

### 1. Lý thuyết Cốt lõi (Khái niệm)

- **Framework là gì?** Laravel là một PHP Framework. Hãy tưởng tượng nó là một "bộ khung" xe đã được lắp ráp sẵn (bánh xe, động cơ, khung sườn). Thay vì tự xây dựng mọi thứ từ đầu (như PHT 2-5), bạn chỉ cần tập trung vào việc "thêm" các tính năng (ghế da, màu sơn) cho chiếc xe. Nó cung cấp các quy tắc và công cụ để làm việc nhanh hơn và an toàn hơn.
- **Composer:** Là công cụ quản lý thư viện (dependency manager) cho PHP. Nó giúp bạn cài đặt Laravel và các thư viện khác (ví dụ: thư viện xử lý thanh toán) chỉ bằng một dòng lệnh.
- **Artisan:** Là "con dao đa năng" của Laravel. Đây là công cụ dòng lệnh (command-line) của Laravel. Thay vì tạo file SinhVienModel.php bằng tay, bạn chỉ cần gõ php artisan make:model SinhVien.
- **Routing (Định tuyến) (6.5):** Đây là "bảng chỉ dẫn" của ứng dụng.
  - Nó nằm trong file routes/web.php.
  - Nó định nghĩa: "Khi người dùng truy cập URL /gioi-thieu thì phải chạy code nào?".
- **Controller (Điều khiển) (6.5):** Đây là "bộ não" xử lý (giống hệt Controller ở Chương 5).
  - Nó nằm trong thư mục app/Http/Controllers/.
  - Đây là nơi chứa code logic (ví dụ: function showGioiThieu() { ... }).

Luồng hoạt động mới:

Người dùng truy cập URL \$rightarrow\$ Laravel kiểm tra routes/web.php \$rightarrow\$ Route gọi đến TenController@tenPhuongThuc \$rightarrow\$ Controller xử lý logic và trả về kết quả.

### 2. Nhiệm vụ Thực hành (BẮT BUỘC)

**Kịch bản:** Cài đặt Laravel, tạo dự án đầu tiên, và xây dựng luồng "Route \$rightarrow\$ Controller" cơ bản.

Code Khởi đầu (Starter Code):

Không có code, vì chúng ta sẽ dùng công cụ dòng lệnh (Terminal/CMD/GitBash).

---

#### A. Cài đặt (6.2, 6.3)

1. // **TODO 1:** Cài đặt Composer (Nếu bạn chưa cài): Truy cập [getcomposer.org](http://getcomposer.org) và cài đặt nó.
2. // **TODO 2:** Mở Terminal (CMD hoặc Git Bash), cd vào thư mục htdocs của XAMPP.
3. // **TODO 3:** Chạy lệnh sau để **tạo dự án Laravel** (đặt tên là cse485\_chapter6). Lệnh này sẽ tự động tải về hàng trăm file cần thiết:

Bash

```
composer create-project --prefer-dist laravel/laravel cse485_chapter6
```

4. // TODO 4: Di chuyển vào thư mục dự án vừa tạo:

Bash

```
cd cse485_chapter6
```

5. // TODO 5: Khởi động server "nhúng" của Laravel (không cần XAMPP Apache):

Bash

```
php artisan serve
```

(Nếu chạy thành công, nó sẽ báo server đang chạy ở <http://127.0.0.1:8000>)

---

## B. Xây dựng Route và Controller (6.4, 6.5)

Mở một Terminal/CMD thứ hai (giữ nguyên Terminal ở TODO 5) và cd vào thư mục cse485\_chapter6.

