# Các thành phần trang web

- header

- navigator

- main-content

+ content

+ side-bar

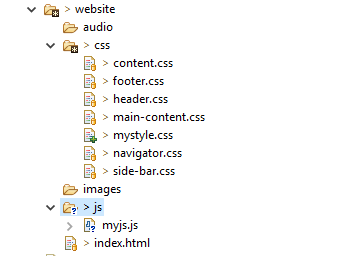
- footer

Cách thành phần chính này : đặt width theo % để có thể có dãn và float để có thể responsive

# Các bước tạo giao diện

## Cấu hình thư mục website

Cấu hình các cấp của thư mục website:



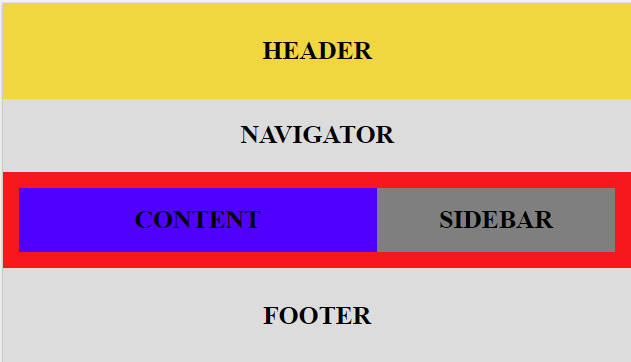
Index.html là trang gọi đầu tiên và nó sẽ gọi các thành phần của các thư mục cùng cấp.

## Định hình cách thành phần của trang chủ

Trong suy nghĩ chúng ta có thể mường tượng các thành phần của trang web.

Dựa vào các thành phần website ở trên và các trang web phổ biến chúng ta chia layout website như sau:

Link: <https://www.w3schools.com/code/tryit.asp?filename=FRNL90KKQ6DX>



Định hình sẽ có 3 phần chính trang web như trên.

## Dựng HTML với các thành phần đã xác định

Chúng ta có thể hoàn toàn sử dụng lại những link trên để có thể sửa và cấu hình mã HTML. Tuy nhiên chúng ta sẽ viết mã mới như sau;

### Tạo nội dung cơ bản

|  |
| --- |
| <!DOCTYPE html>  <html lang=*"en"*>  <head>  <meta charset=*"UTF-8"*>  <meta name=*"viewport"* content=*"width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no"*>  <title>Website quản lý sách</title>  </head>  <body>    </body>  </html> |

### Import các JS, CSS cho head

|  |
| --- |
| <head>  <meta charset=*"UTF-8"*>  <meta name=*"viewport"*  content=*"width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no"*>  <title>Website quản lý sách</title>  <!--CSS LIB START -->  <link rel=*"stylesheet"*  href=*"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"*>  <!--CSS LIB END -->  <!--JS LIB START -->  <script  src=*"https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"*></script>  <!--JS END -->  <!--MY JS CSS START -->  <link rel=*"stylesheet"* type=*"text/css"* href=*"css/mystyle.css"*>  <script type=*"text/javascript"* src=*"js.myjs.js"*></script>  <!--MY JS CSS END -->  </head> |

### Tạo HTML các Grid (khung trong layout) bằng div

Ở đây chúng ta có các div với id mã chúng ta định hình ở phần trước và viết mã chúng trong body

|  |
| --- |
| <body>  <div id=*"main"*>  <!--header START -->  <div id=*"header"*>  <div class=*"container text-center"*>  <h1>Web bán sơn số 1 Việt nam</h1>  <h4>Sơn tuyệt hảo cho mọi gia đình</h4>  </div>  </div>  <!--header END -->  <!--navigator START -->  <div id=*"navigator"*>  <div class=*"container"*>NAVIGATOR</div>  </div>  <!--navigator END -->  <!--main-content START -->  <div id=*"main-content"*>  <div class=*"container"*>  <!--content START -->  <div class=*"content"*>CONTENT</div>  <!--content END -->    <!--side-bar START -->  <div class=*"side-bar"*>SIDE-BAR</div>  <!--side-bar END -->  </div>  </div>  <!--main-content END -->  <!--footer START -->  <div id=*"footer"*>  <div class=*"container"*>FOOTER</div>  </div>  <!--footer END -->  </div>  </body> |

Chúng ta nên có comment START – END để chúng ta có thể dò mã khi code HTML nhiều lên, nếu không có comment đầu cuối chúng ta không thể kiểm sóa được code.

