
LAB 3 MÔN LẬP TRÌNH PHP3: SỬ DỤNG BLADE TEMPLATE

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Tạo blade view và sử dụng các cú pháp blade hỗ trợ
- ✓ Tạo layout và sử dụng các thành phần trong layout
- ✓ Rèn luyện kỹ năng sử dụng Query Builder

NỘI DUNG

Chuẩn bị:

- Tạo project laravel **la3**
- Cấu hình kết nối đến database **la_news**
- Trong phpmyadmin, tạo database **la_news** (nếu chưa tạo) và import file **la_news.sql**

Bài 1: Tạo controller, view và routing

Thực hiện 3 chức năng :

- Trang chủ
- Chi tiết 1 tin (tham số là id của tin)
- Danh sách các tin trong 1 loại (tham số là idLT)

1. Tạo controller

Tạo controller có tên TinController

```
php artisan make:controller TinController
```

2. Định nghĩa các action

Tạo 3 action trong controller mới tạo với yêu cầu như sau:

- **index** : nạp view home
- **chitiet**: tham số là id của tin, nạp view chitiet
- **tintrongloai** : tham số là \$idLT , nạp view tintrongloai

```
class TinController extends Controller
{
    function index(){
        return view("home");
    }
    function chitiet($id=0){
        $data = ['id'=>$id];
        return view("chitiet", $data);
    }
    function tintrongloai($idLT=0){
        $data = ['idLT'=>$idLT];
        return view("tintrongloai", $data);
    }
}
```

3. Tạo các view

Tạo các view trong controller

- Tạo views/home.blade.php

```
<h1>Đây là trang chủ</h1>
```

- Tạo views/chitiet.blade.php

```
<h1>Thông tin chi tiết tin {{$id}}</h1>
```

- Tạo views/tintrongloai.blade.php

```
<h1>Các tin trong loại {{$idLT}}</h1>
```

4. Khai báo các route

Trong routes/web.php

```
use App\Http\Controllers\TinController;
Route::get('/', [TinController::class, 'index']);
Route::get('/tin/{id}', [TinController::class, 'chitiet']);
Route::get('/cat/{id}', [TinController::class, 'tintrongloai']);
```

Test thử:



Bài 2: Tạo layout và chỉ định layout trong các view con

1. Tạo layout

- Tạo views/layout.blade.php và code

```
<head>
  <title>Tiêu đề trang</title>
  <meta charset="utf-8"/>
  <link rel="stylesheet" href=
"https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" />
  <style>
    .container > header { height: 220px}
    .container > nav { height: 60px}
    .container > footer { height: 90px}
    .container > main { display: flex; min-height: 500px}
  </style>
</head>
<body>
<div class="container">
  <header class="bg-primary"></header>
  <nav class="bg-warning">THANH MENU</nav>
  <main>
    <article class="col-9 bg- light">
      NỘI DUNG CHÍNH
    </article>
    <aside class="col-3 bg-info">
      THÔNG TIN BỔ SUNG
    </aside>
  </main>
  <footer class="bg-dark">PHÁT TRIỂN BỞI XYZ</footer>
</div>
</body>
```

