**PENGEMBANGAN APLIKASI WEBSITE**

A logo with a red and grey letter u

Description automatically generated

Disusun Oleh:

Daffa Yusuf Mahendra 1204230015

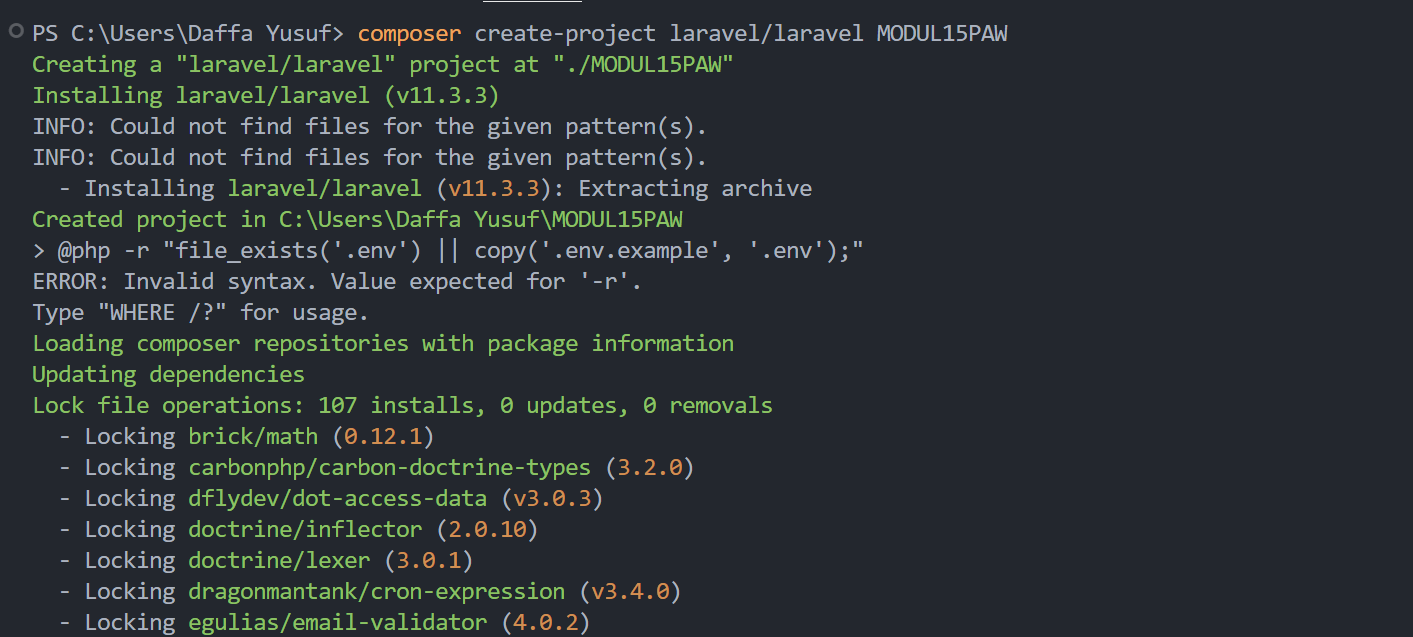
**PROGRAM STUDI S1 SISTEM INFORMASI**

**FAKULTAS REKAYASA INDUSTRI**

**TELKOM UNIVERSITY SURABAYA**

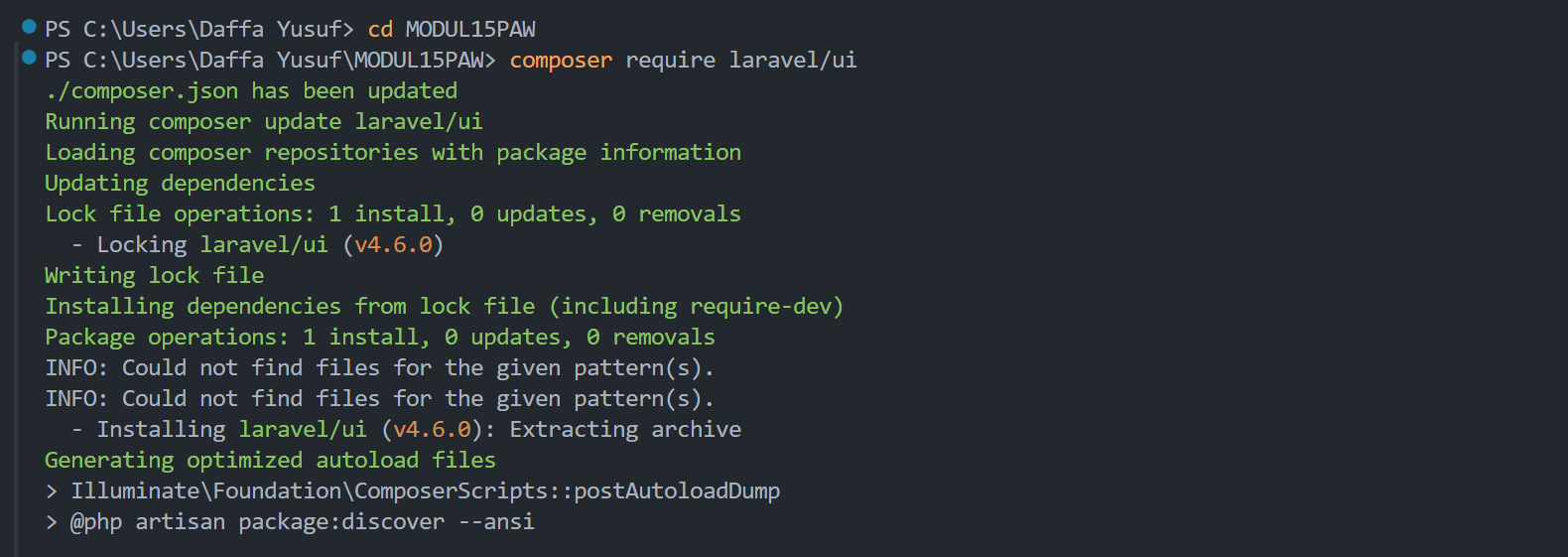
1. **LARAVEL AUTHENTICATION**
2. **Authentication via Laravel UI**

* Buat project Laravel baru dengan command di bawah ini. Perhatikan kondisi project laravel tersebut pada bagian controllers, views, dan routes nya.



