



VIETNAM
AUSTRALIA
Vocational College

Slides 2.3 - Thiết kế Responsive cho các thiết bị Mobile

Mentor: Nguyễn Bá Minh Đạo



Nội dung

1. Tổng quan về thiết kế Responsive
2. Thiết kế Responsive cho các thiết bị Mobile
3. Tạo một bố cục linh hoạt (Fluid Layout)
4. Sử dụng các hình ảnh linh hoạt(Flexible Images)
5. Thiết kế Responsive với chiến lược Mobile-First
6. Thêm các thẻ mô tả Meta
7. Kiểm tra Webpages tại các Viewports khác nhau



Tổng quan về thiết kế Responsive

❑ Giới thiệu:

- ♦ Theo thống kê từ searchenginewatch.com, số lượng tất cả các thiết bị mobile sử dụng để lướt web đã vượt qua số lượng thiết bị desktop lướt web vào năm 2004.
- ♦ Tuy nhiên, các trang web yêu cầu người dùng phải phóng to, thu nhỏ, và cuộn lên xuống để đọc và chuyển hướng trang. -> rất khó sử dụng, vậy phải làm sao?
- ♦ Cách tiếp cận **thiết kế đáp ứng (Responsive Design)** có thể giải quyết vấn đề này.
- ♦ **Responsive design** có khả năng **tạo một trang web hoạt động tốt với các kích thước màn hình và trình duyệt khác nhau** (mobile, tablet, laptop, desktop).

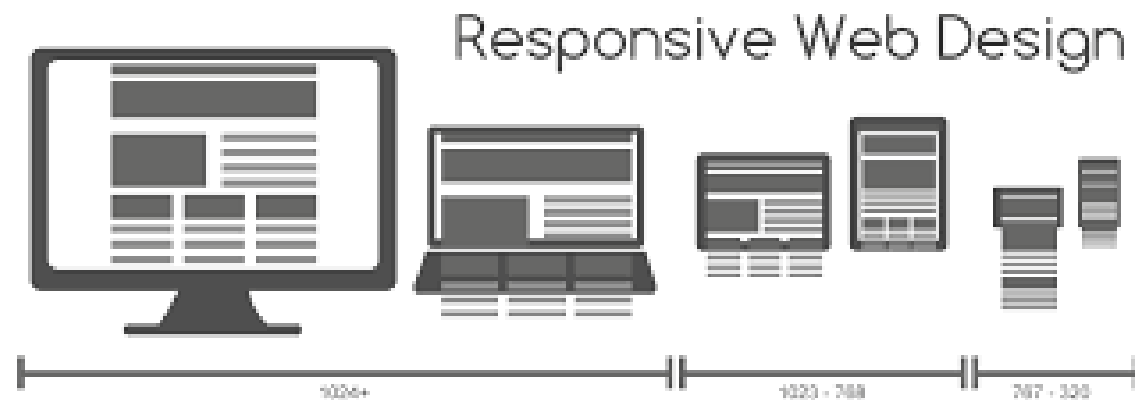




Tổng quan về thiết kế Responsive

❑ Lợi ích:

- ◆ Bằng cách áp dụng các nguyên tắc **Responsive Design**, chúng ta sẽ:
 - **Cung cấp trải nghiệm người dùng tối ưu về trang web** bất kể thiết bị hoặc trình duyệt được sử dụng.
 - **Trang web và nội dung đáp ứng** với **kích thước màn hình thiết bị của người dùng**.
 - **Giảm thiểu việc cuộn** và **thu phóng không cần thiết**, giúp **việc đọc và tương tác** với **trang web** trở nên **thuận tiện** và **trực quan nhất có thể**.

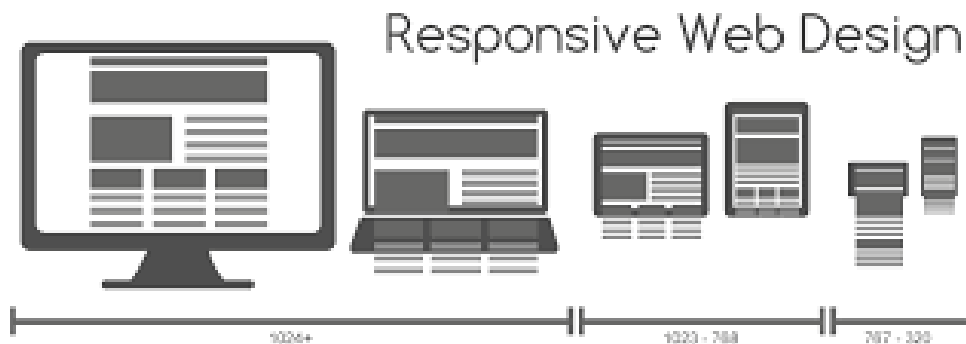




Tổng quan về thiết kế Responsive

❑ Đặc điểm:

- ♦ Responsive Web Design **không phải** là **một bộ quy tắc cụ thể** mà được **đặc trưng** tốt hơn **như một triết lý** được **tinh chỉnh liên tục** khi các tiêu chuẩn, trình duyệt và công nghệ **HTML, CSS** phát triển và cải tiến.
- ♦ Nhưng về cơ bản, Responsive Web Design vẫn sẽ **nổi bật được ba khái niệm** sau:
 - Bố cục linh hoạt (**Fluid layout**)
 - Truy vấn phương tiện truyền thông (**Media Queries**)
 - Hình ảnh linh hoạt (**Flexible images**)

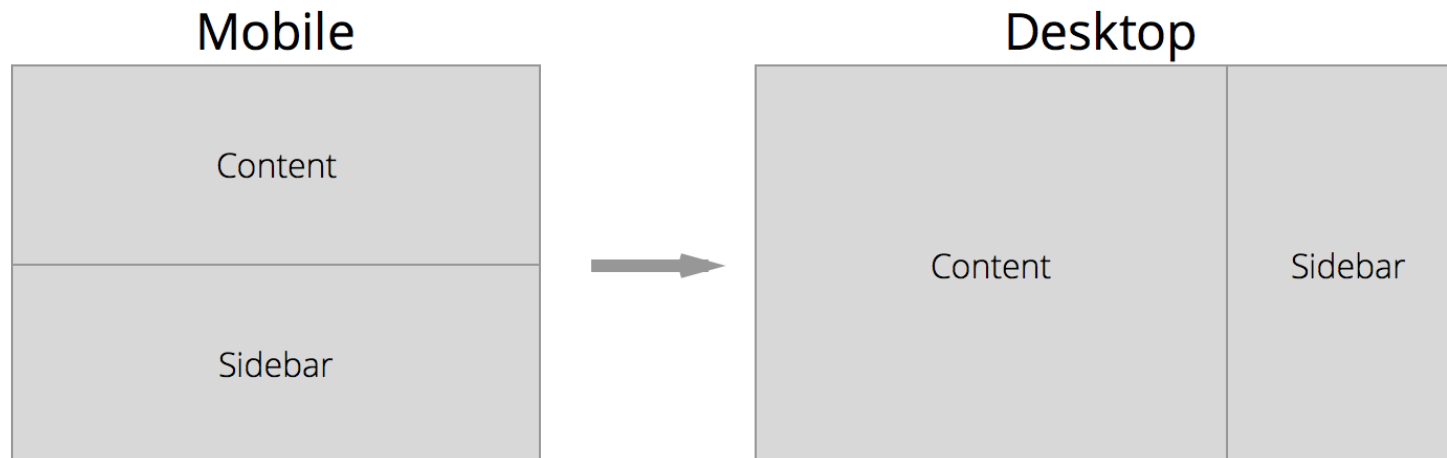




Thiết kế Responsive cho các thiết bị Mobile

❑ Đặt vấn đề:

- ♦ Sự phát triển bùng nổ của các trình duyệt di động nhanh chóng **tạo ra các vấn đề nghiêm trọng** vì các chế độ xem nhỏ hơn **thường yêu cầu phóng to và cuộn quá mức**.
- ♦ Một trong các **kỹ thuật để giải quyết vấn đề** là **xây dựng một trang web hoàn toàn riêng biệt, song song** được tối ưu hóa cho người dùng mobile gọi là trang web mobile.
- ♦ Các trang web di động thường được xác định bằng **tiền tố m.** hoặc **mo.** trong URL.
Ví dụ: <https://m.facebook.com/>

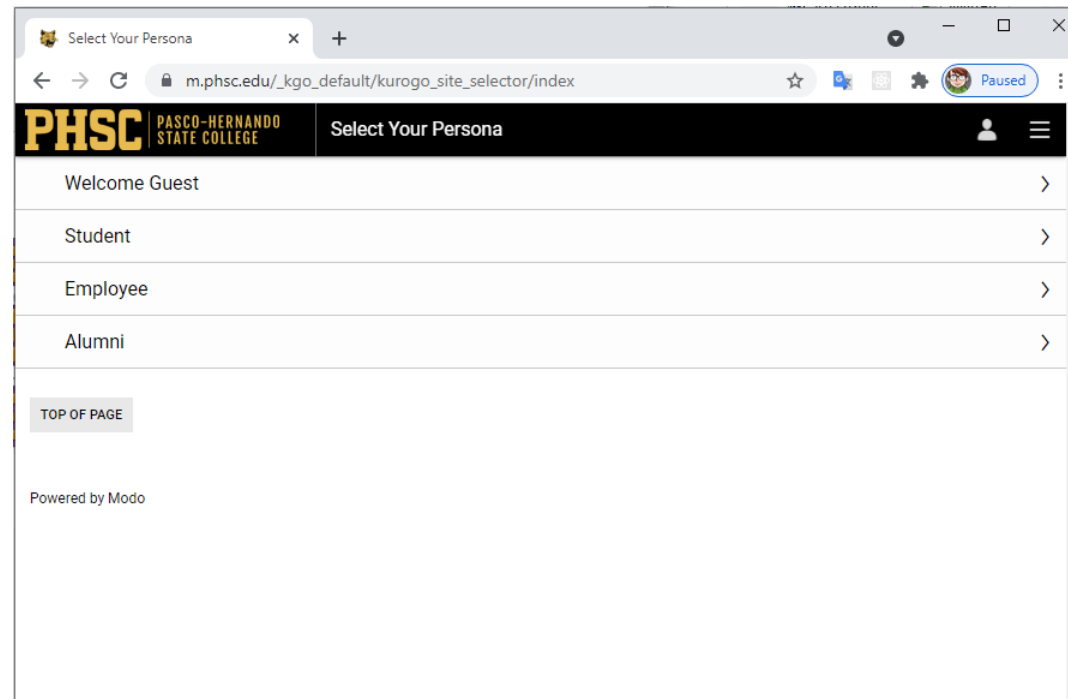




Thiết kế Responsive cho các thiết bị Mobile

❑ Đặt vấn đề:

- ♦ Trang web hiện tại của một tổ chức lớn có thể **chứa** một **lượng lớn các trang đã được phát triển từ lâu** bằng cách dùng các thiết kế không **responsive** dành cho **desktop**.
- ♦ Việc sửa đổi từng trang đó để kết hợp các kỹ thuật **responsive** thường **là một nỗ lực lớn** và **là một giải pháp không khả thi**. -> vậy phải là sao?

