LAPORAN PRAKTIKUM MATA KULIAH

PEMROGRAMAN WEB

TUGAS 13 – Authentication

DISUSUN OLEH:

L0122068 – Gardasvara Mistortoify

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2024

BAB I

ANALISIS SOURCE CODE

1. AuthController.php

```php
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Session;

class AuthController extends Controller
{
    public function showRegisterForm()
    {
        return view('auth.register');
    }

    public function register(Request $request)
    {
        $request->validate([
            'name' => 'required|string|max:255',
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|min:8|confirmed',
        ]);

        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
        ]);

        return redirect()->route('login')->with('success', 'Registration successful. Please login.');
    }

    public function showLoginForm()
    {
        return view('auth.login');
    }

    public function login(Request $request)
    {
        $request->validate([
            'email' => 'required|string|email',
            'password' => 'required|string',
        ]);

        if (Auth::attempt($request->only('email', 'password'))) {
            $request->session()->regenerate();
            return redirect()->route('items.index');
        }

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

    public function logout(Request $request)
    {
        Auth::logout();
        $request->session()->invalidate();
        $request->session()->regenerateToken();
        return redirect()->route('login')->with('success', 'Logged out successfully.');
    }
}
```

Class Declaration dan Import

namespace App\Http\Controllers; dan beberapa use statements digunakan untuk mendefinisikan ruang lingkup dan memanggil fungsi atau class yang diperlukan seperti User, Request, Auth, Hash, dan Session.

Method showRegisterForm

showRegisterForm() mengembalikan tampilan halaman registrasi yang terletak di resources/views/auth/register.blade.php.

Method register

register(Request $request) digunakan untuk menangani data registrasi yang dikirimkan dari form.

validate() memvalidasi input dari pengguna berdasarkan aturan yang ditentukan.

User::create() membuat pengguna baru dan menyimpan data ke dalam database dengan hashing password menggunakan Hash::make().

Setelah registrasi berhasil, pengguna diarahkan ke halaman login dengan pesan sukses.

Method showLoginForm

showLoginForm() mengembalikan tampilan halaman login yang terletak di resources/views/auth/login.blade.php.

Method login

login(Request $request) digunakan untuk menangani data login yang dikirimkan dari form.

validate() memvalidasi input dari pengguna.

Auth::attempt() mencoba melakukan otentikasi dengan kredensial yang diberikan.

Jika otentikasi berhasil, sesi pengguna diregenerasi untuk keamanan dan pengguna diarahkan ke route items.index.

Jika otentikasi gagal, pengguna akan dikembalikan ke halaman login dengan pesan error.

Method logout

logout(Request $request) digunakan untuk menangani proses logout.

Auth::logout() melakukan logout pengguna.