Composer adalah dependency manager untuk PHP, composer akan mengunduh Laravel dari repository resmi dan menyalin file proyek ke dalam folder MODUL15PAW.

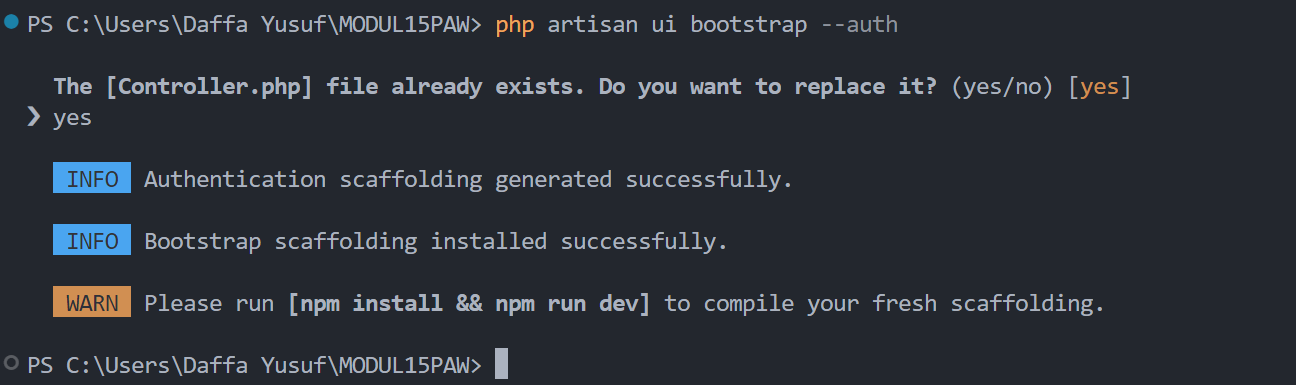
* Setelah itu, masuk pada project Laravel yang baru di buat, dan install package Laravel UI via composer



Require di atas Memberitahu Composer untuk menambahkan paket tertentu (dalam hal ini laravel/ui) ke proyek Laravel MODUL15PAW. Composer akan mencari paket tersebut di repository resmi (Packagist) dan menambahkannya ke file composer.json di bawah bagian require.

