



# Views trong Laravel

Giảng viên: Bùi Quang Đăng

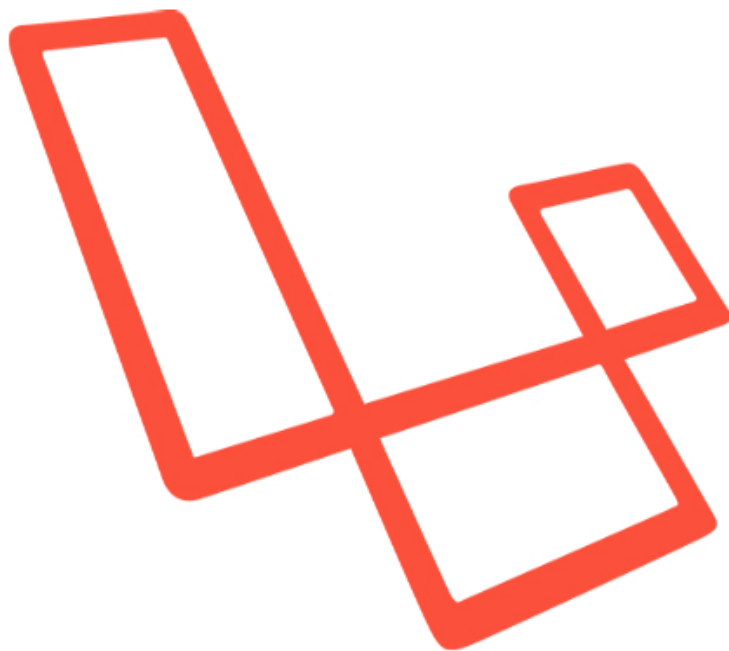


**Làm việc với View**

**Các cú pháp trong Laravel Blade**

**Template trong Laravel**

**Bootstrap trong Laravel**



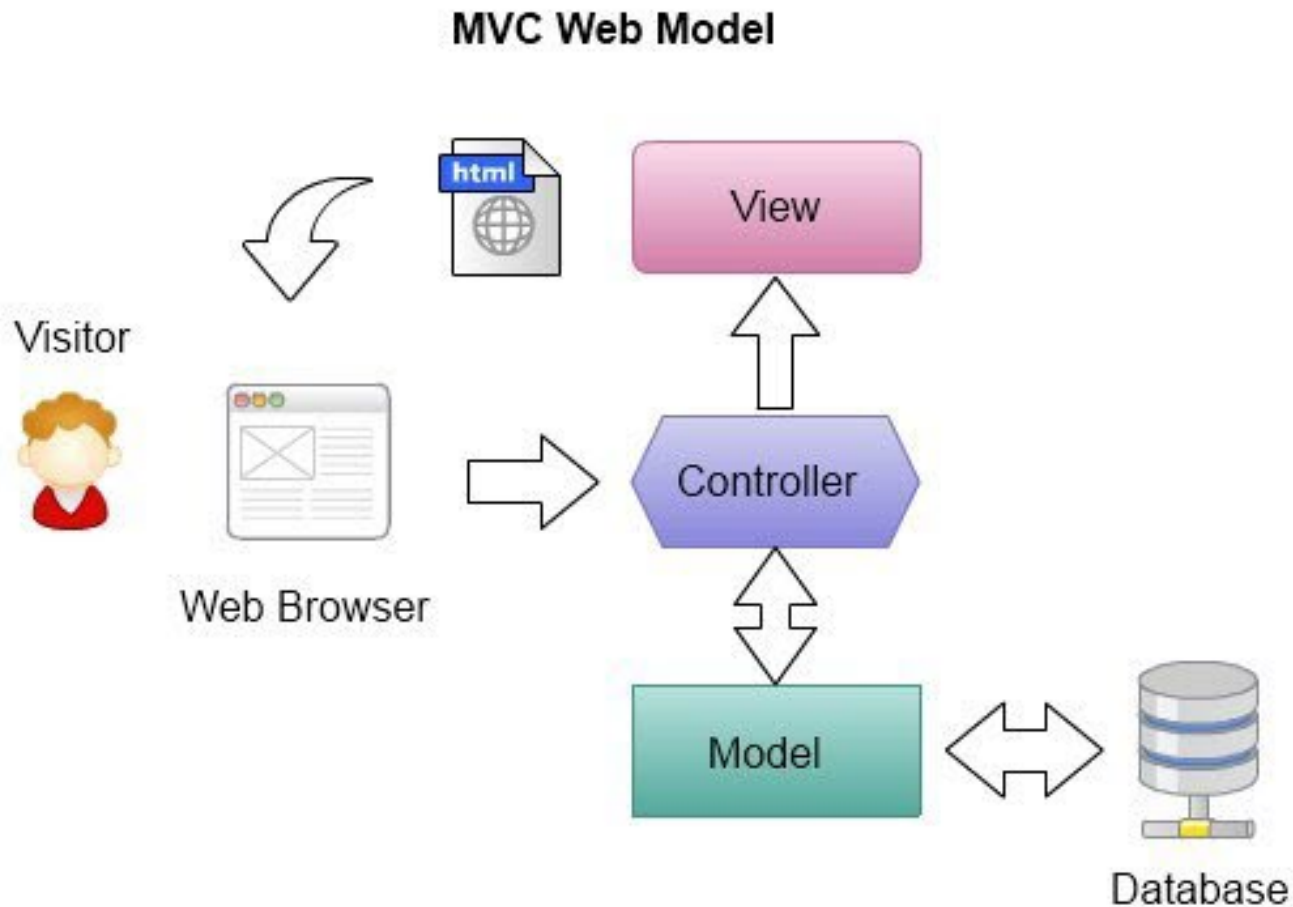
## Custom Laravel Directives

`{{{ }}`



## LÀM VIỆC VỚI VIEWS

## ❖ Giới thiệu về View



## ❖ Giới thiệu về View

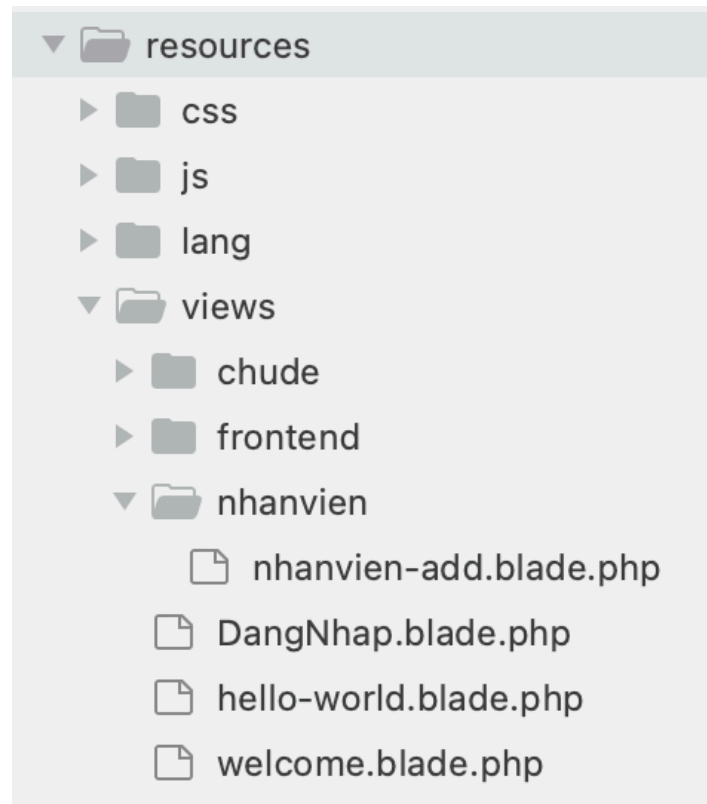
- Trong mô hình MVC, V là chữ viết tắt của View. Nó giúp tách biệt giữa application logic và presentation logic giúp quản lý và phát triển mã nguồn tốt hơn.
- Các view trong Laravel được lưu trữ trong thư mục resources/views.
- Chúng ta có thể tạo thêm các thư mục trong thư mục resources/views để phân cấp quản lý.

## ❖ Giới thiệu về View

- Bạn có thể tạo các thư mục con ở cấp sâu hơn, ví dụ:
  - `resources/views/admin/product`: Các view liên quan đến quản trị sản phẩm trên website.
  - `resources/views/admin/news`: Các view liên quan đến quản trị bài viết trên website.

