**MỤC LỤC**

[A. PHẦN MỞ ĐẦU 4](#_Toc211307852)

[B. PHẦN NỘI DUNG 6](#_Toc211307853)

[Phần 1. BÀI TẬP THIẾT KẾ WEBSITE CÁ NHÂN 6](#_Toc211307854)

[1.1. Mô tả giao diện 6](#_Toc211307855)

[1.2. Chức năng của trang web 7](#_Toc211307856)

[1.3. Chức năng nổi bât 7](#_Toc211307856)

[Phần 2. BÀI TẬP ỨNG DỤNG QUẢN LÝ SẢN PHẨM 12](#_Toc211307860)

[2.1. Mô tả giao diện 12](#_Toc211307861)

[2.2. Chức năng của trang web 14](#_Toc211307865)

[2.3. Chức năng nổi bật 14](#_Toc211307865)

[Phần 3. BÀI TẬP THIẾT KẾ LAYOUT 16](#_Toc211307866)

[3.1. Mô tả giao diện 16](#_Toc211307867)

[3.2. Chức năng của trang web 16](#_Toc211307868)

[3.3. Chức năng nổi bật 23](#_Toc211307869)

[Phần 4. BÀI TẬP HỒ CÁ VUI NHỘN 6](#_Toc211307854)

[4.1. Mô tả giao diện 6](#_Toc211307855)

[4.2. Chức năng của trang web 7](#_Toc211307856)

[2.3. Chức năng nổi bật 14](#_Toc211307865)

[Phần 5. BÀI TẬP THIỆP MỜI SINH ĐỘNG 12](#_Toc211307860)

[5.1. Mô tả giao diện 6](#_Toc211307855)

[5.2. Chức năng của trang web 7](#_Toc211307856)

[5.3. Chức năng nổi bật 14](#_Toc211307865)

[Phần 6. BÀI TẬP THIẾT KẾ MẪU CURRICULUM VITAE - CV 16](#_Toc211307866)

[6.1. Mô tả giao diện 6](#_Toc211307855)

[6.2. Chức năng của trang web 7](#_Toc211307856)

[6.3. Chức năng nổi bật 14](#_Toc211307865)

[D. TÀI LIỆU THAM KHẢO 27](#_Toc211307871)

# A. PHẦN MỞ ĐẦU

**I. LỜI MỞ ĐẦU**

Trong kỷ nguyên số, sự phát triển không ngừng của Internet đã đưa các ứng dụng web trở thành một phần không thể thiếu trong mọi lĩnh vực của đời sống. Cùng với sự tiến bộ đó, yêu cầu về trải nghiệm người dùng ngày càng trở nên khắt khe, đòi hỏi các trang web không chỉ đẹp về mặt thẩm mỹ mà còn phải có bố cục mạch lạc, tốc độ truy cập nhanh và dễ dàng tương tác. Nền tảng chính của việc xây dựng giao diện web hiện là CSS (Cascading Style Sheets).

Tuy nhiên, khi quy mô dự án ngày càng lớn và phức tạp, việc sử dụng CSS thuần bắt đầu bộc lộ nhiều nhược điểm. Lập trình viên thường xuyên phải đối mặt với tình trạng mã nguồn bị lặp lại, khó quản lý, bảo trì và thiếu tính nhất quán. Các tệp CSS với hàng ngàn dòng lệnh trở thành một “cơn ác mộng” mỗi khi cần sửa đổi hay mở rộng chức năng, làm giảm năng suất làm việc và tăng nguy cơ phát sinh lỗi.

Để giải quyết những thách thức trên, các bộ tiền xử lý CSS (CSS Preprocessor) và SASS/SCSS (Syntactically Awesome Style Sheets) nổi lên như một trong những công cụ mạnh mẽ và phổ biến. SASS/SCSS cung cấp các tính năng lập trình cao cấp như biến, câu lệnh điều kiện, vòng lặp, mixin, và khả năng lồng ghép cú pháp, giúp việc tổ chức mã nguồn trở nên logic, có cấu trúc và hạn chế vấn đề lặp mã.