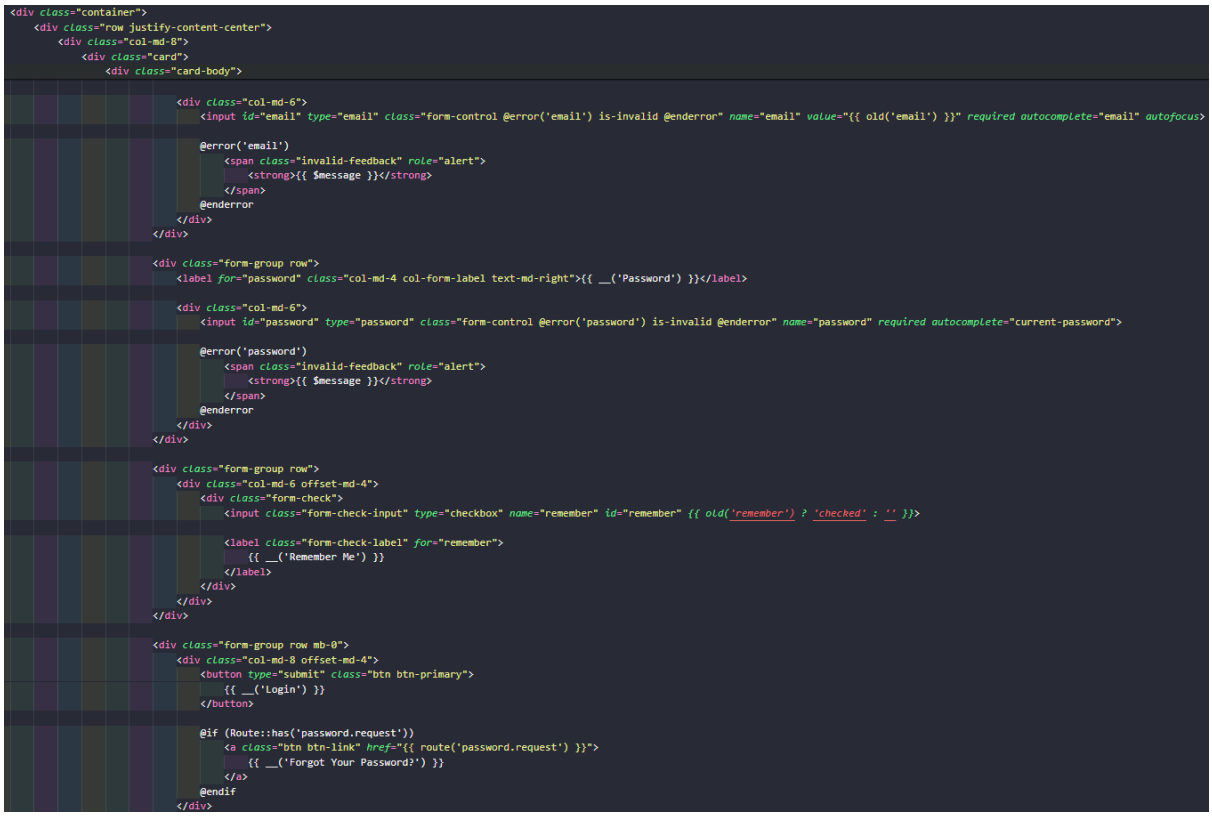
session()->invalidate() dan session()->regenerateToken() digunakan untuk menghapus sesi dan mencegah CSRF.

Setelah logout, pengguna diarahkan ke halaman login dengan pesan sukses.

Kesimpulan

AuthController ini menangani seluruh proses otentikasi pengguna mulai dari registrasi, login, hingga logout. Dengan menggunakan validasi, hashing password, dan manajemen sesi, controller ini memastikan bahwa proses otentikasi berjalan dengan aman dan sesuai dengan praktik terbaik yang dianjurkan oleh Laravel.

2. Auth/Login.blade.php



```
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-body">

                        <div class="col-md-6">
                            <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email" value="{{ old('email') }}" required autocomplete="email" autofocus>

                            @error('email')
                                <span class="invalid-feedback" role="alert">
                                    <strong>{{ $message }}</strong>
                                </span>
                            @enderror
                        </div>
                    </div>

                    <div class="form-group row">
                        <label for="password" class="col-md-4 col-form-label text-md-right">{{ __('Password') }}</label>

                        <div class="col-md-6">
                            <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password" required autocomplete="current-password">

                            @error('password')
                                <span class="invalid-feedback" role="alert">
                                    <strong>{{ $message }}</strong>
                                </span>
                            @enderror
                        </div>
                    </div>

                    <div class="form-group row">
                        <div class="col-md-6 offset-md-4">
                            <div class="form-check">
                                <input class="form-check-input" type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>

                                <label class="form-check-label" for="remember">
                                    {{ __('Remember Me') }}
                                </label>
                            </div>
                        </div>
                    </div>

                    <div class="form-group row mb-0">
                        <div class="col-md-8 offset-md-4">
                            <button type="submit" class="btn btn-primary">
                                {{ __('Login') }}
                            </button>

                            @if (Route::has('password.request'))
                                <a class="btn btn-link" href="{{ route('password.request') }}">
                                    {{ __('Forgot Your Password?') }}
                                </a>
                            @endif
                        </div>
```

Struktur Dasar

@extends('layouts.app'): Menggunakan layout utama dari aplikasi.

@section('content'): Menentukan bagian konten dari halaman yang akan diisi.

Container dan Card

div.container dan div.card: Membungkus konten dalam container yang diatur secara responsif.

div.card-header: Menampilkan header dengan teks "Login".

Handling Errors dan Session

@if ($errors->any()): Menampilkan pesan kesalahan jika ada error saat validasi form.

@if (session('success')): Menampilkan pesan sukses jika ada.

Formulir Login

form method="POST" action="{{ route('login') }}": Mengirimkan data form ke route login menggunakan metode POST.

