TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ GIAO DIỆN WEBSITE GIỚI THIỆU CÁC ĐỊA ĐIỂM DU LỊCH Ở ĐÀ LẠT**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:**  Nguyễn Ngọc Đan Thanh | **Sinh viên thực hiện:**  Mã số sinh viên: 110122188  Họ tên: Nguyễn Văn Tổng  Lớp: DA22TTC |

***Trà Vinh, tháng 4 năm 2024***

TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ GIAO DIỆN WEBSITE GIỚI THIỆU CÁC ĐỊA ĐIỂM DU LỊCH Ở ĐÀ LẠT**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:**  Nguyễn Ngọc Đan Thanh | **Sinh viên thực hiện:**  Mã số sinh viên: 110122188  Họ tên: Nguyễn Văn Tổng  Lớp: DA22TTC |

***Trà Vinh, tháng 4 năm 2024***

**MỤC LỤC**

[CHƯƠNG 1. Tổng quan về nội dung nghiên cứu 1](#_Toc164515441)

[1.1. Quy trình phát triển website 1](#_Toc164515442)

[1.2. Ngôn ngữ HTML 1](#_Toc164515443)

[1.2.1. Giới thiệu 1](#_Toc164515444)

[1.2.2. Cách thức hoạt động của HTML 2](#_Toc164515445)

[1.2.3. Các thành phần cơ bản của một trang HTML. 2](#_Toc164515446)

[1.3. Bảng định kiểu CSS 12](#_Toc164515447)

[1.3.1. Giới thiệu 12](#_Toc164515448)

[1.3.2. Cách thức hoạt động của CSS 13](#_Toc164515449)

[1.3.3. Các cú pháp và đơn vị của CSS 14](#_Toc164515450)

[1.3.3.1 Chọn phần tử: 14](#_Toc164515451)

[1.3.3.2 Thuộc tính CSS: 14](#_Toc164515452)

[1.3.3.3 Các đơn vị: 15](#_Toc164515453)

[1.3.3.4 Bố cục: 15](#_Toc164515454)

[1.4. Ngôn ngữ Javascript 16](#_Toc164515455)

[1.4.1. Giới thiệu 16](#_Toc164515456)

[1.4.2. Cách thức hoạt động 16](#_Toc164515457)

[1.4.3. Cú pháp Js đơn giản 16](#_Toc164515458)

[1.5. Sơ lượt về Bootstrap 20](#_Toc164515459)

[1.5.1. Giới thiệu chung về Bootstrap 20](#_Toc164515460)

[1.5.2. Đặc điểm chính: 20](#_Toc164515461)

[1.5.3. Lợi ích sử dụng Bootstrap: 21](#_Toc164515462)

[1.6. Kết chương 21](#_Toc164515463)

[CHƯƠNG 2. Phân tích thiết kế hệ thống 22](#_Toc164515464)

[2.1. Mô tả bài toán 22](#_Toc164515465)

[2.2. Đặc tả các yêu cầu 22](#_Toc164515466)

[2.2.1. Yêu cầu chức năng 22](#_Toc164515467)

[2.2.2. Yêu cầu phi chức năng 22](#_Toc164515468)

[2.3. Thiết kế giao diện 23](#_Toc164515469)

[2.3.1. Giao diện trang chủ người dùng 23](#_Toc164515470)

[2.3.1.1 Giao diện khi người dùng nhấn bắt đầu 24](#_Toc164515471)

[2.3.1.2 Giao diện cảnh quan 24](#_Toc164515472)

[2.3.2. Kết chương 25](#_Toc164515473)

[CHƯƠNG 3. Cài đặt thực nghiệm 26](#_Toc164515474)

[3.1. Kết quả thực nghiệm 26](#_Toc164515475)

[3.1.1. Chức năng 26](#_Toc164515476)

[3.1.2. Chức năng hệ thống 26](#_Toc164515477)

[3.1.3. Chức năng người dùng 26](#_Toc164515478)

[3.2. Kết chương 26](#_Toc164515479)

[CHƯƠNG 4. Kết luận và hướng phát triển 27](#_Toc164515480)

[4.1. Kết luận 27](#_Toc164515481)

[4.2. Hướng phát triển 27](#_Toc164515482)

[4.2.1. Mở rộng nội dung: 27](#_Toc164515483)

[4.2.2. Nâng cao chất lượng hình ảnh và video: 27](#_Toc164515484)

[4.2.3. Tăng cường tính tương tác: 28](#_Toc164515485)

[4.2.4. Phát triển các tính năng hữu ích: 28](#_Toc164515486)

[4.2.5. Quảng bá trang web: 28](#_Toc164515487)

[4.2.6. Phân tích dữ liệu và theo dõi hiệu quả: 28](#_Toc164515488)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1 HTML 1](#_Toc164515892)

[Hình 1.2 CSS 13](#_Toc164515893)

[Hình 1.3 Javascripts 16](#_Toc164515894)

[Hình 1.4 boostrap 20](#_Toc164515895)

[Hình 2.1 Giao diện trang chủ người dùng 24](#_Toc164515896)

[Hình 2.2 Giao diện khi người dùng nhấn bắt đầu. 25](#_Toc164515897)

[Hình 2.3 Giao diện cảnh quan 25](#_Toc164515898)

[Hình 2.4 Giao diện cảnh quan 26](#_Toc164515899)

# Tổng quan về nội dung nghiên cứu

## Quy trình phát triển website

Các quy trình phát triển của một website bao gồm:

* Thu thập thông tin: Xác định mục đích , đối tượng khách hàng, các tính năng cần thiết của website và ngân sách và thời gian hoàn thành dự án.
* Lập kế hoạch: Lập kế hoạch chi tiết cho dự án, thiết kế giao diện ban đầu.
* Phát triển website: Viết mã nguồn cho trang web và kiểm tra, thừ nghiệm website
* Khởi chạy website: Đưa website vào hoạt động.
* Bảo trì website: Cập nhật trang web thường xuyên, thêm nội dung cho trang web, sửa lỗi nếu có.

## Ngôn ngữ HTML

### Giới thiệu

[**HTML**](https://topdev.vn/viec-lam-it/html-kt75) là viết tắt của cụm từ **Hypertext Markup Language**(tạm dịch là Ngôn ngữ đánh dấu siêu văn bản). **HTML** được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes… và **HTML** không phải là ngôn ngữ lập trình[1].



Hình 1.1 HTML

### Cách thức hoạt động của HTML

Khi một người dùng nhập tên miền vào thanh địa chỉ của trình duyệt, ví dụ như Chrome, sẽ thực hiện quá trình kết nối đến máy chủ web bằng cách sử dụng địa chỉ IP đã xác định thông qua quá trình phân giải tên miền (DNS). Máy chủ web là một máy tính kết nối internet nhận các yêu cầu từ trình duyệt và sau đó gửi lại thông tin cần thiết dưới dạng tài liệu HTML để hiển thị trang web.

