



LARAVEL

PERTEMUAN 6

BY FAERUL SALAMUN

REVIEW 5

- Mempelajari PHP dan Instalasi Laravel

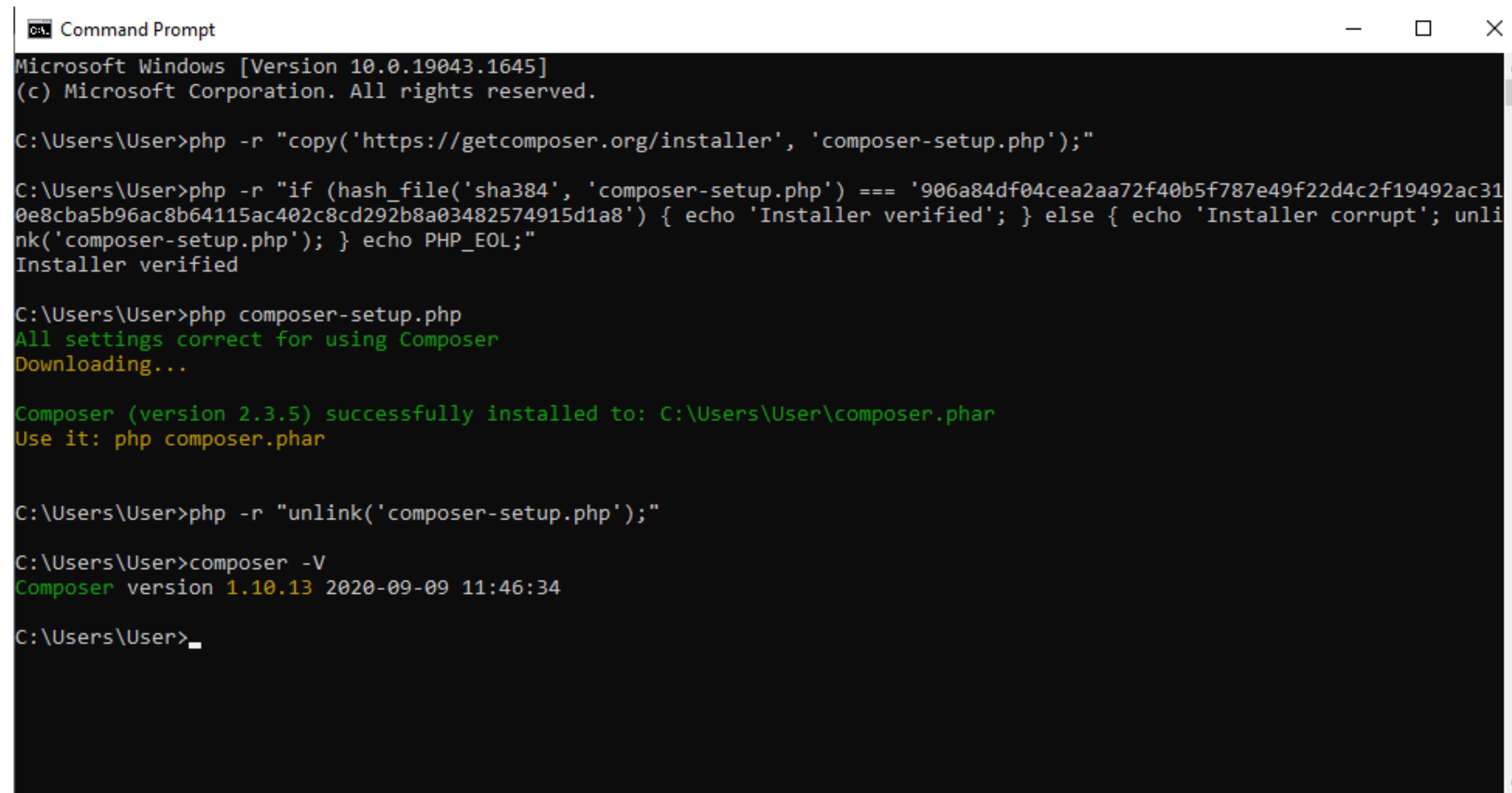
NEXT

- ▶ Laravel

LARAVEL

HOW TO INSTALL LARAVEL ?

- ▶ Pastikan sudah mendownload composer <https://getcomposer.org/download>



The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The window displays the following terminal session:

```
Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"

C:\Users\User>php -r "if (hash_file('sha384', 'composer-setup.php') === '906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8cba5b96ac8b64115ac402c8cd292b8a03482574915d1a8') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
Installer verified

C:\Users\User>php composer-setup.php
All settings correct for using Composer
Downloading...

Composer (version 2.3.5) successfully installed to: C:\Users\User\composer.phar
Use it: php composer.phar

C:\Users\User>php -r "unlink('composer-setup.php');"

C:\Users\User>composer -V
Composer version 1.10.13 2020-09-09 11:46:34

C:\Users\User>
```

HOW TO INSTALL LARAVEL ?

- composer global require laravel/installer

```
faerulsalamun@Faerul-Salamun-MacBook-Pro Downloads % composer global require "laravel/installer"
Changed current directory to /Users/faerulsalamun/.composer
Info from https://repo.packagist.org: #StandWithUkraine
Using version ^4.2 for laravel/installer
./composer.json has been created
Running composer update laravel/installer
Loading composer repositories with package information
Updating dependencies
Lock file operations: 10 installs, 0 updates, 0 removals
  - Locking laravel/installer (v4.2.10)
  - Locking psr/container (2.0.2)
  - Locking symfony/console (v6.0.7)
  - Locking symfony/polyfill-ctype (v1.25.0)
  - Locking symfony/polyfill-intl-grapheme (v1.25.0)
  - Locking symfony/polyfill-intl-normalizer (v1.25.0)
  - Locking symfony/polyfill-mbstring (v1.25.0)
  - Locking symfony/process (v6.0.7)
  - Locking symfony/service-contracts (v3.0.1)
  - Locking symfony/string (v6.0.3)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 10 installs, 0 updates, 0 removals
  - Downloading symfony/process (v6.0.7)
  - Downloading symfony/polyfill-mbstring (v1.25.0)
  - Downloading symfony/polyfill-intl-normalizer (v1.25.0)
  - Downloading symfony/polyfill-intl-grapheme (v1.25.0)
  - Downloading symfony/polyfill-ctype (v1.25.0)
  - Downloading symfony/string (v6.0.3)
  - Downloading psr/container (2.0.2)
  - Downloading symfony/service-contracts (v3.0.1)
  - Downloading symfony/console (v6.0.7)
  - Downloading laravel/installer (v4.2.10)
  - Installing symfony/process (v6.0.7): Extracting archive
  - Installing symfony/polyfill-mbstring (v1.25.0): Extracting archive
  - Installing symfony/polyfill-intl-normalizer (v1.25.0): Extracting archive
  - Installing symfony/polyfill-intl-grapheme (v1.25.0): Extracting archive
  - Installing symfony/polyfill-ctype (v1.25.0): Extracting archive
  - Installing symfony/string (v6.0.3): Extracting archive
  - Installing psr/container (2.0.2): Extracting archive
  - Installing symfony/service-contracts (v3.0.1): Extracting archive
  - Installing symfony/console (v6.0.7): Extracting archive
  - Installing laravel/installer (v4.2.10): Extracting archive
4 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating autoload files
8 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
```

HOW TO INSTALL LARAVEL ?