@csrf: Token untuk melindungi dari serangan CSRF.

div.form-group row: Mengatur field form dalam baris.

Input Fields

label dan input: Menampilkan dan mengatur input untuk email dan password dengan validasi.

@error('email') dan @error('password'): Menampilkan pesan kesalahan validasi untuk email dan password.

Remember Me dan Submit Button

div.form-check: Mengatur checkbox untuk fitur "Remember Me".

button type="submit": Tombol untuk mengirimkan form login.

a.btn.btn-link: Link untuk mengarahkan pengguna ke halaman reset password jika mereka lupa kata sandi.

Penutupan

@endsection: Mengakhiri bagian konten dari halaman.

Kode ini mengatur tampilan halaman login yang terdiri dari form input untuk email, password, dan opsi remember me, serta handling untuk pesan error dan sukses.

3. Auth/Register.blade.php

```blade
<div class="card-body">
    <form method="POST" action="{{ route('register') }}">
        @csrf

        <div class="form-group row">
            <label for="name" class="col-md-4 col-form-label text-md-right">{{ __('Name') }}</label>

            <div class="col-md-6">
                <input id="name" type="text" class="form-control @error('name') is-invalid @enderror" name="name" value="{{ old('name') }}" required autocomplete="name" autofocus>

                @error('name')
                    <span class="invalid-feedback" role="alert">
                        <strong>{{ $message }}</strong>
                    </span>
                @enderror
            </div>
        </div>

        <div class="form-group row">
            <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>

            <div class="col-md-6">
                <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email" value="{{ old('email') }}" required autocomplete="email">

                @error('email')
                    <span class="invalid-feedback" role="alert">
                        <strong>{{ $message }}</strong>
                    </span>
                @enderror
            </div>
        </div>

        <div class="form-group row">
            <label for="password" class="col-md-4 col-form-label text-md-right">{{ __('Password') }}</label>

            <div class="col-md-6">
                <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password" required autocomplete="new-password">

                @error('password')
                    <span class="invalid-feedback" role="alert">
                        <strong>{{ $message }}</strong>
                    </span>
                @enderror
            </div>
        </div>

        <div class="form-group row">
            <label for="password-confirm" class="col-md-4 col-form-label text-md-right">{{ __('Confirm Password') }}</label>

            <div class="col-md-6">
                <input id="password-confirm" type="password" class="form-control" name="password_confirmation" required autocomplete="new-password">
            </div>
        </div>

        <div class="form-group row mb-0">
            <div class="col-md-6 offset-md-4">
                <button type="submit" class="btn btn-primary">
                    {{ __('Register') }}
                </button>
            </div>
        </div>
    </form>
```

Struktur Dasar

@extends('layouts.app'): Menggunakan layout utama dari aplikasi.

@section('content'): Menentukan bagian konten dari halaman yang akan diisi.

Container dan Card

div.container dan div.card: Membungkus konten dalam container yang diatur secara responsif.

div.card-header: Menampilkan header dengan teks "Register".

Formulir Register

form method="POST" action="{{ route('register') }}": Mengirimkan data form ke route register menggunakan metode POST.

@csrf: Token untuk melindungi dari serangan CSRF.

div.form-group row: Mengatur field form dalam baris.

Input Fields

label dan input untuk setiap field (name, email, password, password confirmation): Menampilkan dan mengatur input dengan validasi.

@error('field_name'): Menampilkan pesan kesalahan validasi untuk masing-masing field (name, email, password).

Submit Button

button type="submit": Tombol untuk mengirimkan form register.

Penutupan

@endsection: Mengakhiri bagian konten dari halaman.

Kode ini mengatur tampilan halaman register yang terdiri dari form input untuk nama, email, password, dan konfirmasi password, serta handling untuk pesan error. Kode ini memastikan validasi dan penyampaian pesan kesalahan jika ada input yang tidak valid.