Tập tin HTML chứa các phần tử HTML và thường được lưu dưới đuôi mở rộng là .html hoặc .htm. Khi tập tin HTML được tạo ra, trình duyệt web là sẽ là công cụ thực thi. Trình duyệt đọc và hiểu nội dung HTML từ các thẻ có trong tập tin và chuyển đổi chúng thành dạng văn bản có thể đọc, nghe hoặc hiểu bởi các bot máy tính

### Các thành phần cơ bản của một trang HTML.

Một trang web được cấu thành từ nhiều thành phần khác nhau, dưới đây là một số thành phần cơ bản của một trang web và ý nghĩa[2]:

**Khai báo Doctype:**

Dòng đầu tiên của trang HTML5, thông báo cho trình duyệt phiên bản HTML đang sử dụng.

Ví dụ: <!DOCTYPE html>.

**Phần head:**

Phần head và phần body được đặt trong thẻ <html>

Chứa các thông tin meta về trang web, không hiển thị trực tiếp trên giao diện.

Bao gồm các thẻ như:

<title>: Tiêu đề của trang web.

Ví dụ: <title>Đây là tiêu đề trang web</title>

<meta charset="UTF-8">: Khai báo bộ ký tự sử dụng trên trang web.

<link rel="stylesheet" href="style.css">:Liên kết đến tệp tin CSS để định dạng giao diện.

Thẻ <link> có các thuộc tính sau[3]:

* href**:** Xác định URL của tài nguyên được liên kết, đây là thuộc tính bắt buộc
* rel: Xác định mối quan hệ giữa tài nguyên được liên kết và tài liệu HTML, đây là thuộc tính bắt buộc.

Các giá trị phổ biến của thuộc tính rel:

* stylesheet: Liên kết đến tệp CSS.
* icon: Liên kết đến favicon.
* preload: Tải trước tài nguyên để tăng tốc độ tải trang.
* alternate: Liên kết đến phiên bản khác của tài liệu (ví dụ: HTML5 và XHTML)
* type**:** Xác định loại MIME của tài nguyên được liên kết, thông thường không cần thiết.
* media: Xác định phương tiện truyền thông mà tài nguyên được liên kết nên được áp dụng, ít sử dụng
* integrity: Xác định giá trị băm SHA-256 của tài nguyên được liên kết để đảm bảo tính toàn vẹn. mới trong HTML5.
* crossorigin: Xác định cách trình duyệt xử lý yêu cầu chéo nguồn, mới trong HTML5.
* sizes: Xác định kích thước của tài nguyên được liên kết cho các thiết bị màn hình nhỏ, mới trong HTML5.
* as: Xác định loại tài nguyên được liên kết, mới trong HTML5.

<script src="script.js"></script>: Liên kết đến tệp tin JavaScript để thêm chức năng cho trang web.

Trong thẻ <script> có các thuộc tính sau:

* src: Xác định URL của tệp script được tải. Có thể không cần thiết nếu mã script được viết trực tiếp trong thẻ.
* Type: Xác định loại MIME của mã script. Mặc định text/javascript.
* async: Chỉ thị cho trình duyệt tải script không đồng bộ với việc tải trang, mới trong HTML5. Script được tải sẽ được thực thi ngay khi có sẵn, mà không cần đợi trang tải xong.
* defer: Chỉ thị cho trình duyệt tải script không đồng bộ với việc tải trang, nhưng trì hoãn việc thực thi cho đến khi trang tải xong, mới trong HTML5.
* charset: Xác định mã hóa ký tự của mã script, ít sử dụng.
* language: Xác định ngôn ngữ script. Mặc định là javascript.
* integrity: Xác định giá trị băm SHA-256 của mã script để đảm bảo tính toàn vẹn, mới trong HTML5.
* crossorigin: Xác định cách trình duyệt xử lý yêu cầu chéo nguồn, mới trong HTML5.

Giá trị phổ biến:

* anonymous: Gửi cookie trình duyệt cùng với yêu cầu chéo nguồn.
* use-credentials: Gửi tất cả cookie trình duyệt cùng với yêu cầu chéo nguồn.
* nonce: Xác định giá trị nonce để ngăn chặn tấn công xss, mới trong HTML5.

**Phần body:**

Chứa nội dung chính của trang web, hiển thị trực tiếp trên giao diện.

Bao gồm các thẻ HTML để tạo dựng các thành phần như:

<h1>, <h2>, ..., <h6>: Tiêu đề các cấp.

<p>: Đoạn văn.

<img>: Hình ảnh.

Các thuộc tính của thẻ <img>:

* src: Xác định URL của hình ảnh được hiển thị. Đây là thuộc tính bắt buộc
* alt: Cung cấp văn bản thay thế cho hình ảnh khi không thể hiển thị (ví dụ: do lỗi tải, trình duyệt không hỗ trợ hình ảnh, v.v.).
* width: Xác định chiều rộng của hình ảnh bằng pixel. Có thể hữu ích để điều chỉnh kích thước hình ảnh trên trang.
* height:Xác định chiều cao của hình ảnh bằng pixel. Có thể hữu ích để điều chỉnh kích thước hình ảnh trên trang.
* border: Xác định đường viền xung quanh hình ảnh. Sử dụng các giá trị CSS để định dạng đường viền. Ít được sử dụng hơn trong HTML5 do có thể định dạng đường viền bằng CSS hiệu quả hơn.
* loading: Xác định cách trình duyệt tải hình ảnh.

Giá trị phổ biến:

* lazy: Tải hình ảnh khi cuộn trang đến gần vị trí của nó.
* eager: Tải hình ảnh ngay lập tức, mới trong HTML5.

<a>: Liên kết.

Các thuộc tính riêng của thẻ <a>

* href: Xác định URL của tài nguyên mà liên kết trỏ đến. Đây là thuộc tính bắt buộc.
* target: Xác định nơi mở liên kết khi được nhấp chuột.

Giá trị phổ biến:

* \_blank: Mở liên kết trong tab hoặc cửa sổ mới.
* \_self: Mở liên kết trong cùng tab hoặc cửa sổ.
* \_parent: Mở liên kết trong cửa sổ cha.
* \_top: Mở liên kết trong cửa sổ hàng đầu.
* title: Cung cấp văn bản mô tả ngắn gọn về liên kết khi di chuột qua.

Hiển thị dưới dạng tooltip (công cụ bật lên) cho người dùng.Nên sử dụng để cung cấp thêm thông tin về nội dung mà liên kết dẫn đến.

* rel: Xác định mối quan hệ giữa trang hiện tại và tài nguyên được liên kết, mới trong HTML5.

