Dự Án Website Bán Sách Online

# **PHẦN I – XÂY DỰNG GIAO DIỆN NGƯỜI DÙNG**

**TỔNG QUAN**

1. **Các thư viện bên thứ ba**

* [BootStrap 4 (v4.2.1)](https://getbootstrap.com/docs/4.2/getting-started/introduction/)
* [FontAwesome (v5.9.0)](https://fontawesome.com/download)
* [JQuery (v3.3.1)](https://jquery.com/)
* Facebook SDK

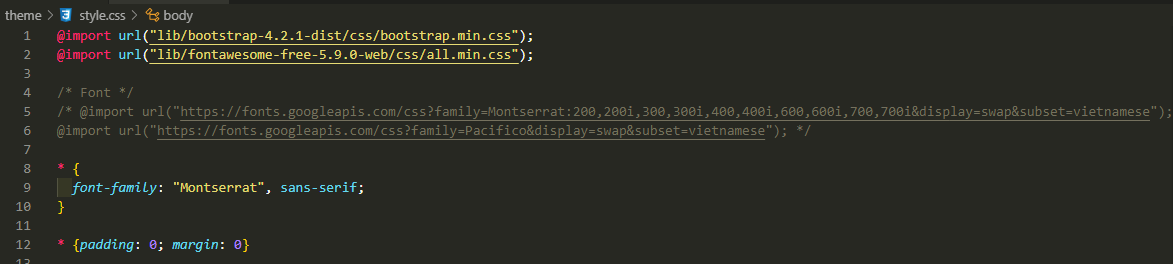
1. **Cấu trúc thư mục**

|  |  |  |
| --- | --- | --- |
|  |  | |\_\_img  |  |\_\_theme  | |  | |\_\_lib  | | |  | | |\_\_ bootstrap-3.3.7-dist  | | |  | | |\_\_ bootstrap-4.2.1-dist  | | |  | | |\_\_ fontawesome-free-5.9.0-web  | |  | |\_\_script.js  | |  | |\_\_style.css  |  |\_\_index.html |
|  |  |  |

**TRIỂN KHAI MÃ**

**1. Import các file CSS và JavaScript.**

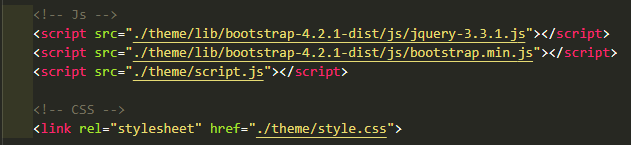
* **CSS:**

****Như đã nói ở phần cấu trúc thư mục, tất cả các file định kiểu khác sẽ được import vào file `*style.css*` để sử dụng, cụ thể ở đây là các file định kiểu được dựng sẵn của Boostrap 4 và FontAwesome.

Import các file CSS của Bootstrap 4 và FontAwesome vào một file duy nhất

* **HTML:**

Tiến hành import các file JavaScript vào file `*index.html*`. Cụ thể ở đây là JQuery và Bootstrap 4.

Để hiển thị được khung fanpage facebook cần import thư viện SDK của Facebook theo [hướng dẫn này](https://developers.facebook.com/docs/plugins/page-plugin).

Import các tệp JavaScript và định kiểu vào HTML

File `*script.js*` sử dụng để chứa các đoạn mã JQuery tùy chỉnh sau này.

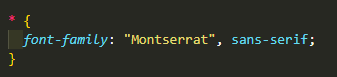
Phần định kiểu sẽ chỉ cần import 1 file duy nhất là `*style.css*`.

Từ giờ trở đi, các thao tác với mã CSS sẽ được hiểu là thực hiện trong file `style.css` còn các thao tác với JavaScript (JQuery) sẽ thực hiện trong file `script.js`

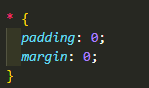
**2. Cài đặt font chữ và đặt lại lề cho trang web.**

Để sử dụng font chữ khác thay vì font chữ mặc định, tiến hành đặt lại font chữ trong `*style.css*`.