Trong bối cảnh đó, việc trang bị kiến thức và kỹ năng sử dụng SASS/SCSS không còn là một lựa chọn mà đã trở thành một yêu cầu cấp thiết đối với các lập trình viên front-end chuyên nghiệp. Việc tìm hiểu sâu về công nghệ này không chỉ giúp tối ưu hóa quy trình làm việc cá nhân mà còn là yếu tố quan trọng góp phần vào sự thành công của các dự án web hiện đại. Chính vì những lý do trên, đề tài “Tìm hiểu CSS SASS/SCSS” được nhóm lựa chọn để nghiên cứu.

# B. PHẦN NỘI DUNG

## Phần 1. BÀI TẬP THIẾT KẾ WEBSITE CÁ NHÂN

## 1.1. Mô tả giao diện

## 1.2. Chức năng của trang web

## 1.2. Chức năng nổi bật

## Phần 2. BÀI TẬP ỨNG DỤNG QUẢN LÝ SẢN PHẨM

## 1.1. Mô tả giao diện

## 1.2. Chức năng của trang web

## 1.2. Chức năng nổi bật

## Phần 2. BÀI TẬP ỨNG DỤNG QUẢN LÝ SẢN PHẨM

## 1.1. Mô tả giao diện

## 1.2. Chức năng của trang web

## 1.2. Chức năng nổi bật

## Phần 2. BÀI TẬP ỨNG DỤNG QUẢN LÝ SẢN PHẨM

## 1.1. Mô tả giao diện

## 1.2. Chức năng của trang web

## 1.2. Chức năng nổi bật

## Phần 2. BÀI TẬP ỨNG DỤNG QUẢN LÝ SẢN PHẨM

## 1.1. Mô tả giao diện

## 1.2. Chức năng của trang web

## 1.2. Chức năng nổi bật

## Phần 2. BÀI TẬP ỨNG DỤNG QUẢN LÝ SẢN PHẨM

## 1.1. Mô tả giao diện

## 1.2. Chức năng của trang web

## 1.2. Chức năng nổi bật

## .2. So sánh tổng thể CSS và SASS/SCSS

**Bảng 2-1** sẽ so sánh trực quan ưu điểm và nhược điểm đặc trưng nhất giữa CSS, SASS và SCSS nhằm giúp người học, người làm web hiểu rõ đặc điểm của từng loại và lựa chọn công cụ 2phù hợp cho dự án của mình.

***Bảng 2-1:*** *so sánh ưu và nhược điểm giữa CSS, SASS và SCSS*

|  |  |  |  |
| --- | --- | --- | --- |
| **So sánh** | **CSS** | **SASS** | **SCSS** |
| **Ưu điểm** | - Đơn giản, dễ học, dễ sử dụng.  - Được hỗ trợ trực tiếp trên tất cả các trình duyệt, không cần cài đặt hay biên dịch.  - Có thể dùng chung 1 tệp CSS cho nhiều trang. | - Cú pháp ngắn gọn  - Có biến, vòng lặp, hàm, mixin giúp tái sử dụng mã.  - Hỗ trợ kế thừa, import, dễ tổ chức mã, rất linh hoạt.  - Giảm trùng lặp và tăng hiệu quả khi làm việc với dự án lớn. | - Cú pháp giống CSS, dễ học, tương thích với mã CSS cũ.  - Có đầy đủ tính năng của SASS, khả năng tái sử dụng mã cao, rất linh hoạt.  - Hỗ trợ tốt trong các công cụ build hiện đại (Webpack, Gulp). |
| **Nhược điểm** | - Thiếu tính năng lập trình (không có biến, vòng lặp, hàm).  - Dễ trùng lặp mã, khó quản lý và bảo trì đối với các dự án lớn.  - Không hỗ trợ chia tệp logic.  - Tính linh hoạt hạn chế, không có khả năng tái sử dụng mã hiệu quả. | - Phải biên dịch sang CSS trước khi sử dụng.  - Cú pháp khác biệt, khó làm quen với người mới. | - Phải biên dịch sang CSS trước khi sử dụng.  - Dễ bị chậm nếu dự án có quá nhiều tệp.  - Phải học thêm cú pháp nâng cao. |