4. layout/App.blade.php (setting Navbar)

```
            <ul class="navbar-nav ml-auto">
                @auth
                    <li class="nav-item">
                        <form method="POST" action="{{ route('logout') }}">
                            @csrf
                            <button type="submit" class="btn btn-link nav-link" style="color: white;">
                                <img src="{{ asset('storage/logout_icon.png') }}" alt="Log Out Icon" class="icon" style="width: 20px; height: 20px;">
                                Log Out
                            </button>
                        </form>
                    </li>
                @else
                    <li class="nav-item">
                        <a class="nav-link" href="{{ route('login') }}">Login</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ route('register') }}">Register</a>
                    </li>
                @endauth
            </ul>
        </div>
    </nav>
```

Bagian navbar ini memiliki mekanisme yang secara dinamis menampilkan elemen-elemen navigasi berdasarkan status autentikasi pengguna. Struktur ini dimulai dengan sebuah <ul> yang menggunakan kelas Bootstrap navbar-nav ml-auto, yang memastikan daftar navigasi ini ditampilkan di bagian kanan navbar.

Di dalam blok <ul>, terdapat direktif Blade @auth yang memeriksa apakah pengguna saat ini sudah login atau belum. Jika pengguna sudah login, maka bagian

kode di dalam blok @auth akan dijalankan. Di sini, sebuah item <li> yang berisi formulir logout ditampilkan. Formulir ini menggunakan metode POST dan mengarahkan ke rute logout. Dalam formulir ini, token CSRF disertakan untuk melindungi dari serangan CSRF.

Formulir tersebut berisi sebuah tombol submit yang tampilannya menyerupai link berkat penggunaan kelas btn btn-link nav-link. Tombol ini diberi ikon logout yang diambil dari penyimpanan lokal (storage/logout_icon.png) dan teks "Log Out". Ikon ini ditampilkan dalam ukuran 20x20 piksel.

Jika pengguna belum login, blok @else akan dijalankan, menampilkan dua item navigasi dalam <li> yang berbeda. Item pertama adalah sebuah link yang mengarahkan pengguna ke halaman login dengan teks "Login", sementara item kedua adalah link yang mengarahkan ke halaman register dengan teks "Register".

Akhirnya, direktif @endauth digunakan untuk menutup blok kondisi autentikasi. Dengan demikian, tampilan navbar ini secara otomatis menyesuaikan diri untuk menampilkan opsi logout bagi pengguna yang sudah login, dan menampilkan opsi login serta register bagi pengguna yang belum login.

5. Routes/Web.php

```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\MovieController;
use App\Http\Controllers\PageController;
use App\Http\Controllers\AuthController;
use Illuminate\Support\Facades\Auth;

Route::get('/', function () {
    if (Auth::check()) {
        return redirect()->route('home'); // Jika sudah login, arahkan ke halaman home
    } else {
        return redirect()->route('register'); // Jika belum login, arahkan ke halaman register
    }
});

// Authentication Routes
Route::get('/register', [AuthController::class, 'showRegisterForm'])->name('register');
Route::post('/register', [AuthController::class, 'register']);
Route::get('/login', [AuthController::class, 'showLoginForm'])->name('login');
Route::post('/login', [AuthController::class, 'login']);
Route::post('/logout', [AuthController::class, 'logout'])->name('logout');

// Protected Routes
Route::middleware(['auth'])->group(function () {
    Route::get('/home', [PageController::class, 'home'])->name('home');
    Route::resource('movies', MovieController::class);
});
```

Default Route

- Route::get('/', function () { ... }): Rute default yang mengarahkan pengguna berdasarkan status autentikasi mereka.
    - if (Auth::check()): Memeriksa apakah pengguna sudah login.
    - return redirect()->route('home'): Jika pengguna sudah login, arahkan ke halaman home.
    - else: Jika pengguna belum login, arahkan ke halaman register.

Authentication Routes

- Route::get('/register', [AuthController::class, 'showRegisterForm'])->name('register');
    - Menampilkan form registrasi menggunakan metode showRegisterForm dari AuthController.
    - Menggunakan nama rute register untuk kemudahan referensi.
- Route::post('/register', [AuthController::class, 'register']);
    - Mengirimkan data registrasi menggunakan metode register dari AuthController.
- Route::get('/login', [AuthController::class, 'showLoginForm'])->name('login');
    - Menampilkan form login menggunakan metode showLoginForm dari AuthController.
    - Menggunakan nama rute login untuk kemudahan referensi.
- Route::post('/login', [AuthController::class, 'login']);
    - Mengirimkan data login menggunakan metode login dari AuthController.
- Route::post('/logout', [AuthController::class, 'logout'])->name('logout');
    - Mengirimkan permintaan logout menggunakan metode logout dari AuthController.
    - Menggunakan nama rute logout untuk kemudahan referensi.