2. Định nghĩa các vùng chứa nội dung động trong layout

- Thay chữ **NỘI DUNG CHÍNH** thành **@yield('noidung')**

- Thay chữ **Tiêu đề trang** thành **@yield('tieudetrang')**

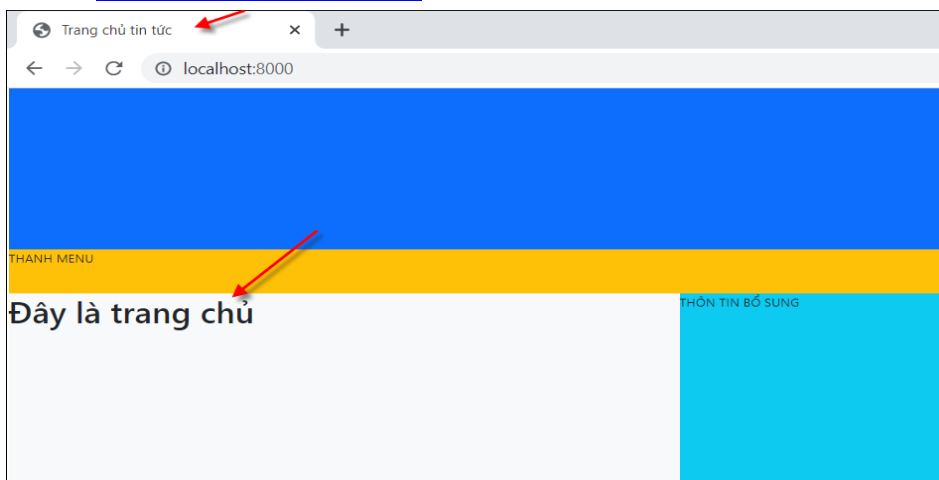
3. Khai báo layout trong các view con

- a. Code lại views/home.blade.php để :

- Khai báo layout
- Khai báo giá trị cho 2 vùng nội dung động trong layout **tieudetrang** và **noidung**

```
@extends('layout')
@section('tieudetrang')
    Trang chủ tin tức
@endsection
@section('noidung')
    <h1>Đây là trang chủ</h1>
@endsection
```

- Test : <http://localhost:8000>



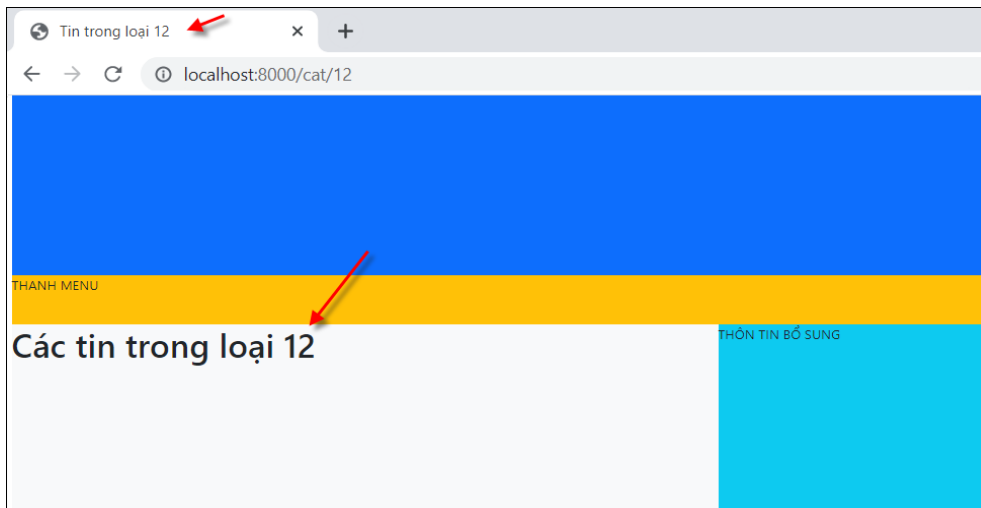
- Tương tự, code lại views/chitiet.blade.php để khai báo layout và giá trị cho 2 vùng nội dung động trong layout **tieudetrang** và **noidung**

```
@extends('layout')
@section('tieudetrang')
    Chi tiết tin {{$id}}
@endsection
@section('noidung')
    <h1>Thông tin chi tiết tin {{$id}}</h1>
@endsection
```

- Test : <http://localhost:8000/tin/1>
- Code lại views/tintrongloai.blade.php để khai báo layout và giá trị cho 2 vùng nội dung động trong layout **tieudetrang** và **noidung**

```
@extends('layout')
@section('tieudetrang')
    Tin trong loại {{$idLT}}
@endsection
@section('noidung')
    <h1>Các tin trong loại {{$idLT}}</h1>
@endsection
```

