
80         $validated = $request->validated();
81
82         $validated = $request->safe()->only(['kategori_kode', 'kategori_nama']);
83         $validated = $request->safe()->except(['kategori_kode', 'kategori_nama']);
84
85
86         KategoriModel::create(
87             [
88                 'kategori_kode' => $request->kodeKategori,
89                 'kategori_nama' => $request->namaKategori
90             ]
91         );
92         return redirect('/kategori');
93     }
94 }
95

```



3. Terapkan validasi juga pada tabel m_user dan m_level

Two screenshots of a web application interface. The top screenshot shows the 'M_User tabel' form with validation errors: 'The username field is required.', 'The name field is required.', and 'The password field is required.'. The bottom screenshot shows the 'M_Level tabel' form with validation errors: 'A title is required' and 'A message is required'. Both forms have input fields for 'Username', 'Nama', 'Password', and 'Level ID' (or 'Level Kode' and 'Level Nama' in the second form), and an 'Add' button. The interface includes a sidebar with 'Krisna Andika' logo, a search bar, and a menu with 'Menu', 'Kategori', and 'User' options.

D. CRUD

1. 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.

```
PS D:\Kuliahe wong_jenius\Semester 4\Web Lanjut\POS> php artisan make:controller POSController --resource  
  
INFO Controller [D:\Kuliahe wong_jenius\Semester 4\Web Lanjut\POS\app\Http\Controllers\POSController.php] created successfully.  
  
PS D:\Kuliahe wong_jenius\Semester 4\Web Lanjut\POS>
```

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

```
76 // POS  
77 Route::resource('m_user', POSController::class);  
78  
79
```

3. Atur pada Models/m_user

```
10 */  
11 public function up(): void  
12 {  
13     Schema::create('useri', function (Blueprint $table) {  
14         $table->id('user_id');  
15         $table->unsignedBigInteger('level_id')->index();  
16         $table->string('username', 20)->unique();  
17         $table->string('nama', 100);  
18         $table->string('password');  
19  
20         $table->timestamps();  
21  
22         $table->foreign('level_id')->references('level_id')->on('m_level');  
23     });  
24 }  
25  
26 /**  
27  * Reverse the migrations.  
28  */  
29 public function down(): void  
30 {  
31     Schema::dropIfExists('useri');  
32 }  
33 };
```

OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

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

```

2 references | 0 implementations
8  class POSController extends Controller
9  {
10     /**
11      * Display a listing of the resource.
12      */
13     0 references | 0 overrides
14     public function index()
15     {
16         //
17         $useri = UserModel::with('level')->get();
18         return view('m_user.index', compact('useri'))->with('i');
19     }
20     /**
21      * Show the form for creating a new resource.
22      */
23     0 references | 0 overrides
24     public function create()
25     {
26         return view('m_user.create');
27     }
28     /**
29      *
30      */
31     0 references | 0 overrides
32     public function store(Request $request)
33     {
34         //
35         $request->validate([
36             'user_id' => 'max 20',
37             'username' => 'required',
38             'nama' => 'required'
39         ]);
40         UserModel::create($request->all());
41         return redirect()->route('m_user.index')->with('success', 'user berhasil ditambahkan');
42     }
43     /**
44      * Display the specified resource.
45      */
46     0 references | 0 overrides
47     public function show(string $id)
48     {
49         $useri = UserModel::findOrFail($id);
50         return view('m_user.show', compact('useri'));
51     }
52 }

```



```

78     UserModel::find($id)->update($request->all());
79     return redirect()->route('m_user.index')->with('success', 'Data berhasil diupdate');
80 }
81
82 /**
83  * Remove the specified resource from storage.
84  */
85 0 references | 0 overrides
86 public function destroy(string $id)
87 {
88     //
89     $useri = UserModel::findOrFail($id)->delete();
90     return redirect()->route('m_user.index')->with('success', 'data berhasil dihapus');
91 }
92
93
94 }
95

```

5. Buatlah folder di Resources/Views/m_user dengan beberapa blade dan isian kode berikut:

1. template.blade.php

```

resources > views > m_user > template.blade.php > html > head
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>CRUD Laravel</title>
5  <link rel="stylesheet"
6  href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.mi
7  n.css">
8  </head>
9  <body>
10 <div class="container">
11 @yield('content')
12 </div>
13 </body>
14 </html>
15

```

2. index.blade.php