Giá trị phổ biến:

* noopener: Ngăn chặn trang đích mở cửa sổ mới với quyền truy cập vào trang hiện tại.
* noreferrer: Ngăn chặn trang đích truyền thông tin giới thiệu về nguồn gốc của yêu cầu.
* download: Chỉ thị cho trình duyệt tải tài nguyên được liên kết xuống thay vì mở nó trong trình duyệt. Thường được sử dụng cho các tệp như PDF, tài liệu Word, hình ảnh, v.v.
* media: Xác định loại phương tiện mà liên kết áp dụng. Ít được sử dụng.
* type: Xác định loại MIME của tài nguyên được liên kết. Ít được sử dụng.
* hreflang: Xác định ngôn ngữ của tài nguyên được liên kết. Giúp trình duyệt hiển thị liên kết phù hợp với ngôn ngữ ưu tiên của người dùng, mới trong HTML5.

<ul>: Danh sách không sắp xếp.

Các thẻ riêng của thẻ <ul>

* type: Xác định kiểu hiển thị của danh sách không thứ tự.

Giá trị phổ biến đối :

* disc: Dấu chấm tròn (mặc định).
* circle: Hình tròn.
* square: Hình vuông.
* none: Không có dấu chấm hoặc biểu tượng.
* compact: Xác định liệu các mục trong danh sách có nên được hiển thị sát nhau hay không.

Giá trị:

* compact: Mục được hiển thị sát nhau.
* Không có giá trị: Mục được hiển thị với khoảng cách mặc định.

<ol> : Danh sách có sắp xếp

Các thuộc tính riêng của thẻ <ol>

* type: Xác định kiểu hiển thị của danh sách không thứ tự.

Giá trị phổ biến đối :

* 1: Số nguyên (mặc định).
* a: Chữ cái thường.
* i: Chữ số La Mã thường.
* A: Chữ cái thường.
* I: Chữ số La Mã thường.
* start: Xác định giá trị bắt đầu cho đánh số. Giá trị mặc định là 1.
* reversed: Xác định thứ tự đánh số ngược. Mục đầu tiên sẽ có số cao nhất.

<table>: Bảng.

Các thuộc tính riêng của thẻ <table>

* summary: Cung cấp tóm tắt ngắn gọn về nội dung của bảng. Hiển thị cho người dùng sử dụng trình đọc màn hình hoặc các công cụ hỗ trợ khác.
* ellspacing:Xác định khoảng cách giữa các ô trong bảng.

Giá trị: Đơn vị pixel hoặc giá trị tương đối. Mặc định là 2px.

* cellpadding: Xác định khoảng cách giữa nội dung ô và đường viền của ô.

Giá trị: Đơn vị pixel hoặc giá trị tương đối. Mặc định là 0px.

* bgcolor: Xác định màu nền cho bảng. Ít được sử dụng hơn so với việc sử dụng CSS để định dạng màu nền.

<form>: Biểu mẫu.

* action: Xác định URL của tập lệnh trên máy chủ sẽ xử lý dữ liệu biểu mẫu khi được gửi. Bắt buộc để biểu mẫu hoạt động.
* method: Xác định phương thức HTTP được sử dụng để gửi dữ liệu biểu mẫu đến máy chủ.

Giá trị phổ biến:

* GET: Gửi dữ liệu qua URL (không an toàn cho dữ liệu nhạy cảm).
* POST: Gửi dữ liệu trong thân yêu cầu (an toàn hơn cho dữ liệu nhạy cảm).

Mặc định là GET.

* enctype: Xác định cách mã hóa dữ liệu biểu mẫu trước khi gửi đến máy chủ.

Giá trị phổ biến:

* application/x-www-form-urlencoded: Mã hóa mặc định.
* multipart/form-data: Sử dụng cho các tệp được tải lên.

Mặc định là application/x-www-form-urlencoded.

* target: Xác định cửa sổ hoặc khung nơi kết quả biểu mẫu sẽ được hiển thị.

Giá trị phổ biến:

* \_self: Hiển thị kết quả trong cùng cửa sổ (mặc định).
* \_blank: Mở kết quả trong tab hoặc cửa sổ mới.
* \_top: Mở kết quả trong cửa sổ hàng đầu.
* name: Xác định tên cho biểu mẫu. Có thể sử dụng để tham chiếu biểu mẫu bằng JavaScript.
* accept-charset: Xác định tập ký tự được sử dụng để mã hóa dữ liệu biểu mẫu.

Giá trị phổ biến:

* utf-8: Mã hóa mặc định.
* ISO-8859-1: Mã hóa Latin-1.
* autocomplete: Bật hoặc tắt tính năng tự động hoàn thành cho các trường nhập trong biểu mẫu.

Giá trị:

* on: Bật tự động hoàn thành (mặc định).
* off: Tắt tự động hoàn thành.
* novalidate: Bỏ qua việc xác thực biểu mẫu mặc định của trình duyệt. JavaScript nên được sử dụng để thực hiện xác thực thay thế.
* Trong <form> thường sử các thẻ như <input>, <label>, <fieldset>, ..

Các thuộc tính trong thẻ <input>:

* type**:** Xác định loại trường nhập.

Giá trị phổ biến:

* text: Trường nhập văn bản (mặc định).
* password: Trường nhập mật khẩu.
* checkbox: Hộp kiểm.
* radio: Nút radio.
* file: Chọn tệp.
* submit: Nút gửi biểu mẫu.
* reset: Nút đặt lại biểu mẫu.
* email: Trường nhập email.
* url: Trường nhập URL.
* number: Trường nhập số.
* date: Trường nhập ngày tháng.
* range: Thanh trượt chọn giá trị trong phạm vi.
* color: Chọn màu.
* search: Trường nhập tìm kiếm.
* name: Xác định tên cho trường nhập. Sử dụng khi gửi dữ liệu biểu mẫu đến máy chủ.
* value**:** Xác định giá trị mặc định cho trường nhập. Hiển thị trong trường khi trang được tải.
* placeholder**:** Cung cấp văn bản gợi ý trong trường nhập khi nó trống. Giúp người dùng hiểu cách sử dụng trường nhập.
* required**:** Yêu cầu người dùng nhập dữ liệu vào trường trước khi gửi biểu mẫu. Trình duyệt sẽ hiển thị thông báo lỗi nếu trường nhập bị bỏ trống.
* disabled**:** Vô hiệu hóa trường nhập, khiến người dùng không thể tương tác với nó. Có thể sử dụng để hiển thị dữ liệu chỉ đọc hoặc ngăn người dùng thay đổi giá trị.
* checked**:** Đánh dấu hộp kiểm hoặc nút radio là được chọn (mặc định cho hộp kiểm).
* maxlength**:** Xác định số lượng ký tự tối đa mà người dùng có thể nhập vào trường.
* minlength**:** Xác định số lượng ký tự tối thiểu mà người dùng phải nhập vào trường.
* pattern**:** Xác định biểu thức chính quy mà giá trị nhập phải khớp. Giúp đảm bảo dữ liệu nhập hợp lệ.
* Các thuộc tính của thẻ <label>:
* for**:** Xác định ID của phần tử biểu mẫu mà nhãn được gắn với. Khi người dùng nhấp vào nhãn, con trỏ chuột sẽ tự động di chuyển đến phần tử được liên kết. Bắt buộc để thẻ label hoạt động chính xác.
* accesskey**:** Xác định phím tắt bàn phím để kích hoạt nhãn và phần tử được liên kết. Ít được sử dụng hơn trong các trình duyệt hiện đại.
* tabindex**:** Xác định thứ tự tab cho nhãn trong trình duyệt. Xác định thứ tự mà người dùng có thể di chuyển giữa các phần tử bằng phím tab.