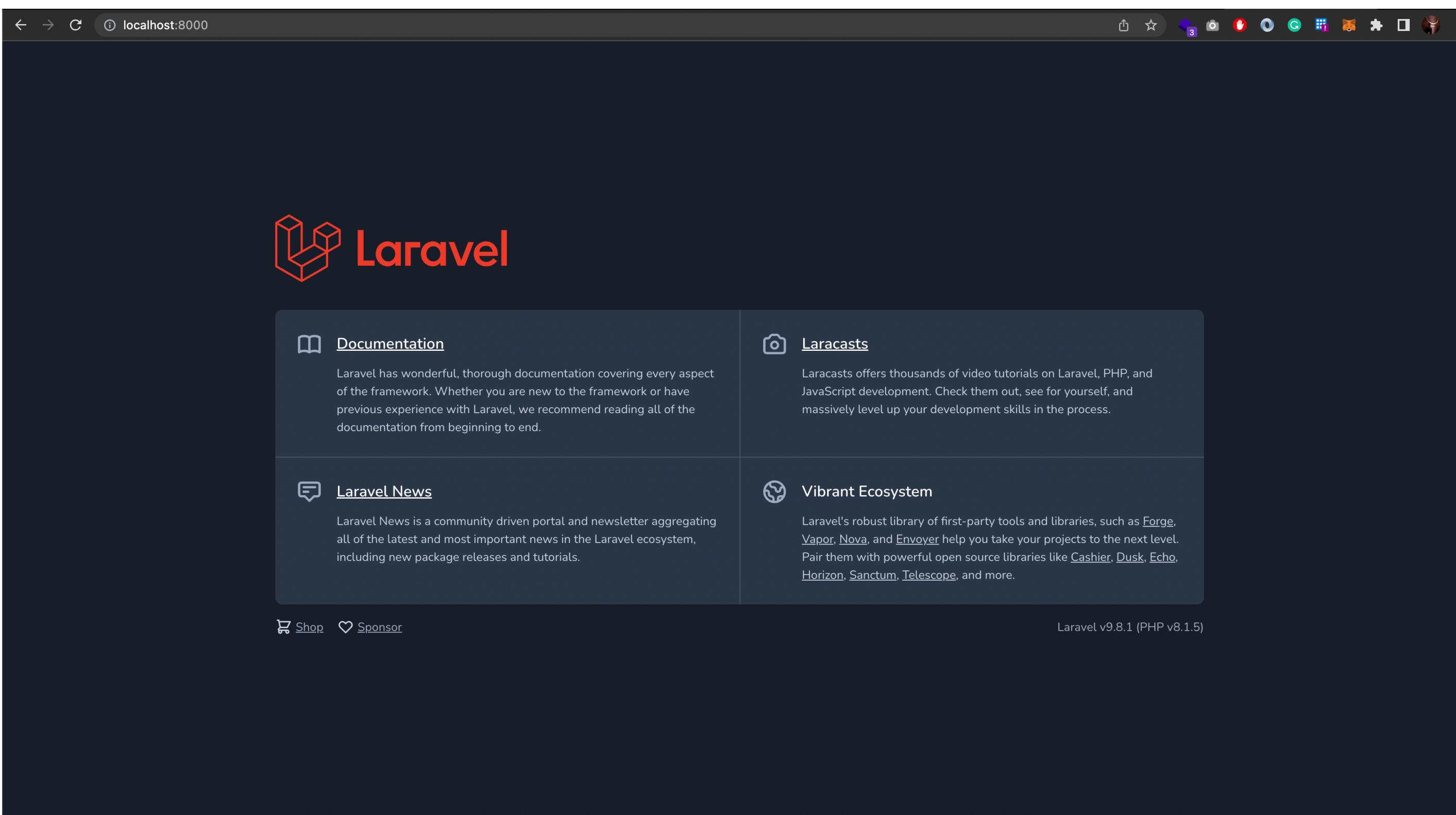
- composer create-project --prefer-dist laravel/laravel gallery

```
[faerulsalamun@Faerul-Salamun-MacBook-Pro Downloads % composer create-project --prefer-dist laravel/laravel gallery
Creating a "laravel/laravel" project at "./gallery"
Installing laravel/laravel (v9.1.5)
  - Downloading laravel/laravel (v9.1.5)
    - Installing laravel/laravel (v9.1.5): Extracting archive
Created project in /Users/faerulsalamun/Downloads/gallery
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 168 installs, 0 updates, 0 removals
  - Locking brick/math (0.9.3)
  - Locking dflydev/dot-access-data (v3.0.1)
  - Locking doctrine/inflector (2.0.4)
  - Locking doctrine/instantiator (1.4.1)
  - Locking doctrine/lexer (1.2.3)
  - Locking dragonmantank/cron-expression (v3.3.1)
  - Locking egulias/email-validator (3.1.2)
  - Locking facade/ignition-contracts (1.0.2)
  - Locking fakerphp/faker (v1.19.0)
  - Locking filp/whoops (2.14.5)
  - Locking fruitcake/php-cors (v1.2.0)
  - Locking graham-campbell/result-type (v1.0.4)
  - Locking guzzlehttp/guzzle (7.4.2)
  - Locking guzzlehttp/promises (1.5.1)
  - Locking guzzlehttp/psr7 (2.2.1)
  - Locking hamcrest/hamcrest-php (v2.0.1)
  - Locking laravel/framework (v9.8.1)
  - Locking laravel/sail (v1.13.9)
  - Locking laravel/sanctum (v2.15.1)
  - Locking laravel/serializable-closure (v1.1.1)
  - Locking laravel/tinker (v2.7.2)
  - Locking league/commonmark (2.3.0)
  - Locking league/config (v1.1.1)
  - Locking league/flysystem (3.0.17)
  - Locking league/nime-type-detection (1.11.0)
  - Locking mockery/mockery (1.5.0)
  - Locking monolog/monolog (2.5.0)
  - Locking myclabs深深 (v1.11.0)
  - Locking nesbot/carbon (2.57.0)
  - Locking nette/schema (v1.2.2)
  - Locking nette/utils (v3.2.7)
  - Locking nikic/php-parser (v4.13.2)
  - Locking nunomaduro/collision (v6.2.0)
  - Locking phar-io/manifest (2.0.3)
  - Locking phar-io/version (3.2.1)
  - Locking phpdocumentor/reflection-common (2.2.0)
  - Locking phpdocumentor/reflection-docblock (5.3.0)
  - Locking phpdocumentor/type-resolver (1.6.1)
  - Locking phppoption/phppoption (1.8.1)
  - Locking phpspec/prophesy (v1.15.0)
  - Locking phpunit/php-code-coverage (9.2.15)
  - Locking phpunit/php-file-iterator (3.0.6)
  - Locking phpunit/php-invoker (3.1.1)
  - Locking phpunit/php-text-template (2.0.4)
  - Locking phpunit/php-timer (5.0.3)
  - Locking phpunit/phpunit (9.5.20)
  - Locking psr/container (2.0.2)
  - Locking psr/event-dispatcher (1.0.0)
  - Locking psr/http-client (1.0.1)
  - Locking psr/http-factory (1.0.1)
  - Locking psr/http-message (1.0.1)
  - Locking psr/log (3.0.0)
  - Locking psr/simple-cache (3.0.0)
```

RUN LARAVEL

- ▶ `php artisan serve`

```
[faerulsalamun@Faerul-Salamun-MacBook-Pro gallery % php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Sun Apr 17 23:45:16 2022] PHP 8.1.5 Development Server (http://127.0.0.1:8000) started
```



STRUCTURE FOLDER LARAVEL

```
> app
> bootstrap
> config
> database
> lang
> public
> resources
> routes
> storage
> tests
> vendor
⚙ .editorconfig
⚙ .env
☰ .env.example
❖ .gitattributes
❖ .gitignore
! .styleci.yml
☰ artisan
{} composer.json
{} composer.lock
{} package.json
RSS phpunit.xml
ⓘ README.md
JS webpack.mix.js
```

- Didalam direktori app terdapat beberapa folder seperti Controller, Service, Model,dll yang digunakan untuk logic sebuah aplikasi.
- bootstrap direktori berguna untuk menyimpan cache dari laravel seperti generate file, dll
- config direktori berguna untuk melakukan setup ataupun konfigurasi pada aplikasi seperti koneksi database, mail, session, dll
- database direktori berguna untuk menyimpan file migration ataupun seeder
- lang direktori berguna untuk menyimpan file bahasa
- public direktori berguna untuk menyimpan assets seperti image, js, css, dll
- resource direktori berguna untuk menyimpan tampilan (html) dengan menggunakan templating blade ataupun yang lainnya
- routes direktori berguna untuk melakukan routing pada sebuah website
- storage direktori berguna untuk menyimpan cache, logs, dll
- test direktori berguna untuk melakukan unit testing
- vendor direktori berguna untuk menyimpan dependecy dari composer