## ❖ Giới thiệu về View

- Các views được tạo trong khu vực sau trong dự án sử dụng Laravel:





## ❖ Làm việc với views

- Để tạo ra một view mới chúng ta sử dụng cú pháp sau:

Tên view **.blade.php**

- Trong đó:
  - Định dạng **.blade.php** là bắt buộc


- Gọi view để sử dụng:





```
return view("Tên view");
```

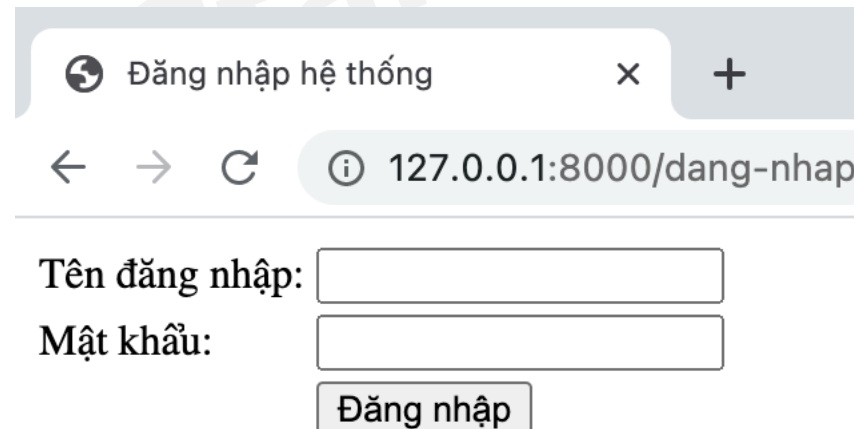
## ❖ Làm việc với views

- Ví dụ: Trang đăng nhập hệ thống như sau:

```
//Hiển thị giao diện đăng nhập
public function hienThiDangNhap()
{
    return view("DangNhap");
}
```

▼  nhanvien

-  nhanvien-add.blade.php
-  DangNhap.blade.php
-  hello-world.blade.php
-  welcome.blade.php



Đăng nhập hệ thống

← → ↻ ⓘ 127.0.0.1:8000/dang-nhap

Tên đăng nhập:

Mật khẩu:

Đăng nhập

## ❖ Làm việc với views

- Tạo view trong một thư mục con của views sử dụng cú pháp sau:

```
chude/Tên view .blade.php
```

- Trong đó:
  - chude là thư mục chứa views được tạo
  - Định dạng **.blade.php** là bắt buộc

- Gọi view để sử dụng:

```
return view("chude.Tên view");
```

## ❖ Làm việc với views

- Hiển thị nội dung từ biến trong code trên view sử dụng cú pháp **{{tên biến}}**:

- Ví dụ:

```
<h1>Tên của bạn là: {{name}}</h1>
```

- Sử dụng kiểm tra khi hiển thị:

```
<h2>Trang chủ</h2>  
<h3>Bạn đang đăng nhập với tài khoản: {{isset($tenDangNhap)  
? $tenDangNhap : "Chưa có"}}</h3>
```

## ❖ Làm việc với views

- Kiểm tra xem view có tồn tại trước khi trả về:

```
Route::get('/', function () {  
    if(View::exists('welcome')){  
        return view('welcome');  
    }  
    else  
    {  
        return 'Trang không tồn tại trên hệ thống';  
    }  
});
```

## ❖ Làm việc với views

- Kiểm tra xem view có tồn tại trước khi trả về:

```
Route::get('/', function () {  
    if(View::exists('welcome')){  
        return view('welcome');  
    }  
    else  
    {  
        return 'Trang không tồn tại trên hệ thống';  
    }  
});
```

## ❖ Làm việc với views

- Truyền và nhận thông tin qua view sử dụng with với dữ liệu đưa vào là một dạng mảng:

```
//Khi nhấn nút đăng nhập
public function xuLyDangNhap(Request $request)
{
    $tenDangNhap = $request->input('tenDangNhap');
    $matKhai = $request->input('matKhai');

    echo "Tên đăng nhập: " . $tenDangNhap . "<br>";
    echo "Mật khẩu: " . $matKhai . "<br>";

    if($tenDangNhap == "stanford" && $matKhai=="123")
    {
        return view("TrangChu")->with(['tenDangNhap'=>$tenDangNhap]);
    }
    else
    {
        return response()->view("DangNhap");
    }
}
```

## ❖ Làm việc với views

- Truyền và nhận thông tin qua view sử dụng with với dữ liệu đưa vào là một dạng mảng:

```
<html>
<head>
  <title>Trang chủ</title>
</head>
<body>
  <h2>Trang chủ</h2>
  <h3>Bạn đang đăng nhập với tài khoản: {{$tenDangNhap}}</h3>
</body>
</html>
```



## ❖ Làm việc với views

- Chia sẻ dữ liệu cho các views bằng việc sử dụng phương thức `share()` viết trong tập tin