Các thuộc tính của thẻ <fieldset>

* disabled: Vô hiệu hóa tất cả các phần tử có thể tương tác trong nhóm trường. Người dùng không thể sửa đổi dữ liệu trong các phần tử này.
* form: Xác định ID của biểu mẫu mà nhóm trường thuộc về. Giúp trình duyệt kết nối nhóm trường với biểu mẫu tương ứng.
* name: Xác định tên cho nhóm trường. Có thể sử dụng để tham chiếu nhóm trường bằng JavaScript. Ít được sử dụng hơn so với việc sử dụng thuộc tính id.
* legend: Cung cấp tiêu đề cho nhóm trường. Hiển thị văn bản mô tả nội dung của nhóm trường. Có thể sử dụng thẻ <legend> bên trong thẻ <fieldset>.

Ngoài ra còn có các thẻ để tạo cấu trúc ngữ nghĩa cho trang web như:

<header>: Tiêu đề trang.

<nav>: Menu điều hướng.

<section>: Khối nội dung.

<aside>: Nội dung phụ.

<article>: Bài viết.

<footer>: Chân trang.

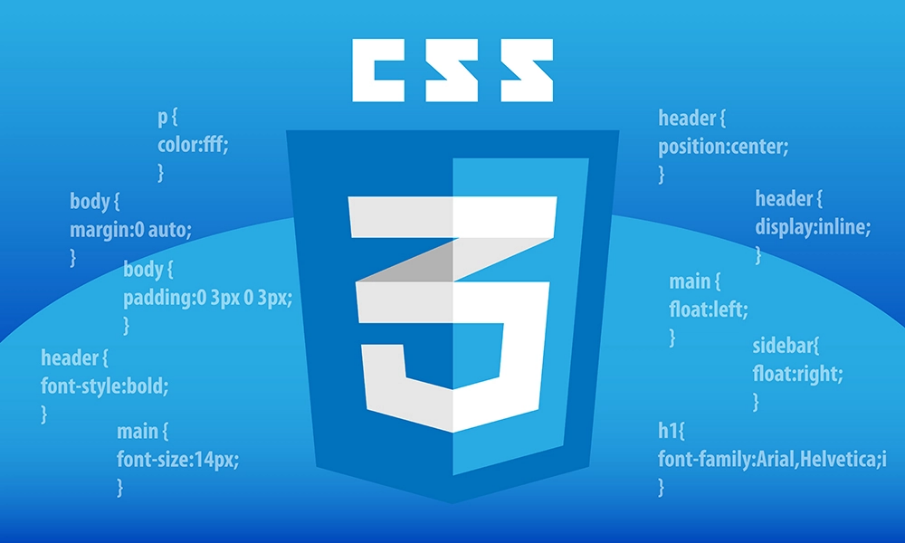
Các thẻ trên đều có các thuộc tính chung như:

* id**:** Xác định một ID duy nhất cho thẻ. Có thể sử dụng để tham chiếu thẻ bằng CSS hoặc JavaScript.
* class**:**Xác định một hoặc nhiều lớp CSS cho thẻ. Có thể sử dụng để áp dụng kiểu CSS cho thẻ.
* style**:** Áp dụng kiểu CSS trực tiếp cho thẻ. Ít được sử dụng hơn so với việc sử dụng các thuộc tính id và class.

## Bảng định kiểu CSS

### Giới thiệu

**CSS** là viết tắt của **Cascading Style Sheets**, là ngôn ngữ định dạng được sử dụng để tạo giao diện cho các trang web. CSS giúp kiểm soát cách hiển thị các phần tử HTML trên màn hình, bao gồm màu sắc, kích thước, kiểu chữ, bố cục, v.v[4].



Hình 1.2 CSS

### Cách thức hoạt động của CSS

CSS hoạt động bằng cách liên kết các quy tắc định dạng với các phần tử HTML.

Có ba cách chính để liên kết CSS với HTML:

**Sử dụng thuộc tính style:**

Thuộc tính style được đặt trược tiếp trong phần tử HTML để định dạng các phần tử đó.

**Sử dụng thẻ**<style>:

 Thẻ <style> được đặt trong phần <head> của tài liệu HTML và chứa các quy tắc CSS áp dụng cho toàn bộ trang web.

**Sử dụng tệp tin CSS ngoài:**

Tệp tin CSS được tạo riêng biệt với tệp tin HTML và liên kết đến tài liệu HTML bằng thẻ <link>.

Khi trình duyệt web hiển thị trang web, nó sẽ áp dụng các quy tắc CSS cho các phần tử HTML tương ứng, từ đó tạo ra giao diện mong muốn.

### Các cú pháp và đơn vị của CSS

#### ****Chọn phần tử:****

Sử dụng thẻ CSS để xác định các phần tử HTML mà bạn muốn định dạng. Có nhiều cách để chọn phần tử, bao gồm:

* Chọn theo thẻ HTML: h1, p, div, img, a, v.v.
* Chọn theo ID: #myElement
* Chọn theo lớp: .myClass
* Chọn theo thuộc tính HTML: [attribute="value"]
* Chọn theo con cháu: parent > child
* Chọn theo anh em: sibling1 ~ sibling2

#### ****Thuộc tính CSS:****

Thuộc tính CSS xác định các đặc điểm cụ thể mà bạn muốn áp dụng cho phần tử đã chọn. Có rất nhiều thuộc tính CSS khác nhau, bao gồm:

* color: Màu chữ
* font-family: Phông chữ
* font-size: Kích thước chữ
* background-color: Màu nền
* padding: Khoảng trống bên trong phần tử,
* margin: Khoảng trống bên ngoài phần tử
* border: Viền xung quanh phần tử
* width: Chiều rộng của phần tử
* height: Chiều cao của phần tử
* display: Kiểu hiển thị của phần tử (block, inline, inline-block, flex, v.v.)
* position: Vị trí của phần tử (static, relative, fixed, absolute)
* float: Xác định cách phần tử trôi trong bố cục (left, right, none)
* text-align: Căn chỉnh văn bản trong phần tử (left, center, right, justify)
* visibility: Hiển thị hoặc ẩn phần tử (visible, hidden)
* transition: Hiệu ứng chuyển tiếp khi thay đổi thuộc tính (color, background-color, font-size, v.v.)