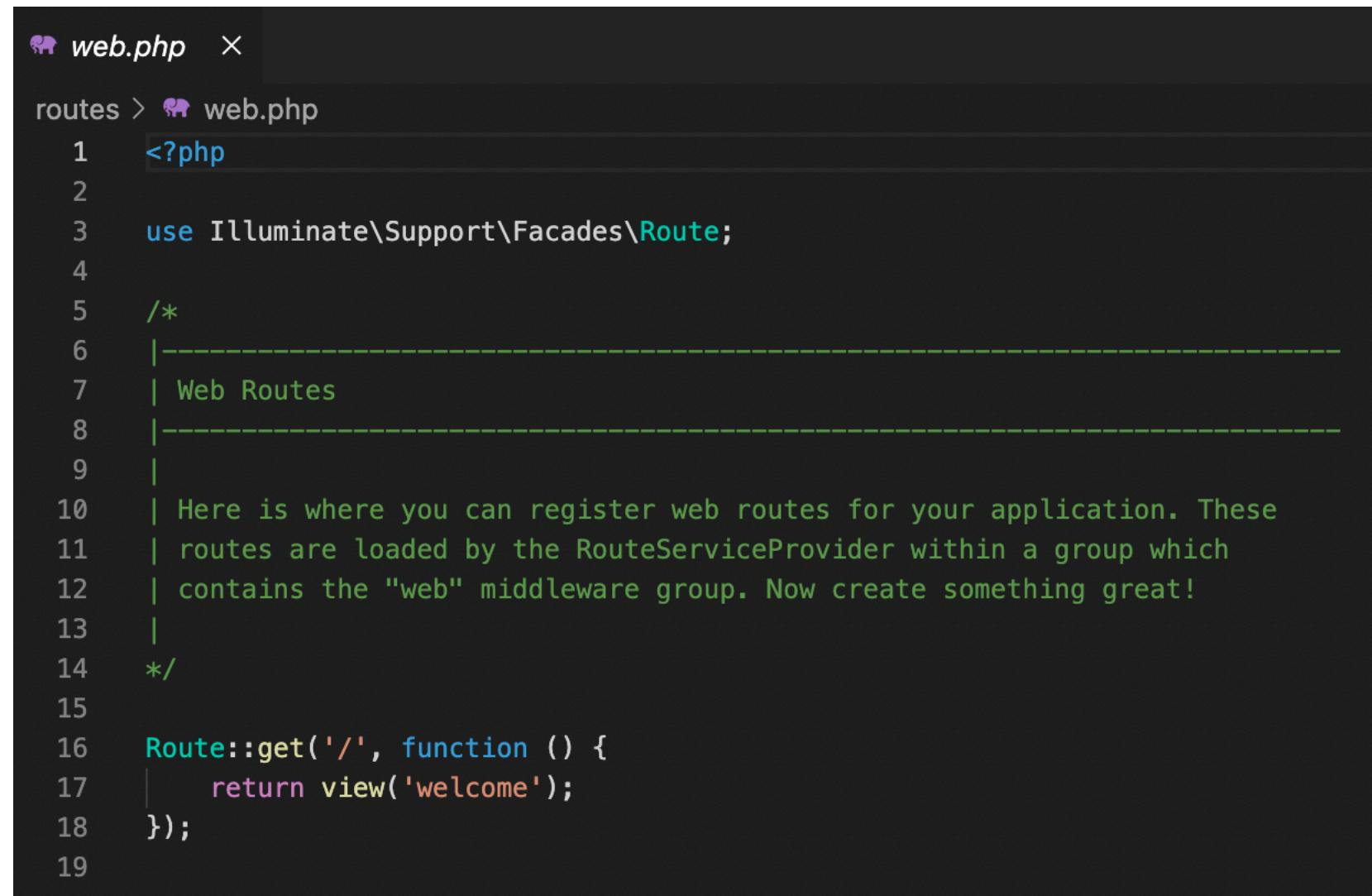
ROUTES

✓ routes

- 🐘 api.php
- 🐘 channels.php
- 🐘 console.php
- 🐘 web.php

Terdapat 2 file yang sering digunakan yaitu api.php dan web.php . File api.php digunakan untuk membuat web service sedangkan web.php digunakan pembuatan website.

ROUTES (WEB.PHP)

A screenshot of a code editor showing the contents of a file named 'web.php'. The file is located in a directory 'routes' and contains PHP code for defining routes. The code includes a block comment for 'Web Routes', a note about registering routes, and a single route definition for the root path '/' that returns the 'welcome' view.

```
web.php ×
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6  |--------------------------------------------------------------------------
7  | Web Routes
8  |--------------------------------------------------------------------------
9  |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
```

Secara default url laravel untuk web adalah localhost:8000 (jika menggunakan php artisan serve)

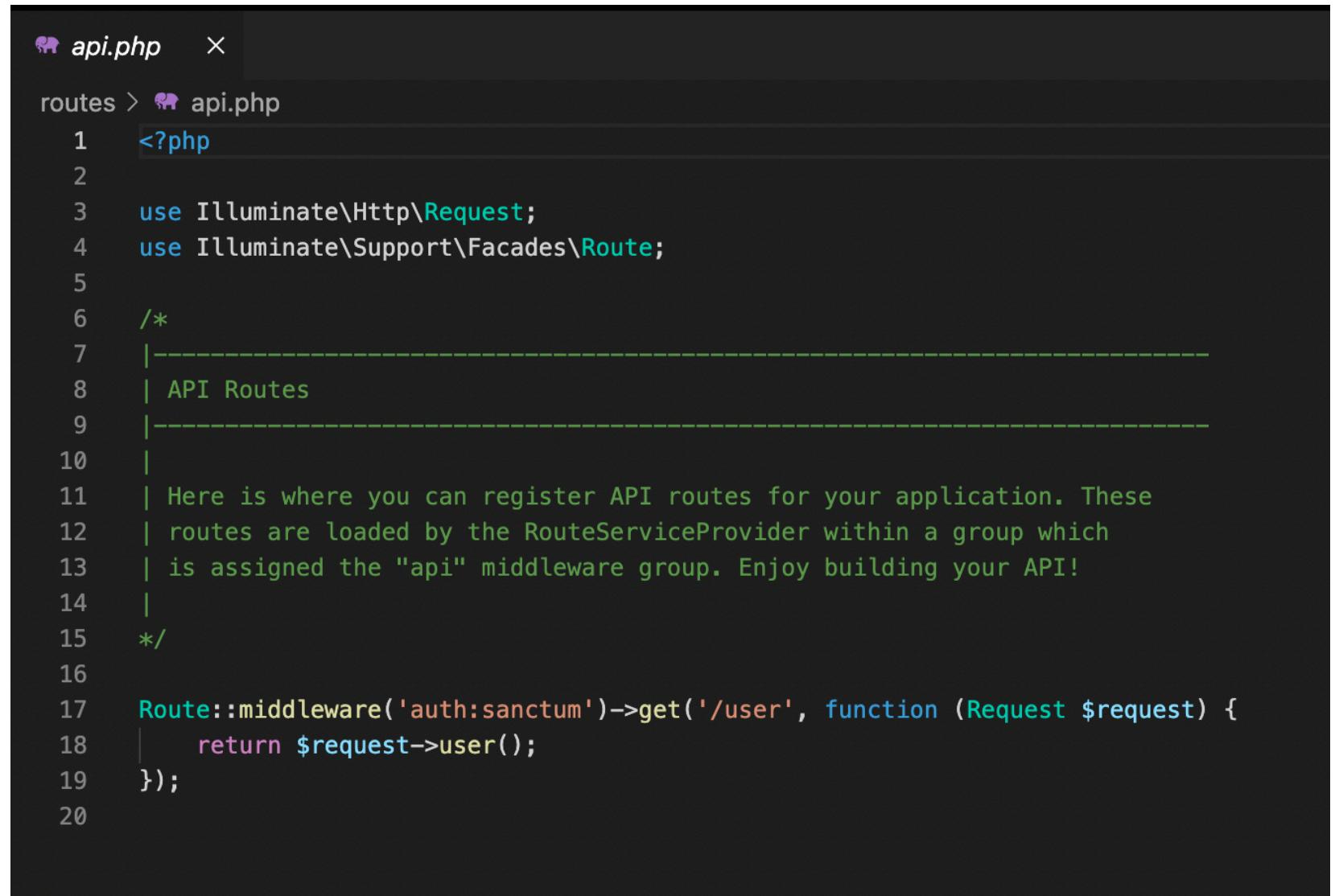
ROUTES (WEB.PHP)