1. // TODO 6: Dùng Artisan để **tạo một Controller mới** tên là PageController:

Bash

```
php artisan make:controller PageController
```

2. // TODO 7: Mở dự án bằng VSCode. Mở tệp Controller vừa tạo tại:

app/Http/Controllers/PageController.php

3. // TODO 8: Bên trong class PageController, thêm một phương thức (method) mới tên là showHomepage():

PHP

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PageController extends Controller
{
    //TODO 8: Thêm phương thức này
    public function showHomepage()
    {
        // TODO 9: Thay vì echo, chúng ta 'return'
        return "Chào mừng bạn đến với PHT Chffóng 6 - Laravel!";
    }
}
```

```
}
```

4. **// TODO 10:** Mở tệp "Routing" tại routes/web.php.

5. **// TODO 11:** Import PageController của bạn ở đầu tệp:

PHP

```
use App\Http\Controllers\PageController;
```

6. **// TODO 12:** Xóa Route Route::get('/', ...); mặc định. Thay vào đó, thêm 2 Route mới:

- o Một Route cho URL / (trang chủ).
- o Một Route cho URL /about.
- o Cả hai đều trỏ đến PageController@showHomepage (chúng ta sẽ dùng chung 1 hàm cho PHT này)

PHP

```
// ... (sau dòng use)
```

```
// TODO 12: Thêm 2 route này
```

```
Route::get('/', [PageController::class, 'showHomepage']); Route::get('/about',
```

```
PageController::class, 'showAbout');
```

### 3. Yêu cầu Bằng chứng (Proof of Work)

Bạn phải nộp lại 3 bằng chứng sau:

#### A. Code đã hoàn thiện:

1. Dán (paste) toàn bộ code của tệp app/Http/Controllers/PageController.php.

```
<?php  
  
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class PageController extends Controller  
{  
    // TODO 8: Thêm phương thức showHomepage  
    public function showHomepage()  
    {  
        // TODO 9: Return chuỗi thay vì echo  
        return "Chào mừng bạn đến với PHT Chương 6 - Laravel!";  
    }  
}
```

2.

3. Dán (paste) toàn bộ code của tệp routes/web.php.

```
<?php  
  
use Illuminate\Support\Facades\Route;
```

```
use App\Http\Controllers\PageController;
```

```
// Trang chủ  
Route::get('/', [PageController::class, 'showHomepage']);
```

```
// Trang giới thiệu  
Route::get('/about', [PageController::class, 'showHomepage']);
```

4.

### B. Ảnh chụp màn hình Kết quả (3 ẢNH):

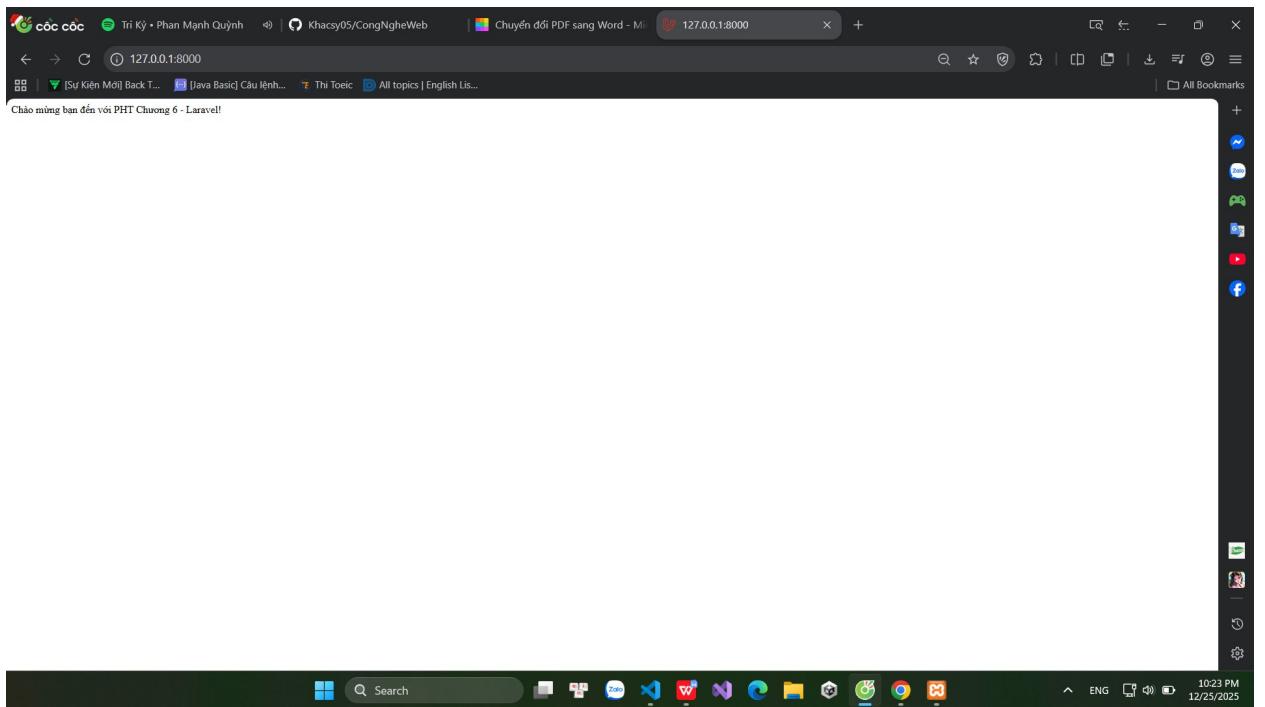
1. **Ảnh 1 (Terminal):** Chụp màn hình Terminal sau khi chạy lệnh php artisan make:controller PageController thành công.

