**BÁO CÁO TRAINNING NGÀY 1**

# I. Html, CSS, Js cơ bản

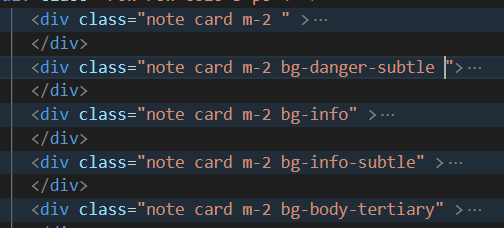
## 1. Class trong CSS

* Class được dùng để xác định các thẻ html có chung các thuộc tính CSS. Sau khi khai báo class, chúng ta khai báo thuộc tính và giá trị trong class đó và định dạng cho các thẻ html bằng cách gán tên class đã khai báo cho thuộc tính “class”.

Ví dụ: Class note được sử dụng để định dạng cho các thẻ div.

Text

Description automatically generated



## 2. Link tương đối là gì ?

* Link tương đối là link dẫn đến một file hay một tài liệu khác trong một hệ thống dựa trên vị trí của file hiện tại.



## 3. Function có ý nghĩa gì ?

* Trong javascript hay bất kỳ một ngôn ngữ lập trình nào cũng đều có những công việc cần lặp lại nhiều lần, mỗi lần như vậy đều phải viết lại code rất mất thời gian và làm code dài dòng. Function giúp chúng ta gom đoạn mã đó lại và tái sử dụng bằng cách gọi tên của chúng. Ngoài ra sử dụng function cũng giúp code ngắn gọn, dễ đọc và dễ bảo trì hơn.

Ví dụ: Hàm display() được gọi mỗi khi người dùng click vào thẻ input.

Text

Description automatically generated

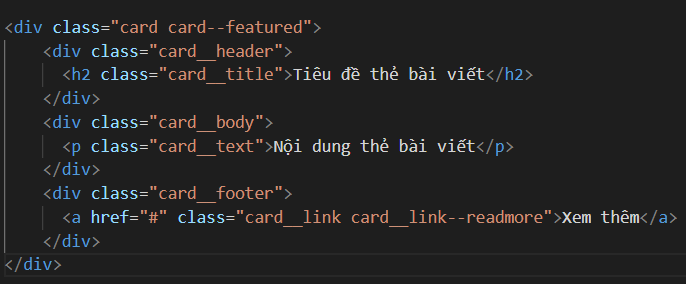
Text

Description automatically generated

## 4. BEM là gì ?

* BEM là viết tắt của "Block, Element, Modifier". Đây là một phương pháp đặt tên class cho các thành phần trong CSS. Trong đó Block là một thành phần lớn chứa các Elements bên trong, Modifier là một lớp đặc biệt dùng để thêm hoặc bớt các thuộc tính của Block hoặc Elements.

Ví dụ:



Ở ví dụ này, card chính là block với các elements là card\_\_header, card\_\_body, card\_\_footer, với modifier là card-featured.

# II. Bootstrap cơ bản

## Flexbox

* Để sử dụng Flexbox trong Bootstrap, cần thêm lớp d-flex cho thẻ cha bao bọc các phần tử bên trong.
* Điều chỉnh kích thước của các phần tử sử dụng lớp flex-\*. Ví dụ, flex-grow-1 sẽ cho phép phần tử tự mở rộng để lấp đầy không gian trống.
* Sử dụng lớp flex-wrap-\* để kiểm soát cách các phần tử xếp chồng lên nhau khi khoảng cách không đủ. Các giá trị có thể được sử dụng bao gồm:
* flex-wrap-nowrap: các phần tử không được kết hợp lại nếu không đủ khoảng cách.
* flex-wrap-wrap: các phần tử sẽ được xếp dọc theo nhiều hàng.
* flex-wrap-wrap-reverse: các phần tử sẽ được xếp từ dưới lên theo nhiều hàng
* Sử dụng lớp flex-row hoặc flex-column để thay đổi hướng của chiều sắp xếp.
* Sử dụng lớp order-\* để kiểm soát thứ tự các phần tử trong container. Các giá trị được sử dụng bao gồm các số từ 0 đến 12, với giá trị càng nhỏ thì phần tử sẽ hiển thị trước.
* Sử dụng lớp align-items-\* để canh chỉnh của các phần tử theo chiều dọc của container. Các giá trị có thể được sử dụng bao gồm:
* align-items-start: căn chỉnh các phần tử từ trên xuống.
* align-items-center: căn chỉnh các phần tử nằm giữa theo chiều dọc.
* align-items-end: căn chỉnh các phần tử từ dưới lên.
* Sử dụng lớp justify-content-\* để điều chỉnh khoảng cách giữa các phần tử theo chiều ngang của container. Các giá trị có thể được sử dụng bao gồm:
* justify-content-start: canh chỉnh các phần tử từ bên trái.
* justify-content-end: canh chỉnh các phần tử từ bên phải.
* justify-content-center: canh chỉnh các phần tử nằm giữa theo chiều ngang.
* justify-content-between: canh chỉnh các phần tử sao cho chúng được căn đều nhưng không có khoảng trống ở hai bên.
* justify-content-around: giống between nhưng có 2 khoảng trống bằng nhau ở 2 bên.

## Grid

* Grid được sử dụng để tạo cột và hàng, giúp phân chia bố cục nội dung trang web một cách dễ dàng. Trong Bootstrap, hệ thống grid gồm 12 cột.
* Đầu tiên, tạo hàng (row) bằng cách sử dụng class row.