#### ****Các đơn vị:****

CSS sử dụng nhiều đơn vị khác nhau để xác định kích thước, vị trí và các giá trị khác. Một số đơn vị phổ biến bao gồm:

* Pixel (px): Đơn vị cơ bản cho kích thước và vị trí trên màn hình.
* Em: Kích thước tương đối dựa trên kích thước phông chữ mặc định của phần tử cha.
* Rem: Kích thước tương đối dựa trên kích thước phông chữ gốc của tài liệu.
* %: Phần trăm của kích thước phần tử cha hoặc kích thước màn hình.
* vw: Phần trăm của chiều rộng khung nhìn.
* vh: Phần trăm của chiều cao khung nhìn.
* auto: Tự động tính toán kích thước dựa trên nội dung hoặc các yếu tố khác.

#### ****Bố cục:****

CSS cung cấp nhiều phương pháp để bố cục các phần tử trên trang, bao gồm:

* Float: Dùng để sắp xếp các phần tử theo hàng dọc hoặc ngang.
* Flexbox: Hệ thống bố cục linh hoạt và mạnh mẽ, cho phép bạn sắp xếp các phần tử theo nhiều cách khác nhau.
* Grid: Hệ thống bố cục dạng lưới, cho phép bạn tạo bố cục phức tạp với nhiều cột và hàng.
* Position: Xác định vị trí chính xác của phần tử trên trang.

## Ngôn ngữ Javascript

### Giới thiệu

**JavaScript** là ngôn ngữ lập trình kịch bản được sử dụng để tạo ra các trang web tương tác và năng động. Nó được nhúng vào trang HTML, cho phép thêm các chức năng và hiệu ứng mà không thể thực hiện chỉ với HTML và CSS[5].



Hình 1.3 Javascripts

### Cách thức hoạt động

JavaScript (Js) hoạt động bằng cách thực thi mã code được nhúng vào trang HTML.

Mã JavaScript có thể được viết trực tiếp trong trang HTML bằng thẻ <script> hoặc được liên kết từ một tệp tin JavaScript ngoài bằng thẻ <link>.

Khi trình duyệt web hiển thị trang web, nó sẽ thực thi mã JavaScript, từ đó tạo ra các chức năng và hiệu ứng mong muốn.

### Cú pháp Js đơn giản

* Khai báo biến:

Sử dụng từ khóa var, let hoặc const để khai báo biến. Ví dụ:

var name = "John Doe";

let age = 30;

const pi = 3.14159;

2. Kiểu dữ liệu:

JavaScript hỗ trợ các kiểu dữ liệu cơ bản như số nguyên (number), chuỗi (string), boolean (boolean), và mảng (array). Ví dụ:

numberVariable = 10;

stringVariable = "Hello, world!";

booleanVariable = true;

arrayVariable = [1, 2, 3, 4, 5];

3. Toán tử:

JavaScript cung cấp các toán tử toán học cơ bản như cộng (+), trừ (-), nhân (\*), chia (/), modulo (%), v.v. Ví dụ:

sum = 10 + 20;

difference = 30 - 15;

product = 5 \* 6;

quotient = 20 / 4;

remainder = 10 % 3;

4. Cấu trúc điều khiển:

JavaScript cho phép bạn kiểm soát luồng chương trình bằng các cấu trúc điều khiển như if, else, switch, for, while, v.v. Ví dụ:

if (age >= 18) { console.log("Bạn đủ tuổi để lái xe."); } else { console.log("Bạn chưa đủ tuổi để lái xe."); }

for (let i = 0; i < 5; i++) { console.log(i); }

while (counter < 10) { console.log(counter); counter++; }

5. Hàm:

Hàm là các khối mã thực hiện một nhiệm vụ cụ thể. Bạn có thể sử dụng hàm để tái sử dụng mã và làm cho chương trình JavaScript của bạn hiệu quả hơn. Ví dụ:

function greet(name) { console.log("Xin chào, " + name + "!"); }

greet("John Doe");

6. Vật thể:

Vật thể là các tập hợp các thuộc tính (key-value pairs) được sử dụng để lưu trữ dữ liệu. Ví dụ:

person = { name: "John Doe", age: 30, occupation: "Software Engineer" };

console.log(person.name); // Output: John Doe

7. DOM (Document Object Model):

DOM là đại diện theo lập trình của một trang web. Nó cho phép bạn truy cập và thao tác các phần tử HTML trong trang.

Các thao tác DOM cơ bản trong Js:

**Truy cập các phần tử:**

* Sử dụng document.getElementById(id) để truy cập phần tử có ID cụ thể. Ví dụ:

const myElement = document.getElementById("myDiv");

* Sử dụng document.getElementsByTagName(tagName) để truy cập danh sách các phần tử có thẻ HTML cụ thể. Ví dụ:

const paragraphs = document.getElementsByTagName("p");

Sử dụng document.querySelector(selector) để truy cập phần tử đầu tiên khớp với bộ chọn CSS. Ví dụ:

const firstButton = document.querySelector("button:first-child");

Sử dụng document.querySelectorAll(selector) để truy cập danh sách các phần tử khớp với bộ chọn CSS. Ví dụ:

const allImages = document.querySelectorAll("img");

**2. Thay đổi nội dung:**

Sử dụng element.textContent để lấy hoặc đặt nội dung văn bản của một phần tử. Ví dụ:

myElement.textContent = "Nội dung mới";

Sử dụng element.innerHTML để lấy hoặc đặt nội dung HTML của một phần tử, bao gồm cả thẻ HTML. Ví dụ:

myElement.innerHTML = "<p>Đây là một đoạn văn mới.</p>";

**3. Thay đổi thuộc tính:**

Sử dụng element.getAttribute(attributeName) để lấy giá trị của một thuộc tính cụ thể. Ví dụ:

const href = myElement.getAttribute("href");

Sử dụng element.setAttribute(attributeName, attributeValue) để đặt giá trị của một thuộc tính cụ thể. Ví dụ:

myElement.setAttribute("href", "https://www.example.com");

**4. Thêm và xóa phần tử:**

Sử dụng element.appendChild(childElement) để thêm một phần tử con vào cuối phần tử cha. Ví dụ:

const newParagraph = document.createElement("p");

newParagraph.textContent = "Đây là một đoạn văn mới.";

myElement.appendChild(newParagraph);

Sử dụng element.removeChild(childElement) để xóa một phần tử con khỏi phần tử cha. Ví dụ:

myElement.removeChild(firstParagraph);