Tóm lại, CSS là nền tảng cơ bản và dễ tiếp cận, phù hợp cho người mới bắt đầu và các dự án có quy mô nhỏ. Trong khi SASS/SCSS lại mang đến nhiều tính năng nâng cao, giúp việc viết mã trở nên nhanh hơn, khả năng tổ chức mã tốt, nâng cao tính tái sử dụng và là giải pháp tối ưu cho các dự án lớn. Đặc biệt, SCSS được xem là lựa chọn cân bằng nhất, vừa giữ cú pháp thân thiện của CSS, vừa kế thừa đầy đủ tính năng của SASS. Việc lựa chọn ngôn ngữ phù hợp với quy mô dự án, yêu cầu kỹ thuật và kỹ năng của người phát triển sẽ giúp việc phát triển web hiệu quả và chuyên nghiệp hơn.

# C**hương** 3. ỨNG DỤNG SASS/SCSS TRONG MỘT DỰ ÁN THIẾT KẾ WEB

## 3.1. Dự án thiết kế Web

Đề bài do nhóm tự đặt ra: Ứng dụng SASS/SCSS để tái lập giao diện trang chủ website *Bảo tàng Lịch sử Thành phố Hồ Chí Minh*.

Quy trình thực hiện dự án:

- *Bước 1*: Phân tích những đặc điểm giao diện, thu thập hình ảnh liên quan.

- *Bước 2*: Tái lập mã nguồn nội dung của website trên nền HTML.

- *Bước 3*: Tái lập giao diện của website bằng bộ tiền xử lý SASS/SCSS.

- *Bước 4*: Tiến hành chạy lệnh website và sửa lỗi nếu có.

Các tư liệu được thu thập và mã nguồn nội dung HTML được lưu trong đường link thư mục *2511COMP180204\_Project\_BTLSTPHCM* đính kèm trong phần Tài liệu tham khảo của bài tiểu luận.

Báo cáo tiểu luận chỉ trình bày lại nội dung lập trình tạo kiểu bằng bộ tiền xử lý SASS/SCSS trong **mục 3.2**.

## 3.2. Định kiểu với SASS/SCSS

Tệp định kiểu tận dụng những tính năng mạnh mẽ đặc trưng để giúp mã lập trình trở nên gọn gàng, dễ bảo trì và hiệu quả hơn so với việc viết truyền thống bằng CSS.

Các tính năng đặc trưng được sử dụng bao gồm: Biến (Variables), Nesting (Lồng cấp), Mixins (@mixin và @include), Kế thừa (@extend), Modules và Functions (@use và các hàm tích hợp), Toán tử (Operators).

@use "sass:color";

// 1. VARIABLES

// Định nghĩa các biến cho màu sắc, font chữ, và các giá trị khác để dễ dàng tái sử dụng.

$primary-color: #911414; // Màu đỏ đất (đã cập nhật)

$secondary-color: #ede3df; // Màu be nhạt

$bg-color: #ffffff; // Màu nền trắng

$text-color: #333333;

$light-text-color: #FFFFFF;

$border-color: #ddd;

$base-font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

$base-border-radius: 12px; // Bo góc

$base-spacing: 1rem;

// 2. MIXINS

// Tạo các mixin cho các đoạn code lặp lại.

@mixin transition($property: all, $duration: 0.3s, $timing: ease-in-out) {

transition: $property $duration $timing;

}

@mixin card-shadow {

box-shadow: 0 4px 15px rgba(0, 0, 0, 0.08);

}

// 3. BASE & GENERAL STYLES

// Các style cơ bản cho toàn trang.

body {

font-family: $base-font-family;

background-color: $bg-color;

color: $text-color;

line-height: 1.7;

}

a {

color: $primary-color;

text-decoration: none;

@include transition;

&:hover {

color: color.scale($primary-color, $lightness: -10%);

text-decoration: none;

}

}

h1, h2, h3, h4, h5, h6 {

font-weight: 600;

}

img {

max-width: 100%;

height: auto;