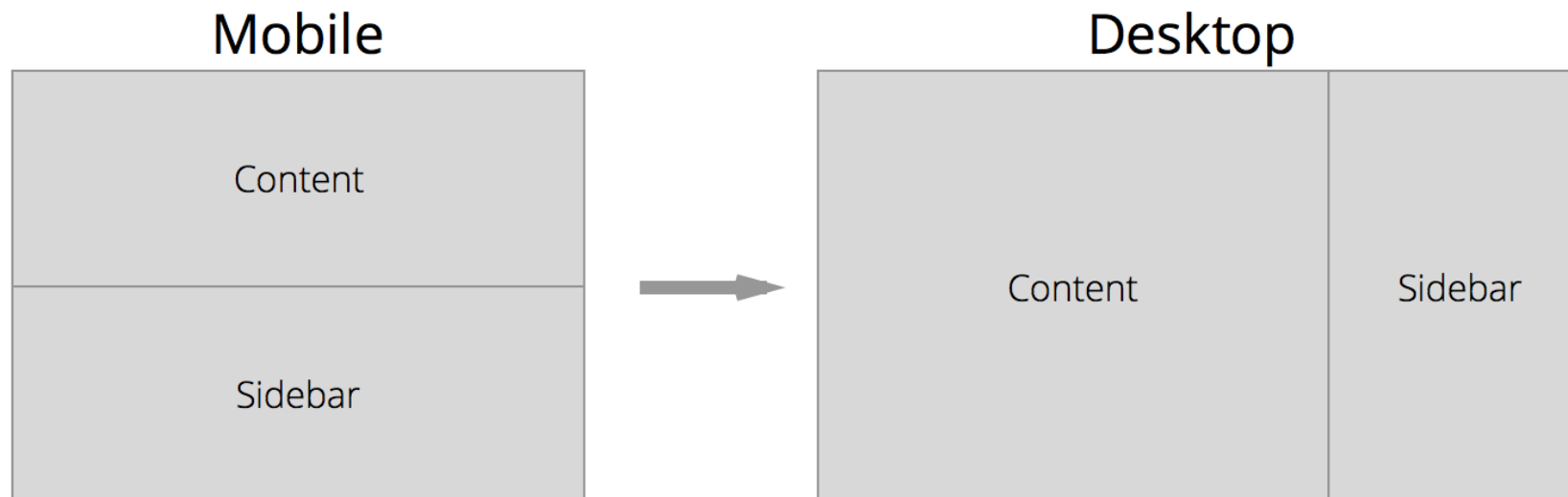




Thiết kế Responsive cho các thiết bị Mobile

❑ Giải pháp và vấn đề phát sinh khác:

- ♦ Phương pháp tiếp cận trang web dành cho mobile đó là **xây dựng một trang web mới**, song song từ đầu, và đặc biệt **thiết kế đó dành cho các mobile viewport**.
- ♦ Cách tiếp cận này **nhANH hơn nhiều** so với việc chuyển đổi một **trang web lớn gồm các trang không responsive** -> **nhưng vẫn sẽ tăng công việc** vì duy trì 2 trang 1 lúc.
- ♦ Vậy phải làm sao để **vẫn hỗ trợ được mobile** và **không cần duy trì 2 trang web** 1 lúc?





Thiết kế Responsive cho các thiết bị Mobile

❑ Giải pháp hiệu quả:

- ♦ Chiến lược **mobile-first** là một giải pháp hiệu quả để các nhà phát triển website phải tập trung vào nội dung cốt lõi, thiết yếu trước tiên cho mobile viewport.
- ♦ Dần dần **thêm nhiều nội dung hơn** khi chế độ xem phát triển.
- ♦ Sau đó, một nhà phát triển web **sử dụng các truy vấn phương tiện** để **thêm kiểu cho các khung nhìn lớn hơn dần dần**, tiến triển từ **tablet -> laptop -> desktop**.





Tạo một bố cục linh hoạt (Fluid Layout)

□ Giới thiệu:

- ♦ Một trang web có bố cục nổi lỏng (**liquid layout**), còn được gọi là bố cục linh hoạt (**fluid layout**), có thể thay đổi chiều rộng dựa trên kích thước của chế độ xem (**viewport**).
- ♦ Thiết kế **responsive** được xây dựng dựa trên bố cục linh hoạt (**fluid layout**).





- ♦ Các nhà phát triển web mới bắt đầu có thể nhằm lẫn khả năng của truy vấn phương tiện (**Media Queries**) với hành vi của bố cục lỏng (**Fluid Layout**).

- ◆ Nhưng **một trang web** có thể có bố cục lỏng (**Fluid Layout**) mà không cần sử dụng truy vấn phương tiện (**Media Queries**).

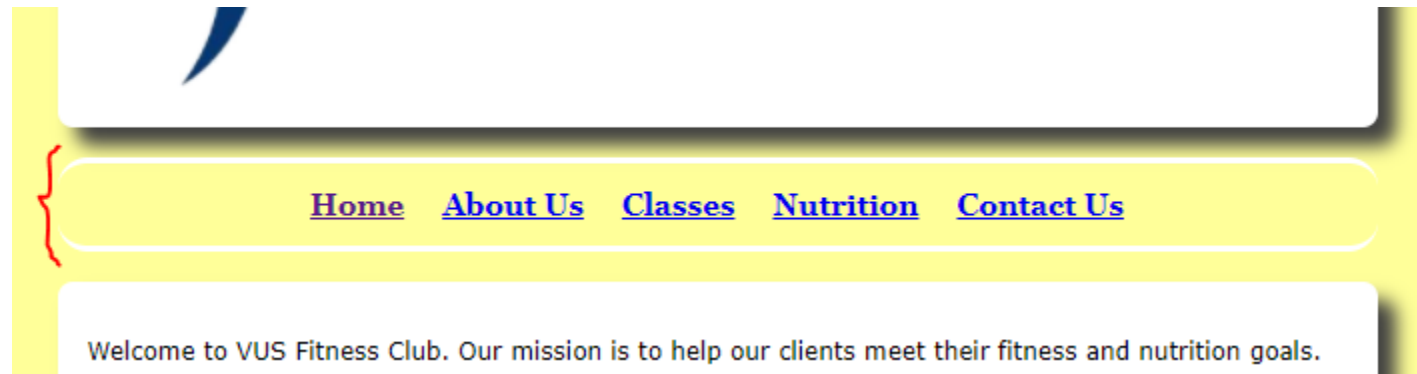




Tạo một bố cục linh hoạt (Fluid Layout)

❑ Các phần tử điều hướng trong Fluid Layouts:

- ♦ Để làm cho các liên kết điều hướng hấp dẫn hơn và dễ tìm thấy trên trang web, bạn **có thể định dạng chúng để chúng trông giống các nút hơn là liên kết văn bản.**
- ♦ Mỗi nút cũng phải có thêm không gian để **hiển thị văn bản liên kết rõ ràng và để tách nó khỏi các nút liền kề.**
- ♦ Hệ thống điều hướng trong chế độ xem điện thoại có thể **sử dụng cùng một bảng màu** nhưng mỗi nút phải lớn hơn.
- ♦ Nút lớn hơn **để người dùng có thể tìm và chọn từng nút** một cách **dễ dàng** trên màn hình nhỏ của điện thoại.

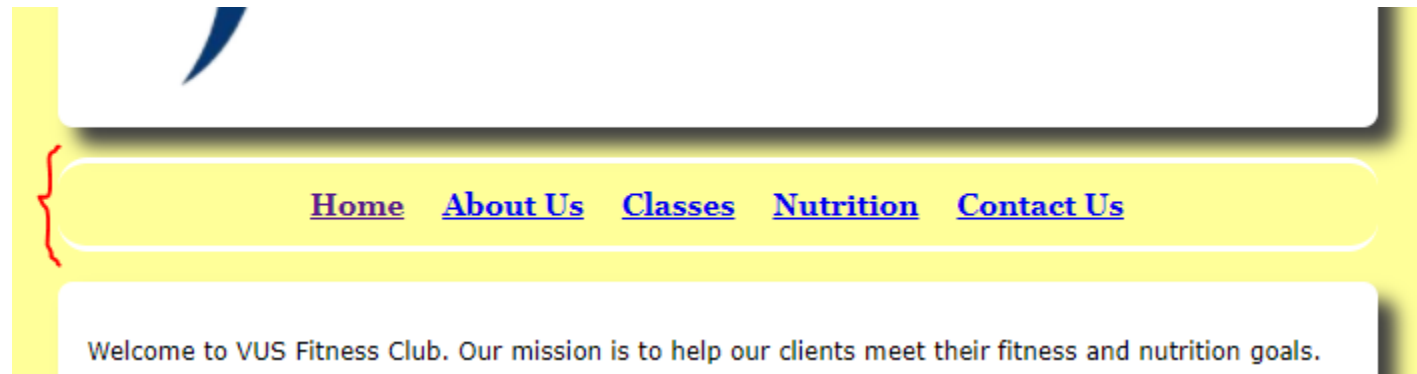




Tạo một bố cục linh hoạt (Fluid Layout)

❑ Các phần tử điều hướng trong Fluid Layouts:

- ♦ Để thực hiện thiết kế Fluid Layout, trước tiên **hãy viết mã các liên kết điều hướng dưới dạng danh sách không có thứ tự**.
- ♦ Đây là một kỹ thuật thiết kế web phổ biến mô tả nội dung theo ngữ nghĩa và **cho phép** bạn **định dạng từng liên kết bằng** các kiểu **CSS**.
- ♦ Khi điều hướng được mã hóa dưới dạng danh sách không có thứ tự, bạn có thể **thêm CSS** để **định dạng các mục** trong **danh sách** mà **không cần hiển thị dấu đầu dòng**.
- ♦ Áp dụng màu nền, màu văn bản và đường viền tròn để **làm cho mỗi liên kết trông giống như một nút**. Sau đó, bạn sẽ loại bỏ các gạch chân ở các liên kết theo mặc định.





Tạo một bố cục linh hoạt (Fluid Layout)

❑ Các phần tử điều hướng trong Fluid Layouts:

♦ Thực hành áp dụng kỹ thuật chỉnh sửa điều hướng để tạo Fluid Layout cho khu vực điều hướng của dự án **VUS Fitness Club** lần lượt tại 3 trang: Home, About Us, Contact Us

```
<nav>
  <a href="index.html">Home</a> &nbsp; &nbsp; &nbsp;
  <a href="about.html">About Us</a> &nbsp; &nbsp; &nbsp;
  <a href="class.html">Classes</a> &nbsp; &nbsp; &nbsp;
  <a href="nutrition.html">Nutrition</a> &nbsp; &nbsp; &nbsp;
  <a href="contact.html">Contact Us</a>
</nav>
```

```
<nav>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="about.html">About Us</a></li>
    <li><a href="class.html">Classes</a></li>
    <li><a href="nutrition.html">Nutrition</a></li>
    <li><a href="contact.html">Contact Us</a></li>
  </ul>
</nav>
```

Quan sát menu
của dự án **VUS
Fitness Club** sau
khi chỉnh sửa!

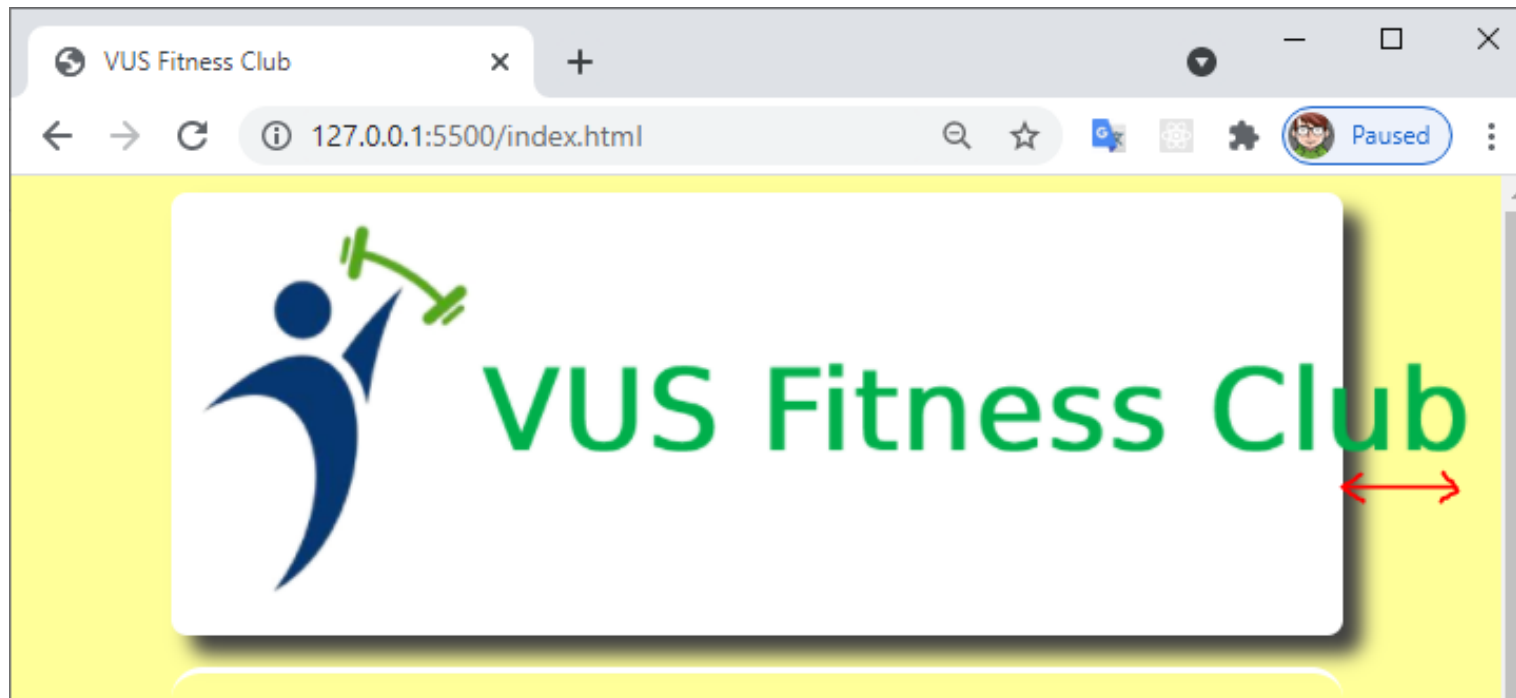
- [Home](#)
- [About Us](#)
- [Classes](#)
- [Nutrition](#)
- [Contact Us](#)



