

A. Identitas Job Sheet

IDENTITAS JOB SHEET

Perguruan Tinggi	: Politeknik Negeri Bengkalis	Pertemuan Ke	:
Jurusan/Program Studi	: Teknik Informatika	Job Ke	: 1
Kode Mata Kuliah	:	Halaman	:
Nama Mata Kuliah	: Pemograman Web Framework	Mulai Berlaku	:

B. Komponen Job Sheet

1.1. Judul Job

JOB I Pengenalan Framework Laravel,

1.2. Tujuan

2. Mahasiswa mampu mengenal lingkungan kerja pemrograman web dengan menggunakan framework laravel.
3. Mahasiswa mampu menerapkan penggunaan framework laravel dalam pemrograman website.
4. Mahasiswa mampu menjalankan website yang dibuat dengan menggunakan framework Laravel.
5. Mahasiswa mampu membuat middleware, membuat controller, konfigurasi route, konfigurasi database, serta memahami blade templating engine laravel.

1.3. Dasar Teori

Laravel adalah sebuah framework yang digunakan dalam pengembangan website yang berbasis bahasa pemrograman PHP untuk memudahkan pembuatan sebuah website sehingga menjadi lebih dinamis. Dengan adanya framework laravel ini, dapat membuat bahasa pemrograman PHP menjadi lebih powerfull, apalagi framework ini akan selalu diupdate sehingga akan ada fitur-fitur baru di setiap update nya. Framework laravel menggunakan struktur MVC (Model View Controller), yang merupakan model aplikasi yang memisahkan antara data dan tampilan yang berdasarkan kepada komponen dari aplikasi. Dengan model ini, pengguna laravel dapat dengan mudah mempelajari framework ini serta menjadikan pembuatan aplikasi berbasis website lebih cepat.

Framework laravel juga menyediakan fitur bawaan yang lengkap, salah satunya adalah fitur autentikasi sehingga membuat framework ini lebih cenderung berfokus kepada level end-user. Framework laravel memiliki keunggulan dari segi penulisan kode yang sederhana atau dari segi tampilan yang membuat web developer untuk membangun sebuah website dengan fitur yang lengkap secara efektif dan efisien.

Dalam bagian ini, anda akan membuat sebuah login dan tampilan blade dalam laravel serta bagaimana caranya untuk melakukan beberapa konfigurasi lainnya seperti mengubah environment variable, menambahkan custom middleware, mengkonfigurasi database, membuat sebuah controller, dan melakukan konfigurasi terhadap route dalam aplikasi laravel.

1.4. Alat dan Bahan

Alat dan bahan yang diperlukan dalam praktikum ini adalah sebagai berikut:

1. computer/laptop dengan windows versi 7 keatas.
2. Code editor visual studio code.
3. XAMPP.
4. Browser.
5. PHP versi 8.2 Keatas.

1.5. Persiapan/Instalasi Composer

Sebelum menggunakan framework Laravel, terlebih dahulu anda harus menginstall composer. Ikuti langkah berikut dengan teliti agar tidak terjadi kesalahan saat melakukan proses instalasi:

1. Kunjungi halaman berikut <https://getcomposer.org>, lalu halaman seperti pada gambar dibawah ini akan muncul. Kemudian klik bagian “Download”.



- Setelah itu akan muncul halaman seperti gambar berikut. Kemudian klik bagian yang didalam kotak merah.

[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

Download Composer Latest: v2.5.8

Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#), it will install the latest composer version whenever it is executed.

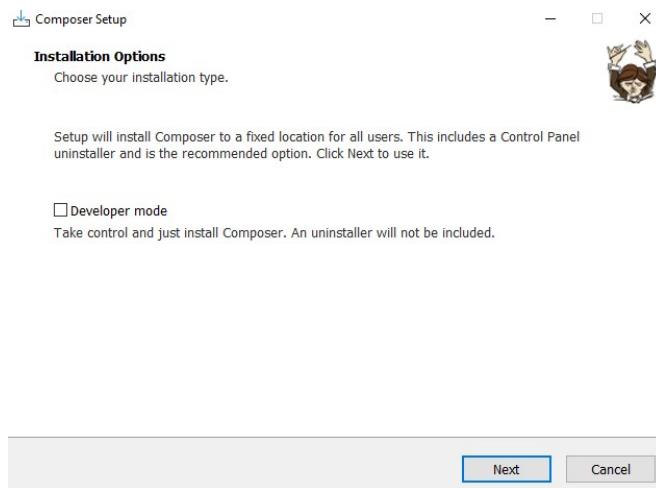
Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

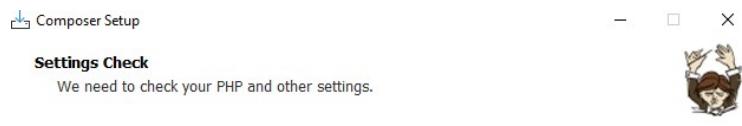
```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === 'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f123'
    php composer-setup.php
    php -r "unlink('composer-setup.php');"
```

This installer script will simply check some `php.ini` settings, warn you if they are set incorrectly, and then download the latest `composer.phar` in the current directory. The 4 lines above will, in order:

- Setelah itu file composer akan di download. Setelah file selesai di download, double click pada file tersebut.
- Jika muncul pesan berikut, klik bagian “Install for all users (recommended)”.
- Maka akan muncul gambar seperti dibawah ini, kemudian klik bagian “Next”.



- Setelah itu akan muncul tampilan berikut, jika pada perangkat sudah terdapat bahasa pemrograman PHP, maka akan otomatis terisi dengan direktori folder PHP yang ada di perangkat. Selanjutnya klik bagian “Next”.

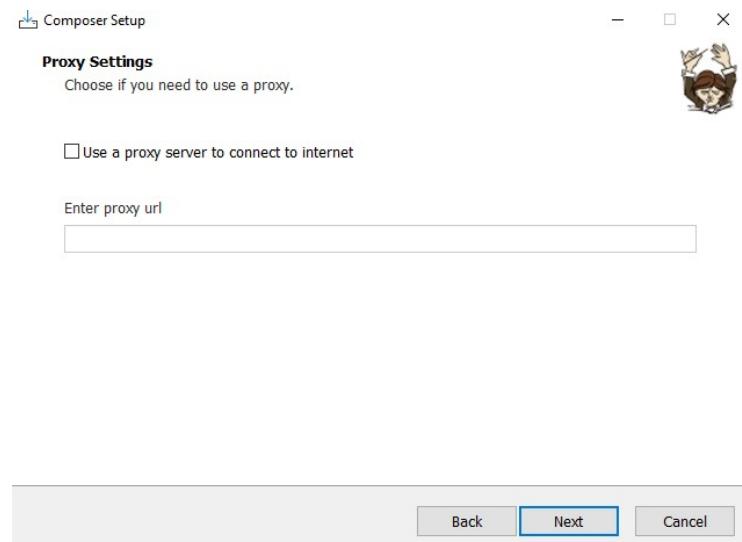


Choose the command-line PHP you want to use:

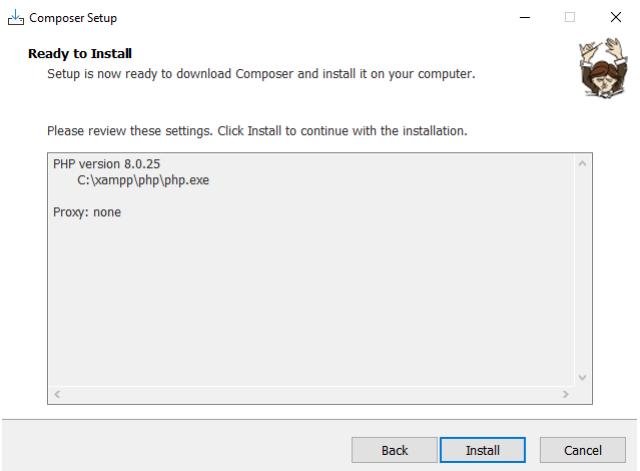
C:\xampp\php\php.exe

This is the PHP in your path. Click Next to use it.

7. Setelah itu akan muncul tampilan berikut, klik bagian “Next”.



8. Setelah itu akan muncul tampilan berikut, klik bagian “Install” maka proses instalasi composer akan dilakukan. Tunggu sampai proses instalasi selesai.



9. Jika proses instalasi composer selesai, maka akan muncul tampilan berikut lalu klik bagian “Next”.



Important

You must open a new command window to use Composer for the first time, because your environment has changed and running programs may not be aware of this.

If this does not work, you will have to do one of the following:

- Close all File Explorer windows, then open a new command window. OR
- Logoff and Logon again, then open a new command window.

As a last resort, you may need to restart your computer.

Next

10. Setelah itu akan muncul tampilan berikut, kemudian klik bagian “Finish”. Proses instalasi composer sudah selesai.



11. Untuk mengetahui apakah composer sudah terinstall kedalam perangkat atau tidak, buka cmd lalu ketikkan perintah berikut "**composer -v**". Jika composer sudah terinstall kedalam perangkat, maka tampilan berikut akan muncul setelah menjalankan perintah diatas.

```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>composer -v

[Logo]

Composer version 2.5.8 2023-06-09 17:13:21

Usage:
  command [options] [arguments]

Options:
  -h, --help          Display help for the given command. When no command is given display help for the list
  command
  -q, --quiet         Do not output any message
  -V, --version       Display this application version
  --ansi=ansi         Force (or disable --no-ansi) ANSI output
  -n, --no-interaction Do not ask any interactive question
  --profile           Display timing and memory usage information
  --no-plugins        Whether to disable plugins.
  --no-scripts        Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache          Prevent use of the cache
  -v[vvvvvv], --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
  3 for debug
```

1.6. Membuat Project Laravel

Setelah berhasil melakukan proses instalasi composer pada perangkat anda, selanjutnya anda akan membuat aplikasi Laravel dengan menggunakan perintah via terminal dari perangkat anda. Pertama-tama, persiapkan folder yang nantinya akan anda gunakan untuk menyimpan aplikasi Laravel anda, jika sudah maka dalam folder tempat anda akan menyimpan aplikasi Laravel anda, buka terminal dalam direktori tersebut kemudian ketikkan perintah dibawah ini:

```
composer create-project laravel/laravel ecommerce-laravel
```

Jika aplikasi Laravel berhasil dibuat, maka akan muncul pesan seperti gambar dibawah ini pada terminal anda:

```
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE

78 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

[INFO] No publishable resources for tag [laravel-assets].  
  
No security vulnerability advisories found.
> @php artisan key:generate --ansi  
  
[INFO] Application key set successfully.  
  
> @php -r "file_exists('database/database.sqlite') || touch('database/database.sqlite');"  
> @php artisan migrate --graceful --ansi  
  
[INFO] Preparing database.  
  
Creating migration table ..... 9.01ms DONE  
  
[INFO] Running migrations.  
  
0001_01_01_000000_create_users_table ..... 20.81ms DONE
0001_01_01_000001_create_cache_table ..... 7.41ms DONE
0001_01_01_000002_create_jobs_table ..... 16.81ms DONE  
  
D:\Project\Dir\Modul Laravel 11>
```

Setelah itu, anda dapat membuka visual studio code lalu buka folder atau aplikasi Laravel yang sudah anda buat sebelumnya. Jika anda perhatikan, maka akan terdapat banyak folder yang memiliki fungsinya masing-masing dengan penjelasan sebagai berikut ini:

1. Folder app: Folder ini berisi inti dari aplikasi Anda, termasuk model-model, controller-controller, dan kode bisnis utama. Model-model digunakan untuk berinteraksi dengan database menggunakan Eloquent ORM. Controller-controller digunakan untuk menangani permintaan HTTP dan menghubungkan model dengan tampilan.
2. Folder bootstrap: Folder ini berisi file yang digunakan untuk memuat framework Laravel dan melakukan konfigurasi awal sebelum aplikasi dimulai. Ini termasuk file app.php yang memuat komponen-komponen penting.
3. Folder config: Konfigurasi aplikasi disimpan di sini. Anda dapat mengatur berbagai pengaturan seperti database, layanan pihak ketiga, dan banyak lagi.
4. Folder database: Berisi file-file migrasi dan pengaturan penyemaian data awal (seeder) untuk mengelola skema database dan data awal.
5. Folder public: Inilah titik masuk bagi permintaan HTTP dari pengguna. File index.php di sini menjadi titik awal eksekusi aplikasi Anda. Selain itu, Anda akan

menemukan asset-asset publik seperti gambar, file CSS, dan JavaScript yang dapat diakses oleh publik.

6. Folder resources: Ini berisi tampilan (views), file asset yang belum diolah (seperti file CSS, JavaScript, atau template), dan terjemahan (translations).
7. Folder routes: Berisi file web.php yang mendefinisikan rute-rute aplikasi berbasis HTTP. Anda juga dapat memiliki file lain di sini untuk mengatur rute-rute lain seperti API.
8. Folder storage: Menyimpan file-file cache, sesi, dan lainnya yang dihasilkan oleh aplikasi. Ini juga termasuk folder app, framework, dan logs.
9. Folder tests: Berisi pengujian aplikasi dengan menggunakan PHPUnit. Anda dapat melakukan pengujian unit, pengujian fitur, dan pengujian integrasi di sini.
10. Folder vendor: Berisi dependensi-dependensi yang dimuat oleh Composer, manajer dependensi PHP. Folder ini dihasilkan oleh Composer dan tidak perlu dimodifikasi secara manual.
11. File .env: File lingkungan (environment) yang berisi pengaturan spesifik proyek seperti kredensial database, konfigurasi email, dan variabel lingkungan lainnya.
12. File composer.json dan composer.lock: File ini berisi konfigurasi dependensi proyek dan dihasilkan oleh Composer.