Graphical user interface

Description automatically generated

* Mỗi hàng được chia thành các cột với các class col-\*, trong đó \* là số thể hiện chiều rộng của cột. Ví dụ: col-6 là cột chiếm 50% chiều rộng của hàng, col-4 là cột chiếm 33.33% chiều rộng của hàng, và col-12 là cột chiếm 100% chiều rộng của hàng.
* Có thể kết hợp các class để tạo ra các tỷ lệ phần trăm khác nhau cho các cột:
* col-\*: Cột theo tỷ lệ phần trăm trên tất cả các kích thước thiết bị.
* col-sm-\*: Cột theo tỷ lệ phần trăm trên thiết bị có độ rộng lớn hơn hoặc bằng 576px.
* col-md-\*: Cột theo tỷ lệ phần trăm trên thiết bị có độ rộng lớn hơn hoặc bằng 768px.
* col-lg-\*: Cột theo tỷ lệ phần trăm trên thiết bị có độ rộng lớn hơn hoặc bằng 992px.
* col-xl-\*: Cột theo tỷ lệ phần trăm trên thiết bị có độ rộng lớn hơn hoặc bằng 1200px.

Text

Description automatically generated

* Trong ví dụ này, khi độ rộng của trang web giảm xuống dưới 768px (điện thoại di động), cột đầu tiên sẽ chiếm 50% chiều rộng của hàng và cột thứ hai sẽ chiếm 50% chiều rộng của hàng. Khi độ rộng của trang web lớn hơn hoặc bằng 768px (màn hình máy tính bảng hoặc máy tính xách tay), cột đầu tiên sẽ chiếm 33.33% chiều rộng của hàng và cột thứ hai sẽ chiếm 66.67% chiều rộng của hàng.

## Position

* Các loại position:
* Relative: Di chuyển dựa trên vị trí gốc là vị trí ban đầu của phần tử
* Absolute: Di chuyển dựa trên vị trí gốc là vị trí của thẻ cha gần nhất có thuộc tính position là relative hoặc absolute.
* Fixed: Nằm ở một vị trí cố định trên trang web, không di chuyển khi người dùng cuộn trang web.

# III. Laravel cơ bản

## **1.Cấu trúc project Laravel**

|  |  |
| --- | --- |
| app | Chứa mã nguồn PHP của ứng dụng Laravel. Thư mục này cũng chứa các thư mục con để phân loại code, chẳng hạn như "Http" (chứa các Controllers, Middleware, Form Requests...), "Providers" (chứa các Service Provider), ... |
| bootstrap | Chứa các file khởi động cho ứng dụng, ví dụ như tạo ra instance của ứng dụng hoặc thiết lập các biến môi trường. |
| config | Chứa các file cấu hình cho ứng dụng, chẳng hạn như thông tin cơ sở dữ liệu, xác thực người dùng, mail,… |
| database | Chứa các migrations (để tạo hoặc chỉnh sửa cấu trúc CSDL), seeds (để khởi tạo dữ liệu mặc định) và các factories (để tạo data giả trong quá trình phát triển). |
| public | Chứa các file giao diện người dùng, JS, CSS, hình ảnh, fonts, ... Đây là nơi cung cấp các tài nguyên tĩnh cho ứng dụng. |
| resources | Chứa các file nguồn để phát triển giao diện người dùng như template blade, Sass/LESS, JS, hình ảnh, fonts, ... Đây là nơi để chỉnh sửa và phát triển các tài nguyên dùng cho giao diện trong ứng dụng. |
| routes | Chứa các file định tuyến của ứng dụng. File web.php quản lý các uri rút gọn và file api.php quản lý các API route. |
| storage | Chứa các file được sinh ra bởi hệ thống như session, file cache hay tệp tin log của ứng dụng. Hãy chắc chắn rằng nó có quyền write để lưu các tệp tin này. |
| tests | chứa các test case để kiểm tra tính ổn định, độ chính xác của ứng dụng Laravel. Laravel hỗ trợ nhiều loại test như unit, feature và browser testing. |

## 2. Một số lệnh trong Laravel

* Tạo project mới: composer create-project laravel/laravel tên-project.
* Tạo controller: php artisan make:controller tên-controller.
* Tạo model: php artisan make:model tên-model.
* Chạy project: php artisan serve.
* Cài đặt composer: composer i.
* Tạo key cho ứng dụng: php artisan key:generate
* Migrate và seed data cho database: php artisan migrate:refresh --seed

## 3. Controller

* Controller là một lớp dùng để xử lý yêu cầu từ route hoặc view, sau đó gọi tới Model để lấy dữ liệu cần thiết rồi trả về View và hiển thị lên giao diện người dùng.

## 4. Thực hành cắt giao diện từ html css vào Lavarel

* Sau khi đã code được một trang web máy tính đơn giản bên html css, tiến hành cắt giao diện vào laravel.
* Trong folder resource/view, tạo một folder layouts để chứa phần giao diện dùng chung, folder pages để chứa các trang html.

**Text

Description automatically generated**

* File welcome: Cắt phần head + body để sử dụng chung.

**Text

Description automatically generated**

Sử dụng @yield để hiển thị nội dung của các blade view kế thừa từ trang này.

* File caculator: Cắt phần giao diện của máy tính.

Graphical user interface, text

Description automatically generated with medium confidence

Dùng @extends để kế thừa blade view welcome. Sử dụng @section để chỉ định phần nội dung sẽ hiển thị ở vùng @yield(‘main’)

* Sau khi đã cắt xong giao diện, tiến hành tạo controller với phương thức index trả về giao diện vừa cắt.

Text

Description automatically generated

* Trong file web.php, đăng ký router trả về phương thức index của AdminController.

Text

Description automatically generated