Protected Routes

- Route::middleware(['auth'])->group(function () { ... }): Mengelompokkan rute yang hanya dapat diakses oleh pengguna yang telah diautentikasi.
    - Route::get('/home', [PageController::class, 'home'])->name('home');

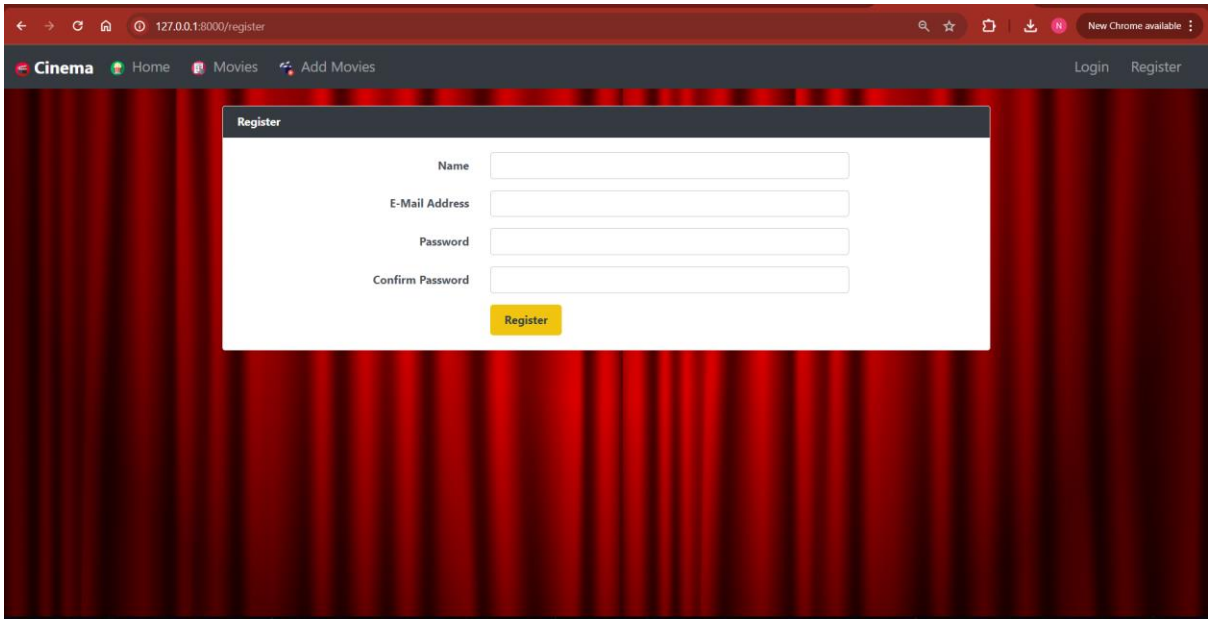
        - Mengarahkan ke halaman home menggunakan metode home dari PageController.
        - Menggunakan nama rute home untuk kemudahan referensi.
    - Route::resource('movies', MovieController::class);
        - Menggunakan resource controller MovieController untuk mengatur CRUD (Create, Read, Update, Delete) operasi pada model Movie.