Selanjutnya, anda akan menjalankan aplikasi Laravel yang sudah anda buat dengan menggunakan terminal yang ada dalam visual studio code. Untuk dapat menjalankan aplikasi Laravel, anda perlu mengetikkan perintah dibawah ini pada terminal tersebut:

```
php artisan serve
```

Jika sudah, anda dapat mengakses url dari hasil perintah yang anda jalankan tersebut dengan menggunakan browser. Berikut ini adalah tampilan awal dari aplikasi Laravel 11:

The screenshot shows the official Laravel website. At the top right is the Laravel logo. Below it is a search bar and a version dropdown. To the left is a sidebar with a navigation menu. The main content area has several sections: 1. **Documentation**: A red circular icon with a book symbol. Text: "Laravel has wonderful documentation covering every aspect of the framework. Whether you are a newcomer or have prior experience with Laravel, we recommend reading our documentation from beginning to end." A red arrow points to the right. 2. **Laracasts**: A red circular icon with a video camera symbol. Text: "Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process." A red arrow points to the right. 3. **Laravel News**: A red circular icon with a newspaper symbol. Text: "Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials." A red arrow points to the right. 4. **Vibrant Ecosystem**: A red circular icon with a globe symbol. Text: "Laravel's robust library of first-party tools and libraries, such as [Forge](#), [Vapor](#), [Nova](#), [Envoyer](#), and [Herd](#) help you take your projects to the next level. Pair them with powerful open source libraries like [Cashier](#), [Dusk](#), [Echo](#), [Horizon](#), [Sanctum](#), [Telescope](#), and more." A red arrow points to the right. At the bottom center of the page is the text "Laravel v11.22.0 (PHP v8.2.12)".

Untuk melanjutkan ke langkah berikutnya, Anda perlu mendownload asset yang dibutuhkan dalam project laravel 11 ini dengan mengunjungi link google drive dibawah ini.

<https://drive.google.com/file/d/1SsscweS2nkbDLpNxRRa5HUwxl2kVbkQC/view?usp=sharing>

Setelah Anda mendownload file asset dari google drive diatas, selanjutnya Anda perlu meng ekstrak file zip tersebut lalu copy folder assets beserta isi yang ada didalam folder tersebut, kemudian paste kan kedalam folder public dalam project laravel 11 Anda.

Untuk langkah selanjutnya, Anda akan membuat fungsi login dengan menggunakan middleware yang terdapat dalam laravel. Anda juga akan mempelajari bagaimana caranya untuk mengimplementasikan multi-authentication pada laravel 11. Simak langkah-langkah nya dengan cermat agar tidak terjadi kesalahan.

Sebelum Anda melanjutkan ke langkah selanjutnya, anda akan menginstall sebuah library alert dengan nama sweetalert. Untuk melakukan proses instalasi library

tersebut, ketikkan perintah berikut dalam terminal direktori project laravel Anda menggunakan terminal vscode seperti berikut ini:

```
composer require realrashid/sweet-alert
```

Jika proses instalasi sudah berhasil, maka akan muncul pesan seperti gambar dibawah ini pada terminal Anda:

```
./composer.json has been updated
Running composer update realrashid/sweet-alert
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking realrashid/sweet-alert (v7.2.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing realrashid/sweet-alert (v7.2.0): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

[INFO] Discovering packages.

laravel/sail ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
realrashid/sweet-alert ..... DONE

79 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

[INFO] No publishable resources for tag [laravel-assets].

No security vulnerability advisories found.
Using version ^7.2 for realrashid/sweet-alert
PS D:\Project\Dir\Modul Laravel 11\ecommerce-laravel>
```

Sebelum dapat menggunakan library sweetalert tersebut, Anda perlu mempublikasikan library tersebut yang terdapat dalam folder vendor agar dapat digunakan. Untuk melakukan proses publikasi folder vendor tersebut, ketikkan perintah dibawah ini dalam terminal direktori project laravel Anda menggunakan terminal vscode seperti berikut ini:

```
php artisan vendor:publish --all
```

Jika proses publikasi folder vendor berhasil, maka akan akan muncul pesan seperti gambar dibawah ini pada terminal Anda:

```
[INFO] Publishing assets.

Copying directory [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\laravel\framework\src\Illuminate\Foundation\Exceptions\views] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\resources\views\errors] DONE
Copying directory [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\laravel\framework\src\Illuminate\Notifications\resources\views] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\resources\views\vendor\notifications] DONE
Copying directory [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\laravel\framework\src\Illuminate\Pagination\resources\views] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\resources\views\vendor\pagination] DONE
Copying directory [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\realrashid\sweet-alert\resources\views] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\resources\views\vendor\sweetalert] DONE
Copying file [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\realrashid\sweet-alert\src\config\sweetalert.php] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\config\sweetalert.php] DONE
Copying directory [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\realrashid\sweet-alert\resources\js] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\public\vendor\sweetalert] DONE
Copying directory [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\laravel\framework\src\Illuminate\Mail\resources\views] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\resources\views\vendor\mail] DONE
Copying directory [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\laravel\sail\runtimes] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\docker] DONE
Copying file [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\laravel\sail\bin\sail] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\sail] DONE
Copying directory [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\laravel\sail\database] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\docker] DONE
Copying file [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\laravel\tinker\config\tinker.php] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\config\tinker.php] DONE
```

Setelah itu, jangan lupa untuk menambahkan kode dibawah ini pada file .env:

```
● ● ●
1 SWEET_ALERT_CONFIRM_DELETE_CONFIRM_BUTTON_TEXT='Ya, hapus!'
2 SWEET_ALERT_CONFIRM_DELETE_CANCEL_BUTTON_TEXT='Tidak, batalkan'
3 SWEET_ALERT_CONFIRM_DELETE_SHOW_CANCEL_BUTTON=true
4 SWEET_ALERT_CONFIRM_DELETE_SHOW_CLOSE_BUTTON=false
5 SWEET_ALERT_CONFIRM_DELETE_ICON='warning'
6 SWEET_ALERT_CONFIRM_DELETE_SHOW_LOADER_ON_CONFIRM=true
```