Sử dụng các hình ảnh linh hoạt (Flexible Images)

❑ Đặt vấn đề:

- ♦ Nếu bạn **giảm chiều rộng** của **cửa sổ trình duyệt**, logo VUS Fitness Club **sẽ bị cắt** vì nó không thay đổi kích thước đối với chế độ xem (viewport).
- ♦ Bạn **có thể giải quyết vấn đề đó** bằng cách **thêm các hình ảnh linh hoạt (Flexible Images)** vào thiết kế hiện tại, **một kỹ thuật chính** khác **của chiến lược Responsive Design**

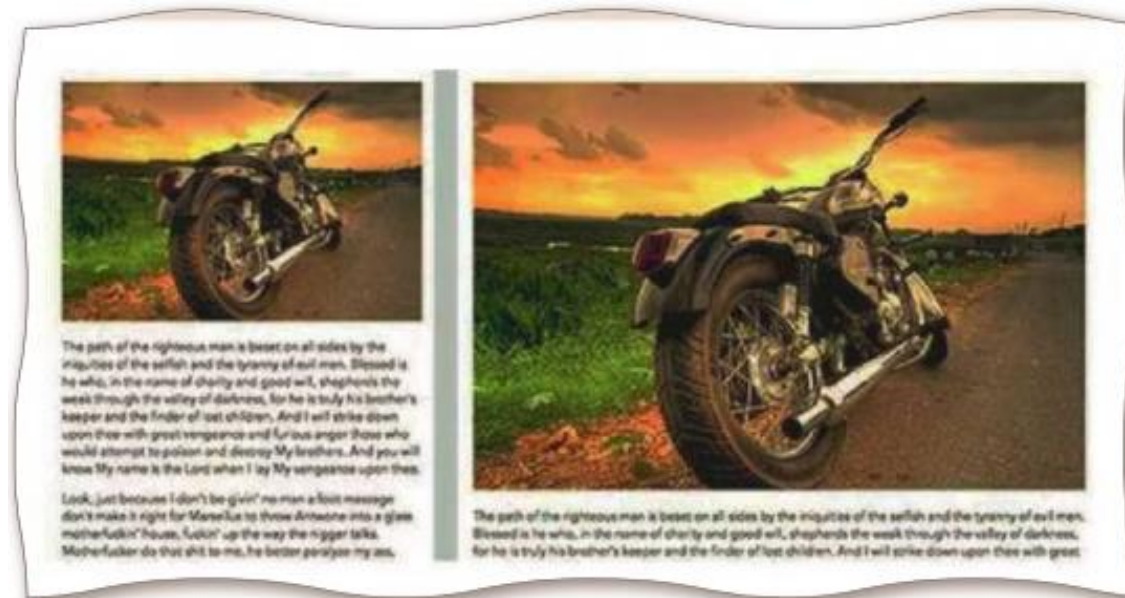




Sử dụng các hình ảnh linh hoạt (Flexible Images)

❑ Đặt vấn đề:

- ◆ Hình ảnh linh hoạt (**Flexiable Image**) tự thay đổi kích thước để phù hợp với kích thước của khung nhìn (**viewport**). Flexiable Image thì khá dễ thực hiện.
- ◆ Để tạo một hình ảnh linh hoạt (flexiale image), ta làm như sau:
 - **Xóa** giá trị 2 thuộc tính **height** và **width** của thẻ **img** trong **file HTML** đi.
 - **Thêm** định kiểu **CSS cho hình ảnh** trong tệp **CSS** để mang lại sự linh hoạt mong muốn như sau: **max-width: 100%;**

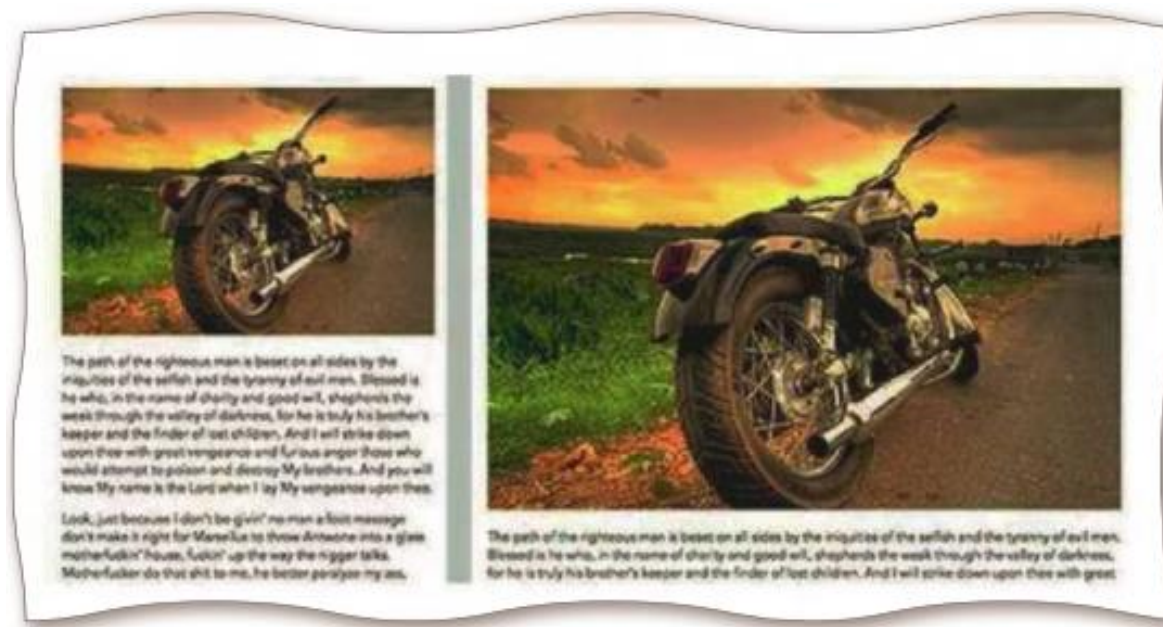




Sử dụng các hình ảnh linh hoạt (Flexible Images)

❑ Đặt vấn đề:

- ◆ Bằng cách **đặt chiều rộng** của hình ảnh thành **100%**, hình ảnh sẽ **tự động kéo dài để lấp đầy 100% chiều rộng** của phần tử vùng chứa.
- ◆ Chiều cao **tự động phát triển** theo **tỷ lệ với chiều rộng**.
- ◆ Sử dụng thuộc tính **max-width** thay vì thuộc tính **width** để giới hạn hình ảnh ở **100%** kích thước thực của nó trong trường hợp chế độ xem thậm chí còn lớn hơn.





Sử dụng các hình ảnh linh hoạt (Flexible Images)

❑ Thêm hình ảnh linh hoạt vào dự án:

♦ Thực hành áp dụng kỹ thuật sử dụng hình ảnh linh hoạt thay thế các hình ảnh trong dự án **VUS Fitness Club** lần lượt tại 3 trang: Home, About Us, Contact Us

- Xóa hết giá trị 2 thuộc tính **height** và **width** của tất cả thẻ **img** ở 3 trang trên đi.

```
<!-- Use the header area for the website name or logo -->
<header>
  <a href="index.html"></a>
</header>
```

```
<h1>Weights</h1>

<p>Our facility includes a weight training area with several weight options. Build lean muscle
```

```
<h1>Cardio</h1>

<p>Burn fat through cardio workouts. If you need to lose 20 lbs or more, include at least 30 min
```

```
<h1>Personal Training</h1>

<p>Our certified personal trainers work with you one on one to help you obtain your fitness goals
```



Sử dụng các hình ảnh linh hoạt (Flexible Images)

❑ Thêm hình ảnh linh hoạt vào dự án:

♦ Thực hành áp dụng kỹ thuật sử dụng hình ảnh linh hoạt thay thế các hình ảnh trong dự án **VUS Fitness Club** lần lượt tại 3 trang: Home, About Us, Contact Us

▪ **Thêm** định kiểu **CSS cho thẻ img** trong tệp CSS để mang lại sự linh hoạt mong muốn cho tất cả các hình ảnh như sau: **max-width: 100%;**

bộ chọn img

mở CSS

chú thích

```
/* Style to create a fluid image */  
img {  
    max-width: 100%;  
}
```

đóng CSS

thuộc tính max-width

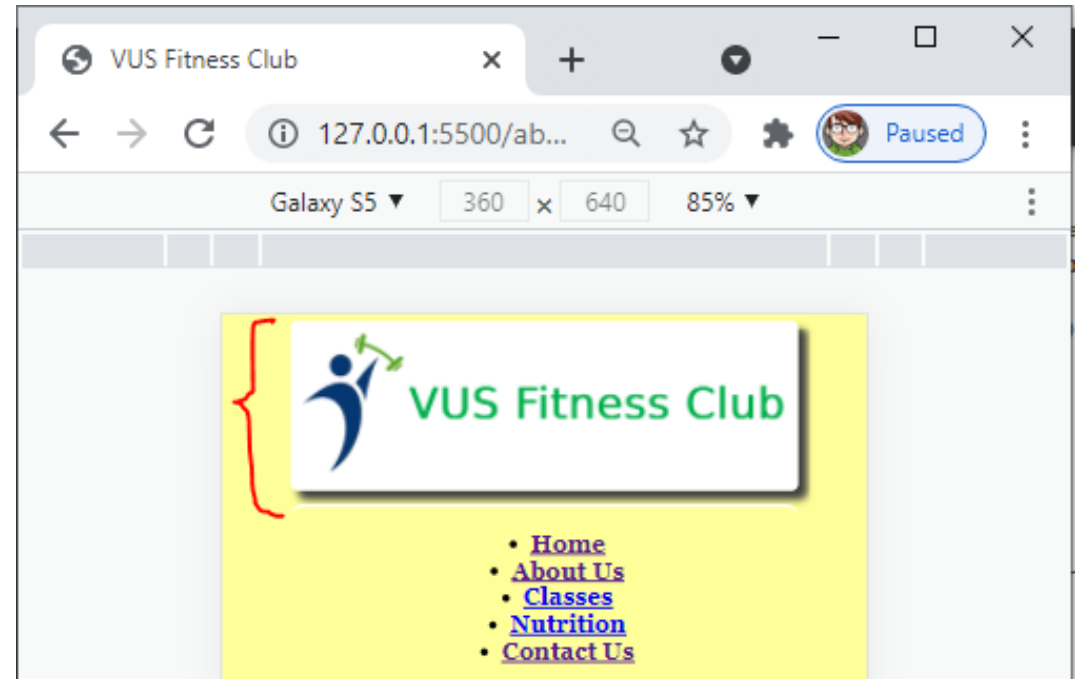
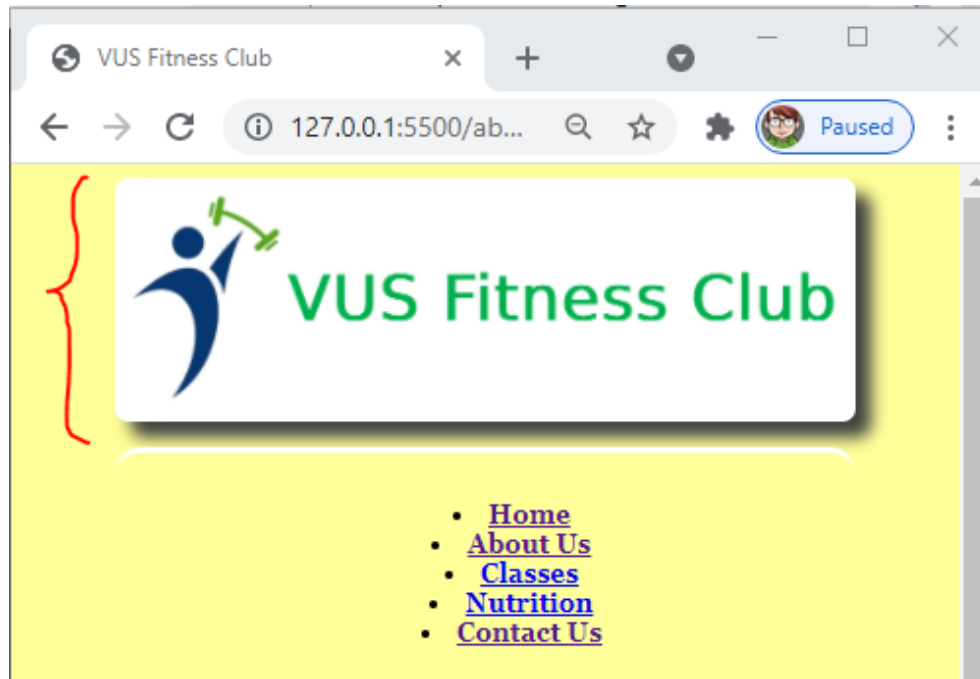
giá trị thuộc
tính max-width



Sử dụng các hình ảnh linh hoạt (Flexible Images)

❑ Thêm hình ảnh linh hoạt vào dự án:

♦ Quan sát tất cả hình ảnh của dự án **VUS Fitness Club** lần lượt tại 3 trang: Home, About Us, Contact Us sau khi chỉnh sửa. Nhất là vùng logo sẽ được hiển thị phù hợp hơn. Không bị vượt quá độ rộng cho phép như lúc đầu nữa.

