

PHIẾU HỌC TẬP CHỦ ĐỘNG

Môn học: CSE485: Công nghệ Web

Họ và tên: Nguyễn Thị Yên – MSV: 2351160567

CHƯƠNG 7: TẠO KHUÔN MẪU VỚI BLADE

1. Lý thuyết Cốt lõi (Khái niệm)

Blade là "chữ V" (View) trong mô hình MVC của Laravel. Nó là các tệp `.blade.php` nằm trong thư mục `resources/views/`.

Tại sao dùng Blade thay vì echo PHP thuần (như PHT Chương 5)?

1. **Cú pháp sạch sẽ:** Dùng `{{ $variable }}` thay vì `<?php echo $variable; ?>`.
2. **Bảo mật (XSS):** `{{ ... }}` tự động lọc dữ liệu (giống `htmlspecialchars`) để chống tấn công XSS.
3. **Kế thừa Layout (Layout Inheritance):** Đây là sức mạnh lớn nhất. Bạn tạo 1 "layout" (khung) chung (như menu, footer) và các trang con (như `trang_chu`, `lien_he`) chỉ cần "nhét" nội dung của mình vào khung đó.

Cú pháp Blade quan trọng:

- `{{ $ten_bien }}`: In ra dữ liệu (an toàn).
- `{!! $ten_bien_chua_html !!}`: In ra HTML (không an toàn, cần thận khi dùng).
- `@if(...)` / `@elseif(...)` / `@else` / `@endif`: Cấu trúc điều khiển.
- `@foreach($mang as $bien)` / `@endforeach`: Vòng lặp.
- `@extends('ten_layout')`: Báo cho view này biết nó "kế thừa" từ 1 file layout.
- `@yield('ten_vung')`: Trong file layout, đây là "lỗ hổng" (placeholder) để các view con nhét nội dung vào.
- `@section('ten_vung')` ... `@endsection`: Trong file view con, đây là khối nội dung sẽ được "nhét" vào `@yield` tương ứng.

2. Nhiệm vụ Thực hành (BẮT BUỘC)

Kịch bản: Chúng ta sẽ nâng cấp PHT Chương 6. Thay vì PageController trả về một chuỗi "Chào mừng...", chúng ta sẽ:

1. Tạo 1 file **layout** chung (`app.blade.php`) chứa `<html>`, `<head>`, `<body>`.
2. Tạo 1 file **view** trang chủ (`homepage.blade.php`) kế thừa layout đó.

3. Sửa PageController để trả về view homepage và **truyền dữ liệu** (như tiêu đề trang, mô tả) cho view đó.

Cấu trúc thư mục (Bắt buộc): Dự án cse485_chapter6 của bạn sẽ có 2 file view mới trong resources/views/:

/resources/views/

|-- layouts/

| |-- app.blade.php (Layout chính - Tập 1)

|-- homepage.blade.php (View trang chủ - Tập 2)

Code Khởi đầu (Starter Code):

Tập 1: resources/views/layouts/app.blade.php (Tạo file MỚI)

HTML

```
<!DOCTYPE html>
<html lang="vi">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{{ $title ?? 'Website Của Tôi' }}</title>
  <style>    body { font-family: sans-
serif; }
    .container { max-width: 960px; margin: 0 auto; padding: 20px; }
    header, footer { background-color: #f4f4f4; padding: 10px; text-align:
center; }    nav { background-color: #333; color: white; padding: 10px; }
nav a { color: white; margin: 0 10px; }
  </style>
</head>
<body>
  <header>
    <h1>Trang Web CSE485 - Chương 7</h1>
  </header>
  <nav>
    <a href="/">Trang Chủ</a>
    <a href="/about">Giới Thiệu</a>
  </nav>
  <div class="container">
    @yield('content')
  </div>
  <footer>
```

```
<p>&copy; 2025 - Khoa CNTT - Trường Đại học Thủy Lợi</p>
</footer>
</body>
</html>
```

Tập 2: resources/views/homepage.blade.php (Tạo file MỚI)

PHP

```
<?php // Xóa dòng này đi, file Blade không bắt đầu bằng
```

```
<?php ?>
```

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

```
@section('content')
```

```
<h2>{{ $page_title }}</h2>
```

```
<p>{{ $page_description }}</p>
```

```
<h3>Danh sách công việc (Lấy từ Controller):</h3>
```

```
<ul>
```

```
@foreach($tasks as $task)
```

```
<li>{{ $task }}</li>
```

```
@endforeach
```

```
</ul>
```

```
@endsection
```

Tập 3: app/Http/Controllers/PageController.php (SỬA file cũ)