}

// Lớp dùng chung cho các khối nội dung chính

.content-block {

margin-bottom: $base-spacing \* 2.5;

}

// 4. LAYOUT: HEADER

.site-header {

padding: $base-spacing 0;

background: $bg-color;

border-bottom: 1px solid $border-color;

// Nested layout

.logo-container {

display: flex;

align-items: center;

.logo {

width: 50px;

height: 50px;

margin-right: $base-spacing;

}

.site-title {

margin: 0;

font-size: 1.5rem;

color: $text-color;

font-weight: 700;

}

}

.main-nav {

ul {

margin: 0;

padding: 0;

list-style: none;

display: flex;

li {

margin-left: $base-spacing \* 2;

a {

padding: 0.5rem 0;

font-weight: 500;

position: relative;

color: $text-color;

// Hiệu ứng gạch chân khi hover hoặc active

&::after {

content: '';

position: absolute;

bottom: 0;

left: 0;

width: 0;

height: 2px;

background-color: $primary-color;

@include transition(width);

}

&:hover::after, &.active::after {

width: 100%;

}

// Đơn giản hóa, chỉ định style cho trạng thái active

&.active {

font-weight: 700;

color: $primary-color;

}

}

}

}

}

}

// 5. BANNER

.banner {

margin-top: $base-spacing;

img {

width: 100%;

height: 500px;

object-fit: cover; // Đảm bảo hình ảnh lấp đầy khung chứa mà không bị méo

display: block;

}

// Điều chỉnh chiều cao cho màn hình nhỏ hơn

@media (max-width: 767.98px) {

img {

height: 250px;

}

}

}

// 6. LAYOUT: SIDEBAR

.widget {

background-color: color.scale($secondary-color, $lightness: 15%);

padding: $base-spacing \* 1.5;

margin-bottom: $base-spacing \* 1.5;

border-radius: $base-border-radius;

border: 1px solid $secondary-color;

.widget-title {

margin-top: 0;

margin-bottom: $base-spacing;

padding-bottom: $base-spacing \* 0.5;

border-bottom: 2px solid $primary-color;

font-size: 1.2rem;

}

// Áp dụng cho nhiều danh sách trong widget

ul {

padding-left: 0;

list-style: none;

margin-bottom: 0;

li {

margin-bottom: $base-spacing \* 0.75;

&:last-child {

margin-bottom: 0;

}

}

}

}

// Search form widget

.search-form {

display: flex;

input {

flex-grow: 1;

border: 1px solid $border-color;

padding: 0.5rem;

border-radius: 6px 0 0 6px;

&:focus {

outline: none;

border-color: $primary-color;

}

}

button {

border: 1px solid $primary-color;

background-color: $primary-color;

color: $light-text-color;

padding: 0.5rem 1rem;

border-radius: 0 6px 6px 0;

cursor: pointer;

@include transition;

&:hover {

background-color: color.scale($primary-color, $lightness: -10%);

}

}

}

// Video widget

.video-container {

position: relative;

padding-bottom: 56.25%; /\* 16:9 aspect ratio \*/

height: 0;

overflow: hidden;

border-radius: 6px;

iframe {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

}

}

// Event list

.event-list .event-date {

background-color: $primary-color;

color: $light-text-color;

padding: 2px 8px;

border-radius: 4px;

margin-right: 8px;

font-weight: bold;

}

// 7. PAGE SPECIFIC: HOME

.image-card {

text-align: center;

img {

border-radius: $base-border-radius;

margin-bottom: $base-spacing \* 0.5;

@include card-shadow;

@include transition;

&:hover {

transform: translateY(-5px);

}

}

}

// 8. PAGE SPECIFIC: COLLECTION

// Sử dụng @extend

.collection-card {

border-radius: $base-border-radius;

overflow: hidden;

background: $bg-color;

height: 100%;

@include card-shadow;

@include transition;

&:hover {

transform: translateY(-8px);

box-shadow: 0 8px 25px rgba(0, 0, 0, 0.12);

}

.card-body {

padding: $base-spacing;

}

.card-title {

color: $primary-color;

}

.card-text {

font-size: 0.9rem;

