

Lab 7: Authentication



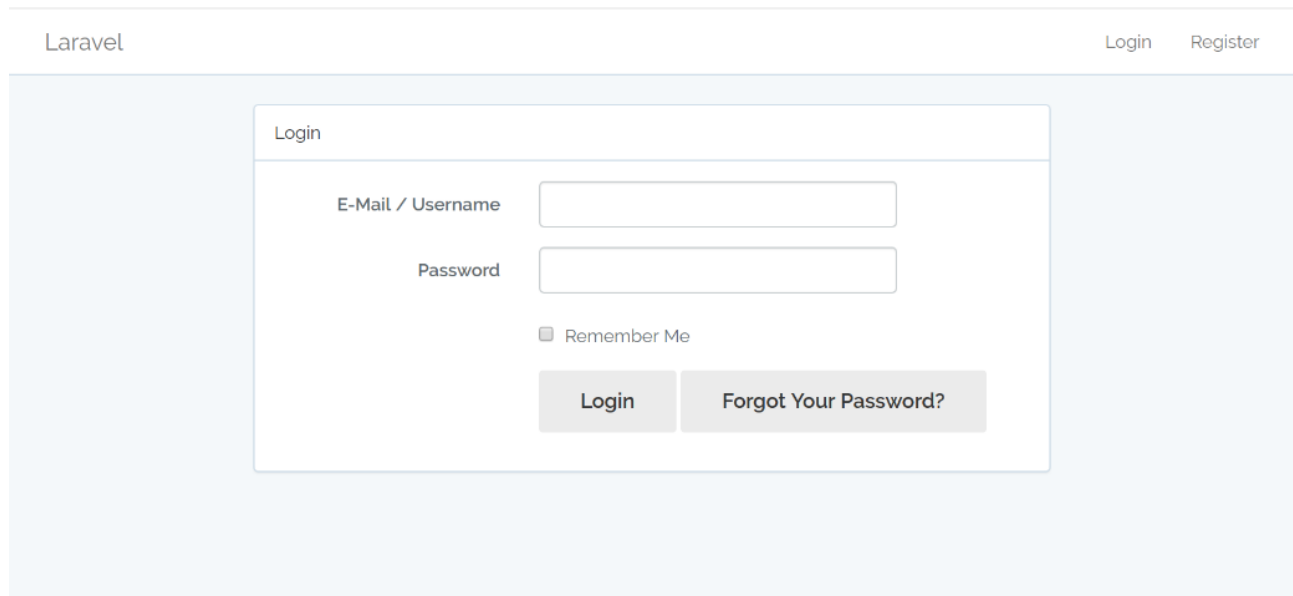
MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hiểu được Authentication
- ✓ Cấu hình và sử dụng Auth
- ✓ Xác thực đăng ký, đăng nhập, thay đổi mật khẩu

BÀI 1 (3 ĐIỂM)

Cấu hình Authentication trong Laravel cho phép login bằng userName hoặc Email dùng User model



HƯỚNG DẪN:

- ✓ Mở thư mục gốc của project và chạy : `php artisan make:auth`
- ✓ Nếu table User chưa tồn tại thì tạo: `php artisan make:migration create_users_table --create="users"`
- ✓ Thêm filed "Username" vào migration

FPT POLYTECHNIC

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('username')->unique();
            $table->string('email')->unique();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('users');
    }
}

```

- ✓ Thực thi migrate database: `php artisan migrate`
- ✓ Mở User Model thêm cột UserName vào mảng `$fillable`

```

protected $fillable = [
    'name', 'email', 'password', 'username',
];

```

FPT POLYTECHNIC

- ✓ Mở `RegisterController.php` thêm `UserName` vào phương thức `create()` và thêm validation cho phương thức `validator()`

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\User;
use Illuminate\Foundation\Auth\RegistersUsers;
use Illuminate\Support\Facades\Validator;

class RegisterController extends Controller
{
    use RegistersUsers;

    /**
     * Where to redirect users after registration.
     *
     * @var string
     */
    protected $redirectTo = '/home';

    /**
     * Create a new controller instance.
     *
     * @return void
     */

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

    /**
     * Get a validator for an incoming registration request.
     *
     * @param array $data
     * @return \Illuminate\Contracts\Validation\Validator
     */
}
```

```
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => 'required|string|max:255',
        'username' => 'required|string|max:20|unique:users',
        'email' => 'required|string|email|max:255|unique:users',
        'password' => 'required|string|min:6|confirmed',
    ]);
}

protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'username' => $data['username'],
        'email' => $data['email'],
        'password' => bcrypt($data['password']),
    ]);
}
}
```