Setelah berhasil melakukan instalasi library sweetalert, maka langkah selanjutnya adalah membuat controller. Controller yang akan Anda buat adalah AuthController, yang berfungsi untuk mengatasi logika autentikasi seperti register, login, dan logout, namun Anda hanya akan membuat fungsi untuk login dan logout. Untuk membuat AuthController tersebut, ketikkan perintah berikut dalam terminal direktori project laravel Anda menggunakan terminal vscode seperti berikut ini:

```
php artisan make:controller Auth/AuthController
```

Setelah berhasil membuat AuthController, tulislah kode dibawah ini pada controller tersebut yang berlokasi dalam **app/Http/Controllers/Auth/AuthController.php** dan usahakan jangan sampai ada yang typo:

```
1 namespace App\Http\Controllers\Auth;
2
3 use App\Http\Controllers\Controller;
4 use Illuminate\Http\Request;
5 use Illuminate\Support\Facades\Auth;
6 use RealRashid\SweetAlert\Facades\Alert;
7 use Illuminate\Support\Facades\Validator;
8
9 class AuthController extends Controller
10 {
11     public function login(Request $request)
12     {
13         $validator = Validator::make($request->all(), [
14             'email' => 'required|email:dns',
15             'password' => 'required|min:8|max:15',
16         ]);
17
18         if ($validator->fails()) {
19             Alert::error('Error', 'Pastikan semua email dan password terisi dengan benar!');
20             return redirect()->back();
21         }
22
23         if (Auth::guard('admin')->attempt(['email' => $request->email, 'password' => $request->password])) {
24             toast('Selamat datang admin!', 'success');
25             return redirect()->route('admin.dashboard');
26         } elseif (Auth::attempt(['email' => $request->email, 'password' => $request->password])) {
27             toast('Selamat datang!', 'success');
28             return redirect()->route('user.dashboard');
29         } else {
30             Alert::error('Login Gagal!', 'Email atau password tidak valid!');
31             return redirect()->back();
32         }
33     }
34
35     public function admin_logout() {
36         Auth::guard('admin')->logout();
37         toast('Berhasil logout!', 'success');
38         return redirect('/');
39     }
40
41     public function user_logout() {
42         Auth::logout();
43         toast('Berhasil logout!', 'success');
44         return redirect('/');
45     }
46 }
```

Setelah berhasil membuat dan menulis kode dalam AuthController tersebut, langkah selanjutnya adalah membuat custom middleware dengan nama AdminMiddleware, yang nantinya akan digunakan untuk menghandle admin yang akan login. Untuk membuat AdminMiddleware tersebut, ketikkan perintah dibawah ini pada terminal vscode Anda:

```
php artisan make:middleware AdminMiddleware
```

Setelah berhasil menambahkan middleware baru dengan nama AdminMiddleware, Anda perlu menuliskan kode berikut ini dan pastikan tidak ada typo sama sekali agar dapat berjalan dengan benar. File AdminMiddleware tersebut terletak dalam **app/Http/Middleware/AdminMiddleware.php**.

```
1 namespace App\Http\Middleware;
2
3 use Closure;
4 use Illuminate\Http\Request;
5 use Illuminate\Support\Facades\Auth;
6 use Symfony\Component\HttpFoundation\Response;
7
8 class AdminMiddleware
9 {
10     /**
11      * Handle an incoming request.
12      *
13      * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
14      */
15     public function handle(Request $request, Closure $next): Response
16     {
17         if (!Auth::guard('admin')->check()) {
18             return redirect('/');
19         }
20
21         return $next($request);
22     }
23 }
```

Setelah menuliskan kode diatas pada AdminMiddleware yang sudah Anda buat, selanjutnya Anda perlu mendaftarkan middleware tersebut agar dikenali oleh laravel. Untuk mendaftarkan middleware tersebut, masuk kedalam file app.php yang terdapat dalam path **bootstrap/app.php** kemudian tulislah kode dibawah ini:

```
● ● ●
```

```
1 use Illuminate\Foundation\Application;
2 use Illuminate\Foundation\Configuration\Exceptions;
3 use Illuminate\Foundation\Configuration\Middleware;
4
5 return Application::configure(basePath: dirname(__DIR__))
6     →withRouting(
7         web: __DIR__.'/../routes/web.php',
8         commands: __DIR__.'/../routes/console.php',
9         health: '/up',
10    )
11    →withMiddleware(function (Middleware $middleware) {
12        $middleware→alias([
13            'admin' ⇒ \App\Http\Middleware\AdminMiddleware::class,
14        ]);
15    })
16    →withExceptions(function (Exceptions $exceptions) {
17        //
18    })→create();
```

Untuk langkah selanjutnya, Anda perlu membuat sebuah model dengan nama Admin dan migration dari model tersebut. Fungsi daripada model tersebut adalah memungkinkan Anda untuk berinteraksi dengan tabel tertentu dalam database tanpa perlu menuliskan langsung raw query, sedangkan migrations berfungsi mengelola skema database yang memungkinkan Anda untuk membuat, memodifikasi, atau menghapus tabel dalam database. Untuk membuat model beserta migration nya, Anda perlu mengetikkan perintah berikut ini dalam terminal vscode Anda:

```
php artisan make:model Admin -m
```

Setelah berhasil membuat model serta migration dari Admin, tuliskan kode dibawah ini dalam file Admin.php yang berada dalam **app/Models/Admin.php** dan pastikan tidak ada kesalahan dalam penulisan kode.

```
● ● ●  
1 namespace App\Models;  
2  
3 use Illuminate\Foundation\Auth\User as Authenticatable;  
4 use Illuminate\Database\Eloquent\Factories\HasFactory;  
5 use Illuminate\Database\Eloquent\Model;  
6  
7 class Admin extends Authenticatable  
8 {  
9     use HasFactory;  
10  
11    protected $fillable = [  
12        'name', 'username', 'email', 'password'  
13    ];  
14 }
```

Selanjutnya, masuk kedalam file migrations admin (sesuaikan nama file nya dengan yang ada dalam project laravel Anda) yang berlokasi dalam **database/migrations/2024_09_10_050343_create_admins_table.php** kemudian tulislah kode dibawah ini:

```
● ● ●
1 use Illuminate\Database\Migrations\Migration;
2 use Illuminate\Database\Schema\Blueprint;
3 use Illuminate\Support\Facades\Schema;
4
5 return new class extends Migration
6 {
7     /**
8      * Run the migrations.
9      */
10    public function up(): void
11    {
12        Schema::create('admins', function (Blueprint $table) {
13            $table->id();
14            $table->string('name');
15            $table->string('username');
16            $table->string('email');
17            $table->string('password');
18            $table->timestamps();
19        });
20    }
21
22    /**
23     * Reverse the migrations.
24     */
25    public function down(): void
26    {
27        Schema::dropIfExists('admins');
28    }
29};
```