Sử dụng element.insertBefore(newElement, referenceElement) để chèn một phần tử mới trước phần tử tham chiếu. Ví dụ:

const newButton = document.createElement("button");

newButton.textContent = "Nhấp vào tôi";

myElement.insertBefore(newButton, firstButton);

**5. Thêm sự kiện:**

Sử dụng element.addEventListener(eventName, eventHandler) để thêm một trình xử lý sự kiện cho một phần tử. Ví dụ:

myButton.addEventListener("click", function() {

alert("Bạn đã nhấp vào nút!");

});

## Sơ lượt về Bootstrap

### Giới thiệu chung về Bootstrap

**Bootstrap** là một framework CSS miễn phí và mã nguồn mở dành cho phát triển web front-end đáp ứng, di động trước tiên. Nó bao gồm các mẫu thiết kế dựa trên HTML, CSS và JavaScript (tùy chọn) cho kiểu chữ, biểu mẫu, nút, điều hướng và các thành phần giao diện khác[6].



Hình 1.4 boostrap

### ****Đặc điểm chính**:**

* **Miễn phí và mã nguồn mở:** Bootstrap được cấp phép theo Giấy phép MIT, cho phép bạn sử dụng và sửa đổi nó cho bất kỳ mục đích nào.
* **Dễ sử dụng:** Bootstrap cung cấp một tài liệu hướng dẫn chi tiết và dễ hiểu, giúp bạn dễ dàng bắt đầu sử dụng.
* **Phản hồi:** Bootstrap được thiết kế để tự động điều chỉnh kích thước và bố cục để phù hợp với mọi kích thước màn hình, từ máy tính để bàn đến thiết bị di động.
* **Ưu tiên di động:** Bootstrap được xây dựng với ưu tiên cho thiết bị di động, đảm bảo các trang web của bạn luôn trông đẹp mắt và hoạt động tốt trên mọi thiết bị.
* **Có thể mở rộng:** Bootstrap có thể được tùy chỉnh và mở rộng dễ dàng để đáp ứng nhu cầu cụ thể của bạn.
* **Phổ biến:** Bootstrap là framework CSS phổ biến nhất trên thế giới, với hàng triệu trang web đang sử dụng nó.

### ****Lợi ích sử dụng Bootstrap**:**

* **Tiết kiệm thời gian và tiền bạc:** Bootstrap giúp bạn tiết kiệm thời gian và tiền bạc bằng cách cung cấp cho bạn các thành phần giao diện được thiết kế và xây dựng sẵn mà bạn có thể sử dụng ngay lập tức.
* **Dễ dàng bảo trì:** Các trang web được xây dựng bằng Bootstrap dễ dàng bảo trì hơn nhờ cấu trúc mã rõ ràng và nhất quán.
* **Tăng khả năng truy cập:** Bootstrap được thiết kế để dễ tiếp cận với mọi người, bao gồm cả người dùng khuyết tật.
* **Trang web:** Bootstrap có thể được sử dụng để xây dựng mọi loại trang web, từ trang web đơn giản đến các ứng dụng web phức tạp.
* **Ứng dụng web di động:** Bootstrap có thể được sử dụng để xây dựng các ứng dụng web di động đáp ứng, hoạt động tốt trên mọi thiết bị.

## Kết chương

Chương 1 đã giới thiệu tổng quan nghiên cứu về HTML, CSS, JavaScripts và các thành phần cơ bản đề phát triển một trang web.

# Phân tích thiết kế hệ thống

## Mô tả bài toán

Hiện nay nhu cầu du lịch của người dân trong và ngoài nước ngày càng gia tăng, đặt biệt là những địa điểm nổi tiếng như Đà Lạt. Nhưng không phải ai cũng có đủ thời gian để tìm kiếm những địa điểm du lịch ở Đà Lạt và trang web giới thiệu địa điểm du lịch Đà Lạt với chất lượng không đồng đều, khiến du khách gặp khó khăn trong việc tìm kiếm thông tin chính xác và đầy đủ nên việc xây dựng một trang web giới thiệu các địa điểm du lịch cho người đi du lịch là cần thiết để tiết kiệm thời gian cho người đi du lịch.

Việc xây dựng một trang web giới thiệu địa điểm du lịch ở Đà Lạt một cách hiệu quả sẽ giúp du khách dễ dàng tìm kiếm thông tin, lên kế hoạch cho chuyến đi và có những trải nghiệm du lịch tuyệt vời nhất.

## Đặc tả các yêu cầu

### Yêu cầu chức năng

* **Hiển thị thông tin rõ ràng, súc tích:**

Hiển thị rỏ ràng các thông tin như tên, giới thiệu về các địa điểm du lịch ở Đà Lạt .

* **Tối ưu hóa cho thiết bị di động:**

Website cần hiển thị tốt trên mọi thiết bị, bao gồm máy tính để bàn, máy tính bảng và điện thoại thông minh.

Sử dụng phông chữ dễ đọc và bố cục hợp lý cho thiết bị di động.

### Yêu cầu phi chức năng

* **Hiệu suất:**

Website phải có tốc độ tải trang nhanh, mượt mà.

Có thể truy cập được trên mọi thiết bị, trình duyệt.

Đảm bảo an toàn, bảo mật cho thông tin của du khách.

* **Khả năng sử dụng:**

Giao diện website đẹp mắt, thân thiện, dễ sử dụng.

Cấu trúc website rõ ràng, logic.

Hệ thống điều hướng đơn giản, dễ tìm kiếm thông tin.