- ✓ Chỉnh sửa tập tin resources/views/auth/register.blade.php nhằm hỗ trợ username và email

```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">Register</div>
                <div class="panel-body">
                    <form class="form-horizontal" method="POST" action="{{ route('register') }}">
                        {{ csrf_field() }}
                        <div class="form-group{{ $errors->has('name') ? ' has-error' : '' }}">
                            <label for="name" class="col-md-4 control-label">Name</label>
```

```

<div class="col-md-6">
    <input id="name" type="text" class="form-control" name="name" value="{{ old('name') }}" required autofocus>
    @if ($errors->has('name'))
        <span class="help-block">
            <strong>{{ $errors->first('name') }}</strong>
        </span>
    @endif
</div>
</div>
<div class="form-group{{ $errors->has('username') ? ' has-error' : '' }}">
    <label for="username" class="col-md-4 control-label">Username</label>
    <div class="col-md-6">
        <input id="username" type="text" class="form-control" name="username" value="{{ old('username') }}" required>
        @if ($errors->has('username'))
            <span class="help-block">
                <strong>{{ $errors->first('username') }}</strong>
            </span>
        @endif
    </div>
</div>
<div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
    <label for="email" class="col-md-4 control-label">E-Mail Address</label>
    <div class="col-md-6">
        <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}" required>
        @if ($errors->has('email'))
            <span class="help-block">
                <strong>{{ $errors->first('email') }}</strong>
            </span>
        @endif
    </div>
</div>
<div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
    <label for="password" class="col-md-4 control-label">Password</label>
    <div class="col-md-6">
        <input id="password" type="password" class="form-control" name="password" required>
        @if ($errors->has('password'))
            <span class="help-block">
                <strong>{{ $errors->first('password') }}</strong>
            </span>
        @endif
    </div>
</div>
<div class="form-group">
    <label for="password-confirm" class="col-md-4 control-label">Confirm Password</label>
    <div class="col-md-6">
        <input id="password-confirm" type="password" class="form-control" name="password_confirmation" required>
    </div>
</div>

```

```
<div class="form-group">  
  <div class="col-md-6 col-md-offset-4">  
    <button type="submit" class="btn btn-primary">  
      Register  
    </button>  
  </div>  
</div>
```

```
</div>  
</form>  
</div>  
</div>  
</div>  
</div>  
@endsection
```

Xem kết quả

Register

Name

Username

E-Mail Address

Password

Confirm Password

Register

BÀI 2 (2 ĐIỂM) : Dựa trên kết quả bài 1 làm tiếp phần login

- ✓ Tiến hành override phương thức `credentials()` trong `LoginController.php`

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Http\Request;

class LoginController extends Controller
{
```

```
use AuthenticatesUsers;

/**
 * Where to redirect users after login.
 *
 * @var string
 */
protected $redirectTo = '/home';

/**
 * Create a new controller instance.
 *
 * @return void
 */
```

```
public function __construct()
{
    $this->middleware('guest', ['except' => 'logout']);
}

/**
 * Get the needed authorization credentials from the request.
 *
 * @param \Illuminate\Http\Request $request
 * @return array
 */
```

FPT POLYTECHNIC

```
protected function credentials(Request $request)
{
    $field = filter_var($request->get($this->username()), FILTER_VALIDATE_EMAIL)
        ? $this->username()
        : 'username';

    return [
        $field => $request->get($this->username()),
        'password' => $request->password,
    ];
}
```

✓ Thêm input field email trong `login.blade.php`

```
<input type="text" class="form-control" name="email" placeholder="E-Mail / Username"
value="{{ old('email') }}" required autofocus>
```

BÀI 3 (3 ĐIỂM): Đăng nhập với attempt()

Vào Route/web.php tạo thêm routes tên logintest

```
Route::post('logintest', 'TestController@check');
```

Sử dụng login.blade.php của Laravel nhưng sẽ sửa action của form thành:

```
<form class="form-horizontal" role="form" method="POST" action="{{ url('/logintest') }}">
```

Tạo thêm một TestController để xử lý đăng nhập:


```
<?php

namespace App\Http\Controllers;

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

class TestController extends Controller
{
    public function check(Request $request)
    {
        $data=[
            'username'=>$request->username,
            'password'=>$request->password,
        ];
        if(Auth::attempt($data)){
            //true
        }else{
            //false
        }
    }
}
```

Chú ý sử dụng namespace :Illuminate\Support\Facades\Auth; cho việc authentication

Sinh viên bổ sung các hành động xảy ra vào phương thức Auth::attempt() khi chứng thực thành công hoặc thất bại

BÀI 4 (2 ĐIỂM):

Giảng viên cho thêm