welcome.blade.php ×

resources > views > welcome.blade.php

```
1  <!DOCTYPE html>
2  <html lang="{{ str_replace('_', '-', app()->getLocale()) }}>
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6
7      <title>Laravel</title>
8
9      <!-- Fonts -->
10     <link href="https://fonts.googleapis.com/css2?family=Nunito:wght@400;600;700&display=swap" rel="stylesheet">
11
12     <!-- Styles -->
13     <style>
14         /*! normalize.css v8.0.1 | MIT License | github.com/necolas/normalize.css */html{line-height:1.15;-webkit-text-size-adjust:100%}body{margin:0}a{background-
15     </style>
16
17     <style>
18         body {
19             font-family: 'Nunito', sans-serif;
20         }
21     </style>
22 </head>
23 <body class="antialiased">
24     <div class="relative flex items-top justify-center min-h-screen bg-gray-100 dark:bg-gray-900 sm:items-center py-4 sm:pt-0">
25         @if (Route::has('login'))
26             <div class="hidden fixed top-0 right-0 px-6 py-4 sm:block">
27                 @auth
28                     <a href="{{ url('/home') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Home</a>
29                 @else
30                     <a href="{{ route('login') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Log in</a>
31
32                     @if (Route::has('register'))
33                         <a href="{{ route('register') }}" class="ml-4 text-sm text-gray-700 dark:text-gray-500 underline">Register</a>
34                     @endif
35                 @endauth
36             </div>
37         @endif
38
39         <div class="max-w-6xl mx-auto sm:px-6 lg:px-8">
40             <div class="flex justify-center pt-8 sm:justify-start sm:pt-0">
41                 <svg viewBox="0 0 651 192" fill="none" xmlns="http://www.w3.org/2000/svg" class="h-16 w-auto text-gray-700 sm:h-20">
42                     <g clip-path="url(#clip0)" fill="#EF3B2D">
43                         <path d="M248.032 44.676h-16.466v100.23h47.394v-14.748h-30.928V44.676zM337.091 87.202c-2.101-3.341-5.083-5.965-8.949-7.875-3.865-1.909-7.75
44                     </g>
45                 </svg>
46             </div>
47
48             <div class="mt-8 bg-white dark:bg-gray-800 overflow-hidden shadow sm:rounded-lg">
49                 <div class="grid grid-cols-1 md:grid-cols-2">
50                     <div class="p-6">
51                         <div class="flex items-center">
52                             <svg fill="none" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" viewBox="0 0 24 24" class="w-8 h-8">
53                                 <div class="ml-4 text-lg leading-7 font-semibold"><a href="https://laravel.com/docs" class="underline text-gray-900 dark:text-white">Do
54                         </div>
```

ROUTES (API.PHP)



```
api.php  x
routes > api.php
1  <?php
2
3  use Illuminate\Http\Request;
4  use Illuminate\Support\Facades\Route;
5
6  /*
7  |-----
8  | API Routes
9  |-----
10 |
11 | Here is where you can register API routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | is assigned the "api" middleware group. Enjoy building your API!
14 |
15 */
16
17 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
18     return $request->user();
19 });
20
```

Secara default url laravel untuk api adalah localhost:8000/api (jika menggunakan php artisan serve)

CONTROLLER

```
✓ app
  > Console
  > Exceptions
  ✓ Http
    ✓ Controllers
      🐾 Controller.php
    > Middleware
      🐾 Kernel.php
    > Models
    > Providers
```

Berguna untuk menyimpan logic dan menghubungkan antara view dan model

GENERATE CONTROLLER

```
 HomeController.php ×  
app > Http > Controllers > HomeController.php  
1  <?php  
2  
3  namespace App\Http\Controllers;  
4  
5  use Illuminate\Http\Request;  
6  
7  class HomeController extends Controller  
8  {  
9  |    //  
10 }
```

Cara membuat controller dapat melakukan secara manual membuat filenya atau generate dengan perintah `php artisan make:controller NamaController` dan Jika berhasil otomatis akan membuat file baru di dalam folder `app > Http > Controllers`

FUNCTION CONTROLLER

```
HomeController.php ×  
app > Http > Controllers > HomeController.php  
1  <?php  
2  
3  namespace App\Http\Controllers;  
4  
5  use Illuminate\Http\Request;  
6  
7  class HomeController extends Controller  
8  {  
9      public function index(){  
10         return view('welcome');  
11     }  
12 }
```

Function dengan nama index berguna untuk memanggil tampilan dengan nama welcome dengan bantuan menggunakan kata kunci view

DATABASE LARAVEL

SUPPORT DATABASE



- MariaDB 10.2+
- MySQL 5.7+
- PostgreSQL 10.0+
- SQLite 3.8.8+
- SQL Server 2017+

INTERACTING DATABASE



- Raw Query digunakan untuk melakukan pengaksesan data yang bersifat kompleks (query kompleks)
- Eloquent ORM merupakan Active Record bawaan Laravel yang dapat memudahkan untuk melakukan pengolahan data (create, read, update dan delete).

CONFIG DATABASE

Untuk melakukan config database dapat mengganti file .env atau file config > database.php

```
⚙ .env      X

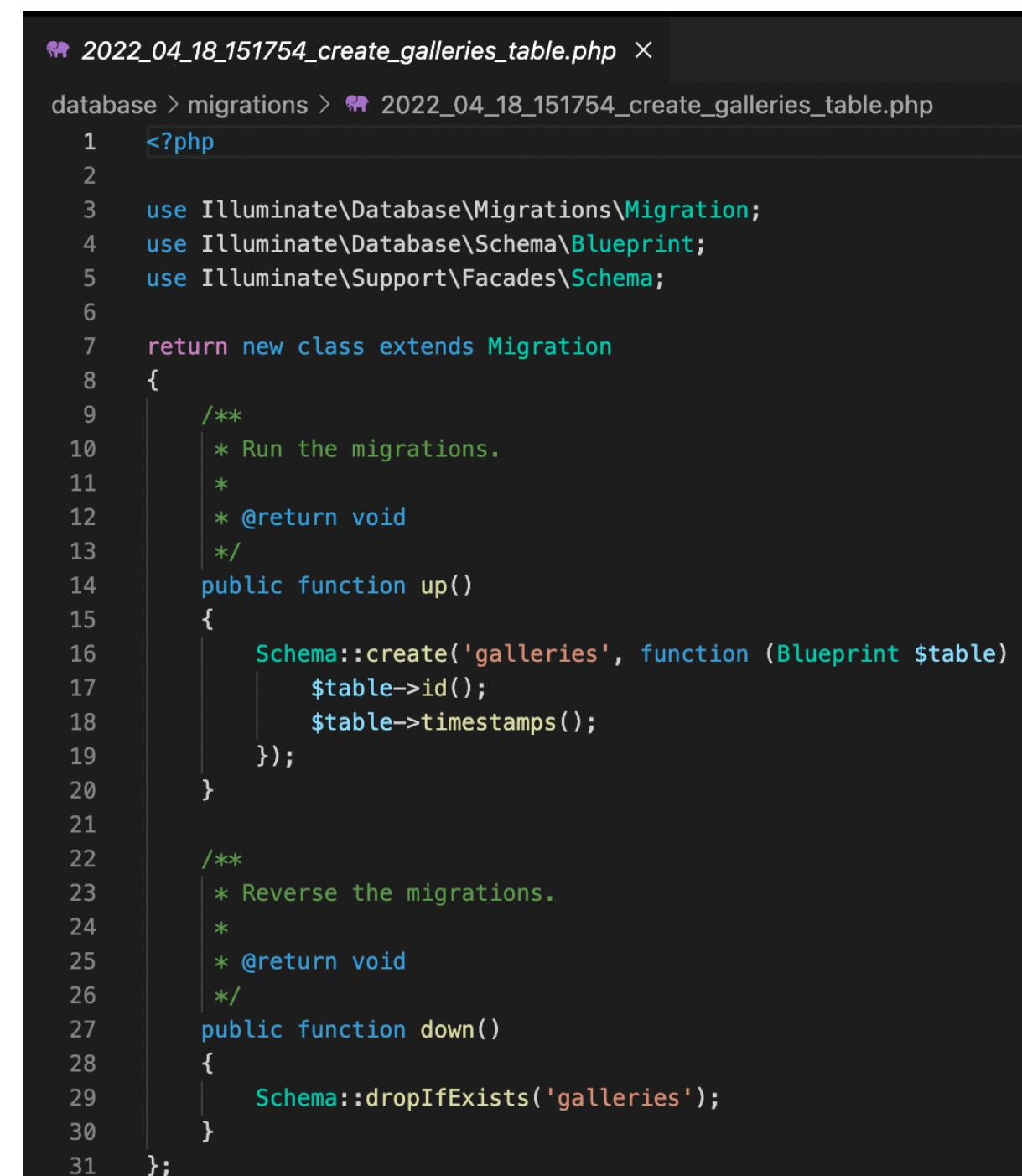
⚙ .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:EEde+zrPbkjvgpC/Nq0Z72K1HgFT/xj3lq/roCGAC0=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=laravel
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smtp
32 MAIL_HOST=mailhog
33 MAIL_PORT=1025
34 MAIL_USERNAME=null
35 MAIL_PASSWORD=null
36 MAIL_ENCRYPTION=null
37 MAIL_FROM_ADDRESS="hello@example.com"
38 MAIL_FROM_NAME="${APP_NAME}"
39
```