**app\Provider\AppServiceProvider.php** như sau:

```
use Illuminate\Support\Facades\View;
```

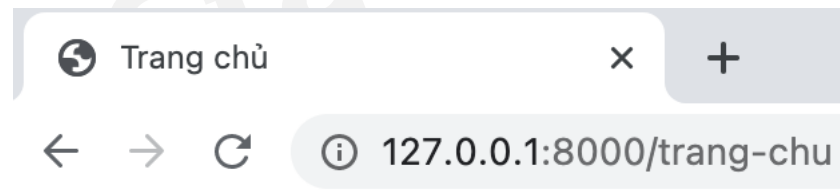
```
View::share('domain', 'stanford.com.vn');
```

## ❖ Làm việc với views

- Chia sẻ dữ liệu cho các views bằng việc sử dụng phương thức `share()` viết trong tập tin

`app\Provider\AppServiceProvider.php` như sau:

```
<h4>Giá trị từ share: {{$domain}}</h4>
```



**Trang chủ**

Giá trị từ share: stanford.com.vn



## LÀM VIỆC VỚI LARAVEL BLADE

## ❖ Làm việc với Laravel Blade

- Sử dụng các lệnh `@if`, `@else`, `@elseif` và `@endif` như trong PHP thuần túy, cú pháp sử dụng như sau:

```
@if (biểu thức điều kiện)  
    [Công việc]  
@else  
    [Công việc]  
@endif
```

## ❖ Làm việc với Laravel Blade

- Vòng lặp sử dụng @for, @while, @foreach:

```
@for($i = 1; $i <= 10; $i++)
```

```
    Giá trị của i = {{ $i }}
```

```
@endfor
```

```
{{ $i = 1 }}
```

```
@while($i <= 10)
```

```
    Giá trị của i = {{ $i++ }}
```

```
@endwhile
```

## ❖ Làm việc với Laravel Blade

- Vòng lặp sử dụng @for, @while, @foreach:

Thuộc tính	Mô tả
<code>\$loop-&gt;index</code>	Trả về chỉ mục vòng lặp hiện hành (bắt đầu bởi 0).
<code>\$loop-&gt;iteration</code>	Trả về số vòng lặp đã thực hiện (bắt đầu bởi 1).
<code>\$loop-&gt;remaining</code>	Trả về số lần lặp còn lại
<code>\$loop-&gt;count</code>	Trả về tổng số lần lặp
<code>\$loop-&gt;first</code>	Trả về true nếu là vòng lặp đầu tiên
<code>\$loop-&gt;last</code>	Trả về true nếu là vòng lặp cuối cùng
<code>\$loop-&gt;depth</code>	Trả về cấp độ lồng nhau của vòng lặp.
<code>\$loop-&gt;parent</code>	Trong vòng lặp lồng nhau, trả về \$loop của vòng lặp cha.

## ❖ Làm việc với Laravel Blade

- Sử dụng comment trong Laravel:

```
{{—Nội dung cần comment --}}
```

- Sử dụng theo kiểu php thuần túy, thẻ **@php**:

```
<?php ?>  
Hoặc  
@php  
    [Nội dung xử lý]  
@endphp
```

## ❖ Sử dụng javascript framework trong Blade

- Có nhiều framework bằng javascript cũng sử dụng {{ }} để in ra giá trị các biến, do đó khi tích hợp một framework Javascript vào dự án mà trong code có những đoạn {{ value }} của Javascript thì nó sẽ không hoạt động do hệ thống mặc định đây là in biến trong Laravel. Để giải quyết vấn đề này bạn sử dụng @, nó sẽ thông báo với Laravel rằng bỏ qua mà không xử lý {{ value }}.

```
<h2>@{{ title }}</h2>  
<p> @{{ content }}. </p>
```



## ❖ Sử dụng javascript framework trong Blade

- Nhiều khối lệnh `{{value}}` có thể code xử lý bằng javascript trong khối lệnh sau:

**@verbatim**

```
<h2>{{ title }}</h2>
```

```
<p> {{ content }}. </p>
```

```
<p> {{ author }}. </p>
```

**@endverbatim**



## TEMPLATE TRONG LARAVEL

## ❖ Làm việc với Laravel Blade

- Bộ máy hiển thị view trong Laravel Framework với định dạng mở rộng `.blade.php`
- Có thể tạo template chung cho trang web sử dụng Laravel Blade
- **@section** để định nghĩa nội dung
- **@yield** để hiển thị nội dung cho section định nghĩa

## ❖ Làm việc với Laravel Blade

```
default.blade.php
<h1>Stanford – Dạy kinh
nghiệm lập trình</h1>
@section('sidebar')
@show

@yield('content');
<p>Website:
Stanford.com.vn</p>
```

