**Chương 6**

**A.CODE**

**1. app/Http/Controllers/PageController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PageController extends Controller

{

    public function showHomepage()

    {

        return "Chào mừng bạn đến với PHT Chương 6 - Laravel!";

    }

}

**2. routes/web.php**

<?php

use Illuminate\Support\Facades\Route;

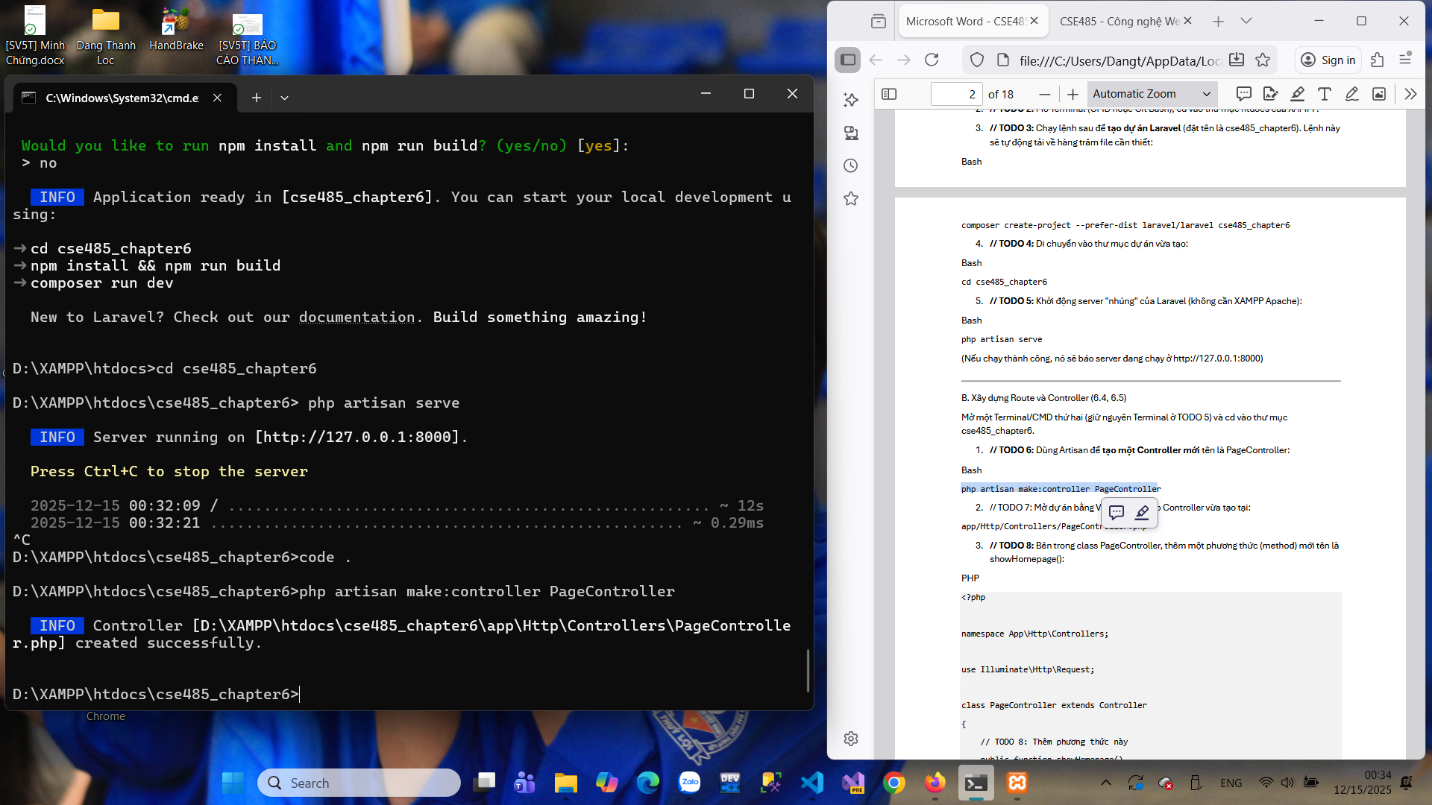
use App\Http\Controllers\PageController;

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

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

**B.Ảnh**

**1.Ảnh 1**



**2.Ảnh 2**

A screenshot of a computer

AI-generated content may be incorrect.