CREATE TABLE

Cara membuat table dapat langsung menggunakan perintah SQL create table atau menggunakan yang telah disediakan oleh laravel yaitu migration dengan perintah php artisan make:migration create_namatable_table

```
[faerulsalamun@Faerul-Salamun-MacBook-Pro gallery % php artisan make:migration create_galleries_table
Created Migration: 2022_04_18_151754_create_galleries_table
```

Jika sudah berhasil maka otomatis akan terdapat file baru pada folder database > migrations



```
2022_04_18_151754_create_galleries_table.php
database > migrations > 2022_04_18_151754_create_galleries_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14      public function up()
15      {
16          Schema::create('galleries', function (Blueprint $table) {
17              $table->id();
18              $table->timestamps();
19          });
20      }
21
22      /**
23      * Reverse the migrations.
24      *
25      * @return void
26      */
27      public function down()
28      {
29          Schema::dropIfExists('galleries');
30      }
31  };
```

FILE MIGRATION

Tambahkan code :

```
$table->string('name');
$table->text('description');
$table->string('url');
```

```
🐘 2022_04_18_151754_create_galleries_table.php ×
database > migrations >🐘 2022_04_18_151754_create_galleries_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14      public function up()
15  {
16          Schema::create('galleries', function (Blueprint $table) {
17              $table->id();
18              $table->string('name');
19              $table->text('description');
20              $table->string('url');
21              $table->timestamps();
22          });
23      }
24
25      /**
26      * Reverse the migrations.
27      *
28      * @return void
29      */
30      public function down()
31  {
32          Schema::dropIfExists('galleries');
33      }
34  };
```

WHY MIGRATION ?

- › Dapat membuat kerja tim menjadi lebih efektif dan efisien.
- › Migration seperti version control untuk database kita.
- › Memudahkan ketika pindah environment coding.

RUN MIGRATION

Dapat menggunakan perintah php artisan migrate

```
Migrating: 2022_04_18_151754_create_galleries_table
Migrated: 2022_04_18_151754_create_galleries_table (2.78ms)
```

 galleries	0	16.00 KB	InnoDB	2022-04-18 22:25:09
---	---	----------	--------	---------------------

ERROR MIGRATION ?

In Connection.php line 664:

```
SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long; max key length is 767 bytes (SQL: alter table `users` add unique `users_email_unique`(`email`))
```

In Connection.php line 458:

```
SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long; max key length is 767 bytes
```

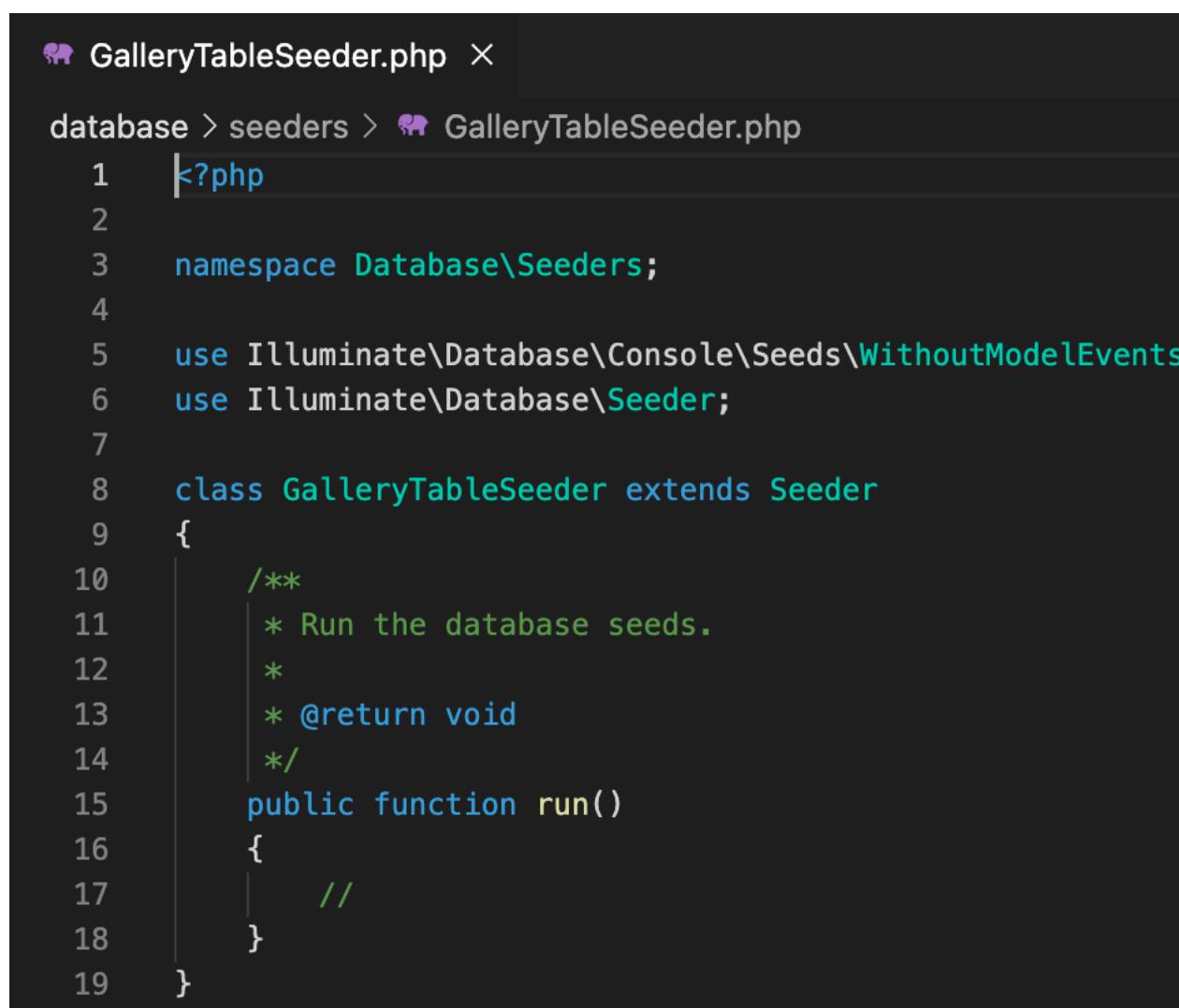
AppServiceProvider.php ×

```
app > Providers > AppServiceProvider.php
1  <?php
2
3  namespace App\Providers;
4
5  use Illuminate\Support\ServiceProvider;
6
7  class AppServiceProvider extends ServiceProvider
8  {
9      /**
10      * Register any application services.
11      *
12      * @return void
13      */
14      public function register()
15      {
16          \Schema::defaultStringLength(191);
17      }
18
19      /**
20      * Bootstrap any application services.
21      *
22      * @return void
23      */
24      public function boot()
25      {
26          //
27      }
28 }
```

INSERT DATA SEEDER

Seeder merupakan fitur untuk mengisi data pada database dengan data sembarang atau data testing. Cara penggunaanya dengan menggunakan perintah php artisan make:seeder NamaTableSeeder

```
[faerulsalamun@Faerul-Salamun-MacBook-Pro gallery % php artisan make:seeder GalleryTableSeeder  
Seeder created successfully.
```



The screenshot shows a code editor window with a dark theme. The file is named 'GalleryTableSeeder.php' and is located in the 'seeders' directory under 'database'. The code is a template for a Seeder class:

```
database > seeders > GalleryTableSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7
8  class GalleryTableSeeder extends Seeder
9  {
10     /**
11      * Run the database seeds.
12      *
13      * @return void
14      */
15     public function run()
16     {
17         //
18     }
19 }
```