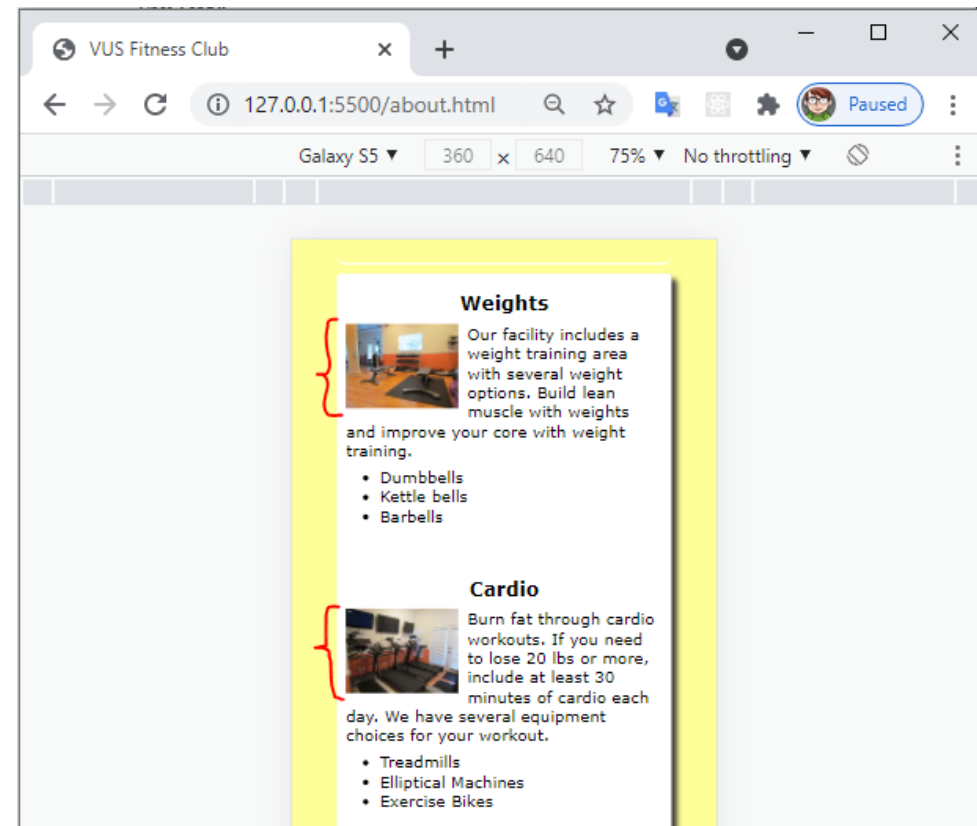
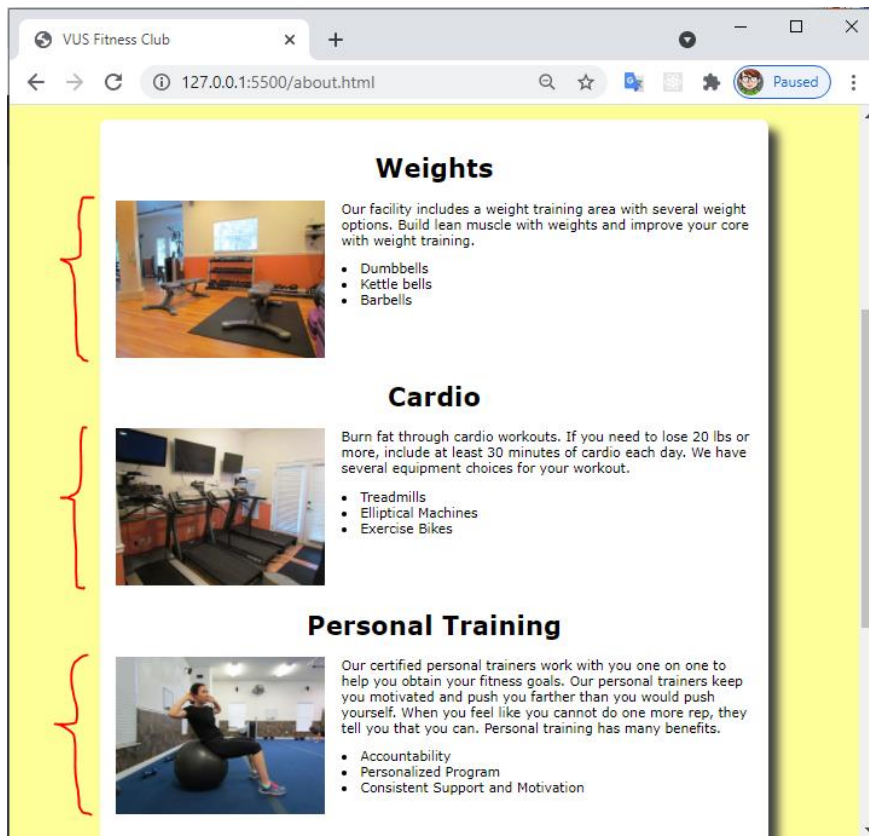




Sử dụng các hình ảnh linh hoạt (Flexible Images)

❑ Thêm hình ảnh linh hoạt vào dự án:

♦ Quan sát tất cả hình ảnh của dự án **VUS Fitness Club** lần lượt tại 3 trang: Home, About Us, Contact Us sau khi chỉnh sửa. Và các hình ảnh ở trang About Us.





Thiết kế Responsive với chiến lược Mobile-First

□ Giới thiệu:

- ♦ Phương pháp thiết kế Web Responsive định kiểu nội dung khác nhau tùy thuộc vào loại thiết bị được sử dụng để xem trang web.
- ♦ Phương pháp ưu tiên thiết bị di động (Mobile-First) tập trung vào việc tạo kiểu nội dung phù hợp cho các thiết bị di động nhỏ hơn.
- ♦ Một mục tiêu cốt lõi khi thiết kế cho thiết bị di động là hiển thị nổi bật nội dung được người dùng truy cập nhiều nhất.
- ♦ Cách tốt nhất là sử dụng bố cục một cột cho màn hình di động, vì điều này ngăn cản việc cuộn theo chiều ngang.

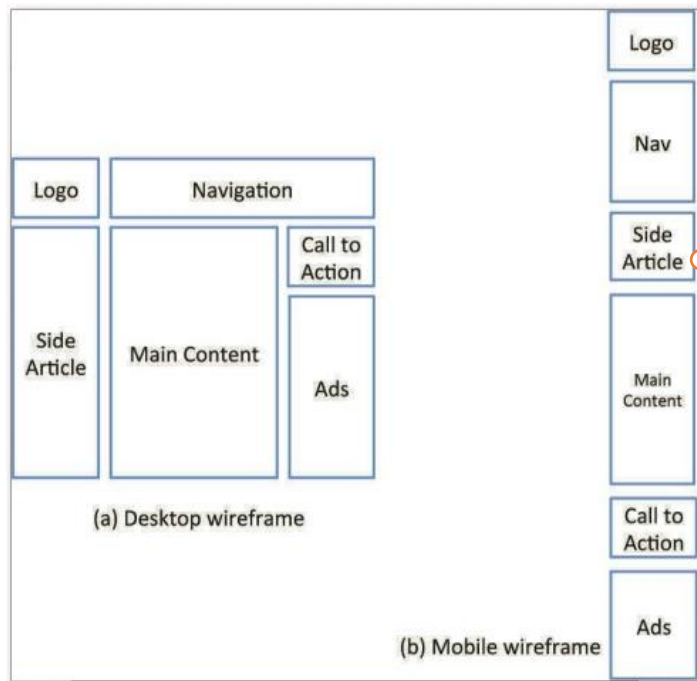




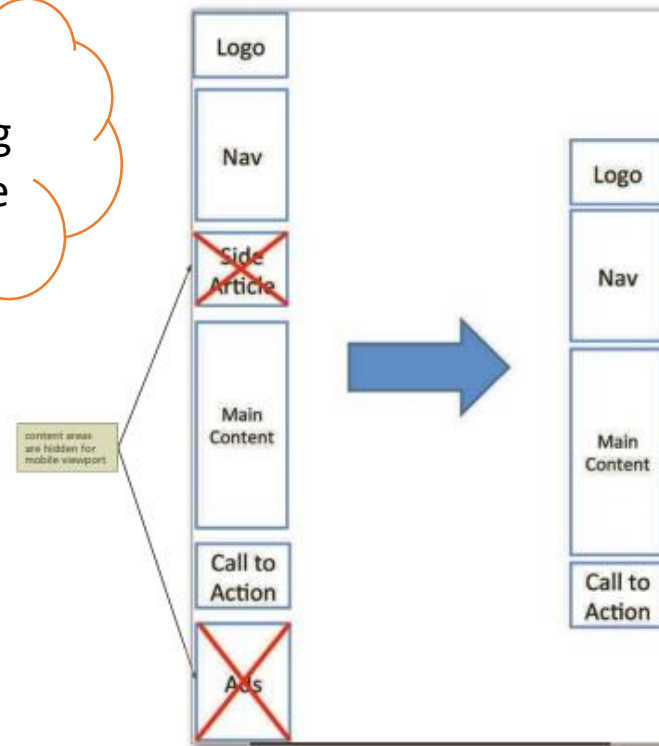
Thiết kế Responsive với chiến lược Mobile-First

❑ Định kiểu nội dung trên các thiết bị Mobile:

- ◆ Nếu bạn **không có** một trang web hiện có, hãy **bắt đầu** bằng cách **xác định** trước tiên **nội dung cần thiết nhất cho trang web** của bạn.
- ◆ Nếu bạn **đã có** nội dung trang, bạn **có thể** cần **phải xóa một số nội dung để có chế độ xem** trên **thiết bị di động**.



Không nên để nội dung trên mobile quá dài!



Nên ẩn các mục không phải trọng yếu đi!



Thiết kế Responsive với chiến lược Mobile-First

❑ Các phương pháp chính tốt nhất của Mobile-First:

♦ Tối ưu hóa giao diện để tối đa hóa trải nghiệm người dùng di động bằng **một số key best practices khi thiết kế** cho **các cửa sổ xem trên thiết bị di động** sau đây:

- Tận dụng 100% không gian màn hình.
- Thiết kế điều hướng dễ dàng và trực quan.
- Giữ thời gian tải ở mức tối thiểu. Nâng cao thời gian tải bằng cách loại bỏ nội dung sử dụng nhiều băng thông và hợp lý hóa mã HTML của bạn.
- Hiển thị nội dung trang cần thiết và ẩn nội dung trang không cần thiết.
- Làm cho nội dung dễ dàng truy cập và đọc.
- Thiết kế một bố cục đơn giản.





Thiết kế Responsive với chiến lược Mobile-First

❑ Các bước hiện thực chiến lược Mobile-First:

♦ Bước 1: Khi bạn muốn triển khai chiến lược **Mobile-First** với một trang web hiện có, trước tiên bạn **phải phân tích từng trang** để **xác định nội dung quan trọng nhất** và ẩn một số **nội dung không phải thiết yếu đi** khi hiển thị trên thiết bị di động.