Template Laravel

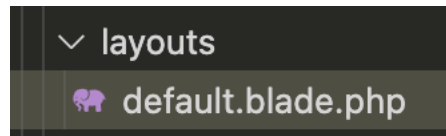
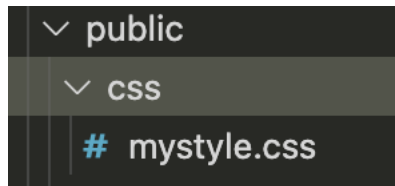
```
TrangChu.blade.php
extends('layouts.default')
```

```
GioiThieu.blade.php
extends('layouts.default')
```

```
LienHe.blade.php
extends('layouts.default')
```

## ❖ Làm việc với Laravel Blade

- Tạo trang chính trong Laravel



```
<html>
<head>
  <title>Stanford.com.vn - @yield('title')</title>
  <link rel="stylesheet" type="text/css" href="{{ asset('css/mystyle.css') }}" />
</head>
<body>
```

## ❖ Làm việc với Laravel Blade

- Tạo các thành phần (component) trong Laravel:
  - Định nghĩa một component trong layouts như sau:

```
<div class="alert {{ $class }}">
  <div class="alert-title"><strong>{{ $title }}</strong></div>
  {{ $message }}
</div>
```

## ❖ Làm việc với Laravel Blade

- Tạo các thành phần (component) trong Laravel:
  - Sử dụng component đã tạo:

```
@component('layouts.alert')  
  
    @slot('class')  
        alert-danger  
    @endslot  
  
    @slot('title')  
        Thông báo lỗi  
    @endslot  
  
    @slot('message')  
        Có lỗi xảy ra trong quá trình thực hiện  
    @endslot  
@endcomponent
```

## ❖ Làm việc với Laravel Blade

- Tạo các thành phần (component) trong Laravel:
  - Sử dụng component đã tạo:

```
@component('layouts.alert')  
  
    @slot('class')  
        alert-success  
    @endslot  
  
    @slot('title')  
        Thông báo  
    @endslot  
  
    @slot('message')  
        Thực hiện thành công  
    @endslot  
  
    Bạn có thể xem lại tại đây  
@endcomponent
```



## ❖ Làm việc với Laravel Blade

- Tạo các thành phần (component) trong Laravel:
  - Sử dụng component đã tạo:

### **Thông báo lỗi**

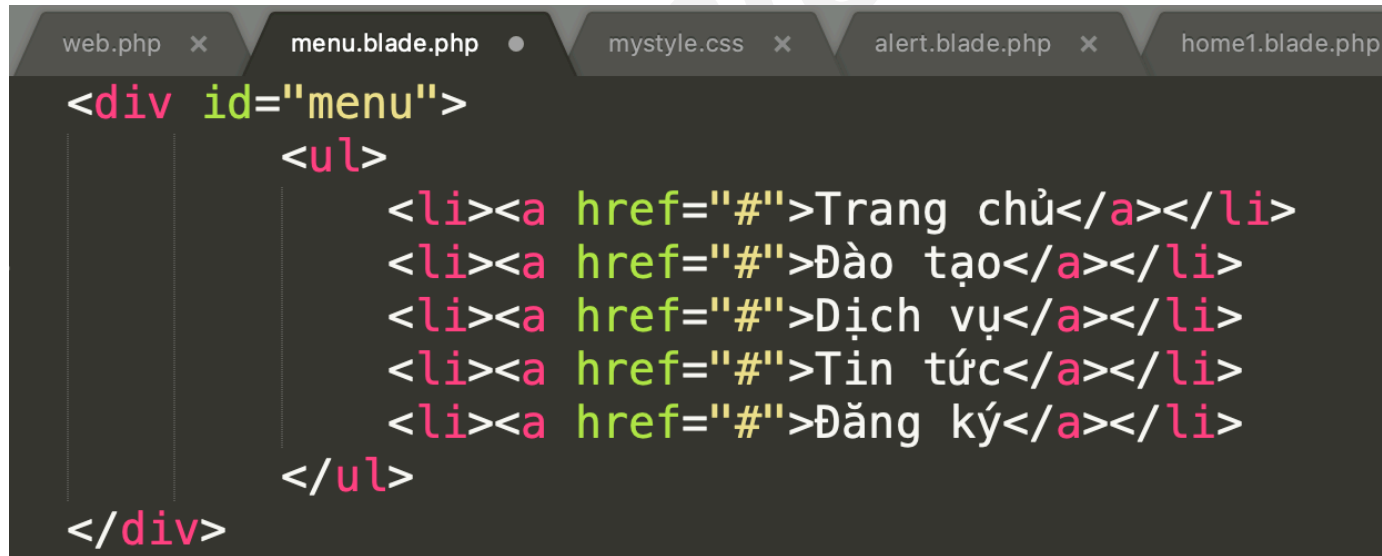
Có lỗi xảy ra trong quá trình thực hiện

### **Thông báo**

Thực hiện thành công

## ❖ Làm việc với Laravel Blade

- **Sử dụng subview trong Laravel:** Tách các ra thành nhiều thành phần và sử dụng `@include` hoặc `@includeIf` kiểm tra xem view có tồn tại không để include vào trang trong Laravel:



```
<div id="menu">
  <ul>
    <li><a href="#">Trang chủ</a></li>
    <li><a href="#">Đào tạo</a></li>
    <li><a href="#">Dịch vụ</a></li>
    <li><a href="#">Tin tức</a></li>
    <li><a href="#">Đăng ký</a></li>
  </ul>
</div>
```

## ❖ Làm việc với Laravel Blade

- **Sử dụng subview trong Laravel:** Tách các ra thành nhiều thành phần và sử dụng `@include` hoặc `@includeIf` kiểm tra xem view có tồn tại không để include vào trang trong Laravel:

```
@include('layouts.menu')
```

```
@includeIf('layouts.menu1')
```

Không bị lỗi  
khi không tìm  
thấy view

## ❖ Làm việc với Laravel Blade

- **Tùy chỉnh lệnh trong Laravel:** Blade cho phép người lập trình bổ sung thêm lệnh bằng việc khai báo trong **AppServiceProvider.php**:

```
use Illuminate\Support\Facades\Blade;
```

```
Blade::directive('datetime', function ($expression) {  
    return "<?php echo ($expression)->format('d/m/Y H:m'); ?>";  
});
```

```
<h3>Ngày tháng hiện tại là:</h3>  
@php  
    $ngay = Now();  
@endphp  
@datetime(Now())
```

## ❖ Làm việc với Laravel HTML

- Thực hiện di chuyển vào thư mục gốc của dự án Laravel để cài đặt gói **LaravelCollective**. Trên bản 4.x thì có sẵn nhưng từ 5.x muốn dùng thì cài thêm:

```
composer require laravelcollective/html
```

- Trong file **app.php** của config: Bổ sung trong mục provider và aliases nội dung sau:

```
Collective\Html\HtmlServiceProvider::class,
```

```
'Form' => Collective\Html\FormFacade::class,
```

```
'Html' => Collective\Html\HtmlFacade::class,
```

## ❖ Làm việc với Laravel HTML

- Một số cú pháp sử dụng:

```

{!! HTML::image('images/logo.png', 'Stanford Logo') !!}
```

```
<link rel="stylesheet" href="/css/mystyle.css">
{!! HTML::style('css/mystyle.css') !!}
```

```
<script src="/javascript/jquery-3.4.min.js"></script>
{!! HTML::script('javascript/jquery-3.4.min.js') !!}
```

## ❖ Làm việc với Laravel HTML

- Một số cú pháp sử dụng:

```
{!! Form::open(array('url' => 'foo/bar')) !!}
//
{!! Form::close() !!}
//
{!! Form::open(array('url' => 'foo/bar', 'method' => 'put'))
!!}
//
{!! Form::open(array('route' => 'route.name')) !!}
//
{!! Form::open(array('route' => array('route.name', $user-
>id))) !!}
{!! Form::token() !!}
```

## ❖ Làm việc với Laravel HTML

- Một số cú pháp sử dụng:

```
{!! Form::label('email', 'Địa chỉ Email') !!}  
//  
{!! Form::label('email', 'Địa chỉ Email', array('class' => 'label-  
success')) !!}  
{!! Form::text($name, $value = null, $attributes =  
array('class'=>'form-control', 'placeholder'=>'Nhập tên của bạn'))  
!!}  
{!! Form::textarea($name, $value, $attributes = array()) !!}  
{!! Form::email($name, $value = null, $attributes = array()) !!}  
{!! Form::password($name) !!}  
{!! Form::file($name, $attributes = array()) !!}
```



## ❖ Làm việc với Laravel HTML

- Một số cú pháp sử dụng:

```
{!! Form::checkbox('$name', '$value', true) !!}  
{!! Form::radio('$name', '$value', true) !!}  
{!! Form::date('post_date', \Carbon\Carbon::now()) !!}  
{!! Form::select('size', array('1' => 'Laravel', '2' => 'Symfony', '3'  
=> 'Yii'), '1') !!}  
{!! Form::submit('Thực hiện') !!}
```