}

}

// 9. PAGE SPECIFIC: CONTACT

.contact-form {

.form-control {

padding: 0.75rem 1rem;

border-radius: 8px;

&:focus {

box-shadow: 0 0 0 0.25rem rgba($primary-color, 0.25);

border-color: $primary-color;

}

}

.form-buttons {

margin-top: $base-spacing;

display: flex;

gap: $base-spacing;

}

.btn {

padding: 0.75rem 1.5rem;

border-radius: 8px;

font-weight: bold;

border: 2px solid;

@include transition;

}

.btn-submit {

@extend .btn; // Áp dụng @extend

background-color: $primary-color;

border-color: $primary-color;

color: $light-text-color;

&:hover {

background-color: color.scale($primary-color, $lightness: -10%);

border-color: color.scale($primary-color, $lightness: -10%);

}

}

.btn-reset {

@extend .btn;

background-color: transparent;

border-color: #6c757d;

color: #6c757d;

&:hover {

background-color: #6c757d;

color: $light-text-color;

}

}

}

.map-container {

border-radius: $base-border-radius;

overflow: hidden;

border: 1px solid $border-color;

@include card-shadow;

}

// 10. LAYOUT: FOOTER

.site-footer {

background-color: $text-color;

color: color.scale($text-color, $lightness: 30%);

padding: $base-spacing \* 2 0;

margin-top: $base-spacing \* 3;

p {

margin: 0;

}

.social-links {

list-style: none;

padding: 0;

margin: 0;

display: flex;

justify-content: flex-end;

gap: $base-spacing;

a {

color: color.scale($text-color, $lightness: 30%);

font-size: 1.2rem;

&:hover {

color: $light-text-color;

}

}

}

}

// 11. RESPONSIVE DESIGN

@media (max-width: 991.98px) {

.site-header .main-nav {

// Basic responsive nav for smaller screens

width: 100%;

margin-top: $base-spacing;

ul {

justify-content: center;

}

}

.logo-container {

justify-content: center;

width: 100%;

}

aside {

margin-top: $base-spacing \* 2;

}

}

@media (max-width: 767.98px) {

.site-header {

.logo-container {

flex-direction: column;

.site-title {

margin-top: $base-spacing \* 0.5;

font-size: 1.2rem;

}

}

.main-nav ul li {

margin-left: $base-spacing;

margin-right: $base-spacing;

}

}

.site-footer {

text-align: center;

.social-links {

justify-content: center;

margin-top: $base-spacing;

}

}

}

## 3.3. Kết quả thực hiện

A red and white website with a flower

AI-generated content may be incorrect.

**Hình 3-1:** *Khối Header và Banner của dự án*

A screenshot of a web page

AI-generated content may be incorrect.

***Hình 3-2:*** *Khối Article và khối Form của dự án*

A screenshot of a computer

AI-generated content may be incorrect.

***Hình 3-3:*** *Khối Widget và khối Footer của dự án*

Nhóm đã tái hiện lại được website *Bảo tàng Lịch sử Thành phố Hồ Chí Minh* một cách tương đối với các khối chính, màu sắc và một số đường dẫn đến các trang con. Các thành phần đặc trưng của SASS/SCSS đã thực hiện đúng chức năng giao diện, khả năng làm mạch lạc các câu lệnh giúp ích rất nhiều trong việc điều chỉnh sau này. Do vấn đề giới hạn về thời gian nên dự án vẫn chưa thực sự được chỉn chu và chưa đáp ứng được mục tiêu ban đầu. Một số tính năng vẫn chưa được sử dụng một cách triệt để, nhóm vẫn cần trau dồi thêm kiến thức, luyện tập và cải thiện khả năng lập trình hơn nữa.