▪ Ví dụ: Trang web **VUS Fitness Club** đã có sẵn nội dung trang, bạn cần phân tích từng trang để chọn ra nội dung cần thiết nhất rồi thiết kế nội dung cho **mobile viewport**.

♦ Bước 2: Mở tệp **CSS** ra, **sửa đổi** và **thêm mới** các quy tắc kiểu **CSS hiện có/CSS thêm mới** để chúng áp dụng cho chế độ xem trên thiết bị di động.

♦ Bước 3: **Tạo** một **Media Queries** với **các kiểu** cho **tablet**, **desktop**. Hoặc **liệt kê** các kiểu **viewports** và **media queries** theo thứ tự: mobile, tablet, laptop, desktop.





Thiết kế Responsive với chiến lược Mobile-First

❑ Thực hành hiện thực chiến lược Mobile-First trên VUS Fitness Club:

- ♦ **Sửa** định kiểu **CSS** của **#container** cho **Mobile Viewports**:

```
/* Style for the container centers the page and specifies the width */
#container {
  width: 80%; → 100%
  margin-left: auto;
  margin-right: auto;
}
```

- ♦ **Sửa** định kiểu **CSS** của **header** cho **Mobile Viewports**:

```
/* Style for the header specifies top margin, background color,
rounded corners, and center-aligns content */
header {
  margin-top: 10px; → 0,2 em
  margin-bottom: 20px;
  background-color: ■ #FFFFFF;
  border-radius: 10px; → 0,5 em
  box-shadow: 12px 12px 12px □ #404040;
  text-align: center;
}
```



Thiết kế Responsive với chiến lược Mobile-First

- ❑ Thực hành hiện thực chiến lược Mobile-First trên VUS Fitness Club:
 - ♦ **Sửa** định kiểu **CSS** của **nav** cho **Mobile Viewports**:

```
/* Style for the nav specifies text properties */
nav {
  font-family: Georgia, 'Times New Roman', Times, serif;
  font-size: 1.25em;
  font-weight: bold;
  text-align: center;
  padding: 15px;
  margin-top: 10px;
  margin-bottom: 10px;
  border-radius: 20px;
  border-top: 5px solid #FFFFFF;
  border-bottom: 5px solid #FFFFFF;
}
```




Thiết kế Responsive với chiến lược Mobile-First

❑ Thực hành hiện thực chiến lược Mobile-First trên VUS Fitness Club:

- ◆ **Thêm** định kiểu **CSS** của **nav ul** cho **Mobile Viewports**:

```
/* Style specifies padding and margins for unordered list */
nav ul {
  padding: 0;
  margin-top: 0.5em;
  margin-bottom: 0.5em;
}
```

- ◆ **Thêm** định kiểu **CSS** của **nav li** cho **Mobile Viewports**:

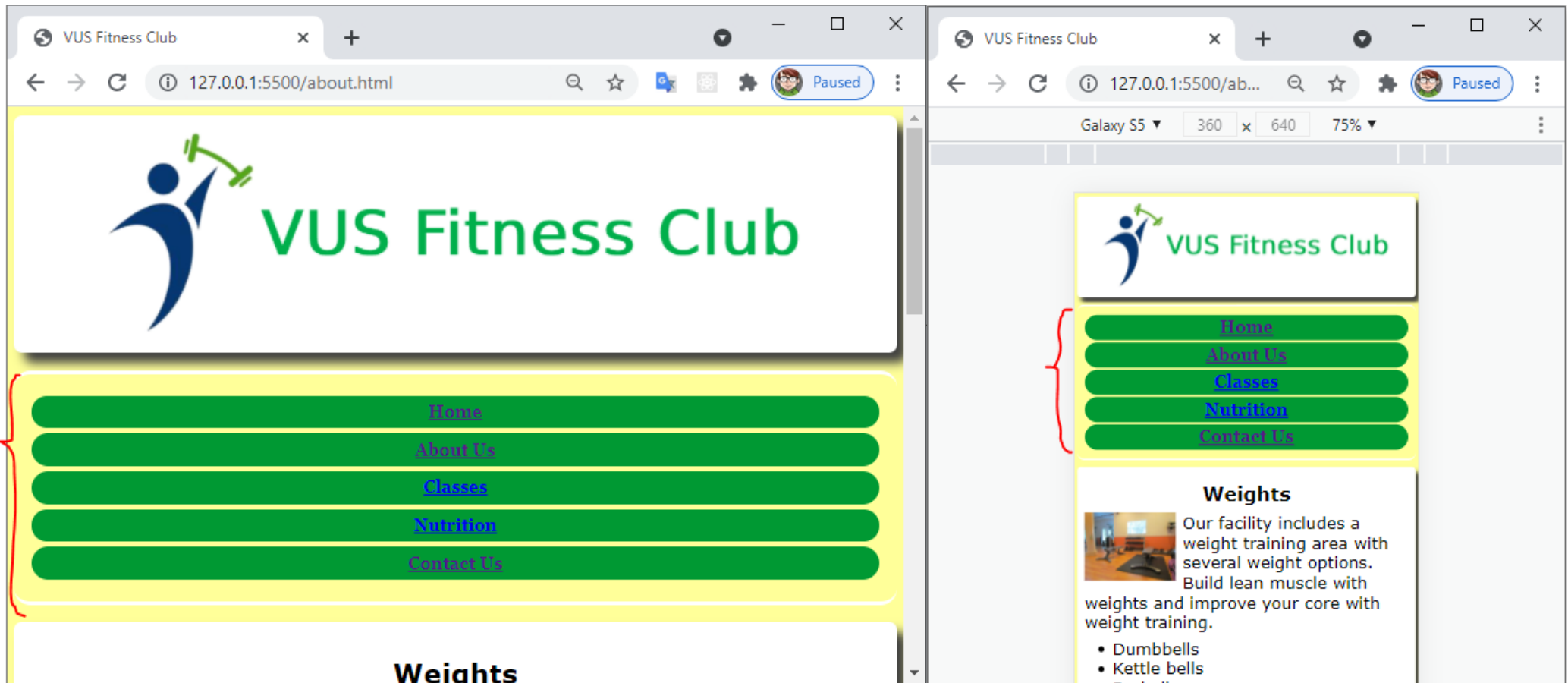
```
/* Style for nav li specifies the background color, rounded corners,
removes bullet style, and applies margins and padding for list items
within the navigation */
nav li {
  background-color:  #009933;
  border-radius: 2em;
  list-style-type: none;
  margin: 0.3em;
  padding: 0.4em;
}
```



Thiết kế Responsive với chiến lược Mobile-First

❑ Thực hành hiện thực chiến lược Mobile-First trên VUS Fitness Club:

♦ **Lưu lại** các **thay đổi** và **quan sát lại** giao diện sau khi chỉnh sửa:



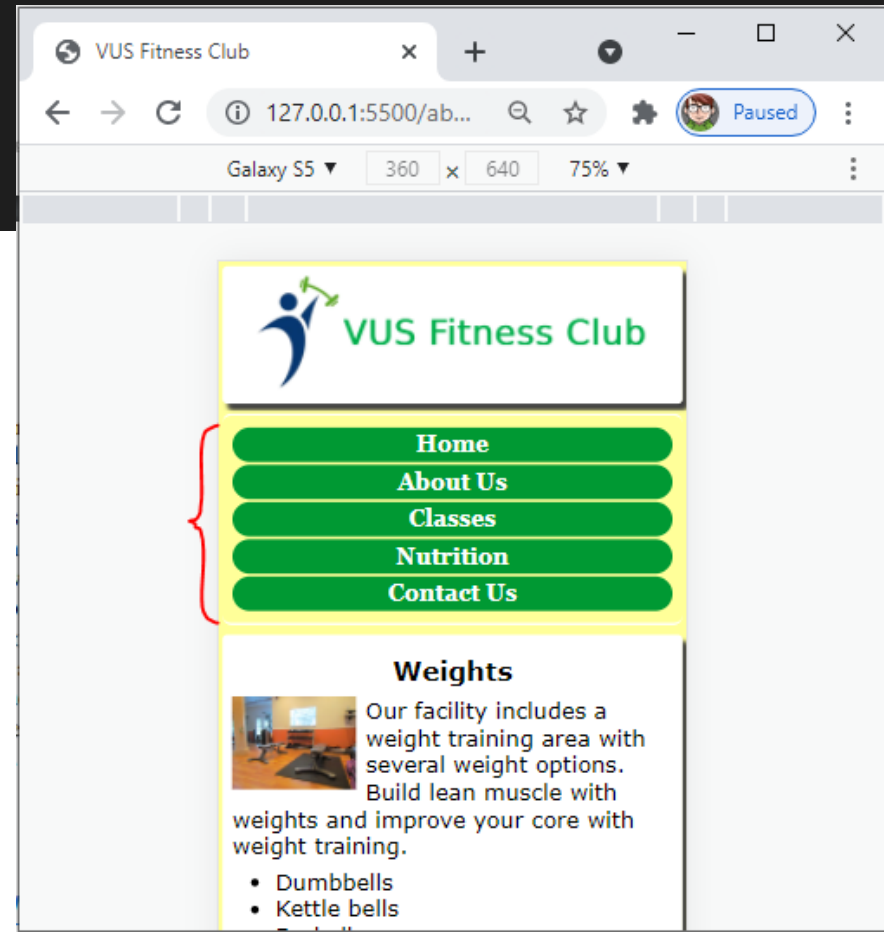
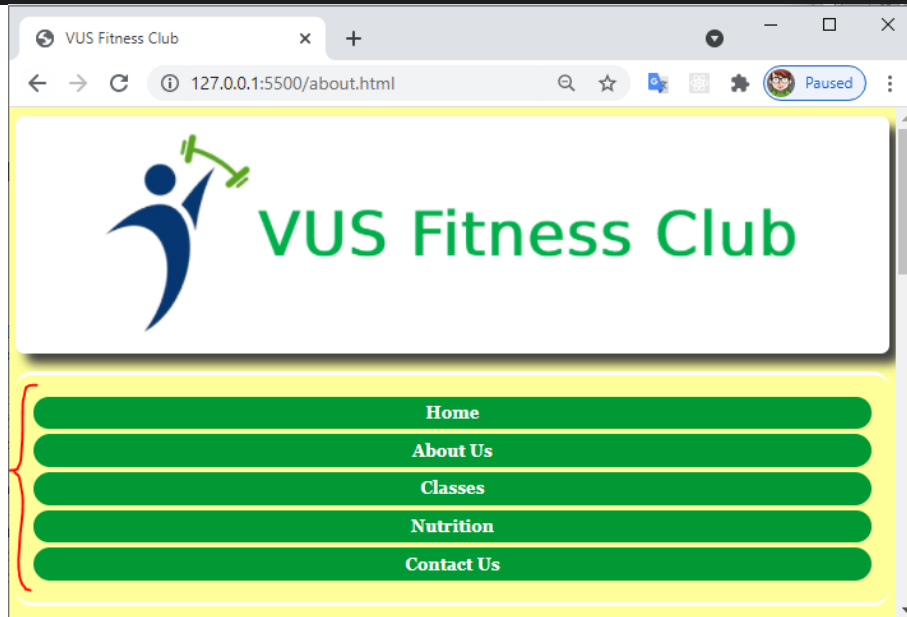


Thiết kế Responsive với chiến lược Mobile-First

❑ Thực hành hiện thực chiến lược Mobile-First trên VUS Fitness Club:

- ♦ **Thêm** định kiểu **CSS** của **nav li a** cho **Mobile Viewports**:

```
/* Style changes navigation link text color to white and removes the underlines */
nav li a {
  color: #FFFFFF;
  text-decoration: none;
}
```





Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang chủ (Home) cho thiết kế Mobile-First:

- ♦ Nội dung trên trang chủ hiện đã được tạo kiểu để **có trải nghiệm xem hiệu quả trên mobile**, nhưng nội dung trang **vẫn đang phù hợp với người dùng desktop**.
- ♦ Thay vì cung cấp thông điệp chào mừng, giới thiệu, hãy **sử dụng nội dung** mà **người dùng thiết bị mobile muốn tìm nhanh chóng**, (Ví dụ: giờ làm việc, số điện thoại của câu lạc bộ thể dục có liên kết điện thoại,...)
- ♦ Bạn **có thể thêm nội dung cần thiết** này **vào trang chủ** mà **không cần xóa nội dung** đối với khách truy cập trang web bằng thiết bị desktop.





Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang chủ (Home) cho thiết kế Mobile-First:

♦ **Chỉ sửa nội dung trên Home:** **thêm thông tin cần thiết** (giới thiệu tuần dùng thử cho thành viên, giờ làm việc, điện thoại liên lạc,...) **phù hợp** cho **mobile** và **desktop**.

```
<!-- Use the main area to add the main content of the webpage -->
<main>
  <p>Welcome to VUS Fitness Club. Our mission is to help our clients meet
  <div class="mobile">
    <h3>FREE One-Week Trial Membership!</h3>
    <p><a href="tel: 02871093333">Call Us Today to Get Started</a></p>

    <h4>Fitness Club Hours:</h4>
    <p>Monday - Thursday: 6:00am - 6:00pm<br>
      Friday: 6:00am - 6:00pm<br>
      Saturday: 8:00am - 6:00pm<br>
      Sunday: Closed</p>
  </div>
  <div class="desktop">
    <p>If you have struggled with getting healthy and need the motivation
    and information to keep you healthy.</p>
    <p>We provide a FREE, one-week membership to experience the benefits
    of a trial!
  </p>
  </div>
</main>
```

```
/* Style display the mobile class */
.mobile {
  display: inline;
}

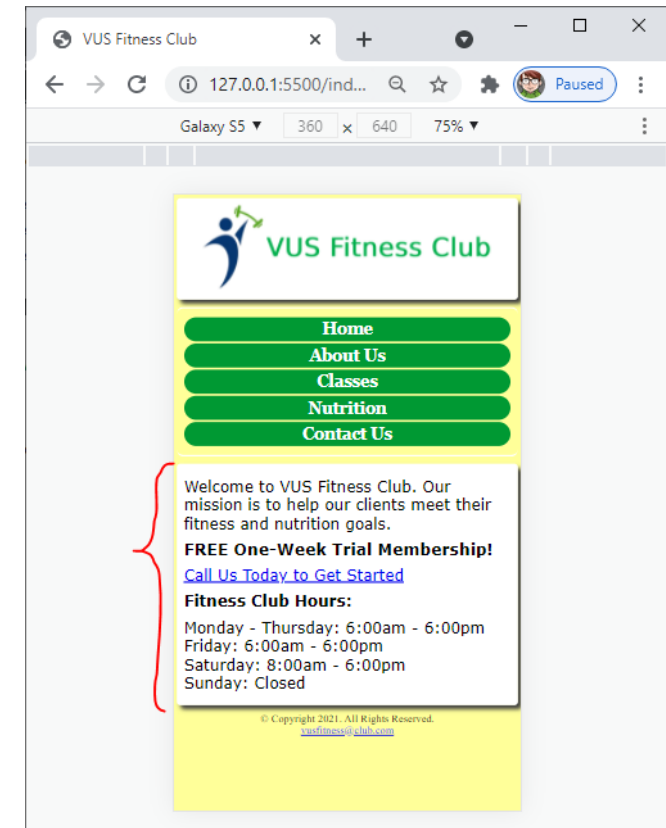
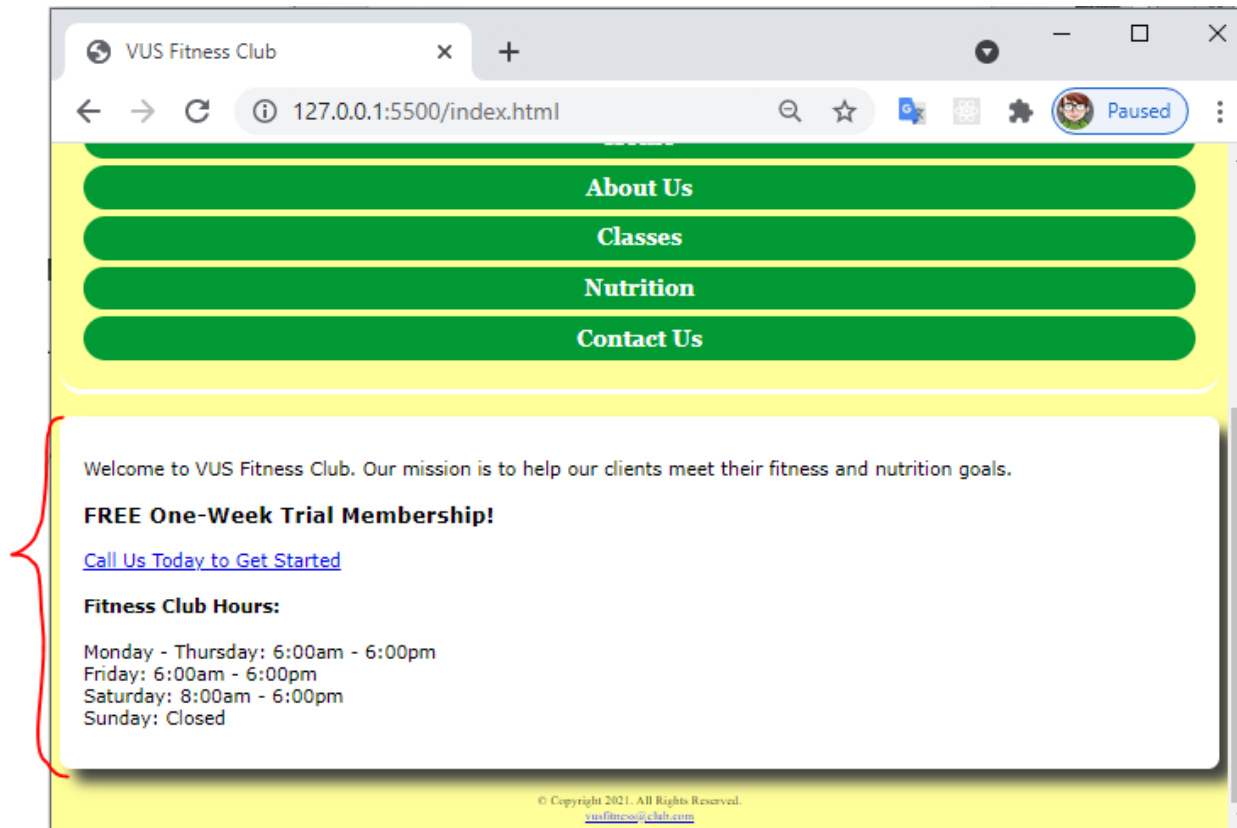
/* Style hides the desktop class */
.desktop {
  display: none;
}
```




Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang chủ (Home) cho thiết kế Mobile-First:

♦ **Chỉnh sửa nội dung trên Home:** **thêm thông tin cần thiết** (giới thiệu tuần dùng thử cho thành viên, giờ làm việc, điện thoại liên lạc,...) **phù hợp** cho **mobile** và **desktop**.





Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang chủ (Home) cho thiết kế Mobile-First:

♦ **Chỉnh sửa nội dung trên Home:** **chỉnh sửa CSS** cho **phần đệm** (padding) và **lề** (margin) **để hiển thị nhiều nội dung hơn** cho viewport **mobile** và **thay đổi đơn vị đo lường tuyệt đối (px) thành tương đối (em)** sử dụng trong **Fluid Responsive Layout**.

```
main {  
  display: block;  
  font-family: Verdana, Arial, sans-serif;  
  font-size: 1em;  
  margin-top: 20px;  
  margin-bottom: 10px;  
  padding: 20px;  
  border-radius: 10px;  
  background-color: #FFFFFF;  
  box-shadow: 12px 12px 12px #404040;  
}
```

```
main {  
  display: block;  
  font-family: Verdana, Arial, sans-serif;  
  font-size: 1em;  
  margin-top: 0.5em;  
  padding: 1em;  
  border-radius: 1em;  
  background-color: #FFFFFF;  
  border-top: solid 0.2em #009933;  
  border-bottom: solid 0.2em #009933;  
}
```

Chỉnh đơn vị đo của
footer luôn!

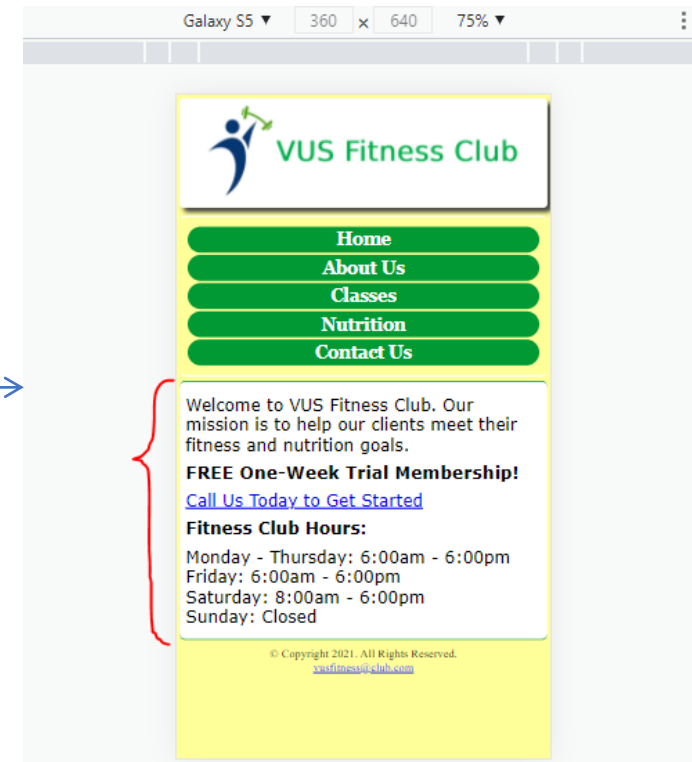
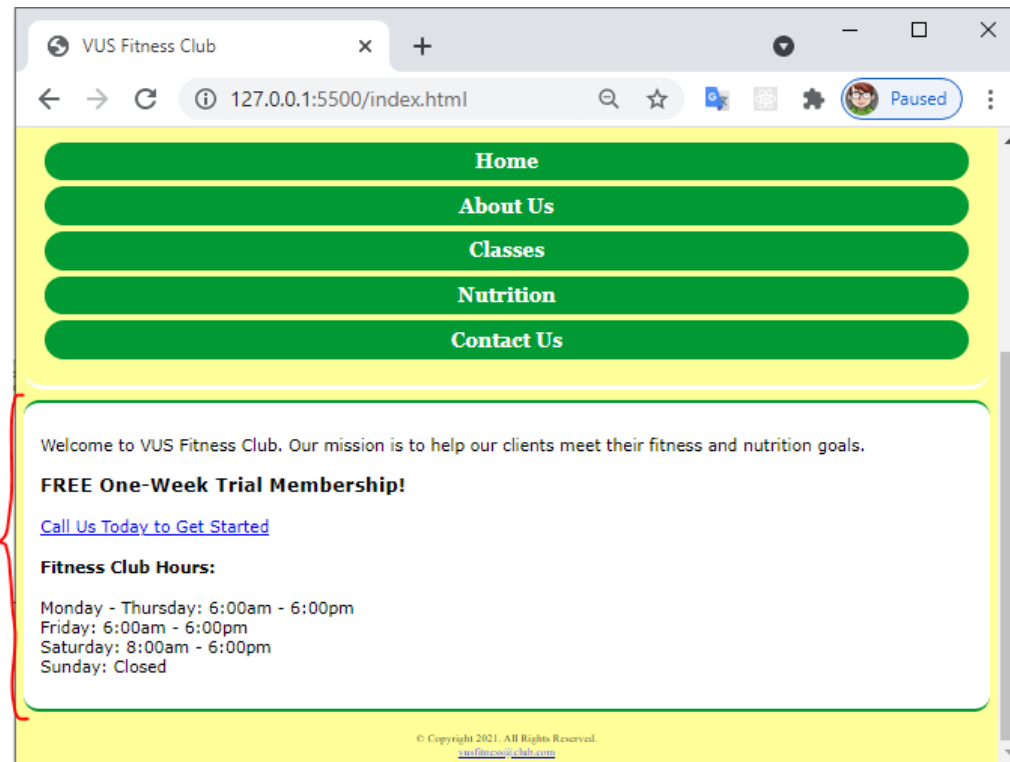
```
footer {  
  font-size: 0.70em;  
  text-align: center;  
  margin-top: 20px; → 2em  
}
```



Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang chủ (Home) cho thiết kế Mobile-First:

♦ **Chỉnh sửa nội dung trên Home:** **chỉnh sửa CSS** cho **phần đệm** (padding) và **lề** (margin) **để hiển thị nhiều nội dung hơn** cho viewport **mobile** và **thay đổi đơn vị đo lường tuyệt đối (px) thành tương đối (em)** sử dụng trong **Fluid Responsive Layout**.





Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang giới thiệu (About Us) cho thiết kế Mobile-First:

- ♦ Trang **About Us** có nhiều nội dung hơn trang chủ. Để giảm thiểu việc cuộn, hãy xem lại nội dung trang và xác định **thông tin cần thiết nhất, mô tả thiết bị và dịch vụ**.
- ♦ Trong trường hợp của trang **About Us** này, ta có thể:
 - **Ẩn ảnh, xóa dấu ngắt dòng và xóa quy tắc kiểu** cho **bộ chọn ul**.
 - **Đặt lề dưới lớn và định vị dấu đầu dòng** cho danh sách.
 - Có thể **ẩn nội dung khác trong yếu tố chính**, bao gồm thông tin bài tập thể dục chung chung và đoạn văn có liên kết ngoài. (không cần thiết đối với người dùng mobile vì người dùng mobile chỉ cần biết những thông tin cần thiết nhất, vd: mô tả thiết bị, dịch vụ)





Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang giới thiệu (About Us) cho thiết kế Mobile-First:

♦ **Chỉnh sửa nội dung trên About Us:** **Loại bỏ** tất cả các ngắt dòng **br** và **thêm** một **div** với một lớp có tên là **tablet**. Sau đó, **thêm** bộ chọn **.tablet** vào CSS để đặt giá trị thuộc tính **display** thành **none** để lớp này không xuất hiện trong **mobile viewport**.

```
<h1>Weights</h1>

<p>Our facility includes a weight training area with several
<ul>...
</ul><br>

<h1>Cardio</h1>

<p>Burn fat through cardio workouts. If you need to lose 20
<ul>...
</ul><br>

<h1>Personal Training</h1>

<p>Our certified personal trainers work with you one on one
    tell you that you can. Personal training has many benefits
<ul>...
</ul><br>
```

```
<h1>Common Exercises</h1>
<p>The following are common exercises that we encourage our
<dl>
  <dt>Burpee</dt>
  <dd>Burpees are a great, full body exercise to increase
    <br></dd>
  <dt>Plank</dt>
  <dd>Planks build your core strength. To perform a plank,
    <br></dd>
  <dt>Mountain Climber</dt>
  <dd>Mountain climbers are a good cardio exercise. Place
    <br></dd>
</dl>
```



Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang giới thiệu (About Us) cho thiết kế Mobile-First:

♦ **Chỉnh sửa nội dung trên About Us:** **Loại bỏ** tất cả các ngắt dòng **br** và **thêm** một **div** với một lớp có tên là **tablet**. Sau đó, **thêm** bộ chọn **.tablet** vào CSS để đặt giá trị thuộc tính **display** thành **none** để lớp này không xuất hiện trong **mobile viewport**.

```
<div class="tablet">
  <h1>Common Exercises</h1>
  <p>The following are common exercises that we encourage our clients
  <dl>
    <dt>Burpee</dt>
    <dd>Burpees are a great, full body exercise to increase your str
      a squatted position.
    </dd>

    <dt>Plank</dt>
    <dd>Planks build your core strength. To perform a plank, get in
    </dd>

    <dt>Mountain Climber</dt>
    <dd>Mountain climbers are a good cardio exercise. Place your han
    </dd>
  </dl>

  <p>For more information about how to stay active, visit <a href="/">
</div>
```

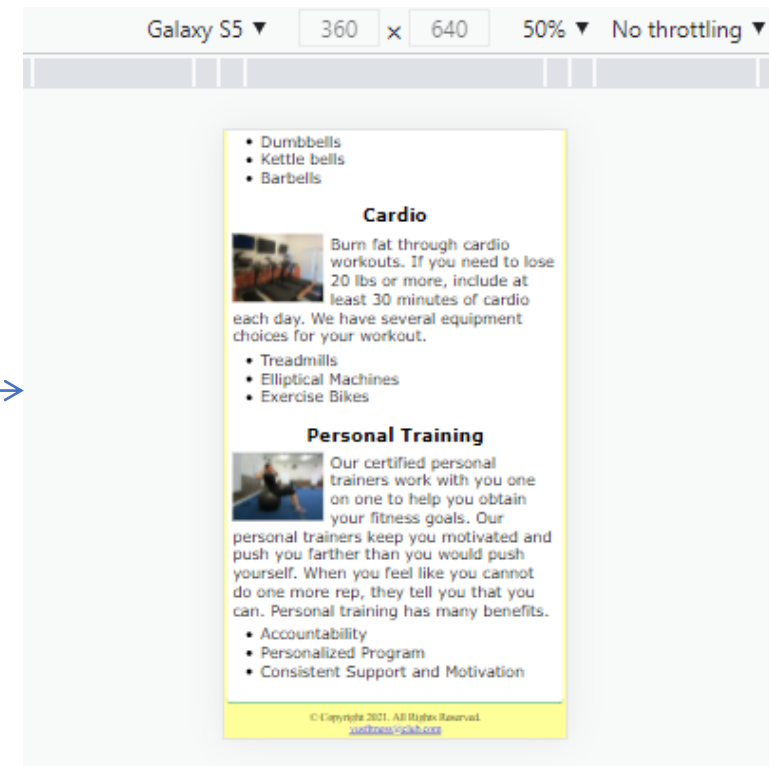
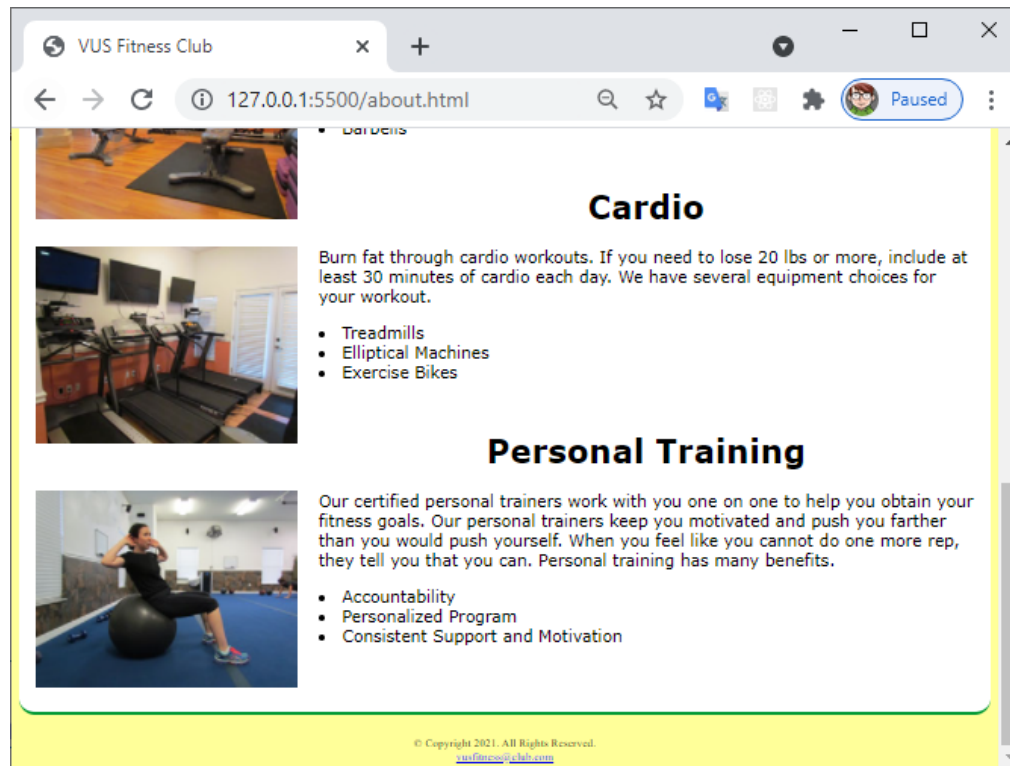
```
/* Style hides the tablet class */
.tablet {
  display: none;
}
```



Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang giới thiệu (About Us) cho thiết kế Mobile-First:

♦ **Chỉnh sửa nội dung trên About Us:** Loại bỏ tất cả các ngắt dòng **br** và **thêm** một **div** với một lớp có tên là **tablet**. Sau đó, **thêm** bộ chọn **.tablet** vào CSS để đặt giá trị thuộc tính **display** thành **none** để lớp này không xuất hiện trong **mobile viewport**.



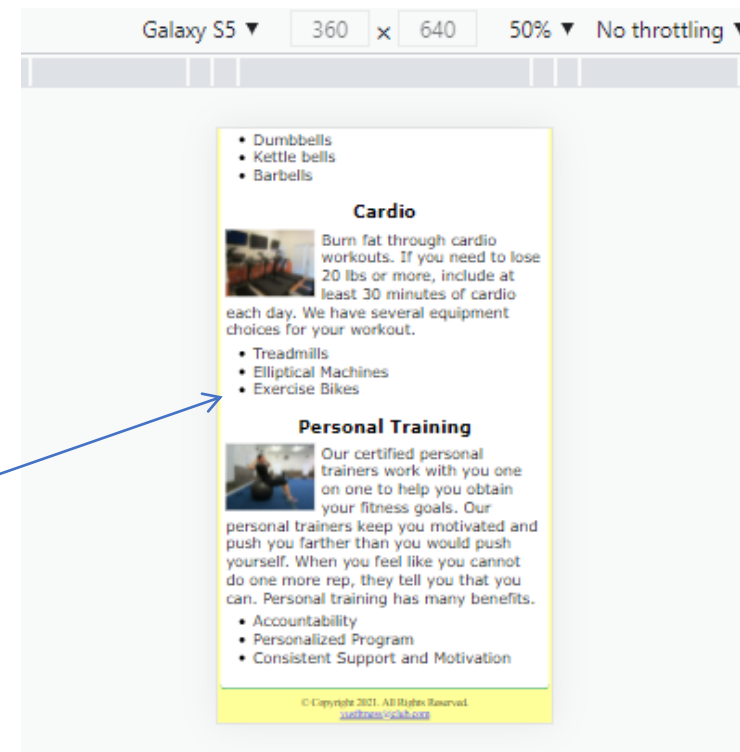
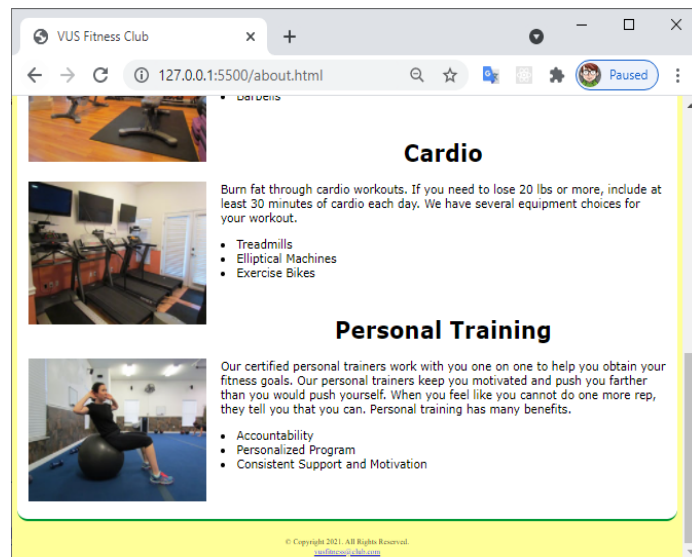


Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang giới thiệu (About Us) cho thiết kế Mobile-First:

♦ **Chỉnh sửa nội dung trên About Us:** **Sửa đổi** quy tắc kiểu CSS **.equip** bằng cách **thêm thuộc tính display** với giá trị **none** để **xóa các quy tắc kiểu ul, dt và dd** khi nhìn trên mobile, lớp này không xuất hiện trong **mobile viewport** nữa.

```
/* Style rule specifies to not display the equip class */  
  
.equip {  
  display: none;  
  float: left;  
  padding-right: 20px;  
}
```