INSERT DATA SEEDER

Selanjutnya jalankan perintah php artisan db:seed

```
[faerulsalamun@Faerul-Salamun-MacBook-Pro gallery % php artisan db:seed
Seeding: Database\Seeders\GalleryTableSeeder
Seeded: Database\Seeders\GalleryTableSeeder (18.29ms)
Database seeding completed successfully.
```

```
❖ GalleryTableSeeder.php ×
database > seeders > ❖ GalleryTableSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use DB;
8  use Illuminate\Support\Str;
9
10 class GalleryTableSeeder extends Seeder
11 {
12     /**
13      * Run the database seeds.
14      *
15      * @return void
16     */
17     public function run()
18     {
19         DB::table('galleries')->insert([
20             'name' => 'Gambar PHP',
21             'description' => 'Ini merupakan Gambar PHP',
22             'url' => 'https://www.php.net'
23         ]);
24         DB::table('galleries')->insert([
25             'name' => 'Gambar Bootstrap',
26             'description' => 'Ini merupakan Gambar Bootstrap',
27             'url' => 'https://getbootstrap.com'
28         );
29         DB::table('galleries')->insert([
30             'name' => 'Gambar Javascript',
31             'description' => 'Ini merupakan Gambar Javascript',
32             'url' => 'https://www.javascript.com'
33         ]);
34     }
35 }
```

```
❖ DatabaseSeeder.php ×
database > seeders > ❖ DatabaseSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
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      * @return void
14     */
15     public function run()
16     {
17         $this->call([
18             GalleryTableSeeder::class
19         ]);
20     }
21 }
```

ELOQUENT ORM

RELATIONSHIP

One to one relationships connect one entity to one other entity:



One to many relationships connect one entity to one or more other entities:



Many to many relationships connect many entities to many other entities:

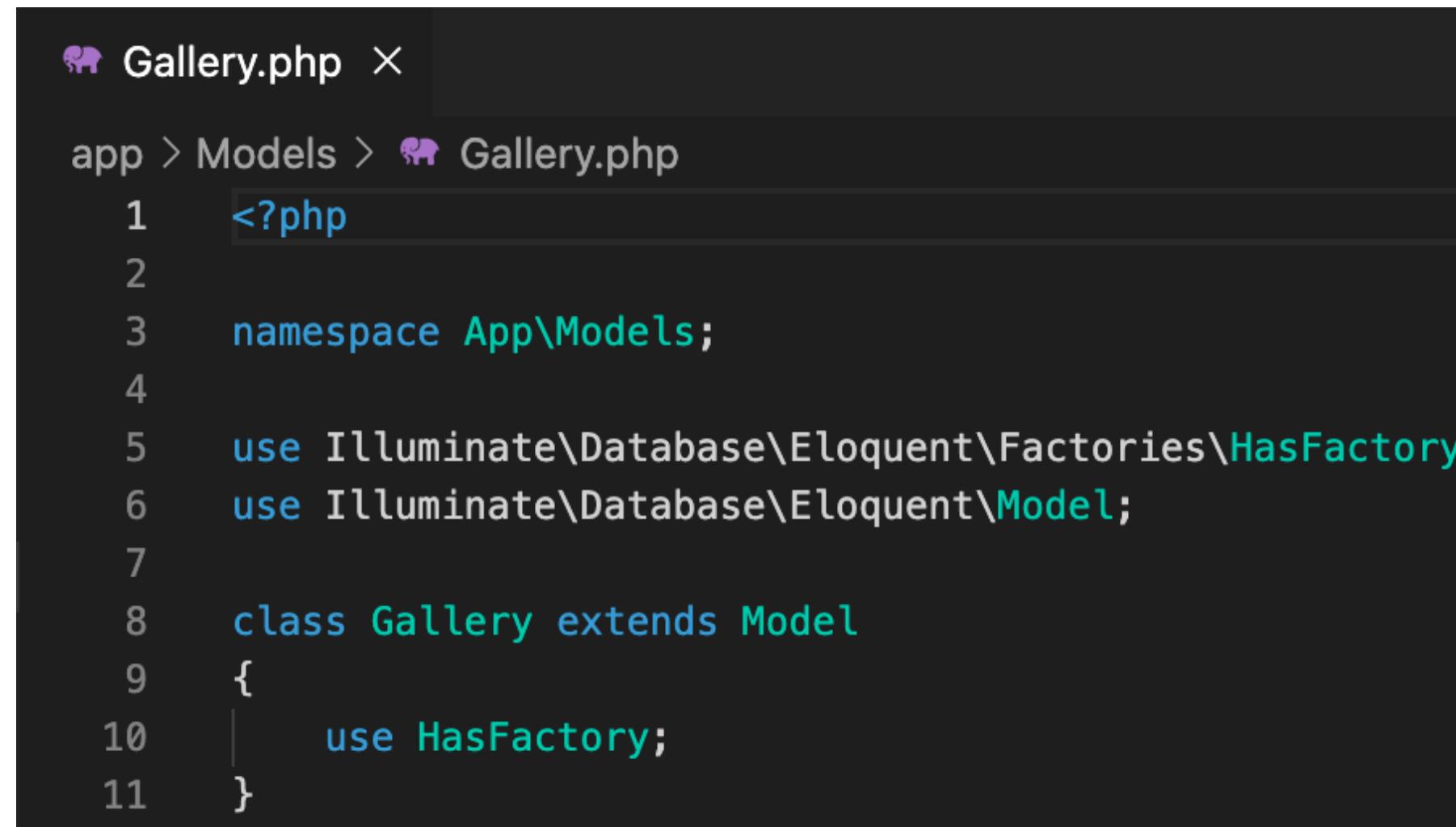


- Relationship adalah hubungan antara table, dibagi menjadi 3 yaitu :
 - One to one dimana sebuah data pada sebuah tabel hanya memiliki relasi ke sebuah data pada tabel yang lain. Misalnya, sebuah data tabel user memiliki relasi 1 profile pada table profile.
 - One to many dimana sebuah data pada sebuah tabel memiliki relasi ke beberapa data pada tabel yang lain. Misalnya, sebuah data tabel user memiliki relasi banyak data alamat di table alamat. Atau dengan kata lain, 1 user memiliki banyak data alamat.
 - Relasi many to one (One to many Inverse) dimana merupakan kebalikan dari relasi one to many. Misalnya kita ingin mengetahui data alamat dimiliki user siapa, maka relasi ini yang akan digunakan.
 - Relasi many to many dimana banyak data pada sebuah tabel memiliki relasi ke banyak data juga pada tabel yang lainnya. Relasi tersebut terbentuk melalui sebuah tabel bantu. Misalnya, banyak data pada tabel user memiliki relasi pembelian ke banyak data pada tabel barang. Relasi tersebut terbentuk dengan tabel bantu bernama table transaksi.

CREATE MODEL

Cara membuat model dengan menggunakan perintah php artisan make:model NamaModel dan jika berhasil File model akan berada di dalam folder app > Models

```
[faerulsalamun@Faerul-Salamun-MacBook-Pro gallery % php artisan make:model Gallery  
Model created successfully.
```



The screenshot shows a code editor window with a dark theme. The title bar says "Gallery.php X". Below it, the path "app > Models > Gallery.php" is shown. The code itself is:

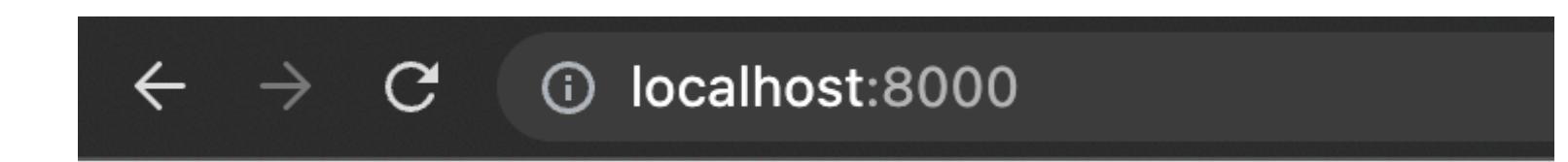
```
1  <?php  
2  
3  namespace App\Models;  
4  
5  use Illuminate\Database\Eloquent\Factories\HasFactory;  
6  use Illuminate\Database\Eloquent\Model;  
7  
8  class Gallery extends Model  
9  {  
10    use HasFactory;  
11 }
```

