

# Pemrograman Framework

Minggu 10  
Laravel Authentication

**Tim Dosen:**  
Purnama Anaking, S.Kom., M.Kom.  
Berlian Rahmy Lidiawaty, S.S., M.MT.



## Authentication

# Kemampuan Akhir yang Direncanakan:

- Mahasiswa mampu menerapkan konsep dasar Authentication pada framework Laravel.

# Laravel Authentication

- Laravel Authentication dipergunakan untuk memproteksi halaman atau fitur dari web yang hanya diakses oleh orang tertentu yang diberikan hak.
- Laravel membuat penerapan otentikasi sangat sederhana dan telah menyediakan berbagai fitur yang dapat dimanfaatkan tanpa perlu melakukan penambahan instalasi modul tertentu.



# Laravel Authentication

- File konfigurasi otentikasi terletak di config / auth.php, yang berisi beberapa opsi yang terdokumentasi dengan baik untuk mengubah konfigurasi dari layanan otentikasi.
- Pada intinya, fasilitas otentikasi Laravel terdiri dari “guards” dan “providers”. Guards menentukan bagaimana pengguna diautentikasi untuk setiap permintaan. Misalnya, Laravel mengirim dengan guards untuk sesi dengan menggunakan penyimpanan session dan cookie.

# Alternatives

## → Laravel's Built-in Browser Authentication Services

- ◆ Laravel Jetstream / Fortify
- ◆ Laravel UI (with `--auth` command )
- ◆ Manually authenticating users

## → Laravel's API Authentication Services

- ◆ Passport
- ◆ Sanctum

# Authentication via Laravel UI

---



# Create Laravel Project with Authentication

→ Start by creating a new Laravel project :

```
composer create-project laravel/laravel your-project-name
```

→ Then cd into the newly created project folder and install the **Laravel UI**

```
composer require laravel/ui
```

→ Generate the **Bootstrap Scaffolding with Authentication**

◆ **php artisan ui bootstrap --auth**

# Authentication (Controllers & Routes)

- That command should be used on **fresh applications** and will install a **layout view**, **registration** and **login views**, as well as **routes** for all authentication endpoints.
- The **laravel/ui** package also generates several pre-built authentication controllers, which are located in the **App\Http\Controllers\Auth** namespace. A **HomeController** will also be generated to handle post-login requests to your application's dashboard.
- The **RegisterController** handles new user registration, the **LoginController** handles authentication, the **ForgotPasswordController** handles e-mailing links for resetting passwords, and the **ResetPasswordController** contains the logic to reset passwords.
- Each of these controllers uses a **trait** to include their necessary **methods**. For many applications, you **will not need to modify** these controllers at all.