```

resources > views > m_user > index.blade.php > table.table.table-bordered
1  @extends('m_user/template')
2  @section('content')
3  <div class="row mt-5 mb-5">
4  <div class="col-lg-12 margin-tb">
5  <div class="float-left">
6  <h2>CRUD user</h2>
7  </div>
8  <div class="float-right">
9  <a class="btn btn-success" href="{{ route('m_user.create') }}"> Input
10 User</a></div>
11 </div>
12 </div>
13 @if ($message = Session::get('success'))
14 <div class="alert alert-success">
15 <p>{{ $message }}</p>
16 </div>
17 @endif
18 <table class="table table-bordered">
19 <tr>
20 <th width="20px" class="text-center">User id</th>
21 <th width="150px" class="text-center">Level id</th>
22 <th width="200px" class="text-center">username</th>
23 <th width="200px" class="text-center">nama</th>
24 <th width="150px" class="text-center">password</th>
25 </tr>
26 @foreach ($useri as $m_user)
27
28 <table class="table table-bordered">
29 <tr>
30 <td>{{ $m_user->user_id }}</td>
31 <td>{{ $m_user->level->level_nama }}</td>
32 <td>{{ $m_user->username }}</td>
33 <td>{{ $m_user->nama }}</td>
34 <td>{{ $m_user->password }}</td>
35 <td>{{ $m_user->m_level }}</td>
36 <td class="text-center">
37 <form class="d-flex gap-3 g-col-6" action="{{ route('m_user.destroy',$m_user->user_id) }}"
38 method="POST">
39 <a class="btn btn-info btn-sm" href="{{ route('m_user.show',$m_user->user_id) }}">Show</a>
40 <a class="btn btn-primary btn-sm" href="{{ route('m_user.edit',$m_user->user_id) }}">Edit</a>
41 @csrf
42 @method('DELETE')
43 <button type="submit" class="btn btn-danger btn-sm" onclick="return
44 confirm('Apakah Anda yakin ingin menghapus data
45 ini?')">Delete</button>
46 </form>
47 </td>
48 </tr>
49 </table>
50 @endforeach
51 </table>
52 @endsection

```

3. create.blade.php


```
2  @section('content')
3  <div class="row mt-5 mb-5">
4  <div class="col-lg-12 margin-tb">
5  <div class="float-left">
6  <h2> Show User</h2>
7  </div>
8  <div class="float-right">
9  <a class="btn btn-secondary" href="{{ route('m_user.index') }}">
10  Kembali</a>
11  </div>
12  </div>
13  </div>
14  <div class="row">
15  <div class="col-xs-12 col-sm-12 col-md-12">
16  <div class="form-group">
17  <strong>User_id:</strong>
18  {{ $useri->user_id }}
19  </div>
20  </div>
21  <div class="col-xs-12 col-sm-12 col-md-12">
22  <div class="form-group">
23  <strong>Level_id:</strong>
24  {{ $useri->level_id }}
25  </div>
26  </div>
27  <div class="col-xs-12 col-sm-12 col-md-12">
```

```

14 <div class="row">
15     <div class="col-xs-12 col-sm-12 col-md-12">
16         <div class="form-group">
17             <strong>Username:</strong>
18             {{ $useri->username }}
19         </div>
20     </div>
21     <div class="col-xs-12 col-sm-12 col-md-12">
22         <div class="form-group">
23             <strong>Nama:</strong>
24             {{ $useri->nama }}
25         </div>
26     </div>
27     <div class="col-xs-12 col-sm-12 col-md-12">
28         <div class="form-group">
29             <strong>Password:</strong>
30             {{ $useri->password }}
31         </div>
32     </div>
33 </div>
34 @endsection
35
36

```

5. edit.blade.php

```

1  @extends('m_user/template')
2  @section('content')
3  <div class="row mt-5 mb-5">
4  <div class="col-lg-12 margin-tb">
5  <div class="float-left">
6  <h2>Edit User</h2>
7  </div>
8  <div class="float-right">
9  <a class="btn btn-secondary" href="{{ route('m_user.index') }}">
10 Kembali</a>
11 </div>
12 </div>
13 </div>
14 @if ($errors->any())
15 <div class="alert alert-danger">
16     <strong>Ops!</strong> Error <br><br>
17     <ul>
18 @foreach ($errors->all() as $error)
19 <li>{{ $error }}</li>
20 @endforeach
21 </ul>
22 </div>
23 @endif
24 <form action="{{ route('m_user.update', $user->user_id) }}"
25 method="POST">
26 @csrf
27 @method('PUT')
28 <div class="row">
29 <div class="col-xs-12 col-sm-12 col-md-12">
30 <div class="form-group">
31 <strong>User_id:</strong>
32 <input type="text" name="userid" value="{{ $user->user_id }}"
33 class="form-control" placeholder="Masukkan user id">
34 </div>
35 </div>
36 <div class="col-xs-12 col-sm-12 col-md-12">
37 <div class="form-group">
38 <strong>Level_id:</strong>
39 <input type="text" name="levelid" value="{{ $user->level_id }}"
40 class="form-control" placeholder="Masukkan level">
41 </div>
42 </div>
43 <div class="col-xs-12 col-sm-12 col-md-12">
44 <div class="form-group">

```