## ❖ Làm việc với Laravel HTML

- Sử dụng Laravel HTML giúp viết code ngắn gọn hơn:

```
@php
$listFramework = ["L" => "Laravel", "S" => "Symfony", "Y" =>
"Yii", "C" => "CodeIgniter"];
@endphp
<select name="framework">
@foreach($listFramework as $frm)
<option>{{ $frm }}</option>
@endforeach
</select>
```

## ❖ Làm việc với Laravel HTML

- Sử dụng Laravel HTML giúp viết code ngắn gọn hơn:

```
@php
$lstFramework = ["L" => "Laravel", "S" => "Symfony", "Y" =>
"Yii", "C" => "CodeIgniter"]; @endphp
//
{!! Form::select('framework', $lstFramework) !!}
```

## ❖ Làm việc với Laravel HTML

- Thiết kế giao diện đăng nhập với Laravel Html:

```
{!!Form::open(array('url'=>'dangNhap1','method'=>'post')) !!}  
<table>  
<tr>  
<td>  
{!!Form::label('label1','Tên đăng nhập') !!}  
</td>  
<td>  
{!!Form::text('tenDangNhap', $tenDangNhap?? '', array('placeholder'=>'Nhập  
tên đăng nhập')) !!}  
</td>  
</tr>
```

## ❖ Làm việc với Laravel HTML

- Thiết kế giao diện đăng nhập với Laravel Html:

```
<tr>
<td>
{!!Form::label('label2','Mật khẩu') !!}
</td>
<td>
{!!Form::password('matKhau') !!}
</td>
</tr>
<tr>
<td></td>
<td>
{!!Form::submit('Đăng nhập') !!}
</td>
</tr>
</table>
{!!Form::close()!!}
```

## ❖ Làm việc với Laravel HTML

- Thiết kế giao diện đăng nhập với Laravel Html:

```
public function xuLyDangNhap1(Request $request)
{
    $tenDangNhap = $request->input('tenDangNhap');
    $matKhai = $request->input('matKhai');

    echo "Tên đăng nhập: " . $tenDangNhap . "<br>";
    echo "Mật khẩu: " . $matKhai . "<br>";

    return view("DangNhapLaraHtml")->with(['tenDangNhap'=>$tenDangNhap])
    ;
}
```

Tên đăng nhập

Mật khẩu



## BOOTSTRAP TRONG LARAVEL

## ❖ Làm việc với Bootstrap

- Di chuyển vào thư mục gốc của dự án, đầu tiên cần cài gói ui trong Laravel như sau:

```
composer require laravel/ui
```

```
[(base) Dang's-MacBook-Pro:Sample1 dangquang$ composer require laravel/ui
PHP Deprecated: Directive 'track_errors' is deprecated in Unknown on line 0

Deprecated: Directive 'track_errors' is deprecated in Unknown on line 0
Using version ^3.2 for laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking laravel/ui (v3.2.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Downloading laravel/ui (v3.2.0)
  - Installing laravel/ui (v3.2.0): Extracting archive
Generating optimized autoload files
```



## ❖ Làm việc với Bootstrap

- Thực hiện cài đặt bootstrap để sử dụng trong dự án Laravel:

```
php artisan ui bootstrap
```

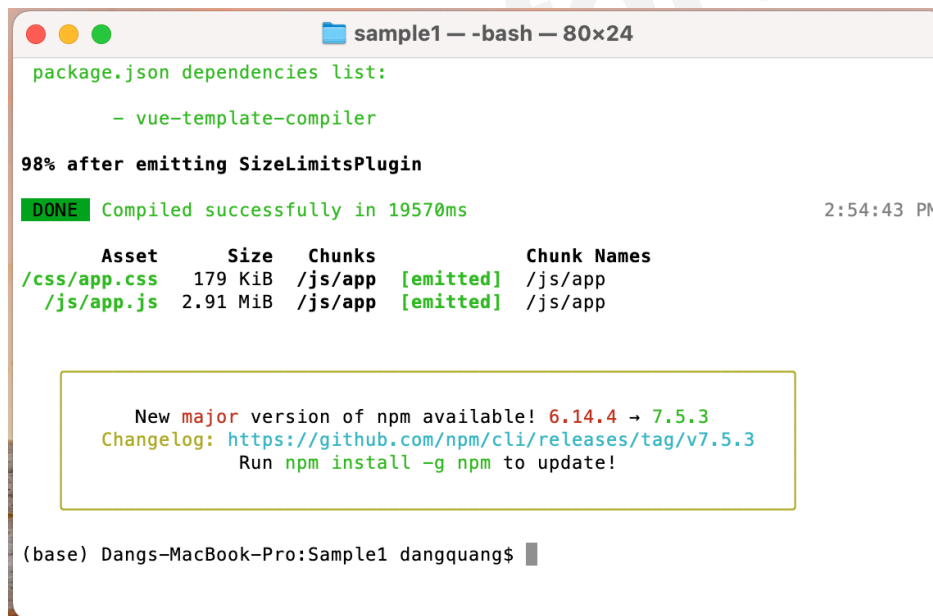
Hoặc

```
php artisan ui bootstrap --auth
```