Selanjutnya, masuk kedalam file migrations user (sesuaikan nama file nya dengan yang ada dalam project laravel Anda) **database/migrations/0001_01_01_000000_create_users_table.php**, kemudian tambahkan kode yang diberikan tanda warna kuning berikut ini:

```
1 Schema::create('users', function (Blueprint $table) {
2     $table->id();
3     $table->string('name');
4     $table->string('email')->unique();
5     $table->timestamp('email_verified_at')->nullable();
6     $table->string('password');
7     $table->bigInteger('point'); $table->bigInteger('point');
8     $table->rememberToken();
9     $table->timestamps();
10});
```

Kemudian, masuk kedalam file models user yang berlokasi dalam **app/models/User.php** lalu tambahkan kode yang diberi tanda wara kuning dibawah ini:

```
1 protected $fillable = [
2     'name',
3     'email',
4     'password',
5     'point',
6];
```

Setelah itu, masuk kedalam file DatabaseSeeder.php yang berlokasi dalam **database/seeders/DatabaseSeeder.php** lalu tulislah kode dibawah ini:

```
1 namespace Database\Seeders;
2
3 use App\Models\User;
4 use App\Models\Admin;
5 // use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7
8 class DatabaseSeeder extends Seeder
9 {
10     /**
11      * Seed the application's database.
12      */
13     public function run(): void
14     {
15         // User::factory(10)->create();
16
17         User::create([
18             'name' => 'user1',
19             'email' => 'user@gmail.com',
20             'password' => bcrypt('123456789'),
21             'point' => 10000,
22         ]);
23
24         Admin::create([
25             'name' => 'admin',
26             'username' => 'Admin',
27             'email' => 'admin@gmail.com',
28             'password' => bcrypt('123456789'),
29         ]);
30     }
31 }
```

Langkah selanjutnya adalah menjalankan perintah migration dalam laravel. Namun sebelum menjalankan perintah tersebut, Anda perlu membuat database baru dengan nama ecommerce_laravel. Setelah itu, masuk kedalam file .env lalu ubah kodennya menjadi seperti berikut ini:

```
1 DB_CONNECTION=mysql
2 DB_HOST=127.0.0.1
3 DB_PORT=3306
4 DB_DATABASE=ecommerce_laravel
5 DB_USERNAME=root
6 DB_PASSWORD=
```

Setelah itu, Anda dapat menjalankan perintah migration untuk menambahkan tabel dalam database yang sudah Anda buat tersebut. Untuk menambahkannya, ketikkan perintah dibawah ini pada terminal Anda:

```
php artisan migrate
```

Jika sudah berhasil, maka tabel akan otomatis terbuat dalam database ecommerce_laravel. Selanjutnya Anda akan memasukkan data dalam tabel users dan admins yang terdapat dalam file DatabaseSeeder.php yang baru saja Anda modifikasi. Untuk mengisi data otomatis kedalam kedua tabel tersebut, ketikkan perintah dibawah ini dalam terminal Anda:

```
php artisan db:seed
```

Jika berhasil, maka tabel admins dan users akan berisi sebuah data yang akan Anda gunakan untuk login.

Menampilkan Data Pada Tabel User dan Tabel Admin

Untuk pengaturan pemanggilan data pada Tabel User dan Admin maka buatlah controller baru dengan nama **ListController**

```
php artisan make:controller ListController
```

Selanjutnya tambahkan sintak berikut pada ListController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Admin;
use App\Models\User;

class ListController extends Controller
{
    public function index()
    {
        $admins = Admin::all();
        $users = User::all();

        return view('welcome', compact('admins', 'users'));
    }
}
```

Agar fungsi index yang sudah di buat pada ListController di eksekusi ketika menjalankan server web maka ubahlah sintak yang terdapat pada **routes/web.php seperti berikut**

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\ListController;

Route::get('/', [ListController::class, 'index']);
```

Langkah terakhir pada View kita lakukan perubahan pada file **welcome.blade.php**

```

<!DOCTYPE html>
<html lang="zxx" class="no-js">

<head>
    <!-- Mobile Specific Meta -->
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <!-- Favicon-->
        <link rel="shortcut icon" href="{{ asset('assets/templates/user/img/fav.png') }}">
    <!-- Author Meta -->
    <meta name="author" content="CodePixel">
    <!-- Meta Description -->
    <meta name="description" content="">
    <!-- Meta Keyword -->
    <meta name="keywords" content="">
    <!-- meta character set -->
    <meta charset="UTF-8">
    <!-- Site Title -->
    <title>Merch Store</title>

    <!--
        CSS
        ===== -->
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/linearicons.css') }}">
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/owl.carousel.css') }}">
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/themify-icons.css') }}">
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/font-awesome.min.css') }}">
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/nice-select.css') }}">
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/nouislider.min.css') }}">
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/bootstrap.css') }}">
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/main.css') }}">
</head>

<body>

    <=====Login Box Area =====>
    <section class="login_box_area section_gap">
        <div class="container">
            <h3 class="text-center mb-5">Selamat Datang Di Halaman Utama</h3>
            <div class="row">
                <div class="col-lg-6">
                    <h5>Data Admin</h5>
                    <table>
                        <tr>
                            <th>Nama</th>
                            <th>Username</th>
                            <th>Email</th>
                        </tr>
                        @forelse ($admins as $item)
                            <tr>
                                <td>{{ $item->name }}</td>
                                <td>{{ $item->username }}</td>
                                <td>{{ $item->email }}</td>
                            </tr>
                        @empty
                            <tr>

```

```

                <td class="text-center">Data Kosong</td>
            </tr>
        @endforelse
    </table>
</div>
<div class="col-lg-6">
    <h5>Data User</h5>
    <table>
        <tr>
            <th>Name</th>
            <th>Email</th>
            <th>Points</th>
        </tr>
        @forelse ($users as $item)
        <tr>
            <td>{{ $item->name }}</td>
            <td>{{ $item->email }}</td>
            <td>{{ $item->point }}</td>
        </tr>
        @empty
        <tr>
            <td class="text-center">Data Kosong</td>
        </tr>
    @endforelse
</table>
</div>
</div>
</div>
</div>
</section>
<!--=====End Login Box Area =====-->

<script src="{{ asset('assets/templates/user/js/vendor/jquery-2.2.4.min.js') }}></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js"
integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCfwBdFNTxtclqqenISfwAzpKaMNFNmj4"
crossorigin="anonymous"></script>
<script src="{{ asset('assets/templates/user/js/vendor/bootstrap.min.js') }}></script>
<script src="{{ asset('assets/templates/user/js/jquery.ajaxchimp.min.js') }}></script>
<script src="{{ asset('assets/templates/user/js/jquery.nice-select.min.js') }}></script>
<script src="{{ asset('assets/templates/user/js/jquery.sticky.js') }}></script>
<script src="{{ asset('assets/templates/user/js/nouislider.min.js') }}></script>
<script src="{{ asset('assets/templates/user/js/jquery.magnific-popup.min.js') }}></script>
<script src="{{ asset('assets/templates/user/js/owl.carousel.min.js') }}></script>
<!--gmaps Js-->
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCjCGmQ0Uq4exrzdcL6rvxywDD0vFAu6eE"></script>
<script src="{{ asset('assets/templates/user/js/gmaps.min.js') }}></script>
<script src="{{ asset('assets/templates/user/js/main.js') }}></script>
</body>
</html>
```