```
43 <div class="col-xs-12 col-sm-12 col-md-12">
44 <div class="form-group">
45 <strong>Username:</strong>
46 <input type="text" value= "{{ $useri->username }}" class="form-control" name="username" placeholder="Masukkan Nomor username"">
47 </div>
48 </div>
49 <div class="col-xs-12 col-sm-12 col-md-12">
50 <div class="form-group">
51 <strong>nama:</strong>
52 <input type="text" value= "{{ $useri->nama }}" name="nama"
53 class="form-control" placeholder="Masukkan nama"></input>
54 </div>
55 </div>
56 <div class="col-xs-12 col-sm-12 col-md-12">
57 <div class="form-group">
58 <strong>Password:</strong>
59 <input type="password" value= "{{ $useri->password
60 }}" name="password" class="form-control" placeholder="Masukkan
61 password"></input>
62 </div>
63 </div>
64 <div class="col-xs-12 col-sm-12 col-md-12 text-center">
65 <button type="submit" class="btn btn-primary">Update</button>
66 </div>
67 </div>
68 </form>
69 @endsection
```

Hasil

CRUD user Input User

User id	Level id	username	nama	password	
1	1	admin	administrator	\$2y\$12\$feLiaoOYohYyF2ZXGRxmUuMV7qUxrbmYXoR1Rn2ExelfYgJr0/Epi	Show Edit Delete
2	2	manager	Manager	\$2y\$12\$PT4sOHCIZYQk/QsIWmlyJuaxlJNOv.v13Cvd8PLKCP6y9IXYdRMbG	Show Edit Delete
3	3	Staff	staff/kasir	\$2y\$12\$GeYigbyc9v.F9biDxQWQSepeJW.G6H1aoWAAglsHvmk0CK4NBYtGq	Show Edit Delete
5	3	Customer-1	pelanggan pertama	\$2y\$12\$BUAdkDEWWPDwxxKDt7ubWeMCDIROuFAhoNjDPnumLLNUoXH.p4yme	Show Edit Delete
6	2	manager_dua	manager 2	\$2y\$12\$qpFj1BXmIcCsreoffCH/vuo709TqMb98K8UYFuZKx57Yt1sl2X9bm	Show Edit Delete
7	2	manager22	Manager Dua Dua	\$2y\$12\$MSd9qdzEpnzerDbL2499SuhoynwEyCPi9U7oVbQReH1ViXTycBDRm	Show Edit Delete
8	2	manager33	Manager Tiga Tiga	\$2y\$12\$I9ygpvOavTy7qJehatCxf.O3WnE84AUmKA7BYRhTZ7ZhXYAxClbqK	Show Edit Delete
9	2	manager56	Manager55	\$2y\$12\$8/9BTfkih08h.ciebdbjMu8KhQUjGy3ump9/zlb.8RMARd6yNI8ja	Show Edit Delete