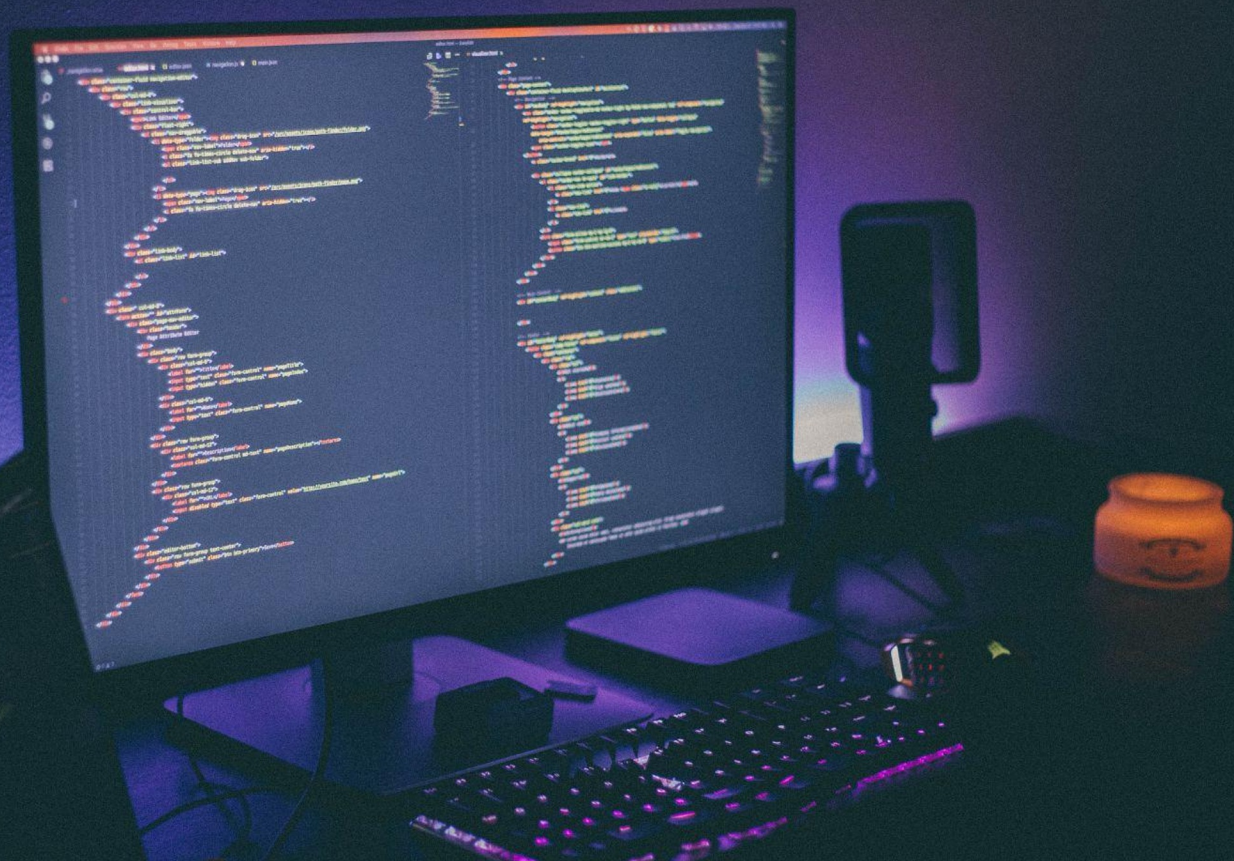


# Authentication (Views)

- As mentioned in the previous section, the **laravel/ui** package's **php artisan ui bootstrap --auth** command will create **all of the views** you need for **authentication** and place them in the **resources/views/auth** directory.
- The **ui** command will also create a **resources/views/layouts** directory containing a **base layout** for your application. All of these views use the **Bootstrap** CSS framework, but you are free to customize them however you wish.

# LIVE CODING SESSION

Let's Code!





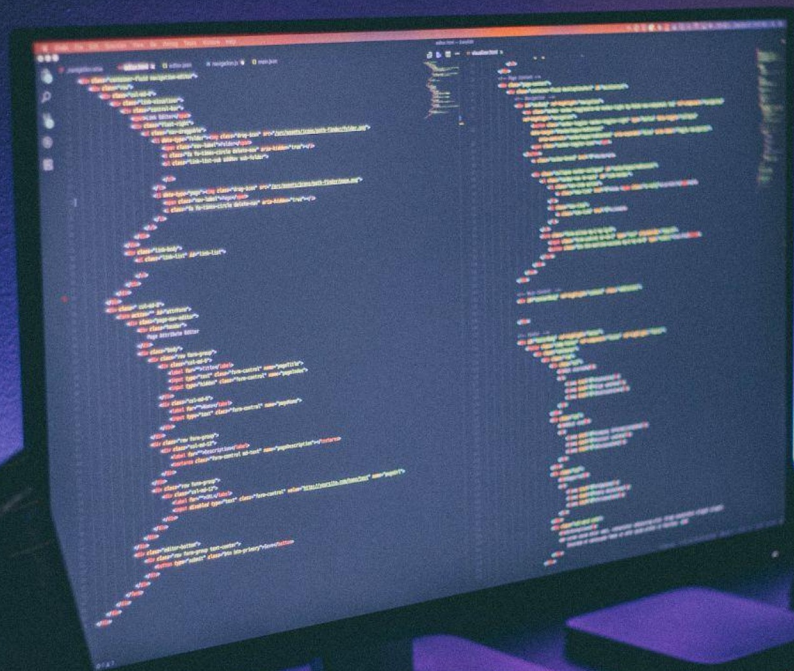
# Authenticating

- Now that you have **routes** and **views** setup for the included authentication **controllers**, you are ready to **register** and **authenticate** new users for your application!
- You may access your application in a **browser** since the authentication controllers already **contain the logic** (via their **traits**) to **authenticate** existing users and **store** new users in the database.



# LIVE CODING SESSION

Let's Code!





# Protecting Routes

- **Route middleware** can be used to only allow authenticated users to access a given route. Laravel ships with an **auth** middleware, which is defined at **Illuminate\Auth\Middleware\Authenticate**. Since this middleware is already registered in your HTTP kernel, all you need to do is attach the middleware to a route definition.
- If you are using **controllers**, you may call the **middleware** method from the controller's **constructor** instead of attaching it in the route definition directly.

```
Route::get('profile', function () {  
    // Only authenticated users may enter...  
})->middleware('auth');
```

```
public function __construct()  
{  
    $this->middleware('auth');  
}
```



# Redirecting Unauthenticated Users

- When the **auth** middleware detects an **unauthorized** user, it will **redirect** the user to the “**login**” named route.
- You may **modify** this behavior by updating the **redirectTo** function in your **app/Http/Middleware/Authenticate.php** file.

