

A. 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.

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with project files like 'PageController.php', 'routes/web.php', and 'web.php'. The main area shows a code editor with the following PHP code:

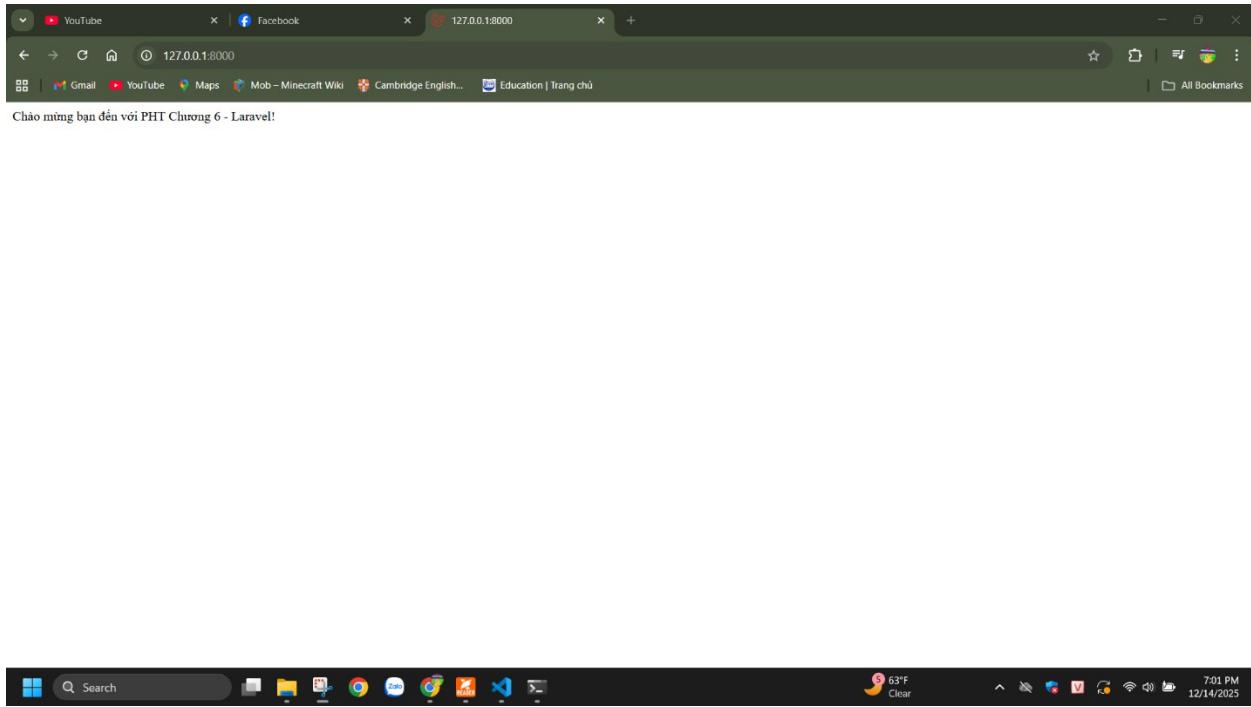
```
routes > routes/web.php
1 <?php
2
3 use App\Http\Controllers\PageController;
4
5 Route::get('/', [PageController::class, 'showHomePage']);
```

Below the code editor is the Terminal tab, which displays the command 'php artisan make:controller PageController' and its output: 'INFO Controller [E:\xampp\htdocs\cse485_chapter6\app\Http\Controllers\PageController.php] created successfully.' The status bar at the bottom shows the file path 'E:\xampp\htdocs\cse485_chapter6' and the date '12/14/2025'.

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

The screenshot shows a Windows PowerShell window titled 'Windows PowerShell'. The command 'cd E:\xampp\htdocs\cse485_chapter6' is run, followed by 'php artisan serve'. The output shows the server is running on 'http://127.0.0.1:8000'. Subsequent log entries show requests for favicon.ico with response times ranging from ~0.13ms to ~584.16ms. The status bar at the bottom shows the file path 'CSE485-CNW-Chuong6-PHT.pdf - Foxit PDF Reader' and the date '12/14/2025'.

3. 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).



B. Câu hỏi Phản biện (Bắt buộc)

Câu hỏi 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 trả lời:

- Cách mà file index.php routing nó sẽ như thế này:

```
if ($_GET['page'] == 'login') {  
  
    loginController();  
  
} elseif ($_GET['page'] == 'register') {  
  
    registerController();  
  
}
```

--> Tạo các điều hướng if-else qua biến truyền URL biến \$_GET['page'];

- **Vấn đề:** File to, nhiều if-else, khó đọc, Thêm chức năng mới → sửa trực tiếp Controller, Khó bảo trì, khó mở rộng, Trộn lẫn trách nhiệm.

- Laravel thì routing qua web.php như này:

```
Route::get('/login', [AuthController::class, 'login']);
```

```
Route::post('/register', [AuthController::class, 'register']);
```

- Nhận xét: URL nào --> đi tới Controller nào, hàm nào. Tuân thủ nguyên tắc Single Responsibility chỉ một việc làm, chỉ có trách nhiệm điều hướng.

- *Lợi ích khác của Laravel tách route riêng: Code sạch, ít if-else, hỗ trợ middleware, bảo mật, phân quyền, phù hợp dự án lớn.*

Câu hỏi của tôi: Nhưng tôi thấy nếu dung index cũng là điều hướng và có động vào controller nếu dung if, else đâu? Như cái ví dụ thể hiện routing của index.php chẳng qua chỉ là truyền vào biến GET tôi thấy nó cũng đơn giản mà hay trong bài thực hành số 2 tôi có điều hướng đơn giản bằng base_uri đc chuẩn hóa qua htaccess + switch, case ?

Câu trả lời:

- index.php KHÔNG hề vi phạm Single Responsibility. Không phải vì index.php *làm sai* Mà vì index.php không nên là nơi con người đọc - chỉnh route.
--> Nó đang làm Front Controller + Router, chứ không làm Controller.

- **Sự khác biệt “then chốt” nằm ở quy mô và vòng đời dự án:**

Khi dự án nhỏ: Điều trên nói đúng hoàn toàn, \$_GET['url'], switch, gọi Controller --> **Đủ xài, dễ hiểu, không thừa**

Khi dự án lớn dần: Những thứ bắt đầu xuất hiện: Route theo **HTTP method**, Route có **tham số**, Route cần **middleware**, Route nhóm theo **module**, Route cần **cache**, Route cho **API, web, admin**.

Câu hỏi của tôi: Vậy front controller + route truyền thống khác gì chỉ route của Laravel?

Câu trả lời: Laravel nó giống như trỏ đường, còn PHP qua index giống như ghi hết luật. Vấn đề này em cảm thấy khá trừu tượng nên mong thầy giải thích thêm về lợi ích của chỉ route khác gì front controller + route?