Laravel UI adalah paket tambahan untuk Laravel yang menyediakan scaffolding (kerangka dasar) frontend. Paket ini digunakan untuk menambahkan autentikasi dan komponen UI berbasis frontend seperti Bootstrap, Vue.js, atau React.js ke proyek Laravel.

* Generate Scafolding Bootstrap dengan fitur Authentication pada project laravel kita. Perhatikan kondisi project laravel tersebut pada bagian controllers, views, dan routes nya.



Perintah kode di atas digunakan untuk menambahkan fitur autentikasi dan integrasi Bootstrap ke proyek Laravel. Ini mencakup halaman login, registrasi, dan fitur autentikasi dasar lainnya.

* Analisis kondisi project laravel antara **sebelum** dan **sesudah** menjalankan tahapan di atas. Jelaskan pada laporan praktikum!

Kondisi awal pada tahap setelah menjalankan perintah composer create-project laravel/laravel MODUL15PAW, kondisi proyek Laravel adalah membuat struktur dasar Laravel seperti :

* Folder utama: app/, routes/, resources/, database/, dll.
* File utama: composer.json, .env, artisan.
* Tidak ada scaffolding autentikasi atau integrasi frontend.

Kondisi setelah melakukan composer require laravel/ui adalah Laravel UI sekarang tersedia di proyek. Perintah php artisan ui dapat digunakan untuk menambahkan scaffolding frontend dan autentikasi.

Kondisi setelah php artisan ui bootstrap --auth adalah :

**Frontend Bootstrap Ditambahkan**:

* Scaffolding frontend berbasis Bootstrap diintegrasikan.
* File resources/views/layouts/app.blade.php dibuat untuk layout utama.

**Autentikasi Ditambahkan**:

* Halaman login (resources/views/auth/login.blade.php) dan registrasi (resources/views/auth/register.blade.php) ditambahkan.
* Rute autentikasi (/login, /register, dll.) otomatis terdaftar.
* Jalankan project laravel tersebut via browser, akan muncul menu & fitur **login** dan **register** pada halaman welcome.

A black background with white text

Description automatically generated

1. **Setup Database**

* Jalankan service MySQL dan Apache via control panel XAMPP.

A screenshot of a computer

Description automatically generated

Menghidupkan **XAMPP Apache** dan **MySQL** sangat penting untuk menjalankan proyek Laravel karena Laravel membutuhkan layanan web server dan database untuk berfungsi.

* Buat database dengan name **“laravel\_auth”** via **PhpMyAdmin**.

A screenshot of a computer

Description automatically generated

* Buka file .env dan atur dan sesuaikan value variabel DB\_DATABASE, DB\_USERNAME, DB\_PASSWORD.

A computer screen shot of a computer code

Description automatically generated

DB\_CONNECT digunakan untuk menentukan jenis database yang digunakan, dalam kasus ini, Laravel diatur untuk menggunakan **MySQL** sebagai database.

DB\_HOST menunjukkan alamat host tempat database dijalankan. 127.0.0.1 adalah alamat lokal (localhost), yang berarti database berjalan pada mesin yang sama dengan server aplikasi.