- Test : <http://localhost:8000/cat/12>



Bài 3: Code hiện chi tiết tin và tin trong loại

1. Hiện chi tiết tin

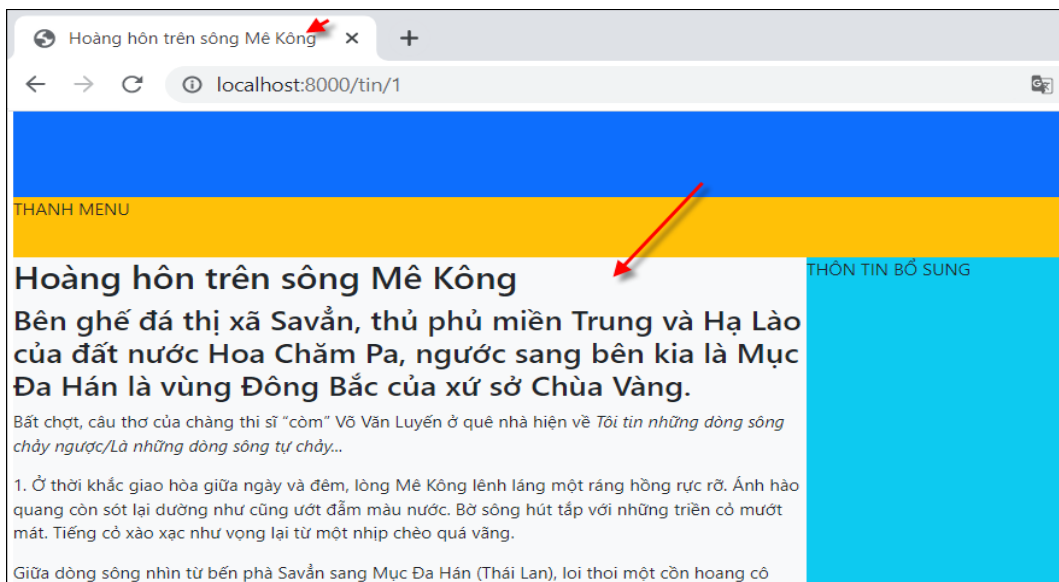
- Trong action chitiet, code lấy tin từ database theo id của tin và đưa kết quả vào view chitiet (biến tin)

```
function chitiet($id=0){
    $tin = DB::table('tin')->where('id', $id)->first();
    $data = ['id'=>$id, 'tin'=>$tin];
    return view("chitiet", $data);
}
```

- Trong views/chitiet.blade.php, hiện biến tin ra trang

```
@extends('layout')
@section('tieudetrang')
    {{$tin->tieuDe}}
@endsection
@section('noidung')
    <h2>{{$tin->tieuDe}}</h2>
    <h3>{{$tin->tomTat}}</h3>
    <div id="noiDung">
        {!! $tin->noiDung !!}
    </div>
@endsection
```

- Test



2. Hiện tin trong loại

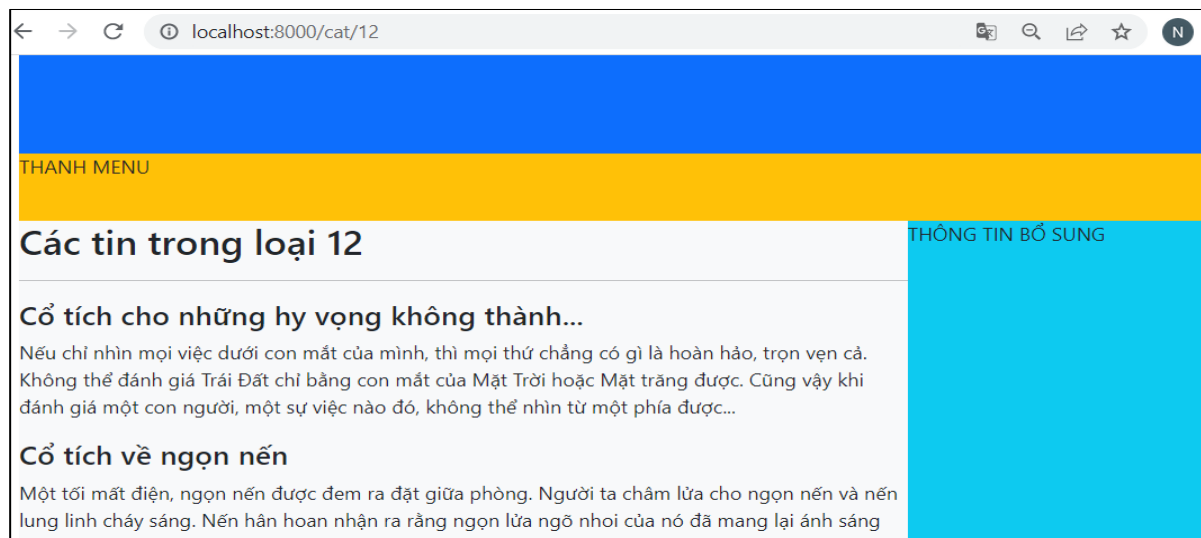
- Trong action tintrongloai, code lấy các tin trong 1 loại từ database theo idLT, đưa kết quả vào view tintrongloai (biến listtin)

```
function tintrongloai($idLT=0){
    $listtin = DB::table('tin')->where('idLT', $idLT)->get();
    $data = ['idLT'=>$idLT, 'listtin'=>$listtin];
    return view("tintrongloai", $data);
}
```