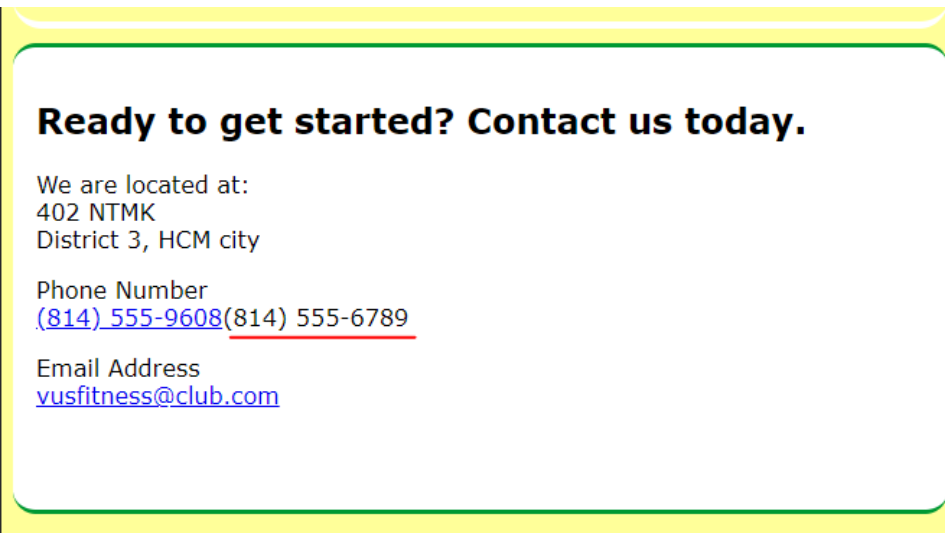




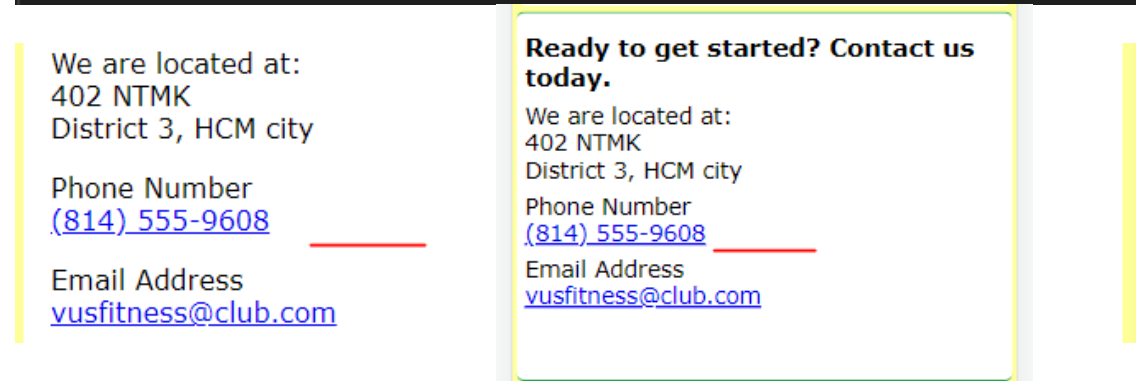
Thiết kế Responsive với chiến lược Mobile-First

❑ Phân tích trang liên hệ (Contact Us) cho thiết kế Mobile-First:

- ♦ Xem lại trang liên hệ với chúng tôi **bao gồm hai số điện thoại**; một có liên kết điện thoại và một không có.
- ♦ Trang liên hệ với chúng tôi **sẽ hiển thị số điện thoại có liên kết và ẩn số điện thoại không có liên kết trong khung nhìn di động**. -> dùng **span** để **chỉnh sửa**!
- ♦ Thêm một phần tử **span** với một lớp tên **desktop** được **đặt bao bọc số điện thoại không có liên kết** trong trang Contact Us.



```
<p>Phone Number<br>  
  <a class="mobile" href="tel:8145556789">(814) 555-9608</a>  
  <span class="desktop">(814) 555-6789</span></p>
```





Thêm các thẻ mô tả Meta

❑ Giới thiệu:

- ◆ Đối với các trang web Responsive, hãy thêm thẻ meta vào phần đầu mỗi trang web.
- ◆ Đoạn **mã meta** sau đây **đảm bảo** rằng **trang tải ban đầu có chiều rộng bố cục phù hợp với chế độ xem của thiết bị**.

`<meta name="viewport" content="width=device-width, initial-scale=1">`

- ◆ Ban đầu **việc tải bố cục chính xác cho từng thiết bị** đặc biệt **hữu ích cho các thiết bị di động** vì nó giúp loại bỏ việc phóng to thêm

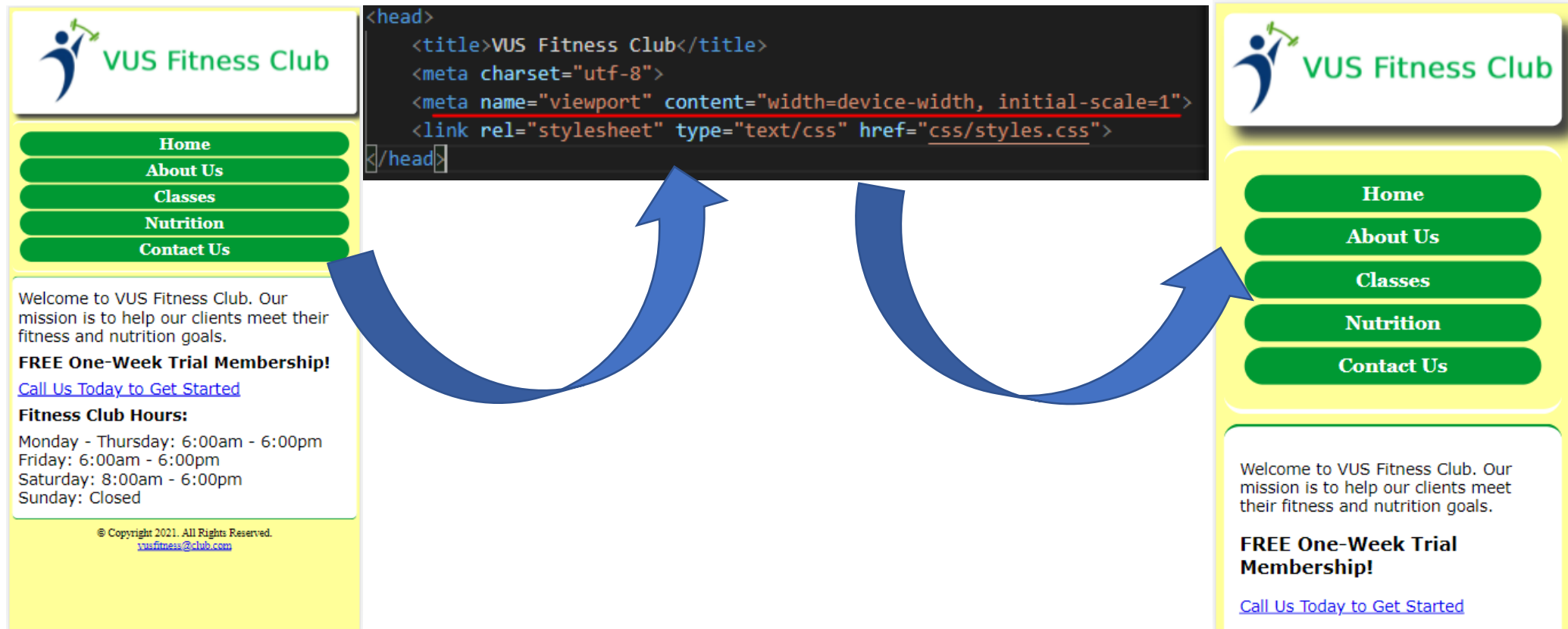




Thêm các thẻ mô tả Meta

❑ Thêm các thẻ mô tả meta cho thiết kế Responsive:

♦ Thêm thẻ **meta viewport** vào phần đầu của Home, About Us, Contact Us **giúp trang web tải ban đầu ở kích thước phù hợp** với kích thước cửa sổ xem của thiết bị hiện tại.

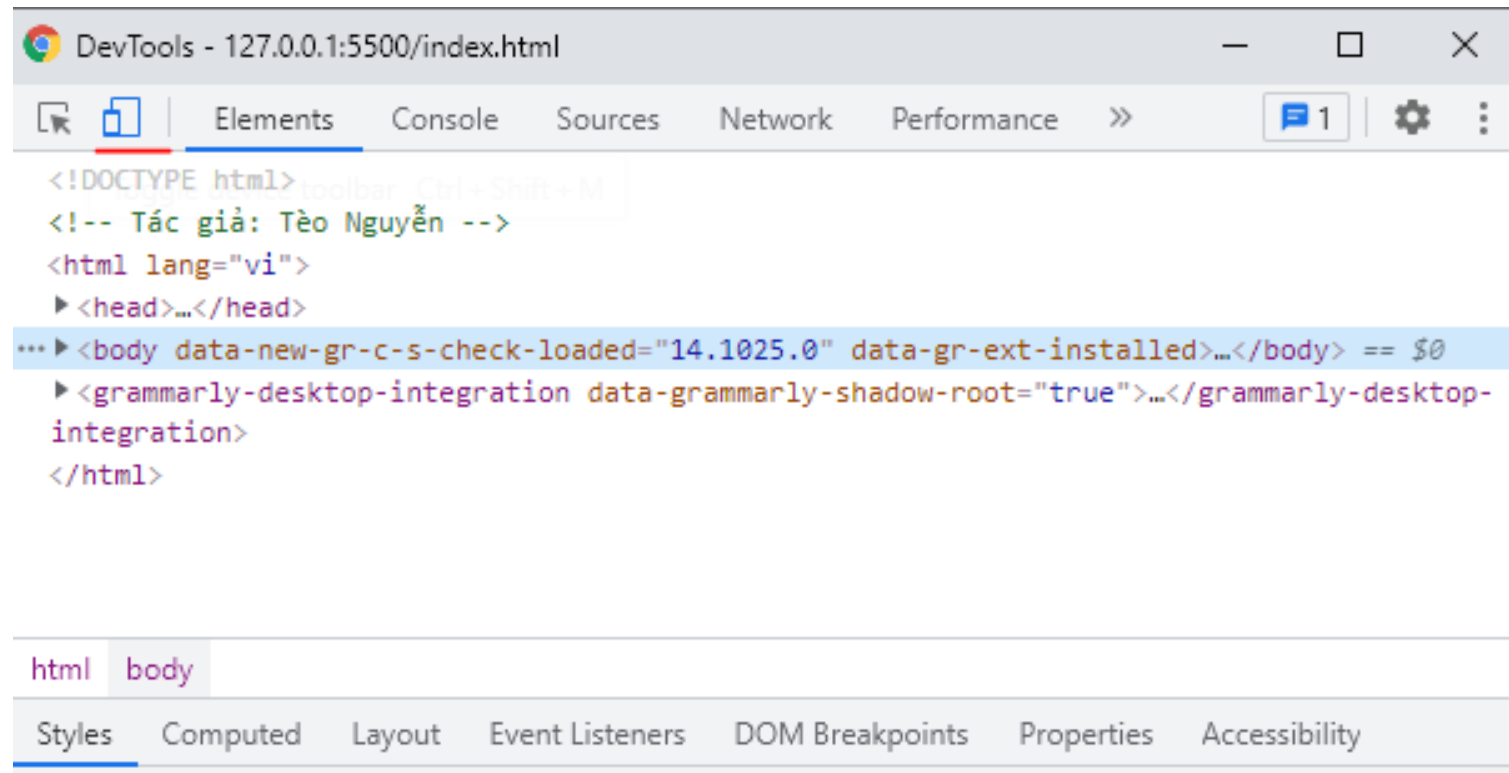




Kiểm tra Webpages tại các Viewports khác nhau

☐ Kiểm tra Webpages bằng simulator của DevTools:

♦ Ta có thể **kiểm tra giao diện website VUS Fitness Club sau khi chỉnh sửa theo chiến lược Mobile-First hiển thị trên nhiều loại màn hình ở nhiều kích cỡ khác nhau**. Và giao diện website sẽ **ưu tiên hiển thị dạng mobile (Mobile-First)**.





Tổng kết nội dung bài học

- ☐ Tổng quan về thiết kế Responsive
- ☐ Thiết kế Responsive cho các thiết bị Mobile
- ☐ Tạo một bố cục linh hoạt (Fluid Layout)
- ☐ Sử dụng các hình ảnh linh hoạt (Flexible Images)
- ☐ Thiết kế Responsive với chiến lược Mobile-First
- ☐ Thêm các thẻ mô tả Meta
- ☐ Kiểm tra Webpages tại các Viewports khác nhau

Let's
Recap

