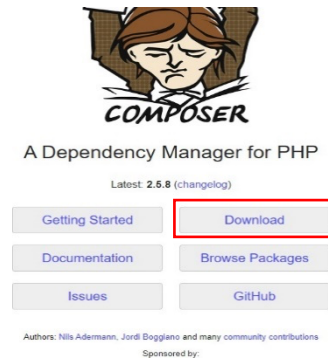


1. Kunjungi halaman berikut <https://getcomposer.org>, lalu halaman seperti padagambar dibawah ini akan muncul. Kemudian klik bagian “Download”.



2. 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.

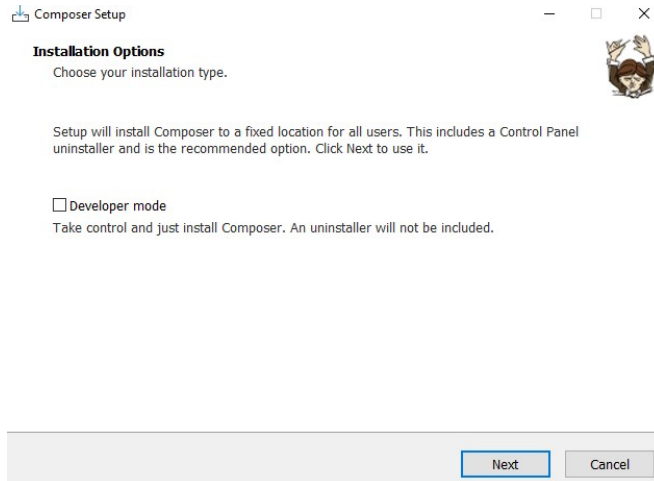
Command-line installation

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](#).