Buatlah perubahan pada **ListController** dan **welcome.blade.php** untuk menampilkan data pada tabel **Distributor**

Halaman Login

Selanjtnya kita membuat halaman login menggunakan data pada tabel **admins** dan **users**. Namun sebelum itu, Anda terlebih dahulu perlu membuat sebuah **file blade** yang nantinya akan ditampilkan sebagai halaman website project Anda. Dalam membuat tampilan tersebut, Anda akan menggunakan blade templating engine untuk membantu Anda agar tidak perlu menuliskan kode yang sama secara berulang. Untuk menerapkan hal tersebut, Anda perlu membuat sebuah template yang nantinya dapat digunakan untuk banyak tampilan blade.

Pada **views** buatlah dua **folder baru** dengan nama **pages** dan **layout**.

- **Folder pages** untuk menyimpan halaman blade
- **folder layouts** yang digunakan untuk menyimpan halaman template.

Didalam masing-masing folder **layouts** dan **pages**, buatlah 2 folder baru lagi dengan nama **admin** dan **user**, karena dalam project laravel 11 ini hanya akan ada 2 tipe pengguna.

Selanjutnya di folder **layouts/user**, tambahkan 5 file (menambahkan file seperti biasa tanpa perintah) dengan nama

1. **footer.blade.php,**
2. **main.blade.php,**
3. **navbar.blade.php,**
4. **script.blade.php,**
5. **style.blade.php.**

Dari kelima file blade tersebut, tulislah kode berikut ini dan jangan sampai ada yang salah:

footer.blade.php

```
● ● ●
1 <footer class="footer-area section_gap">
2     <div class="container">
3         <div class="footer-bottom d-flex justify-content-center align-items-center flex-wrap">
4             <p class="footer-text m-0">
5                 ←!— Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. —>
6                 Copyright &copy;
7                 <script>
8                     document.write(new Date().getFullYear());
9                 </script> All rights reserved | This template is made with <i class="fa fa-heart-o"
10                     aria-hidden="true"></i> by <a href="https://colorlib.com" target="_blank">Colorlib</a>
11                 ←!— Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. —>
12             </p>
13         </div>
14     </div>
15 </footer>
```

main.blade.php

```
● ● ●
1 <!DOCTYPE html>
2 <html lang="zxx" class="no-js">
3
4     <head>
5         ←!— Mobile Specific Meta —>
6         <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7         ←!— Favicon—>
8         <link rel="shortcut icon" href="{{ asset('assets/templates/user/img/fav.png') }}>
9         ←!— Author Meta —>
10        <meta name="author" content="CodePixel">
11        ←!— Meta Description —>
12        <meta name="description" content="">
13        ←!— Meta Keyword —>
14        <meta name="keywords" content="">
15        ←!— meta character set —>
16        <meta charset="UTF-8">
17        ←!— Site Title —>
18        <title>Merch Store</title>
19
20        @include('layouts.user.style')
21    </head>
22    <body>
23        @include('sweetalert::alert')
24
25        @include('layouts.user.navbar')
26
27        @yield('content')
28
29        @include('layouts.user.footer')
30
31        @include('layouts.user.script')
32    </body>
33 </html>
```

navbar.blade.php



```
1 <!-- Start Header Area -->
2 <header class="header_area sticky-header">
3   <div class="main_menu">
4     <nav class="navbar navbar-expand-lg navbar-light main_box">
5       <div class="container">
6         <!-- Brand and toggle get grouped for better mobile display -->
7         <a class="navbar-brand logo_h" href="index.html"></a>
8         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
9           <span class="icon-bar"></span>
10          <span class="icon-bar"></span>
11          <span class="icon-bar"></span>
12        </button>
13        <!-- Collect the nav links, forms, and other content for toggling -->
14        <div class="collapse navbar-collapse offset" id="navbarSupportedContent">
15          <ul class="nav navbar-nav menu_nav ml-auto">
16            <li class="nav-item active"><a class="nav-link" href="index.html">Home</a></li>
17            </ul>
18            <ul class="nav navbar-nav navbar-right">
19              <li class="nav-item"><a class="nav-link" href="{{ route('user.logout') }}>Logout</a></li>
20            </ul>
21          </div>
22        </div>
23      </div>
24    </nav>
25  </div>
26 </header>
27 <!-- End Header Area -->
```

script.blade.php

```
<script src="{{ asset('assets/templates/user/js/vendor/jquery-2.2.4.min.js') }}"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHpOf/4+1nzFPr53nxSS+GLCkfwdFNTxtlqqenISfwAzpKaMNFNmj4"
crossorigin="anonymous"></script>
<script src="{{ asset('assets/templates/user/js/vendor/bootstrap.min.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/jquery.ajaxchimp.min.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/jquery.nice-select.min.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/jquery.sticky.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/nouislider.min.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/jquery.magnific-popup.min.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/owl.carousel.min.js') }}"></script>
<!--gMaps Js-->
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCjCGmQ0Uq4exrzdcL6rvxywDDOvfAu6eE"></script>
<script src="{{ asset('assets/templates/user/js/gmaps.min.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/main.js') }}"></script>
```

style.blade.php

```
● ● ●
1 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/linearicons.css') }}">
2 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/owl.carousel.css') }}">
3 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/themify-icons.css') }}">
4 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/font-awesome.min.css') }}">
5 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/nice-select.css') }}">
6 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/nouislider.min.css') }}">
7 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/bootstrap.css') }}">
8 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/main.css') }}">
9 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/ion.rangeSlider.css') }}" />
10 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/ion.rangeSlider.skinFlat.css') }}" />
11 <link rel="stylesheet" href="{{ asset('assets/templates/user/css/magnific-popup.css') }}">
```