## ❖ Làm việc với Bootstrap

- Thực hiện cài đặt NPM và chạy lệnh sau:

```
npm install  
/  
npm run dev
```



```
sample1 — -bash — 80x24  
package.json dependencies list:  
  - vue-template-compiler  
  
98% after emitting SizeLimitsPlugin  
  
DONE Compiled successfully in 19570ms 2:54:43 PM  
  
Asset      Size  Chunks  Chunk Names  
/css/app.css 179 KiB /js/app [emitted] /js/app  
/js/app.js  2.91 MiB /js/app [emitted] /js/app  
  
New major version of npm available! 6.14.4 → 7.5.3  
Changelog: https://github.com/npm/cli/releases/tag/v7.5.3  
Run npm install -g npm to update!  
  
(base) Dangs-MacBook-Pro:Sample1 dangquang$
```

## ❖ Làm việc với Bootstrap

- Khai báo thư viện bootstrap trên để sử dụng như sau:

```
<!-- Scripts -->
<script src="{{ asset('js/app.js') }}"
defer></script>
<!-- Styles -->
<link href="{{ asset('css/app.css') }}"
rel="stylesheet">
```

## ❖ Làm việc với Bootstrap

- Sử dụng bootstrap trên trang web trong dự án:
  - Trang Danh sách



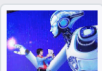
```
<div class="container" style="text-align:center;">
<h1>Danh sách thông tin sách</h1>
</div>
<div style="text-align:right;padding-right:10px;padding-bottom:5px;"><a
class="btn btn-primary" href="/sach-add">Thêm mới</a></div>
<table class="table table-striped table-bordered">
```

## ❖ Làm việc với Bootstrap

- Sử dụng bootstrap trên trang web trong dự án:
  - Trang Danh sách

### Danh sách thông tin sách

[Thêm mới](#)

Ảnh	Id	Tên sách	Mô tả	Giá	Ngày tạo	Tác giả	
	1	Đặc Nhân Tâm	Sách dạy về kỹ năng sống để thành đạt hơn trong cuộc sống của mỗi con người	250000	2020-12-02	Dale Carnegie	<a href="#">Sửa</a> <a href="#">Xoá</a>
	2	Cha giàu cha nghèo	Sách nói về kỹ năng quản lý tài chính của 2 người cha giàu và cha nghèo của tác giả	200000	2020-12-02	Robert T.Kiyosaki	<a href="#">Sửa</a> <a href="#">Xoá</a>
	3	Lập trình C++	Sách dạy về ngôn ngữ lập trình c++ cho người mới học	150000	2020-12-02	Phạm Văn Ất	<a href="#">Sửa</a> <a href="#">Xoá</a>
	4	Trí tuệ nhân tạo	Sách nói về trí tuệ nhân tạo áp dụng trong cuộc sống hiện tại	200000	2020-12-07	Sim mon	<a href="#">Sửa</a> <a href="#">Xoá</a>

## ❖ Làm việc với Bootstrap

- Sử dụng bootstrap trên trang web trong dự án:
  - Trang Thêm mới thông tin sách

```
<form method="post" action="/themSach" enctype="multipart/form-data">
<fieldset class="container">
<legend>Nhập thông tin sách</legend>
<div class="row">
<label class="col-md-2">
Tên sách:
</label>
<div class="col-md-10">
<input type="hidden" name="_token" value="<?php echo csrf_token()?>"/>

<input type="text" name="TenSach" class="form-control"/>
</div>
</div>
```

## ❖ Làm việc với Bootstrap

- Sử dụng bootstrap trên trang web trong dự án:
  - Trang Thêm mới thông tin sách

Thêm mới thông tin sách

Nhập thông tin sách

Tên sách:

Mô tả:

Ảnh:

Choose File

No file chosen

Giá sách:

Tác giả:

Thêm mới

Trở lại



# Thank You !

[www.stanford.com.vn](http://www.stanford.com.vn)