**3.Ảnh 3**

A screenshot of a computer

AI-generated content may be incorrect.

**C. Câu hỏi**

Nếu Laravel gộp chung Routing và Controller vào cùng một file như mô hình PHT Chương 5, điều gì sẽ xảy ra khi ứng dụng mở rộng lên hàng trăm route và hàng chục controller? Hãy phân tích những khó khăn có thể phát sinh và giải thích vì sao việc tách Routing ra file routes/web.php giúp kiến trúc ứng dụng sạch hơn và dễ bảo trì hơn

**Chương 7**

**A.Code**

**1.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;

        }

        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>

**2.homepage.blade.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

**3.PageController**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PageController extends Controller

{    public function showHomepage()

    {

        // TODO 7: Thay vì "return 'Chào mừng...'", chúng ta sẽ

        $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

        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'));

}

**B.Ảnh**

**A screenshot of a computer

AI-generated content may be incorrect.**

**C.Câu hỏi**

Hãy phân tích sự khác biệt giữa @extends('app'), @yield('content') và @section('content'). Nếu thiếu một trong ba thành phần này trong cấu trúc Blade, điều gì sẽ xảy ra với quá trình render HTML cuối cùng? Bạn hãy giải thích bằng ví dụ cụ thể.

**Chương 8**

**A.Code**

**1.Hàm up()**

 public function up(): void

    {

        Schema::create('sinh\_viens', function (Blueprint $table) {

            $table->id();

            $table->string('ten\_sinh\_vien', 255);

            $table->string('email', 255)->unique();

            $table->timestamps();

        });

    }

**2. app/Models/SinhVien.php.**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class SinhVien extends Model

{

    use HasFactory;

    // TODO 8: Thêm mảng $fillable

    protected $fillable = [

        'ten\_sinh\_vien',

        'email',

    ];

}

**3. app/Http/Controllers/SinhVienController.php.**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

// TODO 10: Import Model SinhVien

use App\Models\SinhVien;

class SinhVienController extends Controller

{

    public function index()

    {

        // TODO 11: Dùng Eloquent ::all() để lấy toàn bộ sinh viên

        $danhSachSV = SinhVien::all();

        // TODO 12: Trả về 1 view 'sinhvien.list' và truyền $danhSachSV

        return view('sinhvien.list', compact('danhSachSV'));

    }

    public function store(Request $request)

    {

        // TODO 13: Lấy toàn bộ dữ liệu từ form

        $data = $request->all();

        // TODO 14: Dùng Eloquent ::create() để lưu vào CSDL

        SinhVien::create($data);

        // TODO 15: Chuyển hướng về trang danh sách

        return redirect()->route('sinhvien.index');

    }

}

**4. routes/web.php.**

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\PageController;

use App\Http\Controllers\SinhVienController; // TODO 16: import controller

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

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

// TODO 16: Tạo 2 route cho SinhVien (GET index, POST store)

Route::get('/sinhvien', [SinhVienController::class, 'index'])->name('sinhvien.index');

Route::post('/sinhvien', [SinhVienController::class, 'store'])->name('sinhvien.store');

**B.Ảnh**

**1.Terminal**

**A computer screen shot of a black screen

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**2.phpMyadmin**

**A screenshot of a computer

AI-generated content may be incorrect.**

**3.Trình duyệt**

**A screenshot of a computer

AI-generated content may be incorrect.**

**C.Câu hỏi**

Hãy phân tích sự khác biệt về cách tiếp cận giữa Eloquent (ORM) và PDO/SQL thuần. Theo bạn, Eloquent mang lại lợi ích lớn nhất ở điểm nào trong việc phát triển ứng dụng web? Ngược lại, trong những tình huống nào việc sử dụng PDO/SQL thuần sẽ hiệu quả hơn Eloquent? Hãy đưa ra ví dụ minh họa cho từng trường hợp