10	2	manager11	Manager11	\$2y\$12\$EXMfbfPIUSdtWmomfI4UtOvTj33Qqnp1eVxmM.8Fqk9Mh0VfCicrO	Show Edit Delete
12	2	manager13	Manager13	\$2y\$12\$tel83ZeMrX.eKxloiRw9tOddd7Y6A1HG4OHcM5UaZXn1U1pXq3cC	Show Edit Delete

Soal

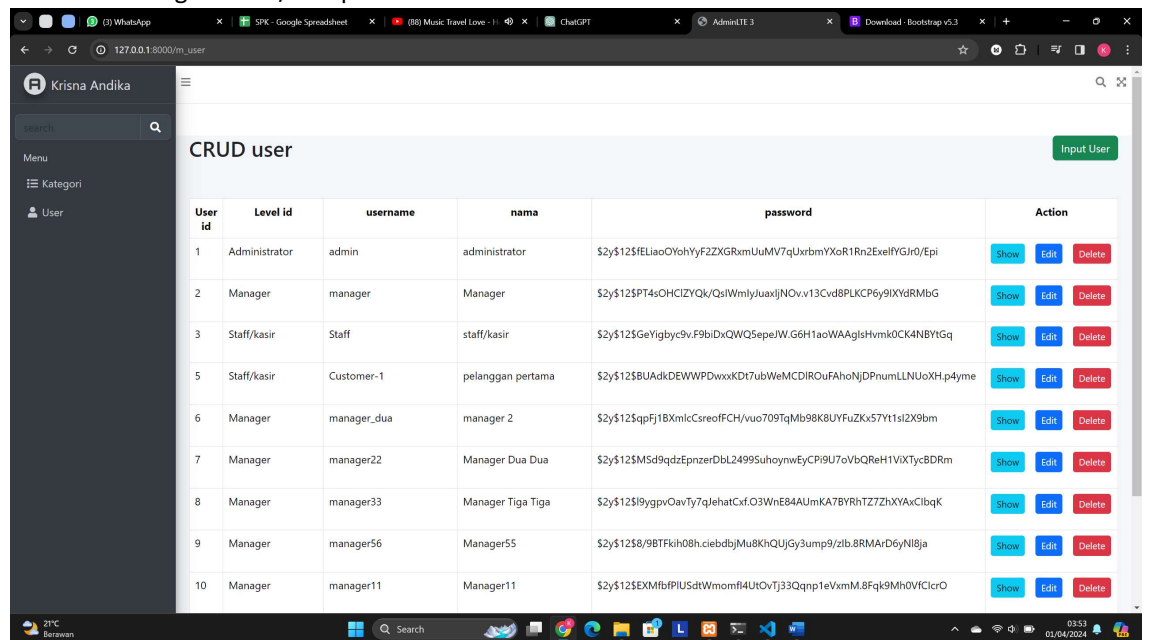
1. Coba tampilkan level_id pada halaman web tersebut dimana field ini merupakan foreign key

CRUD user

Input User

User id	Level id	username	nama	password	
1	Administrator	admin	administrator	\$2y\$12\$fELiaoOYohYyF2ZXGRxmUuMV7qUxrbmYXoR1Rn2ExelfYGJr0/Epi	Show Edit Delete
2	Manager	manager	Manager	\$2y\$12\$PT4sOHCIZYQk/QsIWmlyJuaxlJNOv.v13Cvd8PLKCP6y9IXYdRMBG	Show Edit Delete
3	Staff/kasir	Staff	staff/kasir	\$2y\$12\$GeYigbyc9v.F9biDxQWQ5epeJW.G6H1aoWAAGlsHvmk0CK4NBYtGq	Show Edit Delete
5	Staff/kasir	Customer-1	pelanggan pertama	\$2y\$12\$BUAdkDEWWDWxxKDt7ubWeMCDIROuFAhoNjDPnumLLNUoXH.p4yme	Show Edit Delete
6	Manager	manager_dua	manager 2	\$2y\$12\$qpFj1BXmIcCsreofFCH/vuo709TqMb98K8UYFuZKx57Yt1sl2X9bm	Show Edit Delete
7	Manager	manager22	Manager Dua Dua	\$2y\$12\$MSd9qdzEpnzerDbL2499SuhoynwEyCPi9U7oVbQReH1ViXTycBDRm	Show Edit Delete
8	Manager	manager33	Manager Tiga Tiga	\$2y\$12\$9ygpvOavTy7qJehatCxf.O3WnE84AUmKA7BYRhTZ7ZhXYaXCibqK	Show Edit Delete
9	Manager	manager56	Manager55	\$2y\$12\$8/9BTFkih08h.ciebdbjMu8KhQUjGy3ump9/zlb.8RMARd6yNI8ja	Show Edit Delete
10	Manager	manager11	Manager11	\$2y\$12\$EXMfbfPIUSdtWmomfI4UtOvTj33Qqnp1eVxmM.8Fqk9Mh0VfClcrO	Show Edit Delete
12	Manager	manager13	Manager13	\$2y\$12\$tel83ZeMrX.eKxloiRw9tOoddb7Y6A1HG4OHcM5UaZXn1U1pXq3cC	Show Edit Delete

2. Modifikasi dengan tema/ template kesukaan Anda



3. 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 diisi.
 Untuk error apabila terdapat error maka akan tersimpan dalam variable tersebut.
 Dan untuk menampilkannya digunakan alert