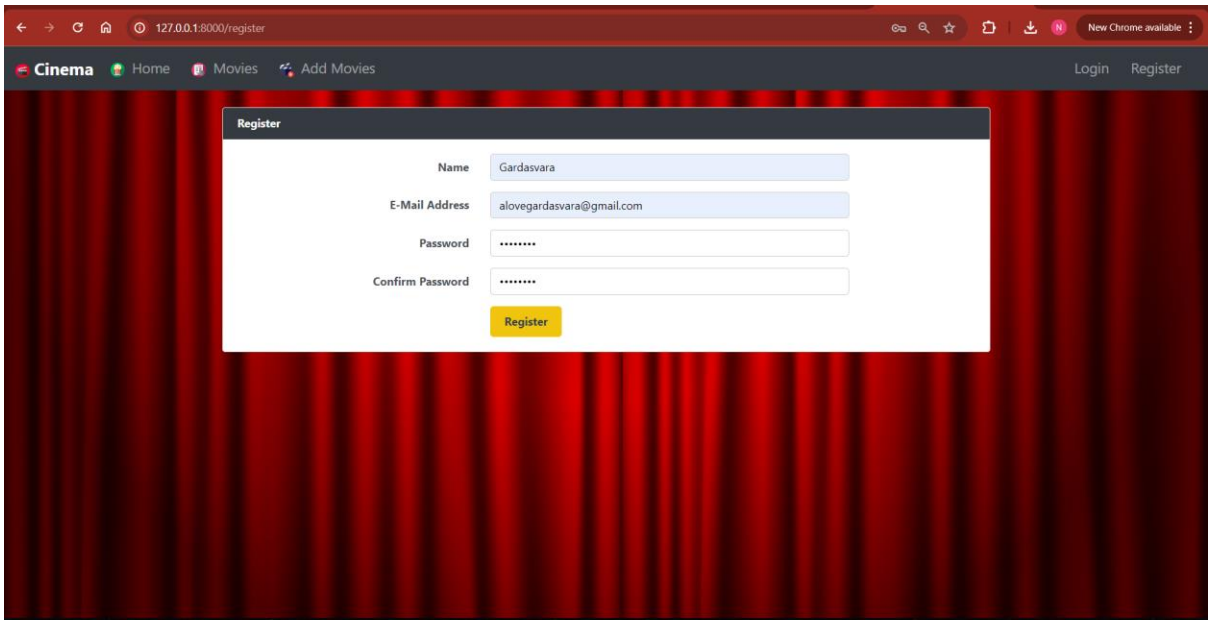
Kode ini mengatur rute untuk autentikasi (login, register, logout) serta rute yang dilindungi oleh middleware auth, memastikan hanya pengguna yang terautentikasi dapat mengakses halaman home dan resource movies.

BAB II

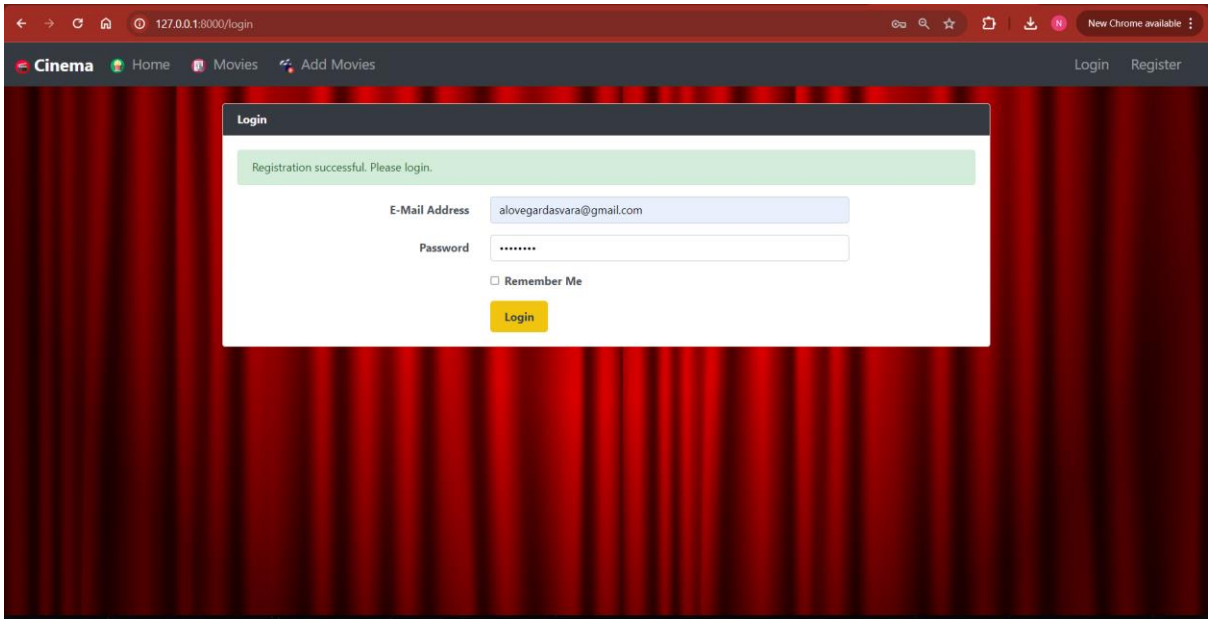
Analisis Output

Tampilan Saat pertama kali di run akan ke laman register





Tampilan Setelah berhasil register akan direct ke laman login



Setelah berhasil login maka user akan diarahkan ke halam utama

Jika ingin logout bisa mengklik yang ada pada navbar