GET DATA

```
use App\Http\Controllers\HomeController;
```



```
HomeController.php X
app > Http > Controllers > HomeController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\Gallery;
7
8 class HomeController extends Controller
9 {
10     public function index(){
11         $galleries = Gallery::all();
12
13         foreach($galleries as $gallery){
14             echo "Nama : $gallery->name <br>";
15             echo "Description : $gallery->description <br>";
16             echo "URL : $gallery->url <br><br>";
17         }
18     }
19 }
```



Nama : Gambar PHP
Description : Ini merupakan Gambar PHP
URL : <https://www.php.net>

Nama : Gambar Bootstrap
Description : Ini merupakan Gambar Bootstrap
URL : <https://getbootstrap.com>

Nama : Gambar Javascript
Description : Ini merupakan Gambar Javascript
URL : <https://www.javascript.com>

LIMIT GET DATA

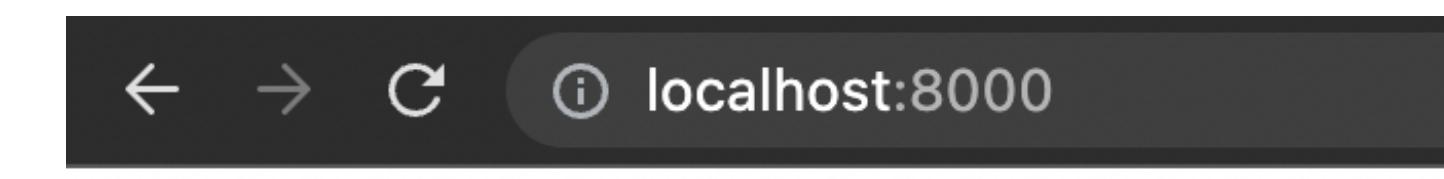
```
🐘 HomeController.php ×  
app > Http > Controllers > 🐘 HomeController.php  
1  <?php  
2  
3  namespace App\Http\Controllers;  
4  
5  use Illuminate\Http\Request;  
6  use App\Models\Gallery;  
7  
8  class HomeController extends Controller  
9  {  
10     public function index()  
11     {  
12         $galleries = Gallery::offset(0)  
13             ->limit(2)  
14             ->get();  
15  
16         foreach($galleries as $gallery){  
17             echo "Nama : $gallery->name <br>";  
18             echo "Description : $gallery->description <br>";  
19             echo "URL : $gallery->url <br><br>";  
20         }  
21     }  
22 }
```



```
← → ⌂ ⓘ localhost:8000  
  
Nama : Gambar PHP  
Description : Ini merupakan Gambar PHP  
URL : https://www.php.net  
  
Nama : Gambar Bootstrap  
Description : Ini merupakan Gambar Bootstrap  
URL : https://getbootstrap.com
```

LIMIT GET DATA WHERE

```
 HomeController.php X
app > Http > Controllers > HomeController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Gallery;
7
8  class HomeController extends Controller
9  {
10     public function index(){
11         $galleries = Gallery::where('name','Gambar PHP')
12                     ->offset(0)
13                     ->limit(2)
14                     ->get();
15
16         foreach($galleries as $gallery){
17             echo "Nama : $gallery->name <br>";
18             echo "Description : $gallery->description <br>";
19             echo "URL : $gallery->url <br><br>";
20         }
21     }
22 }
```



Nama : Gambar PHP
Description : Ini merupakan Gambar PHP
URL : https://www.php.net

SHOW DATA IN VIEW

```
HomeController.php ×  
app > Http > Controllers > HomeController.php  
1  <?php  
2  
3  namespace App\Http\Controllers;  
4  
5  use Illuminate\Http\Request;  
6  use App\Models\Gallery;  
7  
8  class HomeController extends Controller  
9  {  
10     public function index(){  
11         $galleries = Gallery::all();  
12  
13         return view('gallery',["galleries"=>$galleries]);  
14     }  
15 }
```



```
gallery.blade.php ×  
resources > views > gallery.blade.php  
1  @foreach ($galleries as $gallery)  
2      Nama : {{$gallery->name }}<br>  
3      Description : {{$gallery->description }}<br>  
4      URL : {{$gallery->url }}<br><br>  
5  @endforeach
```



← → C ⓘ localhost:8000

Nama : Gambar PHP
Description : Ini merupakan Gambar PHP
URL : <https://www.php.net>

Nama : Gambar Bootstrap
Description : Ini merupakan Gambar Bootstrap
URL : <https://getbootstrap.com>

Nama : Gambar Javascript
Description : Ini merupakan Gambar Javascript
URL : <https://www.javascript.com>

CREATE DATA



CREATE DATA

```
gallery-add.blade.php ×  
resources > views > gallery-add.blade.php  
1  <form action="/" method="POST">  
2    @csrf  
3    <input type="text" name="name" placeholder="Name"/><br><br>  
4    <input type="text" name="description" placeholder="Description"/><br><br>  
5    <input type="text" name="url" placeholder="URL"/><br><br>  
6    <input type="submit" name="submit" placeholder="submit"/><br>  
7  </form>
```

The diagram illustrates the flow of data from the view code to the user interface and finally to the controller implementation.

On the left, a code editor window shows the `gallery-add.blade.php` file with its Blade template syntax. A red arrow points from this code to the center, where a browser screenshot displays the form fields: Name, Description, URL, and Submit.

On the right, another red arrow points from the browser screenshot to a code editor window showing the `processAdd` method implementation. This method creates a new `Gallery` object, sets its properties from the request, saves it, and then redirects to the root path.

```
localhost:8000/add  
Name  
Description  
URL  
Submit
```

```
public function processAdd(Request $request){  
    $gallery = new Gallery();  
    $gallery->name = $request->get('name');  
    $gallery->description = $request->get('description');  
    $gallery->url = $request->get('url');  
    $gallery->save();  
  
    return redirect('/');  
}
```

UPDATE DATA



UPDATE DATA

```
gallery-edit.blade.php x
resources > views > gallery-edit.blade.php
1  <form action="/{$gallery->id}" method="POST">
2    @csrf
3    <input type="hidden" name="id" value="{{$gallery->id}}"/>
4    <input type="text" name="name" placeholder="Name" name="name" value="{{$gallery->name}}"/><br><br>
5    <input type="text" name="description" placeholder="Description" name="description" value="{{$gallery->description}}"/><br><br>
6    <input type="text" name="url" placeholder="URL" name="url" value="{{$gallery->url}}"/><br><br>
7    <input type="submit" name="submit" placeholder="submit"/><br>
8  </form>
```

localhost:8000/1

Gambar PHP

Ini merupakan Gambar

https://www.php.net

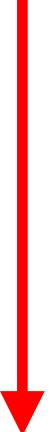
Submit

```
public function processEdit($id, Request $request){
    $gallery = Gallery::find($id);
    $gallery->name = $request->get('name');
    $gallery->description = $request->get('description');
    $gallery->url = $request->get('url');
    $gallery->save();

    return redirect('/');
}
```

DELETE DATA

```
Route::get('/delete/{id}', [HomeController::class, 'processDelete']);
```



```
gallery.blade.php ×  
resources > views > gallery.blade.php  
1 <?php @foreach ($galleries as $gallery)  
2 |     Nama : {{$gallery->name }}<br>  
3 |     Description : {{$gallery->description }}<br>  
4 |     URL : {{$gallery->url }}<br>  
5 |     Action : <a href="/{$gallery->id }">Edit</a> || <a href="/delete/{{$gallery->id }}">Delete</a><br><br>  
6 @endforeach  
7  
8 <a href="add">Add</a>
```



localhost:8000

Nama : Gambar PHP
Description : Ini merupakan Gambar PHP
URL : https://www.php.net
Action : [Edit](#) || [Delete](#)

Nama : Gambar Bootstrap
Description : Ini merupakan Gambar Bootstrap
URL : https://getbootstrap.com
Action : [Edit](#) || [Delete](#)

Nama : Gambar Javascript
Description : Ini merupakan Gambar Javascript
URL : https://www.javascript.com
Action : [Edit](#) || [Delete](#)