Setelah menuliskan kode untuk masing-masing file diatas, kemudian dalam folder **layouts/admin**, tambahkan 5 file lagi dengan nama **footer.blade.php**, **main.blade.php**, **navbar.blade.php**, **script.blade.php**, dan **sidebar.blade.php**. Dari kelima file blade tersebut, tulislah kode berikut ini dan jangan sampai ada yang salah:

footer.blade.php

```
● ● ●
1 <footer class="main-footer">
2   <div class="footer-left">
3     Copyright ©;
4     <script>
5       var CurrentYear = new Date().getFullYear()
6       document.write(CurrentYear)
7     </script>
8     , designed & developed by <a href="#" class="text-muted">RPL</a>
9   </div>
10 </footer>
```

main.blade.php

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta content="width=device-width, initial-scale=1, maximum-scale=1, shrink-to-fit=no" name="viewport">
6
7     <!-- General CSS Files -->
8     <link rel="stylesheet" href="{{ asset('assets/templates/admin/modules/bootstrap/css/bootstrap.min.css') }}>
9     <link rel="stylesheet" href="{{ asset('assets/templates/admin/modules/fontawesome/css/all.min.css') }}>
10
11    <!-- CSS Libraries -->
12
13    <!-- Template CSS -->
14    <link rel="stylesheet" href="{{ asset('assets/templates/admin/css/style.css') }}>
15    <link rel="stylesheet" href="{{ asset('assets/templates/admin/css/components.css') }}>
16    <!-- Start GA -->
17    <script async src="https://www.googletagmanager.com/gtag/js?id=UA-94034622-3"></script>
18    <script>
19        window.dataLayer = window.dataLayer || [];
20        function gtag(){dataLayer.push(arguments);}
21        gtag('js', new Date());
22
23        gtag('config', 'UA-94034622-3');
24    </script>
25    <title>@yield('title')</title>
26 </head>
27 <body>
28     @include('sweetalert::alert')
29
30     <div id="app">
31         <div class="main-wrapper main-wrapper-1">
32             @include('layouts.admin.navbar')
33
34             @include('layouts.admin.sidebar')
35
36             @yield('content')
37
38             @include('layouts.admin.footer')
39         </div>
40     </div>
41
42     @include('layouts.admin.script')
43 </body>
44 </html>
```

navbar.blade.php

```
1 <div class="navbar-bg"></div>
2 <nav class="navbar navbar-expand-lg main-navbar">
3     <form class="form-inline mr-auto">
4         <ul class="navbar-nav mr-3">
5             <li><a href="#" data-toggle="sidebar" class="nav-link nav-link-lg"><i class="fas fa-bars"></i></a></li>
6         </ul>
7     </form>
8     <ul class="navbar-nav navbar-right">
9         <li>
10             <li class="dropdown"><a href="#" data-toggle="dropdown" class="nav-link dropdown-toggle nav-link-lg nav-link-user">
11                 
12                 <div class="d-sm-none d-lg-inline-block">Hi, Admin</div></a>
13                 <div class="dropdown-menu dropdown-menu-right">
14                     <a href="{{ route('admin.logout') }}" class="dropdown-item has-icon text-danger">
15                         <i class="fas fa-sign-out-alt"></i> Logout
16                     </a>
17                 </div>
18             </li>
19         </ul>
20     </nav>
```

script.blade.php

```
<!-- General JS Scripts -->
<script src="{{ asset('assets/templates/admin/modules/jquery.min.js') }}"></script>
<script src="{{ asset('assets/templates/admin/modules/popper.js') }}"></script>
<script src="{{ asset('assets/templates/admin/modules/tooltip.js') }}"></script>
<script      src="{{      asset('assets/templates/admin/modules/bootstrap/js/bootstrap.min.js') }}></script>
<script      src="{{      asset('assets/templates/admin/modules/nicescroll/jquery.nicescroll.min.js') }}></script>
<script src="{{ asset('assets/templates/admin/modules/moment.min.js') }}"></script>
<script src="{{ asset('assets/templates/admin/js/stisla.js') }}"></script>

<!-- JS Libraies -->

<!-- Page Specific JS File -->

<!-- Template JS File -->
<script src="{{ asset('assets/templates/admin/js/scripts.js') }}"></script>
<script src="{{ asset('assets/templates/admin/js/custom.js') }}"></script>
```

sidebar.blade.php



```
1 <div class="main-sidebar sidebar-style-2">
2   <aside id="sidebar-wrapper">
3     <div class="sidebar-brand">
4       <a href="#">Teknik Informatika | RPL</a>
5     </div>
6     <div class="sidebar-brand sidebar-brand-sm">
7       <a href="#">RPL</a>
8     </div>
9     <ul class="sidebar-menu">
10       <li class="menu-header">Menu</li>
11       <li class="{{ Route::is('admin.dashboard') ? 'active' : '' }}><a class="nav-link"
12           href="{{ route('admin.dashboard') }}><i class="fas fa-home"></i> <span>Dashboard</span></a>
13       </li>
14     </ul>
15   </aside>
16 </div>
```

Setelah berhasil menambahkan kode diatas, selanjutnya Anda akan membuat sebuah file blade yang digunakan sebagai halaman utama dari tiap role (admin & user). Untuk membuat kedua file tersebut, tambahkan file baru dengan nama yang sama yakni **index.blade.php** kedalam folder **pages/admin** dan **pages/user**, lalu tulislah kode dibawah ini:

index.blade.php Admin

```
@extends('layouts.admin.main')
@section('title', 'Admin Dashboard')
@section('content')
    <div class="main-content">
        <section class="section">
            <div class="section-header">
                <h1>Dashboard</h1>
                <div class="section-header-breadcrumb">
                    <div class="breadcrumb-item active"><a href="#">Dashboard</a></div>
                </div>
            </div>

            <div class="row">
                <div class="col-lg-3 col-md-6 col-sm-6 col-12">
                    <div class="card card-statistic-1">
                        <div class="card-icon bg-primary">
                            <i class="far fa-user"></i>
                        </div>
                        <div class="card-wrap">
                            <div class="card-header">
                                <h4>Total Pengguna</h4>
                            </div>
                            <div class="card-body">
                                200
                            </div>
                        </div>
                    </div>
                </div>
                <div class="col-lg-3 col-md-6 col-sm-6 col-12">
                    <div class="card card-statistic-1">
                        <div class="card-icon bg-danger">
                            <i class="far fa-newspaper"></i>
                        </div>
                        <div class="card-wrap">
                            <div class="card-header">
                                <h4>Total Produk</h4>
                            </div>
                            <div class="card-body">
                                200
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </section>
    </div>
@endsection
```

index.blade.php User

```
@extends('layouts.user.main')
@section('content')
<!-- start banner Area -->
<section class="banner-area">
    <div class="container">
        <div class="row fullscreen align-items-center justify-content-start">
            <div class="col-lg-12">
                <div class="">
                    <!-- single-slide -->
                    <div class="row">
                        <div class="col-lg-5 col-md-6">
                            <div class="banner-content">
                                <h1>Nike New <br>Collection!</h1>
                                <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation.</p>
                            </div>
                        </div>
                        <div class="col-lg-7">
                            <div class="banner-img">
                                