PHP

```
<?php namespace
```

```
App\Http\Controllers;
```

```
use
```

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

```
class PageController extends Controller
{
    // Sửa phương thức showHomepage() của Chương 6
    public function showHomepage()
    {
        // TODO 7: Thay vì "return 'Chào mừng...'", chúng ta sẽ
        // chuẩn bị dữ liệu để truyền cho View

        $viewTitle = 'PHT Chương 7 - Blade Template';
        $pageTitle = 'Chào mừng bạn đến với Blade!';
        $pageDescription = 'Đây là trang chủ được render bằng Blade Template Engine.';

        $tasks = [
            'Cài đặt Laravel',
            'Hiểu về Routing & Controller',
            'Tạo Layout với Blade',
            'Truyền dữ liệu cho View'
        ];

        // TODO 8: Trả về một View
        // Gợi ý: Dùng hàm view('ten_view', $data_array)
        // 'homepage' tương đương 'homepage.blade.php'
        return
        view('homepage', [
            'title' => $viewTitle,
```

```
'page_title' => $pageTitle,  
'page_description' => $pageDescription,  
'tasks' => $tasks  
]);  
  
// TODO 9: (Cách khác) Dùng hàm compact() cho gọn  
// return view('homepage', compact('viewTitle', 'pageTitle',  
'pageDescription', 'tasks'));  
  
// (Lưu ý: khi dùng compact, tên biến ở Controller và View phải khớp  
nhau)
```

```
}  
}
```

3. Yêu cầu Bằng chứng (Proof of Work)

Bạn phải nộp lại 2 bằng chứng sau:

A. Code đã hoàn thiện: Dán (paste) toàn bộ code của 3 tệp bạn đã tạo/sửa:

1 resources/views/layouts/app.blade.php

```
<!DOCTYPE html>  
<html lang="vi">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>{{ $title ?? 'Website Của Tôi' }}</title>  
<style>  
  body {  
    font-family: sans-serif;  
  }  
  .container {  
    max-width: 960px;  
    margin: 0 auto;  
    padding: 20px;  
  }  
  header, footer {  
    background-color: #f4f4f4;  
    padding: 10px;
```

```
        text-align: center; }
nav {
    background-color: #333;
    color: white;
    padding: 10px;
    text-align: center;
}
nav a {
    color: white;
    margin: 0 10px;
    padding: 10px;
    text-decoration: none;
}
</style>
</head>
<body>
    <header>
        <h1>Trang Web CSE485 - Chương 7</h1>
    </header>
    <nav>
        <a href="/">Trang Chủ</a>
        <a href="/about">Giới Thiệu</a>
    </nav>
    <div class="container">
        @yield('content')
```

```
</div>

<footer>

    <p>&copy; 2025 - Khoa CNTT - Trường Đại học Thủy Lợi</p>

</footer>

</body>

</html>
```

2 resources/views/homepage.blade.php

```
<!-- báo cho view này biết nó kế thừa từ layout nào -->
@extends('layouts.app')

<!-- tên vùng nội dung để chèn vào layout -->
@section('content')

<!-- Tên biến trong view và controller phải giống nhau -->
    <h2>{{ $page_title }}</h2>
    <p>{{ $page_description }}</p>

    <h3>Danh sách công việc (Lấy từ Controller):</h3>
    <ul>
        @foreach($tasks as $task)
            <li>{{ $task }}</li>
        @endforeach
    </ul>

    @endsection
```