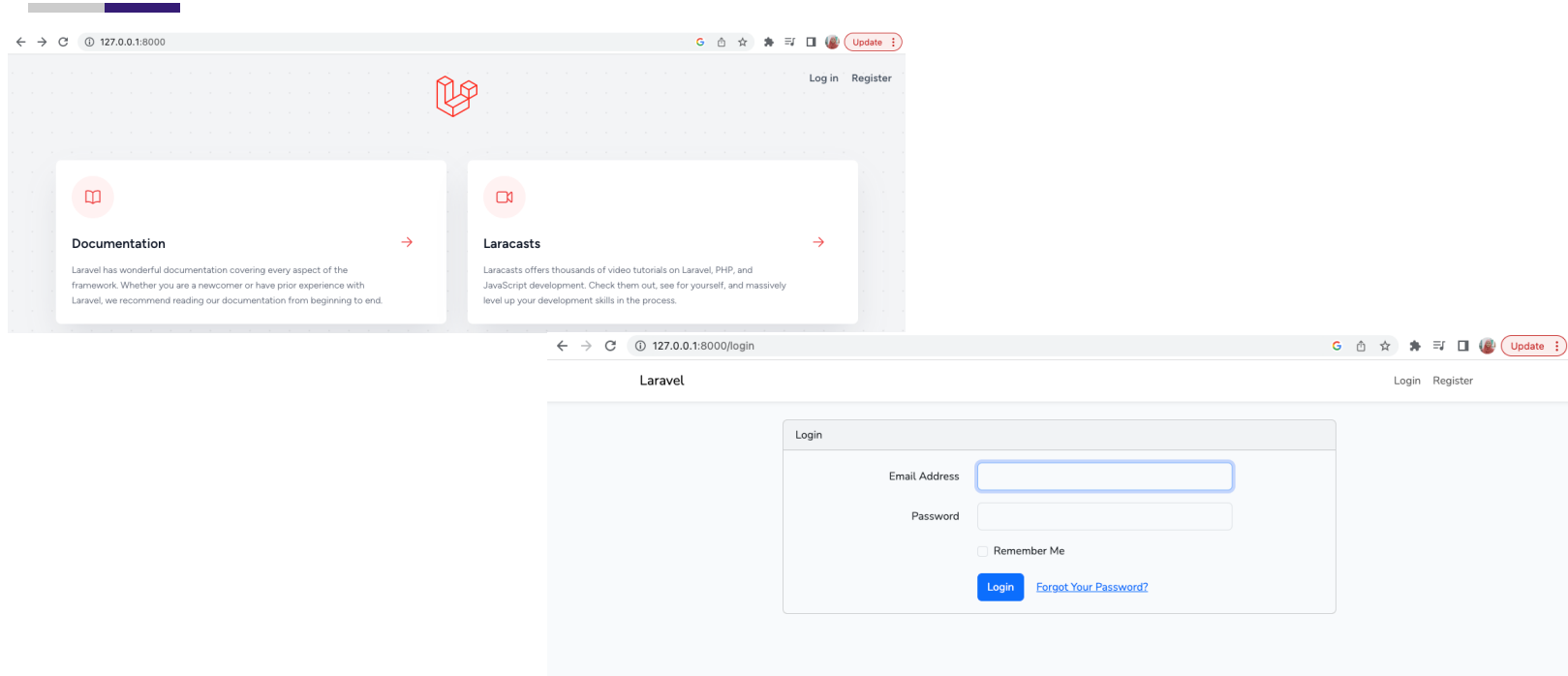
```
/**
 * Get the path the user should be redirected to.
 *
 * @param \Illuminate\Http\Request $request
 * @return string
 */
protected function redirectTo($request)
{
    return route('login');
}
```



# What Happen if Error? When click login/register?

- Check your vite installment
- Open terminal at your project:
- Run: `npm i`
- Run: `npm run dev`