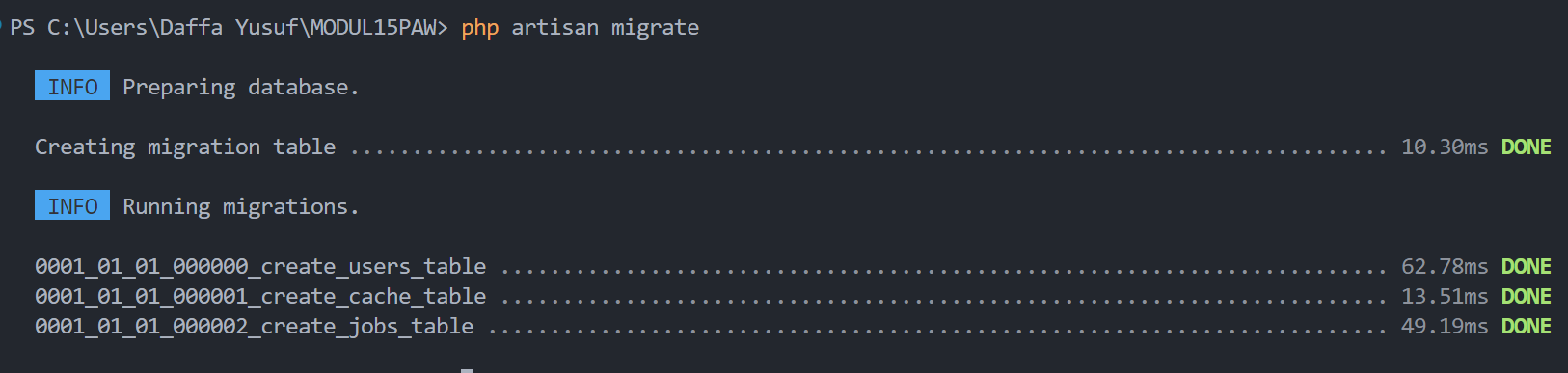
DB\_PORT adalah port yang digunakan untuk koneksi ke MySQL. **3306** adalah port default untuk MySQL.

DB\_DATABASE adalah nama database yang akan digunakan aplikasi Laravel. Dalam kasus ini, nama database adalah data\_masster.

DB\_USERNAME adalah username untuk autentikasi ke database. Default-nya adalah **root** untuk MySQL.

DB\_PASSWORD adalah untuk autentikasi ke database. Dalam pengaturan default XAMPP, password untuk **root** biasanya kosong (tidak diisi).

* Lakukan laravel migration



Perintah php artisan migrate digunakan untuk mengeksekusi migrasi database di Laravel. Migrasi adalah fitur Laravel yang memungkinkan pengembang untuk mengelola struktur database menggunakan skrip PHP.

1. **Menjalankan Fitur Register dan Login**

* Jalankan project laravel via browser dan coba fitur **Register** untuk membuat akun user baru pada sistem.

A screenshot of a computer

Description automatically generated

* Coba fitur **Login** menggunakan akun user yang baru saja dibuat.

A screenshot of a computer

Description automatically generated

* Analisis alur program ketika menjalankan fitur login dan register. Bagaimana hubungan antara file-file **views**, **controllers** dan juga **routes** yang ada. Jelaskan pada laporan praktikum!

Alur program untuk fitur login dan register :

1. Route (web.php)

Routes mendefinisikan URL atau endpoint yang akan diakses oleh pengguna dan menentukan controller mana yang akan menangani permintaan tersebut. File ini terletak di routes/web.php.

1. Controllers (LoginController dan RegisterController)

Controller menerima permintaan dari routes dan memproses logika bisnis, seperti validasi input, autentikasi, dan penyimpanan data pengguna. Controller kemudian mengarahkan pengguna ke halaman yang sesuai (misalnya, dashboard atau halaman login kembali) atau menampilkan form jika ada kesalahan.A screen shot of a computer program

Description automatically generated

Kode yang di atas adalah implementasi dari fungsi authenticate() dalam file login controller. Sebuah controller Laravel yang bertanggung jawab untuk menangani proses autentikasi pengguna, terutama dalam fitur login.

Selanjutnya yaitu register controller method ini menangani pendaftaran pengguna baru. Di sini, data yang diterima dari form registrasi disimpan

A screen shot of a computer code

Description automatically generated

1. Views (Blade Templates)

Views menampilkan formulir atau pesan kesalahan kepada pengguna berdasarkan data yang dikirim oleh controller. View login dan registrasi berfungsi untuk mengumpulkan input dari pengguna dan mengirimkan data ke controller melalui permintaan POST.

1. **Proses Authentication Secara Manual**

* Buka file **LoginController**, tambahkan function dengan nama **authenticate()** menjadi seperti di bawah ini.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Kode di atas merupkana controller yang digunakan untuk mengatur seluruh proses login menggunakan fitur bawaan Laravel. Dengan menggunakan trait **AuthenticatesUsers**, banyak logika autentikasi yang sudah disediakan oleh Laravel, sehingga hanya perlu mengkonfigurasi sedikit saja. Metode authenticate() adalah tempat untuk menangani validasi kredensial dan mengelola alur autentikasi, serta memberikan pengalaman pengguna yang aman dan mudah.