Sau khi kết thúc ta có toàn mã như sau: <https://www.w3schools.com/code/tryit.asp?filename=FRNLA0UR4S9H>

## Tạo CSS cơ bản

### Import CSS và reset CSS

Ở bước này cần tạo các file .css (nếu thêm) và import chúng vào trang web

Nội dung như sau:

|  |
| --- |
| @charset *"UTF-8"*;  /\* IMPORT START\*/  @import *url("footer.css")*;  @import *url("header.css")*;  @import *url("navigator.css")*;  @import *url("main-content.css")*;  @import *url("sidebar.css")*;  @import *url("content.css")*;  @import *url("footer.css")*;  /\* IMPORT END\*/  /\* RESET CSS START \*/  \*{  padding: *0px*;  margin: *0px*;  box-sizing: *border-box*;  }  **a**{  text-decoration: *none*;  }  **ul** {  list-style-type: *none*;  }  /\* RESET CSS END \*/  /\* CUSTOM CSS CLASS LIKE BOOTSTRAP START \*/  *.container* {  padding-right: *15px*;  padding-left: *15px*;  margin-right: *auto*;  margin-left: *auto*;  }  *.clearfix*{  clear: *both*;  }  *.left*{  float: *left*;  }  *.right*{  float: *right*;  }  *.text-center*{  text-align: *center*;  }  /\* CUSTOM CSS CLASS LIKE BOOTSTRAP END \*/ |

Ở trên chúng ta reset các CSS mặc định. Ví dụ thẻ a không được gạch chân.

Ngoài ra chúng ta đã import các file ”.css” thay trong file ”mystyle.css” thay vì import chúng trong file ”inde.html”.

Ta có thể custom một số class để tạo dựng độ rộng theo %

|  |
| --- |
| *.container* {  padding-right: *15px*;  padding-left: *15px*;  margin-right: *auto*;  margin-left: *auto*;  }  *.clearfix*{  clear: *both*;  }  *.left*{  float: *left*;  }  *.right*{  float: *right*;  }  *.column1*{  width: *8.33%*;  }  *.column2*{  width: *16.6%*;  }  *.column3*{  width: *24.9%*;  }  *.column4*{  width: *33.2%*;  }  *.column5*{  width: *41.5%*;  }  *.column6*{  width: *49.8%*;  }  *.column7*{  width: *58.1%*;  }  *.column8*{  width: *66.4%*;  }  *.column9*{  width: *74.7%*;  }  *.column10*{  width: *83%*;  }  *.column11*{  width: *91.3%*;  }  *.row*{  width: *100%*;  }  *.text-center*{  text-align: *center*;  } |



### Tạo CSS để dựng khung

Sử dụng các class custom ta có thể tạo một khung giao diện như sau:



Điều chú ý nhất là chúng ta sử dụng width theo % bằng cách sử dụng cách class %.

Chúng ta sử dụng class left và right để định vị #content và #side-bar cho nội dung web.

Điều thứ 2 đó là chúng ta sử dụng một div class=”clear” để clear float trong #main-content.

Giao diện hiện tại ngắn và màu sắc xấu. Khi ấy chúng ta có thể sửa sau này.

## Responsive cho layout khi còn chưa có nội dung

Để sử dụng responsive ta đã sử dụng:

|  |
| --- |
| <meta name=*"viewport"*  content=*"width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no"*> |

Tiếp theo chúng ta sẽ CSS cho chúng bằng @media (Media Query trong CSS)

|  |
| --- |
| /\*RESPONSIVE SCREEN START\*/  @media only screen and (max-width: 768px){  *.left***,** *.right*{  float: *none*;  width: *100%*;  }  }  /\*RESPONSIVE SCREEN END\*/ |

Với đoạn code này chúng ta đã clear thuộc tính float right và left. Và kéo chúng 100% để có thể chung một hang dọc với nhau trên thiết bị nhỏ max-width: 768px.



Kết quả cho việc sử dụng responsive

## Tạo HTML CSS và nội dung cho các thành phần

Chúng ta sẽ chia nhỏ từng thành phần và làm lần lượt chúng. #main-content là phần chứa nội dung website phức tạp nhất nên chúng ta sẽ để chúng cuối cùng.

Và khi thay đổi sửa chúng ta cũng làm lần lượt như vậy. Cứ mỗi lần sửa thì trang chủ của chúng ta sẽ hoàn thiện hơn

Chúng ta sẽ tạo dựng header và navigator.

### Tạo navigator

Trong navigator thường chứa menu. Nên chúng ta sẽ sử dụng menu ngang để đưa vào.

Chúng ta sẽ phải đưa mã HTML, CSS vào #navigator. Và để đảm bảo trang giữ được responsive thì chúng ta phải tạo một menu responsive.

Hãy làm theo hướng dẫn này bạn sẽ đạt được navigator responsive.

***Tạo navigator với menu ngang reponsive.docx***

Tuy sau khi thành công với navigator thì bạn cần phải cắt lóc CSS, JS cho vào các file quan trọng.

CSS bạn đưa vào navigator.css (Nhớ cắt phần dành riêng cho navigator thôi nhé)



### Tạo header

Thực ra bạn đã tạo một header khi ở hướng dẫn trên thành công rồi. Nhưng để dự án web thì header không đơn thuần như vậy.

Tài liệu tiếp theo hướng dẫn các bạn tạo header responsive:

***Tạo header đơn giản reponsive.docx***

### Tạo footer

### Tạo main-content

### Tạo side-bar

### Tạo content

## Tái cấu trúc CSS cho các class và lược bỏ những code thừa

# Link

1. Hướng dẫn sơ bộ tạo responsive

<https://thachpham.com/web-development/html-css/lam-giao-dien-respoonsive.html>

1. Hướng dẫn kĩ về kĩ thuật responsive: <https://codetot.net/responsive-web-design/>