# Successful Installment





# LIVE CODING SESSION

Silakan dicoba saat praktikum



# Manually Authenticating Users

---

## Manually Authenticating Users

- We will access Laravel's authentication services via the **Auth** facade, so we'll need to make sure to **import the Auth facade at the top of the class**.
- Next, let's check out the **attempt** method. The **attempt** method is normally used to handle authentication attempts from your application's **"login"** form.
- If authentication is successful, you should **regenerate** the user's **session** to prevent **session fixation**.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Http\RedirectResponse;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    /**
     * Handle an authentication attempt.
     */
    public function authenticate(Request $request): RedirectResponse
    {
        $credentials = $request->validate([
            'email' => ['required', 'email'],
            'password' => ['required'],
        ]);

        if (Auth::attempt($credentials)) {
            $request->session()->regenerate();

            return redirect()->intended('dashboard');
        }

        return back()->withErrors([
            'email' => 'The provided credentials do not match our records.',
        ]->onlyInput('email'));
    }
}
```

# Specifying Additional Conditions

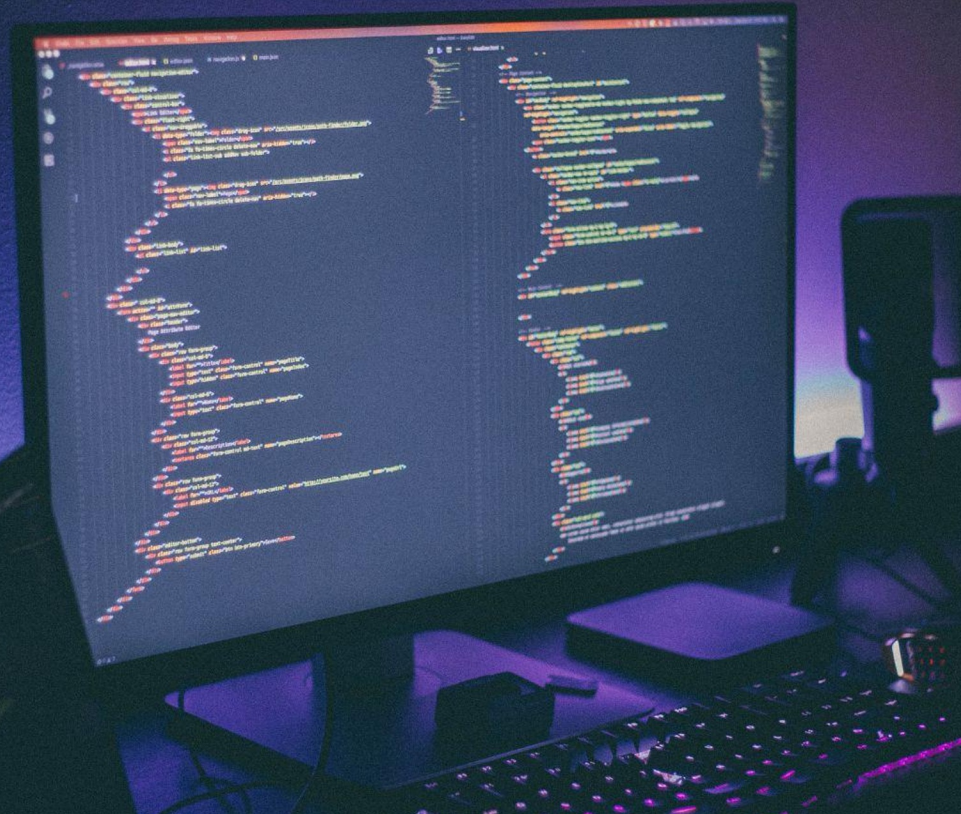
- If you wish, you may also **add extra query conditions** to the authentication query in addition to the user's email and password.
- To accomplish this, we may simply **add the query conditions to the array** passed to the **attempt** method.
- For example, we may verify that the user is marked as **"active"**.

```
if (Auth::attempt(['email' => $email, 'password' => $password, 'active' => 1])) {  
    // Authentication was successful...  
}
```



# LIVE CODING SESSION

Silakan dicoba saat praktikum



# Any Questions?

---

# Doa Memohon Kemudahan

Semoga Allah memudahkan proses belajar anda semua.

Dari Anas bin Malik, beliau berkata bahwa Nabi shallallahu 'alaihi wa sallam bersabda,

اللَّهُمَّ لَا سَهْلَ إِلَّا مَا جَعَلْتَهُ سَهْلًا وَأَنْتَ تَجْعَلُ  
الْحَزْنَ إِذَا شِئْتَ سَهْلًا

“Allahumma laa sahla illa maa ja’altahu sahlaa, wa anta taj’alul hazna idza syi’ta sahlaa”

[artinya: Ya Allah, tidak ada kemudahan kecuali yang Engkau buat mudah. Dan engkau menjadikan kesedihan (kesulitan), jika Engkau kehendaki pasti akan menjadi mudah].

Hadits ini dikeluarkan oleh Ibnu Hibban dalam Shahihnya (3/255). Dikeluarkan pula oleh Ibnu Abi ‘Umar, Ibnus Suni dalam ‘Amal Yaum wal Lailah. (Lihat Jaami’ul Ahadits, 6/257, Asy Syamilah)

Sumber: <https://rumaysho.com/1182-doa-memohon-kemudahan.html>