```
Microsoft Windows [Version 10.0.26100.7171]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\xampp\htdocs\cse485_chapter6>php artisan make:controller PageController  
  
INFO Controller [C:\xampp\htdocs\cse485_chapter6\app\Http\Controllers\PageController.php] created successfully.  
  
C:\xampp\htdocs\cse485_chapter6>
```

2. 3. **Ảnh 2 (Terminal):** Chụp màn hình Terminal đang chạy lệnh php artisan serve.

```
C:\xampp\htdocs\cse485_chapter6>php artisan serve  
  
INFO Server running on [http://127.0.0.1:8000].  
  
Press Ctrl+C to stop the server
```

4. 5. **Ảnh 3 (Trình duyệt Web):** Chụp ảnh màn hình trình duyệt truy cập vào <http://127.0.0.1:8000/> (phải thấy thông điệp chào mừng bạn đã return ở TODO 9).



6.

(Dán Code A1, A2 và 3 Ảnh B1, B2, B3 của bạn vào đây)

#### 4. Câu hỏi Phản biện (Bắt buộc)

Sau khi hoàn thành Phần 2 C 3, hãy đặt 01 câu hỏi tư duy.

(Gợi ý: "Hãy so sánh file index.php (Controller) ở PHT Chương 5 và file routes/web.php ở PHT này. Tại sao Laravel lại tách 'bảng chỉ dẫn' (Routing) ra khỏi 'bộ não' (Controller)? Lợi ích của việc này là gì?").

**Câu hỏi của tôi là:** Một phương thức trong Controller của Laravel có thể được nhiều Route khác nhau sử dụng.

Cách thiết kế này khác gì so với mô hình mỗi trang là một file PHP riêng biệt và nó góp phần nâng cao khả năng tái sử dụng mã nguồn như thế nào?

## 5. Kết nối Đánh giá (Rất quan trọng)

Kỹ năng sử dụng **Artisan**, định nghĩa **Route**, và tạo **Controller** là 3 kỹ năng **bắt buộc** để bắt đầu làm việc với Laravel.

Đây là nền tảng cho **Bài tập trên lớp (Phần Laravel)**<sup>2</sup> (chiếm 20%, dự kiến Tuần 7) và là cấu trúc xương sống của toàn bộ **Bài tập lớn theo nhóm** (chiếm 50%)<sup>3</sup>. Nếu bạn không thể làm PHT này, bạn sẽ không thể làm được các PHT tiếp theo (Chương 7, 8).