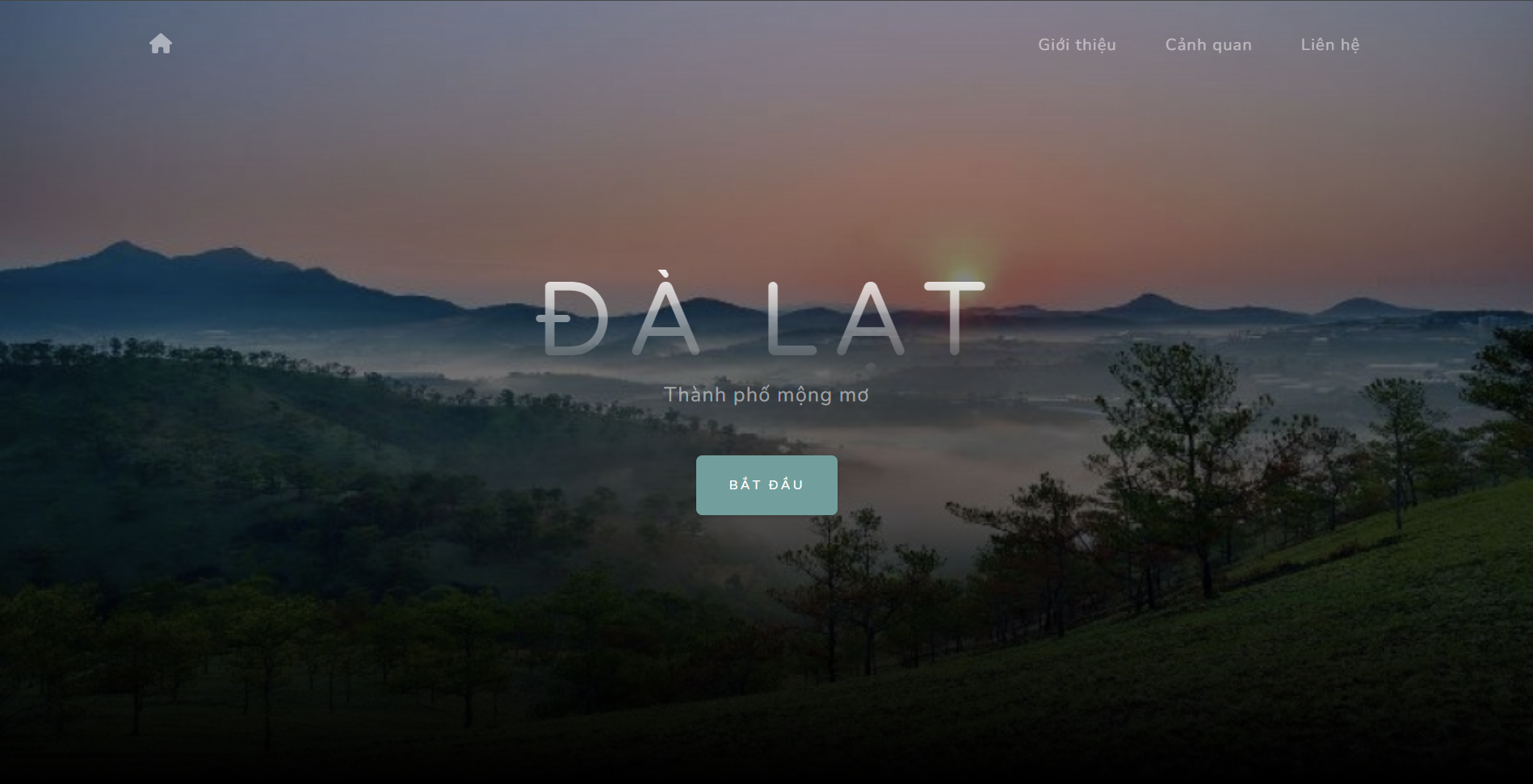
* **Khả năng mở rộng:**

Website có thể dễ dàng mở rộng để đáp ứng nhu cầu phát triển trong tương lai.

Có thể tích hợp thêm các tính năng mới một cách dễ dàng.

## Thiết kế giao diện

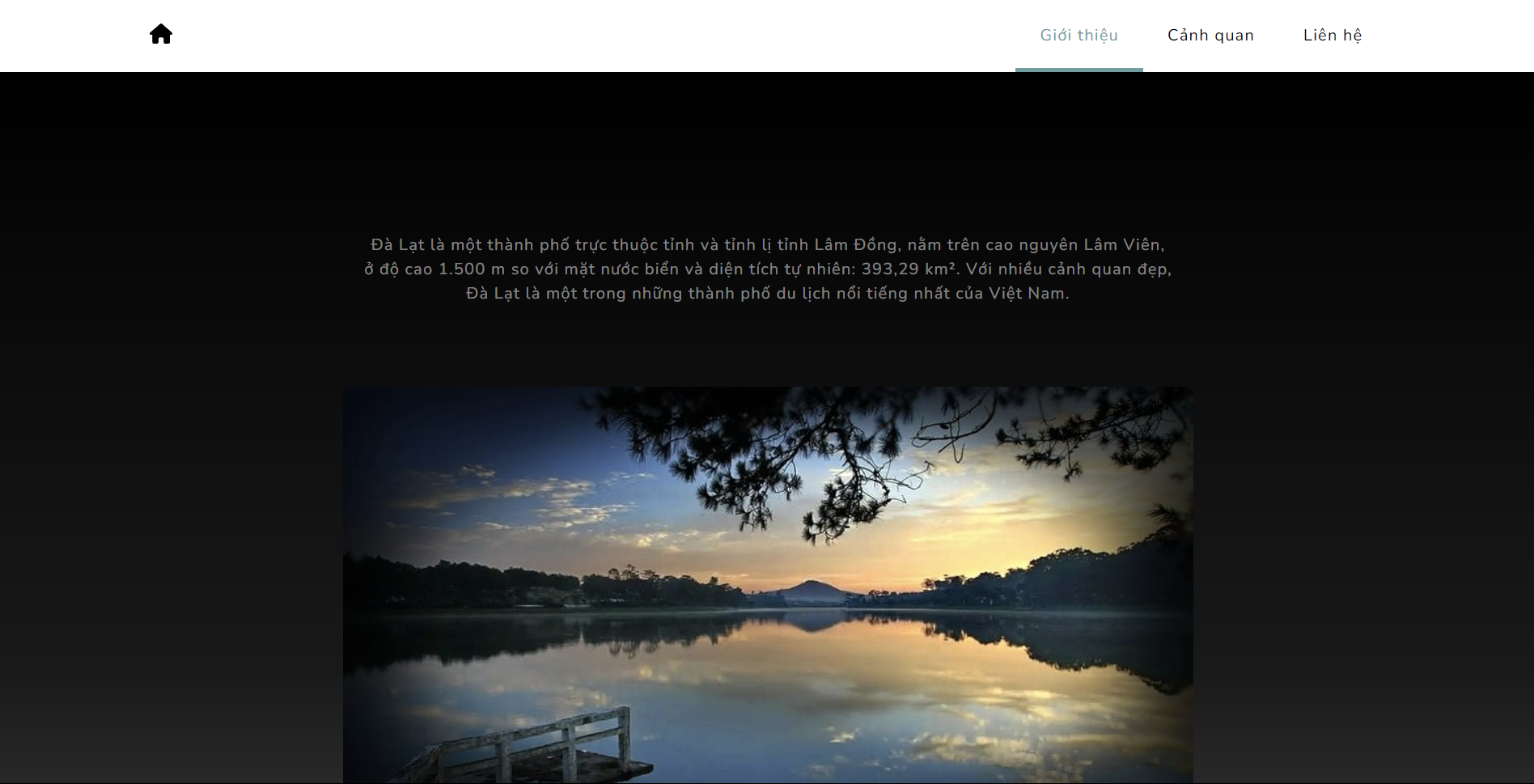
### Giao diện trang chủ người dùng



Hình 2.1 Giao diện trang chủ người dùng

Giao diện người dùng gồm thanh điều hướng có các chức năng quay về trang chủ, và đi đến các phần của nội dung như: Giới thiệu, cảnh quan, và liên hệ. Ở giữa là tiêu đề chính của trang web và một nút để người dùng bắt đầu khám phá trãng web