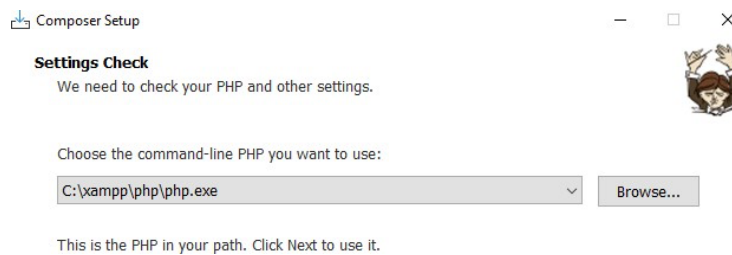
```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === 'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f123
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

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:

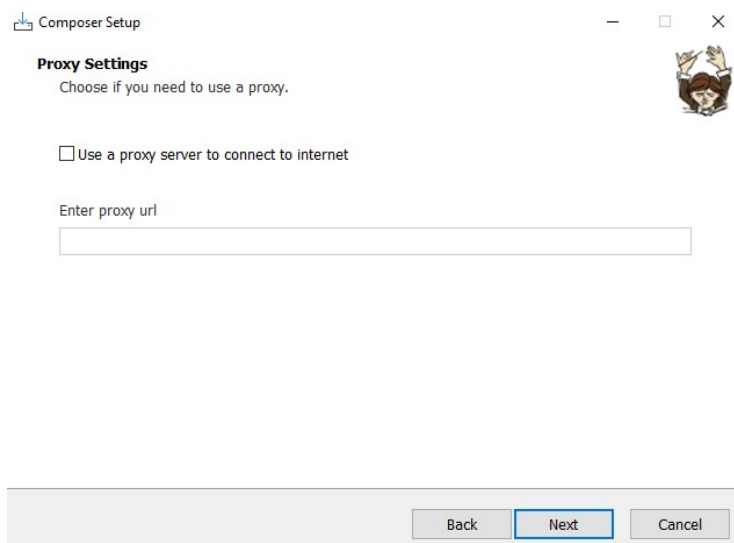
3. Setelah itu file composer akan di download. Setelah file selesai di download, double click pada file tersebut.
4. Jika muncul pesan berikut, klik bagian “Install for all users (recommended)”.
5. Maka akan muncul gambar seperti dibawah ini, kemudian klik bagian “Next”.



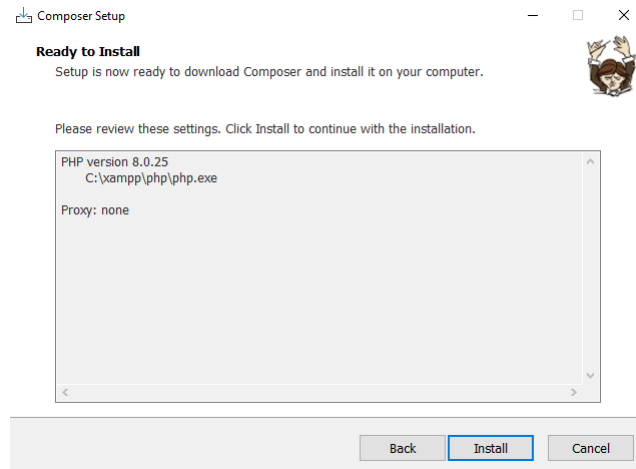
- Setelah itu akan muncul tampilan berikut, jika pada perangkat sudah terdapat bahasa pemrograman PHP, maka akan otomatis terisi dengan direktori folder PHP yang ada di perangkat. Selanjutnya klik bagian “Next”.



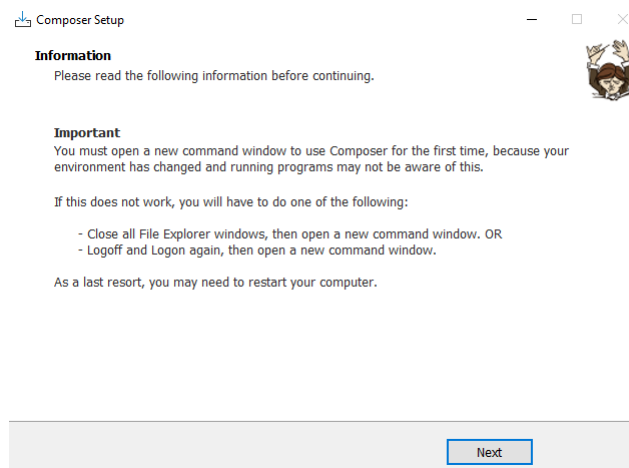
- Setelah itu akan muncul tampilan berikut, klik bagian “Next”.



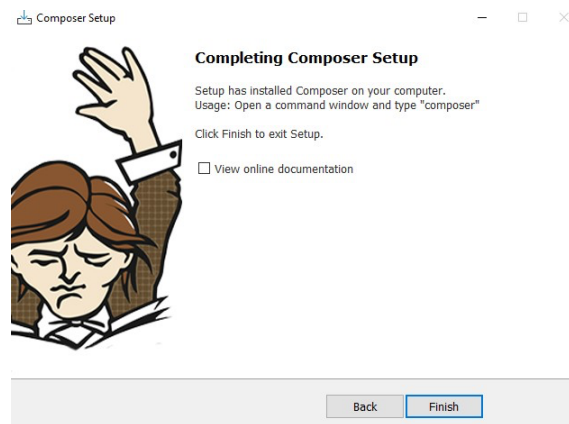
8. Setelah itu akan muncul tampilan berikut, klik bagian “Install” maka proses instalasi composer akan dilakukan. Tunggu sampai proses instalasi selesai.



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



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

Composer
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
  -q, --quiet               Do not output any message
  -V, --version              Display this application version
  --ansi|--no-ansi          Force (or disable) 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|vv|vvv, --verbose       Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
                              3 for debug
```

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:

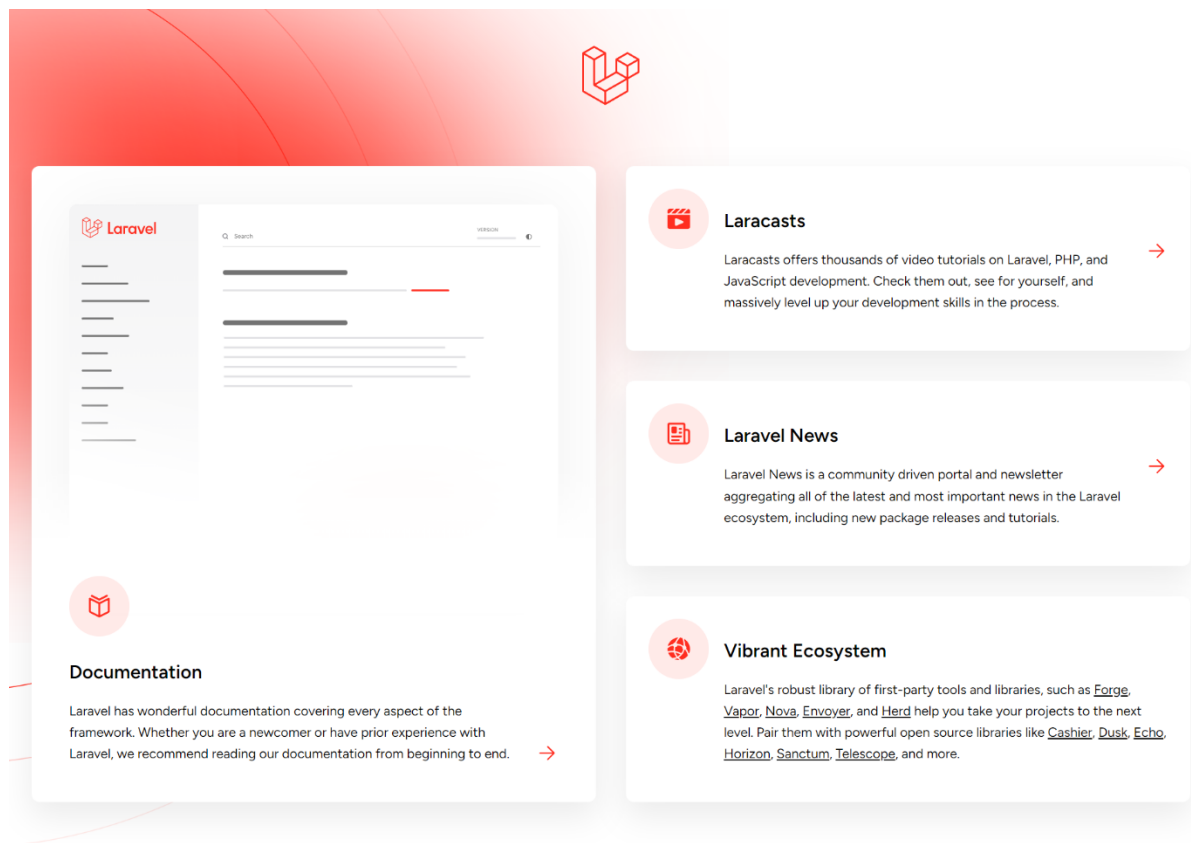
1. Folder app: Folder ini berisi inti dari aplikasi Anda, termasuk model-model, controller-controller, dan kode bisnis utama. Model-model digunakan untuk berinteraksi dengan database menggunakan Eloquent ORM. Controller-controller digunakan untuk menangani permintaan HTTP dan menghubungkan model dengan 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 aset-aset publik seperti gambar, file CSS, dan JavaScript yang dapat diakses oleh publik.
6. Folder resources: Ini berisi tampilan (views), file aset 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:



Laravel v11.22.0 (PHP v8.2.12)

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

https://drive.google.com/file/d/1SsscweS2nkbDLpNxRRa5HUwxl2kVbkQC/view?usp=s_haring