# C. KẾT LUẬN VÀ KIẾN NGHỊ

Thông qua quá trình tìm hiểu và triển khai các nội dung trong ba chương, đề tài “Tìm hiểu CSS SASS/SCSS” đã cơ bản hoàn thành những mục tiêu cơ bản của nhóm đã đề ra và đi đến những kết luận sau:

Thứ nhất, đề tài đã khái quát hóa được cách vận hành và các tính năng đặc trưng của bộ tiền xử lý SASS/SCS, làm rõ những hạn chế cố hữu của việc sử dụng CSS thuần trong các dự án web quy mô lớn, đặc biệt là các vấn đề về sự lặp lại mã nguồn, khó bảo trì và thiếu tính tổ chức. Qua đó, khẳng định sự ra đời của các bộ tiền xử lý như SASS/SCSS là một giải pháp tất yếu nhằm giải quyết các bài toán thực tiễn trong ngành phát triển front-end.

Thứ hai, thông qua việc phân tích và so sánh chi tiết, đề tài đã làm rõ được các tính năng vượt trội của SASS/SCSS, đạt mục tiêu xây dựng một hướng tiếp cận cho người mới bắt đầu học thiết kế web hiểu hơn về cách vận hành của một bộ tiền xử lý. Việc triển khai các đoạn mã nguồn mẫu đã cho thấy rõ hiệu quả của việc sử dụng biến (variables) để quản lý giá trị một cách tập trung, cú pháp lồng (nesting) giúp mã nguồn có cấu trúc trực quan, mixin giúp tái sử dụng code một cách linh hoạt. Các tính năng này không chỉ giúp giảm đáng kể số lượng dòng code mà còn nâng cao tính rõ ràng, mạch lạc của toàn bộ stylesheet.

Thứ ba, kết quả ứng dụng SASS/SCSS trên một dự án của nhóm khẳng định rằng, việc áp dụng SASS/SCSS vào quy trình làm việc mang lại lợi ích to lớn về năng suất cho lập trình viên và chất lượng cho dự án. Nó giúp rút ngắn thời gian phát triển, giảm thiểu lỗi, tạo ra một hệ thống mã nguồn nhất quán và dễ dàng cho việc bảo trì, nâng cấp trong tương lai. Có thể kết luận rằng, SASS/SCSS không còn là một công cụ tùy chọn mà đã trở thành một kỹ năng thiết yếu, một chuẩn mực trong phát triển giao diện web

Từ những kết quả đã đạt được và những khó khăn gặp phải trong quá trình thực hiện đề tài, nhóm xin đưa ra kiến nghị về vấn đề nguồn tài liệu học tập SASS/SCSS bằng tiếng Việt. Trong quá trình tìm kiếm và thu thập thông tin, nhóm nhận thấy một thực tế rằng các tài liệu học tập chuyên sâu, có hệ thống về SASS/SCSS được biên soạn bằng tiếng Việt còn rất hạn chế. Hầu hết các nguồn tài liệu chất lượng cao đều bằng tiếng Anh, bao gồm tài liệu chính thức, các khóa học, và bài viết chuyên ngành. Điều này tạo ra một rào cản không nhỏ cho sinh viên và những người mới bắt đầu tiếp cận công nghệ này, đặc biệt là những người có khả năng ngoại ngữ còn hạn chế.

Vì vậy, nhóm xin kiến nghị cần có thêm nhiều hoạt động chia sẻ kiến thức như viết blog, dịch thuật các tài liệu quan trọng, thực hiện các chuỗi video hướng dẫn bằng tiếng Việt để xây dựng một hệ sinh thái học liệu phong phú hơn. Cũng như khuyến khích việc biên soạn giáo trình, tài liệu tham khảo bằng tiếng Việt về các công nghệ front-end hiện đại như SASS/SCSS để sinh viên có nguồn tài liệu chính thống, dễ tiếp cận. Những điều này sẽ giúp sinh viên sớm làm quen với quy trình làm việc chuyên nghiệp, rèn luyện tư duy tổ chức mã nguồn và chuẩn bị tốt hơn cho các yêu cầu của thị trường lao động sau khi ra trường.

# D. TÀI LIỆU THAM KHẢO

1. Trần Sơn Hải, Lương Trần Hy Hiến, Lê Minh Triết, Lương Trần Ngọc Khiết (2024). *Thiết kế Web*. Nhà xuất bản: Đại học Sư Phạm Thành phố Hồ Chí Minh.