- Trong views/tintrongloai.blade.php, hiện biến danh sách các tin ra trang

```
@extends('layout')
@section('tieudetrang')
    Tin trong loại {{$idLT}}
@endsection
@section('noidung')
    <h1>Các tin trong loại {{$idLT}}</h1>
    @foreach ($listtin as $t)
        <div class="row">
            <h3>{{$t->tieuDe}}</h3>
            <p>{{$t->tomTat}}</p>
        </div>
    @endforeach
@endsection
```

- Test

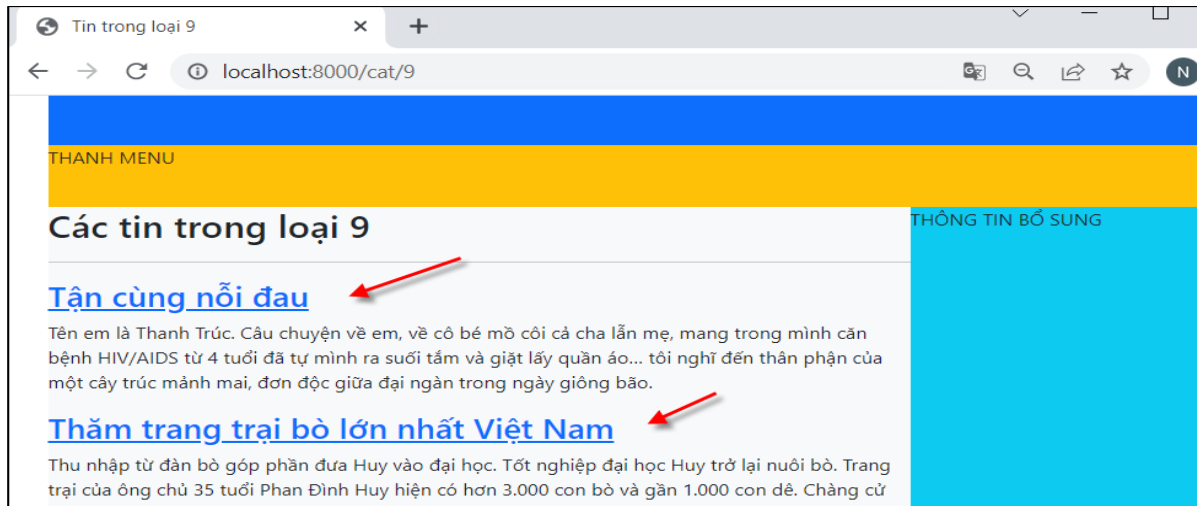


3. Tạo liên kết cho các tiêu đề tin

- Trong views/tintrongloai.blade.php, bổ sung tag a bao quanh tiêu đề tin để tạo liên kết như sau : (xóa h3 cũ và code lại như gợi ý trong hình)

```
@section('tieudetrang')
    Tin trong loại {{$idLT}}
@endsection
@section('noidung')
    <h1>Các tin trong loại {{$idLT}}</h1>
    @foreach ($listtin as $t)
        <div class="row">
            <h3>
                <a href="{{ url('/tin',[$t->id]) }}">
                    {{$t->tieuDe}}
                </a>
            </h3>
            <p>{{$t->tomTat}}</p>
        </div>
```