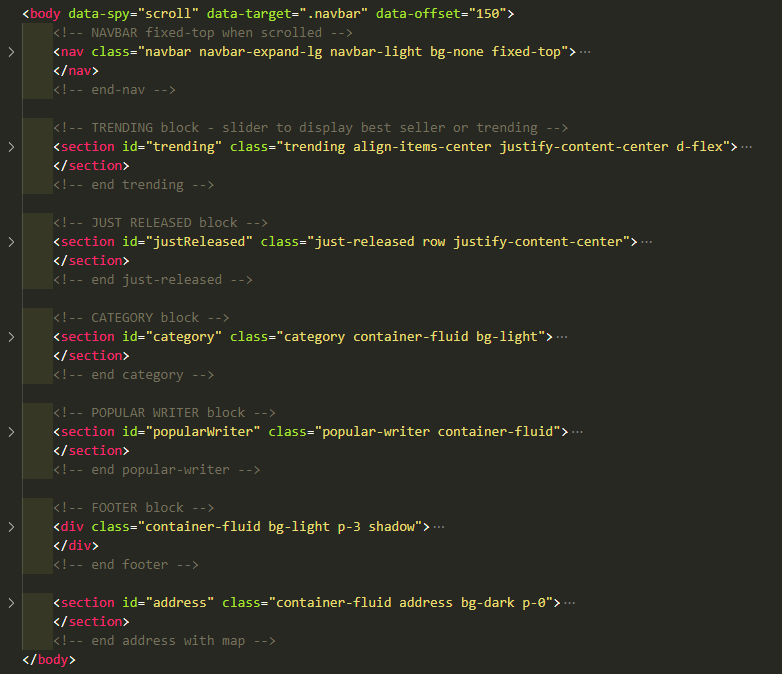
Import font chữ không có trong máy từ Internet (nếu cần)



Đặt lại font chữ cho toàn bộ văn bản trong trang web

Mục đích việc đặt lại các thông số margin và padding nhằm thuận tiện cho việc căn chỉnh lề sau này. Ngoài ra việc căn lề thủ công nhằm tránh các căn lề không đúng ý muốn thiết kế làm vỡ bố cục trang web khi sử dụng các thư viện định kiểu từ bên thứ 3.

Đặt các thông số căn lề về 0

**3. Cấu trúc các khối trong thân của trang Web.**

Hình trên cho thấy các khối chính sẽ xuất hiện trong phần body của trang web kèm theo các id của chúng. Cung cấp các id cho mỗi phần nhằm thực hiện định kiểu dễ dàng hơn đồng thời phục vụ cho thanh điều hướng được nói ngay sau đây.

**a. Thanh điều hướng (navbar).**



Thanh điều hướng trong suốt ở trạng thái ban đầu (màu nền là của phần hình ảnh bên dưới)

Bootstrap 4 cung cấp sẵn các class và thuộc tính để tạo hiệu ứng và định kiểu cho thanh navbar. Bên dưới là một số các class và thuộc tính chính được sử dụng kèm theo ý nghĩa của chúng:

data-spy="scroll" data-target=".navbar" data-offset="150"

Theo dõi hành động cuộn chuột và mục tiêu theo dõi là đối tượng chứa class navbar. Thuộc tính data-offset là vị trí bắt đầu tác dụng hiệu ứng khi cuộn chuột, cụ thể ở đây là 150px trước khi cuộn tới đối tượng (mặc định sẽ là 10px). Cụ thể khi cuộn chuột Bootstrap sẽ dựa vào id (được nêu ở phần trước đó) để phát hiện người dùng đang ở phần nào của trang web mà có hành động thích hợp. Bên dưới là một trong những hiệu ứng mà Bootstrap cung cấp cho thấy người dùng đang ở phần THỂ LOẠI. 

Thanh điều hướng cho thấy người dùng đang ở đâu trên trang nhờ hiệu ứng scrollspy

<nav class="navbar navbar-expand-lg navbar-light bg-none fixed-top">

navbar-expand-lg: thanh navbar sẽ mở rộng khi khung nhìn đạt kích thước large(lg) thay vì thu nhỏ lại thành dạng một nút toggle;

navbar-light: màu chữ và màu nền sáng;

bg-none: thanh navbar trong suốt;

fixed-top: thanh navbar sẽ luôn ở vị trí cao nhất của khung nhìn khi cuộn chuột.

    /\*\*

     \* Scroll Event Handling

     \*/

    $(window).scroll(*function* () {

*var* top = $('html,body').scrollTop();

*var* navHeight = $('.navbar').height();

        if (top > 0) {

            $('.navbar').removeClass('bg-none');

            $('.navbar').addClass('small-navbar bg-white shadow');

            $('#input-search').removeClass('w-75');

            $('#input-search').addClass('form-control-sm');

            $('.form-control-feedback').addClass('sm');

        }

Bên trên là một đoạn mã xử lý cuộn chuột bằng JQuery cho thanh điều hướng. Khi cuộn qua vị trí 0px thanh điều hướng sẽ có 2 hiệu ứng chính là đổ bóng và chuyển nền màu trắng (thay vì trong suốt) (ngoài ra còn thay đổi cỡ chữ, màu chữ, và kích thước ô tìm kiếm) tất cả đều dựa vào 2 hàm đơn giản là addClass/removeClass.