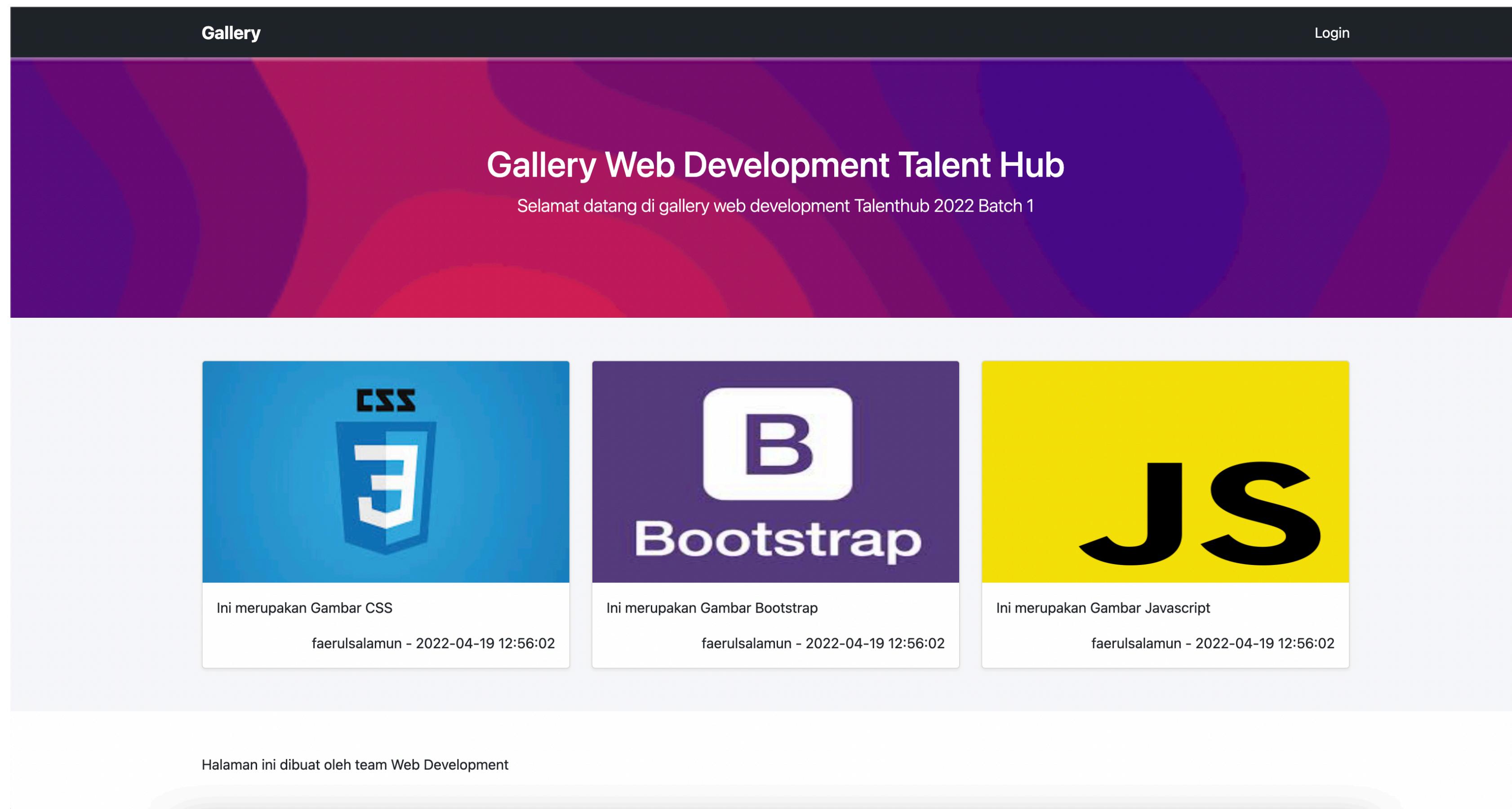
[Add](#)

DELETE DATA

```
public function processDelete($id){  
    $gallery = Gallery::find($id);  
    $gallery->delete();  
  
    return redirect('/');  
}
```

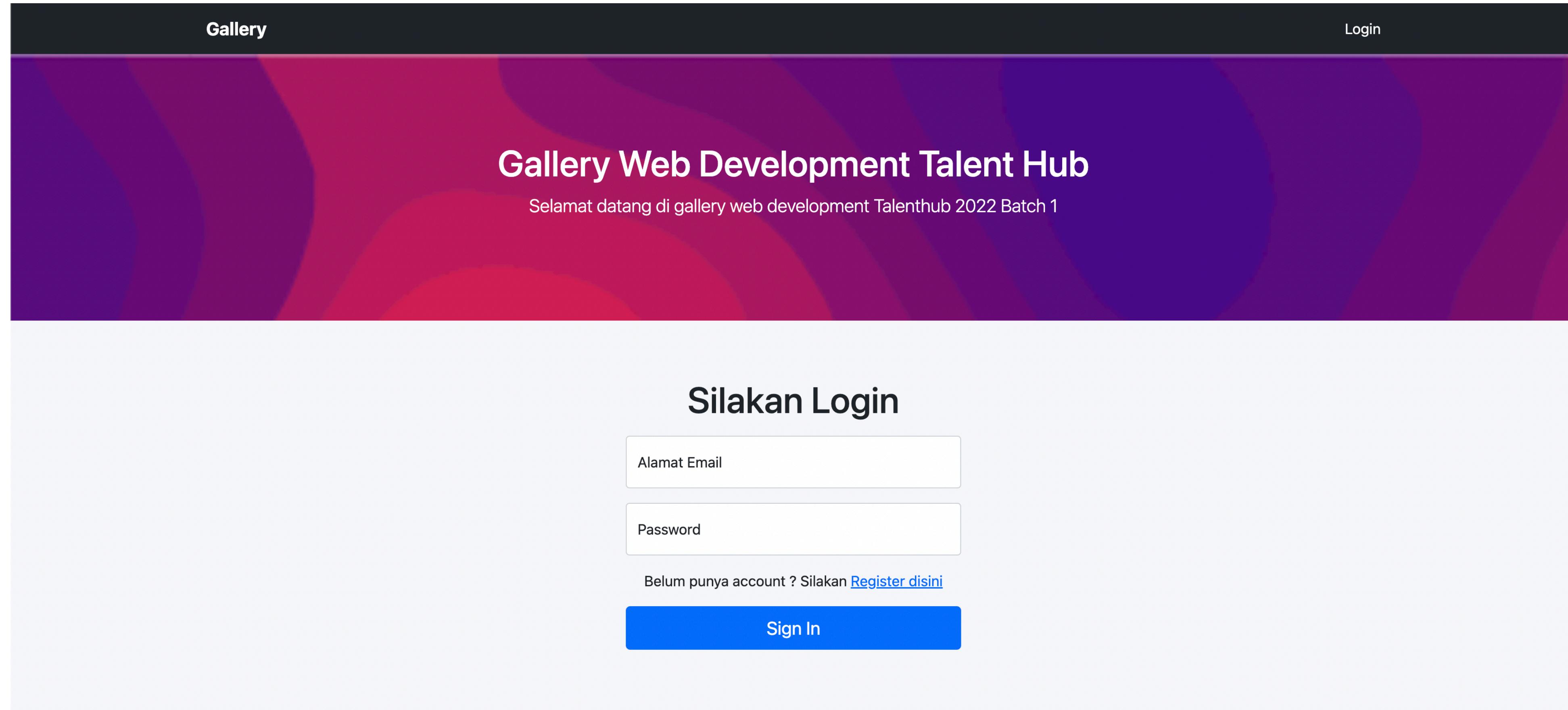
NEXT ?

- › Integrasi website gallery dengan menggunakan Laravel



NEXT ?

.....



NEXT ?

.....

The screenshot shows a web application interface. At the top, there is a dark header bar with the word "Gallery" on the left and "Login" on the right. Below the header is a large, abstract background image featuring overlapping circles in shades of purple, red, and blue. In the center of this background, the text "Gallery Web Development Talent Hub" is displayed in a large, bold, white font. Below this, a smaller line of text reads "Selamat datang di gallery web development Talenthub 2022 Batch 1". The main content area is a light gray box containing the heading "Silakan Register" in a large, bold, black font. Below this heading are three input fields: "Nama" (Name), "Alamat Email" (Email Address), and "Password". Underneath these fields is a link that says "Sudah punya account ? Silakan [Login disini](#)". At the bottom of the form is a large blue button with the word "Register" in white.

Gallery

Login

Gallery Web Development Talent Hub

Selamat datang di gallery web development Talenthub 2022 Batch 1

Silakan Register

Nama

Alamat Email

Password

Sudah punya account ? Silakan [Login disini](#)

Register