Setelah Anda mendownload file aset 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 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\sweetalertt] DONE
Copying file [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\vendor\realrashid\sweet-alert\src\config\sweetalertt.php] to [D:\Project\Dir\Modul Laravel 11\ecommerce-laravel\config\sweetalertt.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\sweetalertt] 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 handle 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:

```

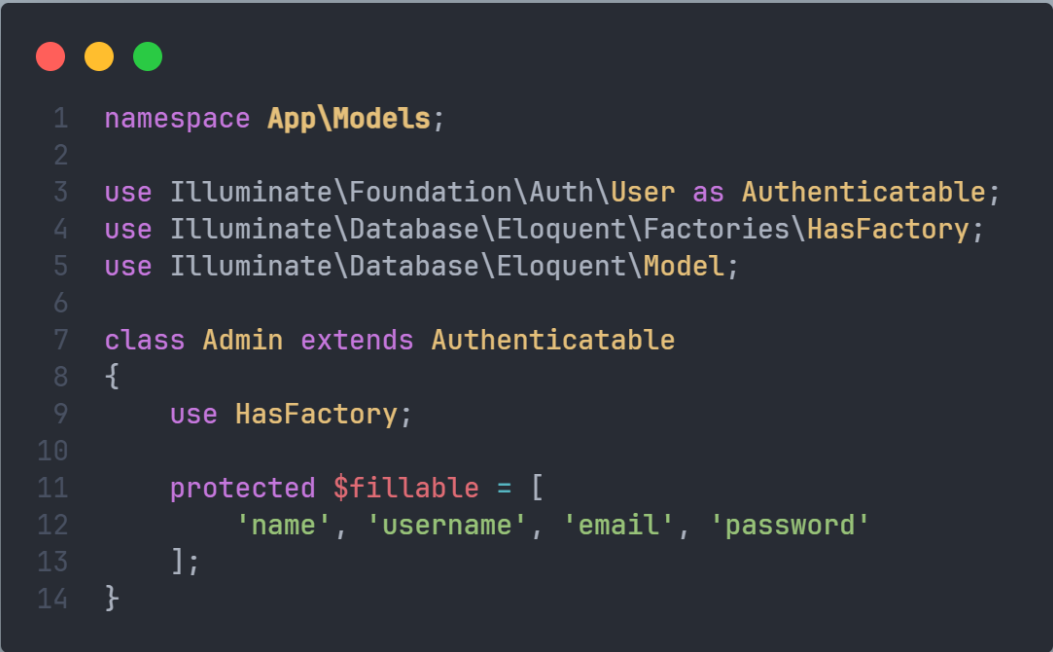
1 use Illuminate\Foundation\Application;
2 use Illuminate\Foundation\Configuration\Exceptions;
3 use Illuminate\Foundation\Configuration\Middleware;
4
5 return Application::configure(basePath: dirname(__DIR__))
6     →withRouting(
7         web: __DIR__.'/../routes/web.php',
8         commands: __DIR__.'/../routes/console.php',
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');
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:

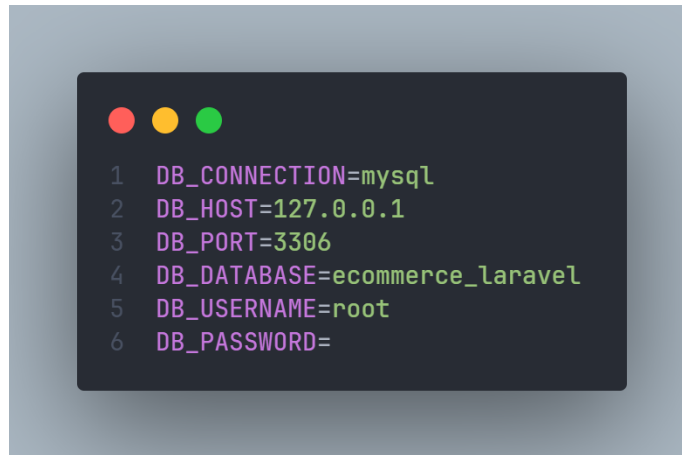
```

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

```

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 kodenya menjadi seperti berikut ini:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It displays six lines of configuration code for a database connection.

```
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.

Berdasarkan langkah untuk membuat tabel Admin, buatlah tabel **distributor** dengan ketentuan fieldnya sebagai berikut:

- id
- nama_distributor
- lokasi
- kontak
- email

Setelah itu, masuk kedalam file DatabaseSeeder.php yang berlokasi dalam **database/seeders/DatabaseSeeder.php** lalu tuliskan 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 }
```

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.

Lakukan hal yang sama untuk mengisi data dummy di tabel Distributor