Thanh điều hướng khi cuộn xuống với background trắng và hiệu ứng đổ bóng (shadow)

**b. Khối #trending**

* + 1. **Phần Hero image**

Phần mở đầu (khối #trending) của trang web sử dụng một [Hero image](https://en.wikipedia.org/wiki/Hero_image) làm background. Do đó khối chứa nó sẽ được căn chỉnh kích thước luôn khớp đúng bằng 100% khung nhìn của người dùng.

Tablet devides

Mobile devides

Desktop devides

***Mã CSS:***

#trending {

*background-image*: url("../img/slider.jpg");

*background-size*: cover;

*background-repeat*: no-repeat;

*width*: 100%;

*height*: 100%;

}

Ngoài ra, dưới đây là đoạn mã ***JQuery*** xử lý việc co dãn kích thước khối theo khung nhìn khi người co giãn cửa sổ trình duyệt:

    /\*\*

     \* Hero background scaling

     \*/

    $('.trending').css({ 'height': heightScreen });

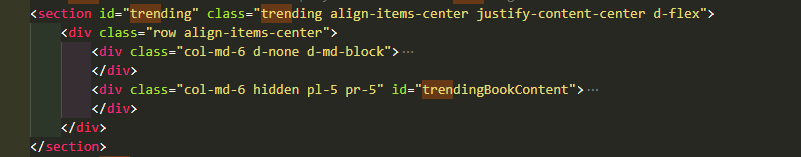
    $(window).resize(*function* () {

*var* heightScreen = $(window).height();

        $('.trending').css({ 'height': heightScreen });

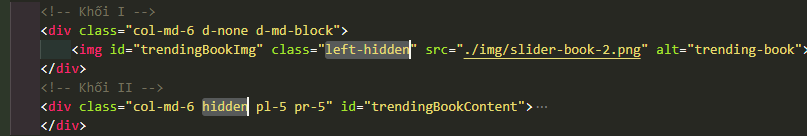
    });

* + 1. **Phần bố cục**

Khối #trending được chia thành hai khối nhỏ hơn bằng nhau như hình dưới. Theo quy ước của Bootstrap một hàng luôn được chia thành 12 cột, do đó tiến hành chia khối #trending thành 2 cột với chiều rộng bằng nhau là 6. Ngoài ra, chỉ định khối thứ I sẽ chỉ hiển thị trên các khung nhìn có kích cỡ lớn hơn trung bình (md: chiều rộng 992px trở lên) do đó sử dụng 2 class “d-none d-md-block” cho khối thứ I.

* + 1. **Hiệu ứng Lazyload khi cuộn chuột**

Trong trường hợp này, hai khối con sẽ được giấu ở bên trái khung nhìn(hình ảnh của Khối I) của người dùng hoặc ẩn đi (Khối II) bằng cách sử dụng CSS, và sẽ chỉ xuất hiện(tải) khi được trang được xong sau 30ms bằng cách sử dụng JQuery. (\*)



Đoạn mã ***Jquery*** dưới đây sẽ tạo hiệu ứng lazyload cho 2 khối con trending:

   setTimeout(() *=>* {

        /\*\*

         \* Display trending content immediately after the page loads

         \*/

        $('#trendingBookContent, #trendingBookImg').removeClass(allHiddenClass);

        $('#trendingBookContent, #trendingBookImg').addClass('shown');

    }, 30);

***CSS:***

.shown {

*opacity*: 1 !important;

*transition*: all cubic-bezier(0, 0, 0.21, 0.84) 0.8s;

}

.left-hidden {

*opacity*: 0;

*transform*: translateX(-100%);

*width*: 0;

}

/\*

  Others hidden (top, right, bottom)...

\*/

.hidden {

*opacity*: 0;

}

***Các khối còn lại của trang có cách xử lý tương tự.***

**Bài viết có sử dụng các Tài liệu tham khảo sau:**

* [**Nhúng Map sử dụng API của Google**](https://developers.google.com/maps/documentation/embed/start?hl=vi)
* [**Nhúng fanpage sử dụng plugin của Facebook**](https://developers.facebook.com/docs/plugins/page-plugin/?locale=vi_VN)