Implementing Custom Authentication in Laravel 11

Step 1: Install Laravel

1. **Open Command Prompt or Windows Terminal**: Press **`Win + R`**, type **`cmd`**, and press **`Enter`**.
2. **Navigate to your web directory**:

```
cd path\to\your\directory
```

3. **Install Laravel via Composer**:

```
composer create-project laravel/laravel auth-custom-<your_account> --
prefer-dist
```

4. **Navigate to the Laravel project directory**:

```
cd auth-custom-<your account
```

Step 2: Set Up Database

1. **Configure `.env` file**: Open the **`.env`** file in your Laravel project directory and update the database configuration.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=authcustom-<your_account>
DB_USERNAME=your_username
DB_PASSWORD=your_password
```

Step 3: Run Migrations

1. **Run Migrations**:

```
php artisan migrate
```

Step 4: Create Authentication Routes and Controller

1. **Generate Authentication Controller**:

```
php artisan make:controller AuthController
```

2. **Define Routes in `web.php`**: Open the **`routes/web.php`** file and define routes for authentication.

```
use Illuminate\Support\Facades\Route;
```

```php
use App\Http\Controllers\AuthController;

Route::get('/', function () {
    return view('welcome');
});

Route::get('/register', [AuthController::class, 'showRegistrationForm'])->name('register');
Route::post('/register', [AuthController::class, 'register']);

Route::get('/login', [AuthController::class, 'showLoginForm'])->name('login');
Route::post('/login', [AuthController::class, 'login']);

Route::get('/logout', [AuthController::class, 'logout']);
Route::post('/logout', [AuthController::class, 'logout'])->name('logout');
```

Step 5: Implement Authentication Logic

1. **Update `AuthController`**: Implement methods for registration, login, and logout.

```php
<?php

namespace App\Http\Controllers;

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

use App\Models\User;

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

    public function register(Request $request)
    {

        // $email = 'user@domain.com';

        if($request->generate_email || ($request->email && $request->generate_email)){
             $email = fake()->unique()->safeEmail();
        } else {
```

```
            $email = $request->email;
        }

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

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

        return redirect()->route('login');
    }

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

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

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

            return redirect()->intended('/home');
        }

        return back()->withErrors([
            'username' => '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');
    }
}
```

Step 6: Create Views

1. **Create Blade Views for Registration and Login**:
    - Create **`resources/views/mainLayout.blade.php`**:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>@yield('page-title')</title>

    <style>
        * {
            font-family: calibri;
        }

        .auth-labels {
              display:inline-block;
              width: 8em;
        }

        .auth-textbox {
            /* display: inline-block; */
            margin-bottom: .5em;
        }
    </style>
</head>
<body>
    @if(!Auth::check())
       @yield('auth-content')
    @else
       @yield('page-content')
    @endif
</body>
</html>
```

    - Create **`resources/views/auth/register.blade.php`**:

```
@extends('mainLayout')

@section('page-title','Account Registration')

@section('auth-content')
    <form method="POST" action="{{ route('register') }}">
        @csrf
        <div>
            <label class="auth-labels">Username</label>
            <input type="text" name="name" value="{{ old('username') }}" required class="auth-
textbox">
            @error('name')
                <span>{{ $message }}</span>
            @enderror
        </div>
        <div>
            <label class="auth-labels">Email</label>
            <input type="email" name="email" class="auth-textbox">
            <input type="checkbox" name="generate_email">Generate
            @error('email')
                <span>{{ $message }}</span>
            @enderror
        </div>
        <div>
            <label class="auth-labels">Password</label>
            <input type="password" name="password" required class="auth-textbox">
            @error('password')
                <span>{{ $message }}</span>
            @enderror
        </div>
        <div>
            <label class="auth-labels">Confirm Password</label>
            <input type="password" name="password_confirmation" required class="auth-textbox">
        </div>
        <button type="submit">Register</button>
    </form>
@endsection
```

- Create **`resources/views/auth/login.blade.php`**:

```
@extends('mainLayout')

@section('page-title','Account Login')

@section('auth-content')
```

```
    <form method="POST" action="{{ route('login') }}">
        @csrf
        <div>
            <label class="auth-labels">Username</label>
            <input type="text" name="name" value="{{ old('username') }}" required class="auth-
textbox">
            @error('name')
                <span>{{ $message }}</span>
            @enderror
        </div>
        <div>
            <label class="auth-labels">Password</label>
            <input type="password" name="password" required class="auth-textbox">
            @error('password')
                <span>{{ $message }}</span>
            @enderror
        </div>
        <button type="submit">Login</button>
    </form>
@endsection
```

- Create **`resources/views/homepage.blade.php`**:

```
@extends('mainLayout')

@section('page-title','Main Landing Page')

@section('page-content')
<h1>Welcome to the Site</h1>
<br>
{{-- <a href="{{ route('logout') }}">Logout</a> --}}
<form action="{{ route('logout') }}">
    @csrf
    <button type="submit">
        Logout
    </button>
</form>
@endsection
```

Step 7: Test Authentication

1. **Start the Laravel Development Server**:

```
php artisan serve
```

2. **Open your web browser** and navigate to **`http://localhost:8000`**.

You should be able to register a new user at **`/register`**, log in at **`/login`**, and log out. Upon successful login, you will be redirected to **`/home`**. When you log out you will be redirected to the login page.

Step 8: Secure Routes

1. **Add Middleware to Secure Routes**:
    - In **`routes/web.php`**, secure routes using the **`auth`** middleware.

```
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\AuthController;

Route::get('/', function () {
    return view('welcome');
});

Route::get('/register', [AuthController::class, 'showRegistrationForm'])->name('register');
Route::post('/register', [AuthController::class, 'register']);

Route::get('/login', [AuthController::class, 'showLoginForm'])->name('login');
Route::post('/login', [AuthController::class, 'login']);

Route::get('/logout', [AuthController::class, 'logout']);
Route::post('/logout', [AuthController::class, 'logout'])->name('logout');

Route::middleware('auth')->group(function(){
    Route::view('/home','homepage');
});
```

This completes the custom authentication activity. Take note that you need to implement the elements needed for login and logout processes because you are no longer making use of Laravel's out-of-the-box authentication framework. This gives you more control over how your authentication will look or behave in your application.