3 app/Http/Controllers/PageController.php

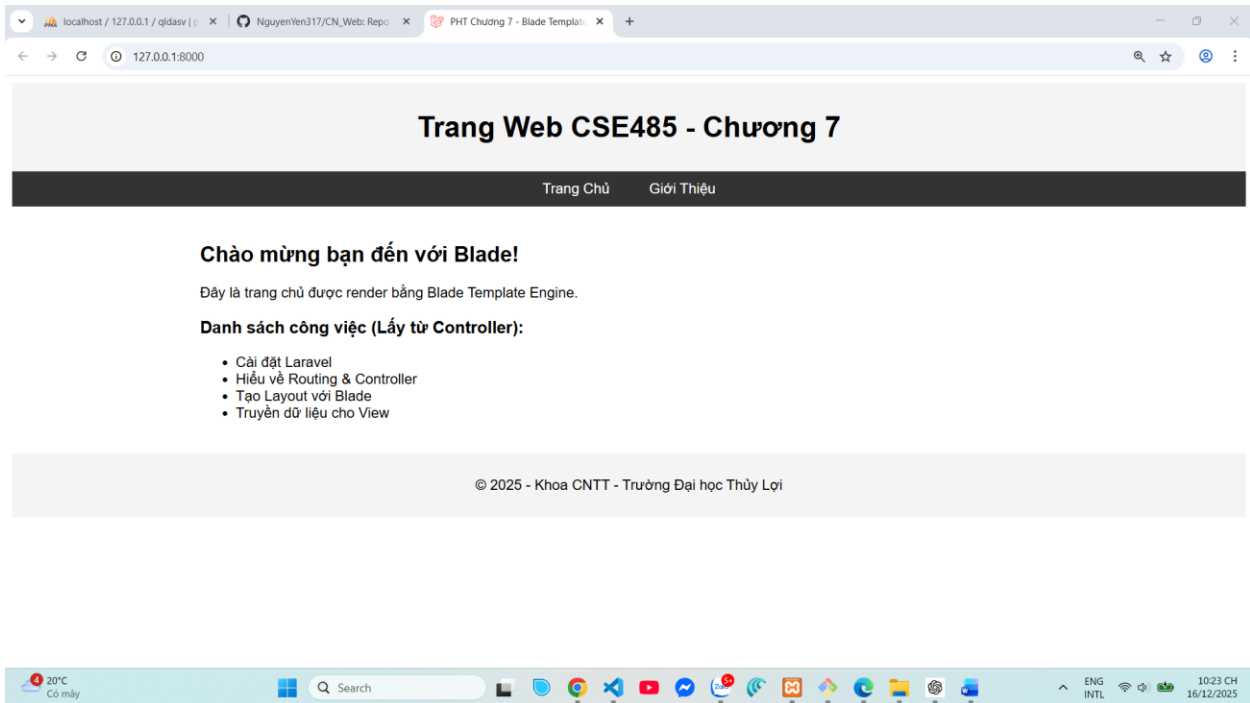
```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class PageController extends Controller
{
    // Sửa phương thức showHomepage() của Chương 6
    public function showHomepage()
    {
        // TODO 7: Thay vì "return 'Chào mừng...'", chúng ta sẽ
        // chuẩn bị dữ liệu để truyền cho View
        $viewTitle = 'PHT Chương 7 - Blade Template';
        $pageTitle = 'Chào mừng bạn đến với Blade!';
        $pageDescription = 'Đây là trang chủ được render bằng Blade Template Engine.';
        $tasks = [
            'Cài đặt Laravel',
            'Hiểu về Routing & Controller',
            'Tạo Layout với Blade',
            'Truyền dữ liệu cho View'
        ];
        // TODO 8: Trả về một View
        // Gợi ý: Dùng hàm view('ten_view', $data_array)
        // 'homepage' tương đương 'homepage.blade.php'
```

```
return view('homepage', [  
    'title' => $viewTitle,  
    'page_title' => $pageTitle,  
    'page_description' => $pageDescription,  
    'tasks' => $tasks  
]);
```

// TODO 9: (Cách khác) Dùng hàm compact() cho gọn, biến truyền sang cho view phải trùng tên

```
//return view('homepage', compact('viewTitle', 'pageTitle',  
'pageDescription', 'tasks'));  
  
// (Lưu ý: khi dùng compact, tên biến ở Controller và View phải khớp  
nhau)  
}  
}
```

B.Ảnh chụp màn hình Kết quả (Trình duyệt Web): Chạy php artisan serve và truy cập <http://127.0.0.1:8000/>. Chụp ảnh màn hình trình duyệt hiển thị kết quả. (Phải thấy rõ layout chung (header/footer) và phần nội dung (danh sách công việc) được nạp ở giữa).



4. Câu hỏi Phản biện (Bắt buộc)

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


(Gợi ý: "Hãy giải thích chính xác vai trò của `@yield('content')` (trong file `app.blade.php`) và `@section('content')` (trong file `homepage.blade.php`). Chúng làm việc cùng nhau như thế nào để ghép thành file HTML cuối cùng mà trình duyệt nhận được?").

Câu hỏi của tôi là:

1. Vì sao Laravel không cho bạn viết thẳng HTML trong Controller?
2. Vì sao: `{{ $content }}` an toàn hơn `<?php echo $content; ?>`
3. Điều gì xảy ra nếu: Layout có `@yield('content')` nhưng View con **không có** `@section('content')`? Và ngược lại?
4. Có thể `@extends` nhiều layout trong 1 view không? Vì sao?
5. `@yield` và `@section` khác nhau ở vai trò như thế nào?

5. Kết nối Đánh giá (Rất quan trọng)

Kỹ năng sử dụng **Blade** (tách layout, truyền dữ liệu) là kỹ năng **bắt buộc** để xây dựng giao diện người dùng (Frontend) trong Laravel.



Bạn sẽ vận dụng trực tiếp PHT này để hoàn thành **Bài tập trên lớp (Phần Laravel)**, chiếm 20% tổng điểm, dự kiến vào Tuần 7. Nó cũng là phần "V" (View) không thể thiếu trong **Bài tập lớn theo nhóm (50%)**.