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>
<!-- End banner Area -->

<!-- start product Area -->
<section class="section_gap">
    <!-- single product slide -->>
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-lg-6 text-center">
                <div class="section-title">
                    <h1>Latest Products</h1>
                    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
                </div>
            </div>
        </div>
    </div>
    <div class="row">
        <!-- single product -->
        <div class="col-lg-3 col-md-6">
            <div class="single-product">
                
                <div class="product-details">
                    <h6>addidas New Hammer sole</h6>
                </div>
            </div>
        </div>
    </div>
</section>
```

Selanjutnya, Anda akan mengubah kode dari file **welcome.blade.php** menjadi tampilan login yang akan digunakan oleh pengguna (admin & user) untuk melakukan proses login untuk masuk ke dalam halaman sesuai dengan role nya masing-masing. Dalam file **welcome.blade.php**, tulislah kode dibawah ini:

welcome.blade.php

```
<!DOCTYPE html>
<html lang="zxx" class="no-js">

<head>
    <!-- Mobile Specific Meta -->
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Favicon-->
    <link rel="shortcut icon" href="{{ asset('assets/templates/user/img/fav.png') }}">
    <!-- Author Meta -->
    <meta name="author" content="CodePixel">
    <!-- Meta Description -->
    <meta name="description" content="">
    <!-- Meta Keyword -->
    <meta name="keywords" content="">
    <!-- meta character set -->
    <meta charset="UTF-8">
    <!-- Site Title -->
    <title>Merch Store</title>

    <!--
        CSS
        =====
    -->
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/linearicons.css') }}>
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/owl.carousel.css') }}>
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/themify-icons.css') }}>
    <link rel="stylesheet" href="{{ asset('assets/templates/user/css/font-awesome.min.css') }}>
```

```

<link rel="stylesheet" href="{{ asset('assets/templates/user/css/nice-select.css') }}">
<link rel="stylesheet" href="{{ asset('assets/templates/user/css/nouislider.min.css') }}">
<link rel="stylesheet" href="{{ asset('assets/templates/user/css/bootstrap.css') }}">
<link rel="stylesheet" href="{{ asset('assets/templates/user/css/main.css') }}">
</head>

<body>
    @include('sweetalert::alert')

    <!--=====Login Box Area =====-->
    <section class="login_box_area section_gap">
        <div class="container">
            <div class="row">
                <div class="col-lg-6">
                    <div class="login_box_img">
                        
                        <div class="hover">
                            <h4>New to our website?</h4>
                            <p>There are advances being made in science and technology everyday, and a good example of this is the</p>
                            <a class="primary-btn" href="registration.html">Create an Account</a>
                        </div>
                    </div>
                    <div class="col-lg-6">
                        <div class="login_form_inner">
                            <h3>Log in to enter</h3>
                            <form class="row login_form" action="/post-login" method="POST" id="contactForm" novalidate="novalidate">
                                @csrf
                                <div class="col-md-12 form-group">
                                    <input type="email" class="form-control" id="email" name="email" placeholder="Email" onfocus="this.placeholder = ''" onblur="this.placeholder = 'Email'">
                                </div>
                                <div class="col-md-12 form-group">
                                    <input type="password" class="form-control" id="password" name="password" placeholder="Password" onfocus="this.placeholder = ''" onblur="this.placeholder = 'Password'">
                                </div>
                                <div class="col-md-12 form-group">
                                    <button type="submit" value="submit" class="primary-btn">Log In</button>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>
    <!--=====End Login Box Area =====-->

    <script src="{{ asset('assets/templates/user/js/vendor/jquery-2.2.4.min.js') }}"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHpOf/4+1nzFPr53nxSS+GLCfwBdFNTxtclqqenISfwAzpKaMNFNmj4" crossorigin="anonymous"></script>
    <script src="{{ asset('assets/templates/user/js/vendor/bootstrap.min.js') }}"></script>
    <script src="{{ asset('assets/templates/user/js/jquery.ajaxchimp.min.js') }}"></script>
    <script src="{{ asset('assets/templates/user/js/jquery.nice-select.min.js') }}"></script>

```

```

<script src="{{ asset('assets/templates/user/js/jquery.sticky.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/nouislider.min.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/jquery.magnific-popup.min.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/owl.carousel.min.js') }}"></script>
<!--gMaps Js-->
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCjCGmQ0Uq4exrzdcL6rvxywDDOvfAu6
eE"></script>
<script src="{{ asset('assets/templates/user/js/gmaps.min.js') }}"></script>
<script src="{{ asset('assets/templates/user/js/main.js') }}"></script>
</body>
</html>

```

Langkah terakhir adalah mengatur route dalam file **web.php**.

File **web.php** tersebut merupakan sebuah file dimana anda dapat mendefinisikan routes atau jalur aplikasi laravel anda. Routes tersebut biasa digunakan untuk menentukan URL mana yang akan menjalankan kode tertentu dalam aplikasi laravel anda, atau dengan kata lain file tersebut mengatur bagaimana aplikasi merespons permintaan (request) dari pengguna seperti membuka halaman web, memproses form, menampilkan data, dan lain sebagainya. Agar logika login yang sudah Anda buat dapat dijalankan, tulislah kode dibawah ini dan jangan sampai terjadi kesalahan:



```

1 use App\Http\Controllers\Auth\AuthController;
2 use Illuminate\Support\Facades\Route;
3
4 // Guest Route
5 Route::group(['middleware' => 'guest'], function() {
6     Route::get('/', function () {
7         return view('welcome');
8     });
9
10    Route::post('/post-login', [AuthController::class, 'login']);
11 })->middleware('guest');
12
13 // Admin Route
14 Route::group(['middleware' => 'admin'], function() {
15     Route::get('/admin', function() {
16         return view('pages.admin.index');
17     })->name('admin.dashboard');
18
19     Route::get('/admin-logout', [AuthController::class,
20     'admin_logout'])->name('admin.logout');
21 })->middleware('admin');
22
23 // User Route
24 Route::group(['middleware' => 'web'], function() {
25     Route::get('/user', function() {
26         return view('pages.user.index');
27     })->name('user.dashboard');
28
29     Route::get('/user-logout', [AuthController::class, 'user_logout'])
30 })->middleware('web');

```

Setelah itu, Jalankan project anda dengan perintah PHP ARTISAN SERVE

Jika sudah, coba lakukan login dengan email dan password yang dapat Anda lihat pada **DatabaseSeeder.php**. Jika yang melakukan adalah admin maka akan diarahkan kedalam halaman admin, sebaliknya jika yang melakukan login adalah user maka akan diarahkan kedalam halaman user.