A screenshot of a computer program

Description automatically generated

Name space di atas digunakan untuk mendefinisikan namespace dari controller ini, yang merupakan bagian dari controller untuk otentikasi (Auth).

Selain itu, terdapat use yang menyertakan berbagai class yang dibutuhkan seperti :

* **Controller**: Parent class untuk semua controller di Laravel.
* **RouteServiceProvider**: Menyediakan jalur atau rute default untuk aplikasi.
* **AuthenticatesUsers**: Trait yang disediakan oleh Laravel untuk menangani fungsionalitas login dan logout secara otomatis.
* **Request**: Digunakan untuk menangani input dari pengguna.
* **Auth**: Facade yang menyediakan akses ke layanan autentikasi di Laravel.

A screenshot of a computer program

Description automatically generated

Kode di atas di fungsikan setelah pengguna berhasil login, mereka akan diarahkan ke halaman yang ditentukan oleh variabel ini, dalam hal ini /home. Variabel ini adalah properti yang sudah disediakan oleh **AuthenticatesUsers**, tetapi dapat Anda ubah sesuai kebutuhan aplikasi.

* Buka file **routes/web.php** kemudian buat route baru dan arahkan ke function **authenticate()** pada **LoginController** seperti di bawah ini.

A black background with yellow text

Description automatically generated

Route di atas di fungsikan untuk mendefinisikan sebuah route yang merespons permintaan **POST**. Permintaan POST umumnya digunakan untuk mengirimkan data yang dapat mengubah status aplikasi, seperti mengirimkan data melalui formulir. Dalam hal ini, data yang dikirim adalah informasi login pengguna. Untuk “**/login”** Ini adalah URL atau endpoint yang akan dipanggil saat pengguna mengaksesnya dengan metode POST. Dalam hal ini, /login adalah alamat URL yang akan menerima permintaan login dari pengguna.

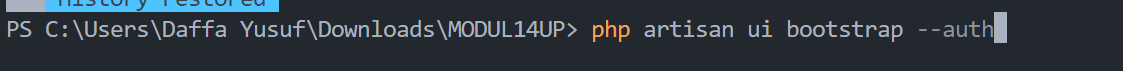
Selanjutnya, LoginControler::class Menunjukkan kelas **LoginController** yang menangani logika untuk autentikasi login.

* Jalankan kembali fitur **Login** via Browser.