- Test: Nhấp vào các tiêu đề tin, phải hiện chi tiết của tin



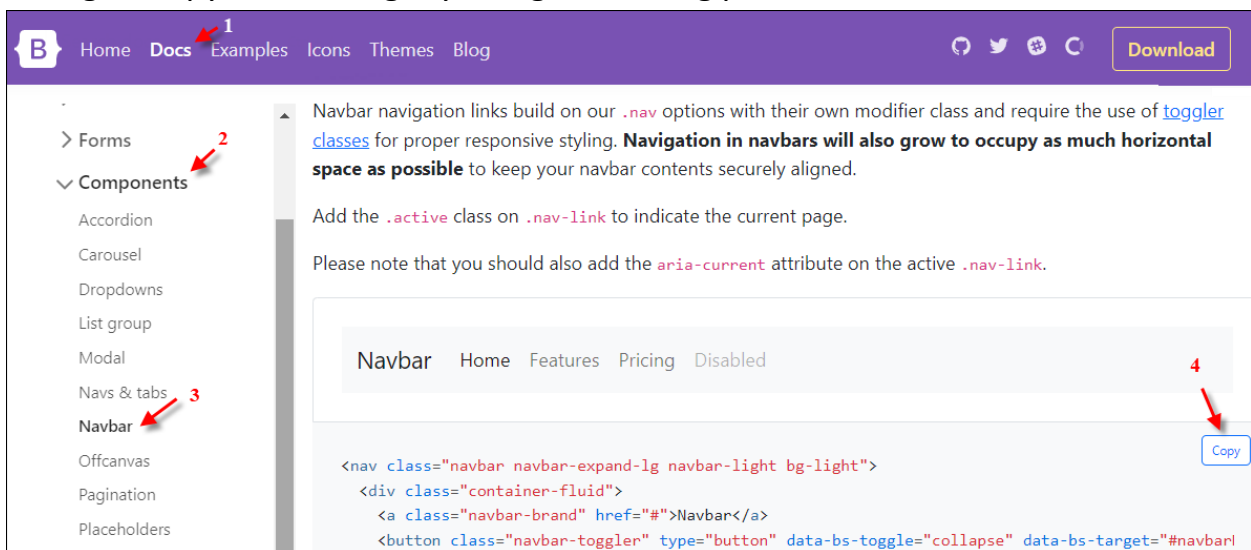
Bài 4: Thanh menu trong layout

1. Nhúng file menu vào layout

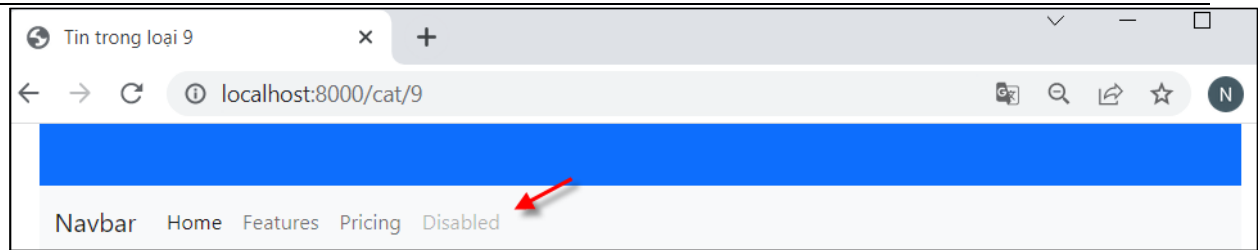
- Mở layout.blade.php, và code trong tag nav (xóa chữ **THANH MENU** và code lại)

```
<nav class="bg-warning">
    @include('menu')
</nav>
```

- Tạo views/menu.blade.php
- Vào <https://getbootstrap.com/> → Docs → Components → Navbar, cuộn xuống và copy code như gợi ý trong hình, xong paste vào file menu mới tạo.



- Xem thử sẽ phải thấy menu hiện ra



- Xóa 2 tag li chứa chữ Disabled, Pricing

2. Lấy các loại tin từ database và hiện trong view

- Đầu view menu , code để lấy các loại tin có AnHien là 1, sắp thuTu tăng dần

```
<?php
    $loaitin_arr = DB::table('loaitin')->select('id', 'ten')
        ->orderby('thuTu','asc')
        ->where('AnHien','=', '1')
        ->limit(5)->get();
?>
```

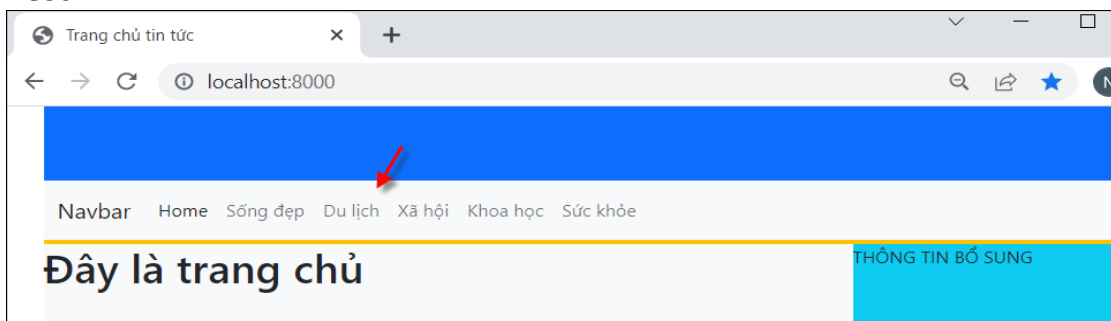
- Code lặp bao quanh tag li chứa chữ Features

```
@foreach ($loaitin_arr as $lt)
<li class="nav-item">
    <a class="nav-link" href="#">Features</a>
</li>
@endforeach
```

- Thay chữ Features và # thành code như hình

```
@foreach ($loaitin_arr as $lt)
<li class="nav-item">
    <a class="nav-link" href="#">Features</a>
</li>
@endforeach
    {{ url('/cat',[$lt->id]) }}
```

- Test



3. Cải thiện các view

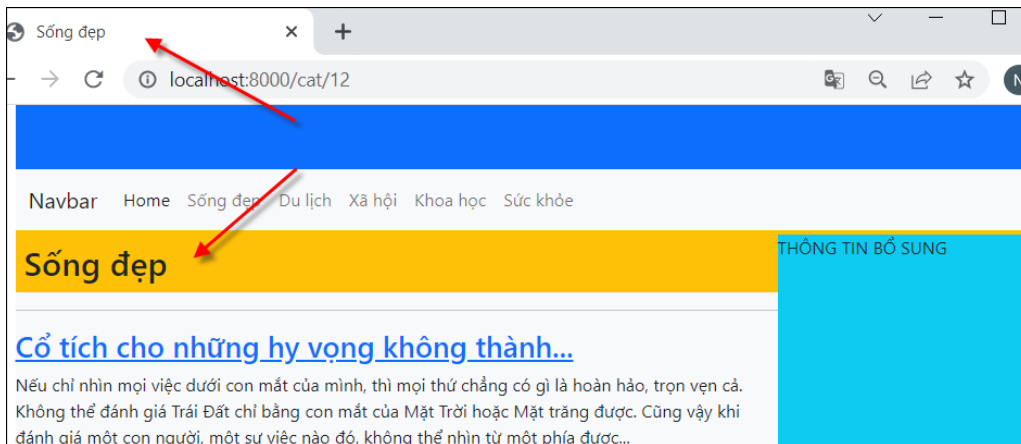
Mời sinh viên thực hiện các yêu cầu sau:

a. Chức năng tin trong loại:

Hiện tên loại tin thay thế tại các vị trí như hình



Kết quả như sau:



b. Định dạng các view chitiet, tintrongloai, home , layout cho đẹp

*** Yêu cầu nộp bài:

SV nén file (hoặc share thư mục google drive) bao gồm các yêu cầu đã thực hiện trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM.

--- Hết ---