1. **Latihan**

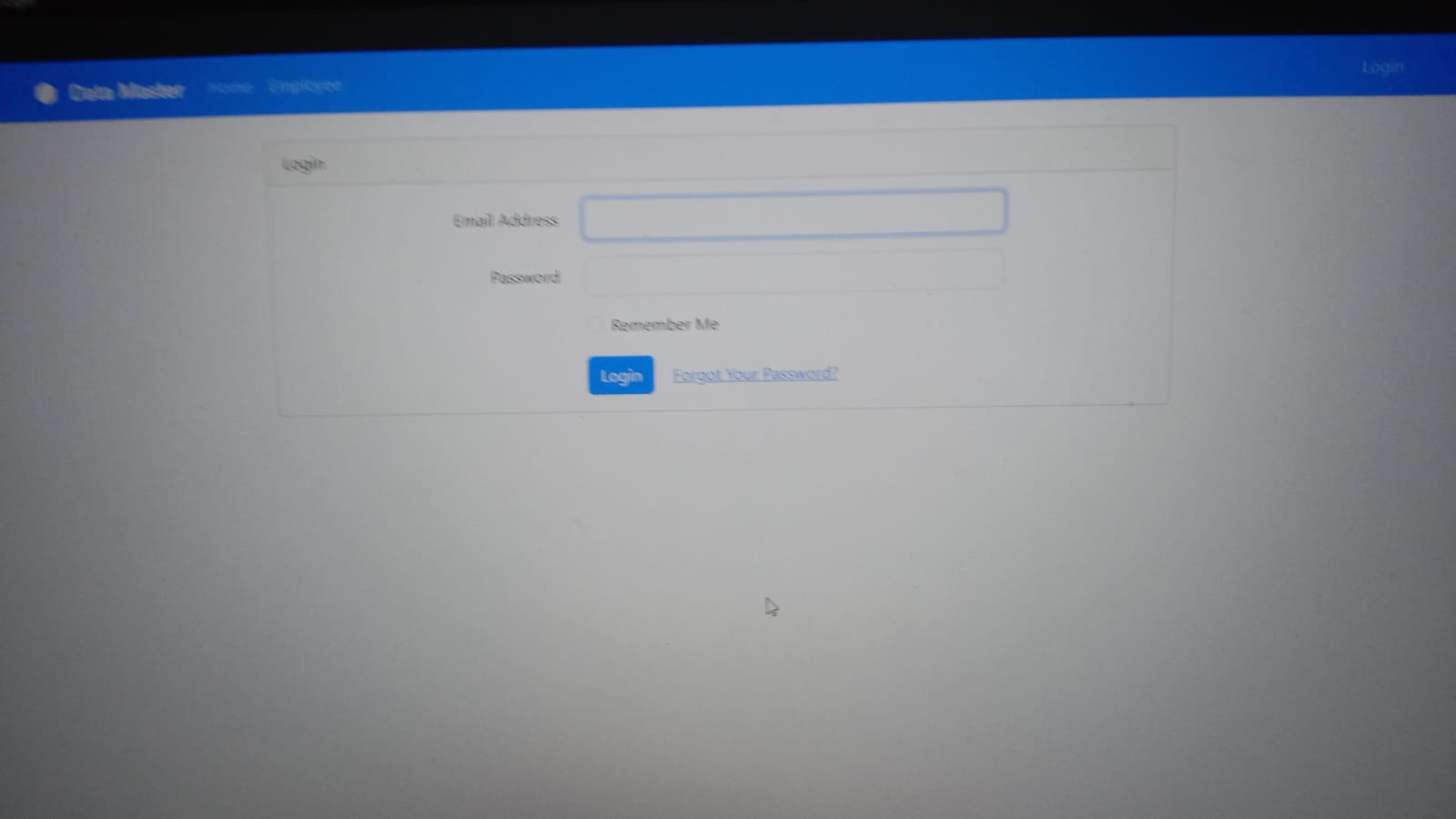
* **Lanjutkan (bukan buat dari awal lagi)** Project Laravel yang kita kerjakan pada **Modul 14 Praktikum (aplikasi sederhana data master employee)** dengan menerapkan Laravel Authentication yang sudah dipelajari pada project laravel tersebut.
* Hasil akhir yang diinginkan adalah sebuah aplikasi sederhana data master employee yang telah memiliki:
* Halaman **Login** dan fitur **Authentication** (tidak ada halaman & fitur Register)
* Halaman Login langsung muncul atau tampil ketika user mengakses [**http://localhost:8000**](http://localhost:8000)via browser.
* Menu **Logout** pada bagian **Navbar**
* Ketika berhasil login, user diarahkan ke halaman **home**.
* Semua halaman aplikasi hanya bisa diakses **setelah login**.
* Buat satu data **user awal**, via **Seeder** dengan:
* **name:** Administrator
* **email/username:** admin@admin
* **password:** adminadmin
* Password yang disimpan di database adalah password yang terenkripsi
* Gunakan method atau function **bcrypt()** untuk mengenkripsi text string password anda ketika akan melakukan insert data user ke database.

1. **Penerapan**
2. Generate Scafolding Bootstrap dengan fitur Authentication pada project laravel kita.



Sama seperti kode sebelumnya, Perintah kode di atas digunakan untuk menambahkan fitur autentikasi dan integrasi Bootstrap ke proyek Laravel. Ini mencakup halaman login, registrasi, dan fitur autentikasi dasar lainnya.

1. Setelah generate maka otomatis akan muncul tampilan login dan register seperti pada gambar di bawah ini :



Gambar di atas merupakan tampilan login default dari bawaan auth.

Selanjutnya, mengubah code pada file **login.blade.php** agar tampilan login sama seperti contoh pada modul.

A screen shot of a computer program

Description automatically generated



Untuk memberikan style pada kotak login disini menggunakan card yang memiliki padding **p-4**, lebar 350px, sudut membulat (border-radius: 15px), dan efek bayangan (box-shadow).



Selanjutnya, code di atas merupakan formulasi login dimana :

Formulir Login:

* Menggunakan metode POST untuk mengirimkan data login.
* action="{{ route('login') digunakan untuk menyebutkan rute login di Laravel yang dipetakan ke fungsi autentikasi di LoginController.
* @csrf: Laravel directive untuk menambahkan token CSRF (Cross-Site Request Forgery) ke form, untuk menghindari serangan CSRF.

Input Email:

* <input id="email" type="email" ...>: Kolom input untuk email pengguna.
* @error('email'): Menampilkan pesan error jika validasi gagal (misalnya, email tidak valid atau tidak ada).

Input Password:

* <input id="password" type="password" ...>: Kolom input untuk password pengguna.
* @error('password'): Menampilkan pesan error jika password tidak valid.

Tombol Login:

* <button type="submit" class="btn btn-primary w-100">: Tombol untuk mengirimkan formulir login.

Hasil tampilan

A screen shot of a login

Description automatically generated

1. Selanjutnya masuk ke file web.php untuk setting route agar ketika pertama kali mengakses localhost maka langsung ke halaman login

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

Kode di atas merupakan definisi rute untuk beerapa halaman penting seperti halaman login yang diarahkan ke login route,

A screen shot of a computer code

Description automatically generated

Halaman beranda (home) dan profil (profile), yang dilindungi oleh middleware autentikasi.

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

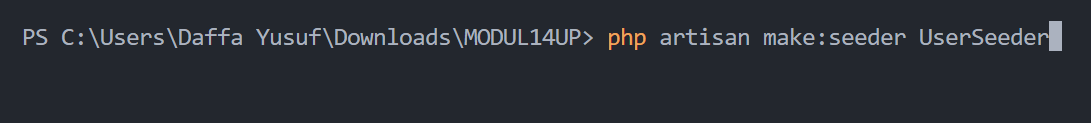
Rute resource untuk mengelola data karyawan (employees), termasuk rute untuk CRUD karyawan.

A screen shot of a computer code

Description automatically generated

Menyediakan rute untuk login dan registrasi otomatis menggunakan **Auth::routes()**.

1. Membuat data user awal dengan mengetikkan perintah



Perintah php artisan make:seeder UserSeeder digunakan untuk membuat sebuah **seeder** baru di dalam aplikasi Laravel. Seeder ini digunakan untuk mengisi (seeding) database dengan data awal (dummy data) saat pengembangan atau testing aplikasi.

Setelah itu masuk pada class UserSeeder dan tambahkan kode berikut :



Name space di atas digunakan untuk Menentukan namespace untuk seeder ini. File ini berada dalam folder database/seeders.

Selain itu, terdapat use yang menyertakan berbagai class yang dibutuhkan seperti :

* Seeder : Mengimpor kelas **Seeder** dari Laravel, yang memungkinkan kita untuk membuat dan menjalankan seeder.
* DB : Mengimpor facade **DB** yang digunakan untuk berinteraksi langsung dengan database, seperti memasukkan data ke dalam tabel.
* Hash : Pada kode ini tidak digunakan, meskipun biasanya digunakan untuk mengenkripsi password. Namun, di sini password dienkripsi dengan menggunakan bcrypt langsung.



Kode dalam public function run di atas menggunakan facade **DB** untuk melakukan query **INSERT** ke tabel users.

* **name**: Nama pengguna yang akan dimasukkan ke tabel, di sini diberi nilai 'Administrator'.
* **email**: Email pengguna yang akan dimasukkan, di sini diberi nilai 'admin@admin'.
* **password**: Password yang akan dimasukkan ke dalam tabel users. Dalam hal ini, password 'adminadmin' dienkripsi menggunakan fungsi **bcrypt()** untuk meningkatkan keamanan penyimpanan password.
* **created\_at** dan **updated\_at**: Timestamp yang menyimpan waktu pembuatan dan pembaruan data. Fungsi **now()** digunakan untuk mendapatkan waktu saat ini.

1. URL GITHUB

LINK GITHUB MODUL 15 PRAKTIKUM

<https://github.com/daffayusufm/WebModul15Prak/tree/main/MODUL15PRAK>

LINK GITHUB MODUL 15 TUGAS