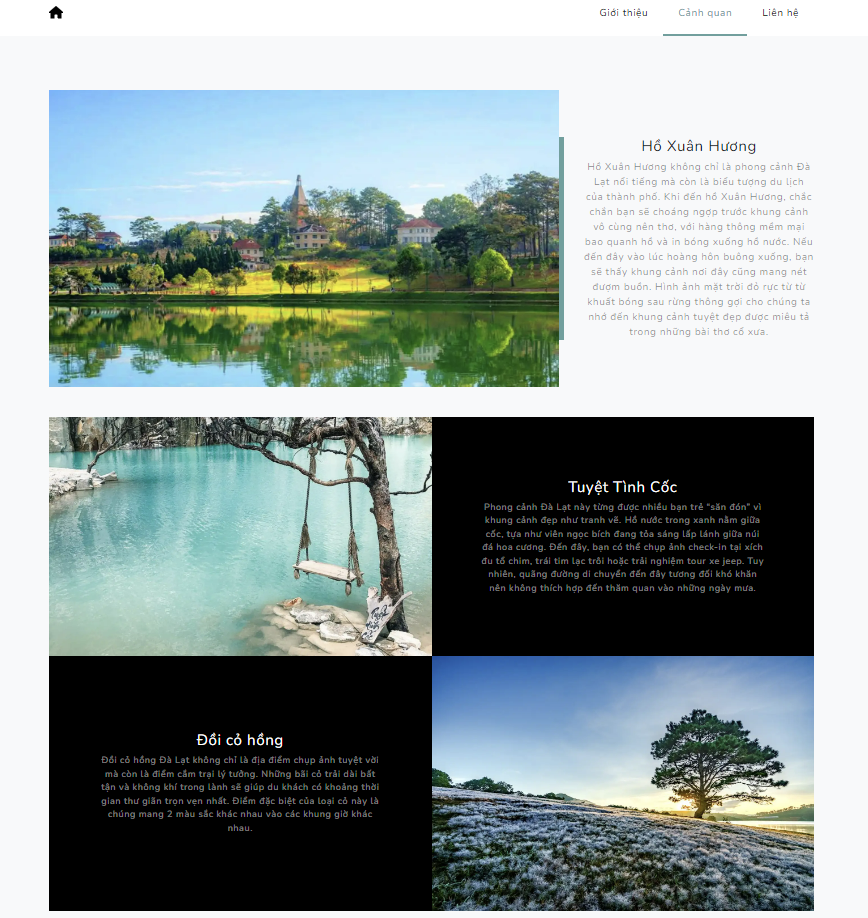
#### Giao diện khi người dùng nhấn bắt đầu



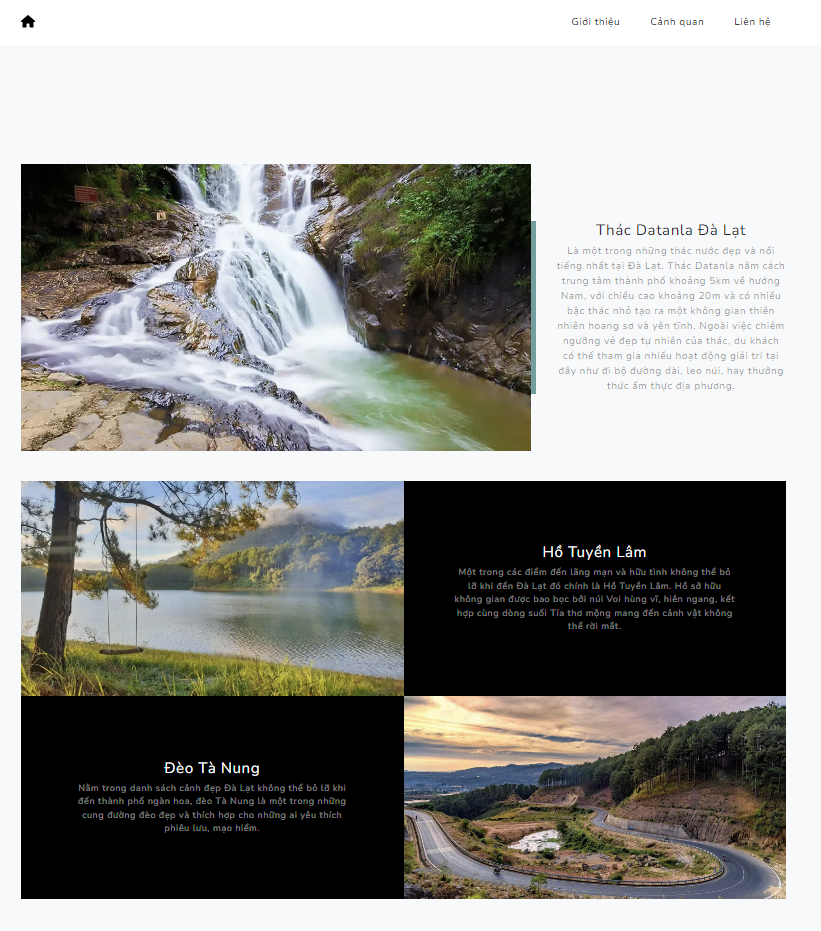
Hình 2.2 Giao diện khi người dùng nhấn bắt đầu.

Ở trang này sẽ giới thiệu cho người dùng sơ lượt về Đà Lạt.

#### Giao diện cảnh quan



Hình 2.3 Giao diện cảnh quan



Hình 2.4 Giao diện cảnh quan

Ở trang này sẽ giới thiệu cho người đọc các địa điểm du lịch nổi tiếng ở Đà Lạt.

### Kết chương

Chương 2 đã giói thiệu sơ lượt về đề tài và

# Cài đặt thực nghiệm

## Kết quả thực nghiệm

### Chức năng

### Chức năng hệ thống

Website đảm bảo

### Chức năng người dùng

## Kết chương

# Kết luận và hướng phát triển

## Kết luận

**Xây dựng trang web giới thiệu địa điểm du lịch ở Đà Lạt cho người đi du lịch** là một đề tài có tính thực tiễn cao và tiềm năng phát triển lớn. Trang web này sẽ giúp du khách dễ dàng tìm kiếm thông tin về các địa điểm du lịch ở Đà Lạt, lên kế hoạch cho chuyến đi và có những trải nghiệm du lịch tuyệt vời nhất.

Để xây dựng một trang web hiệu quả, cần tiến hành nghiên cứu kỹ lưỡng về nhu cầu và sở thích của du khách, lựa chọn công nghệ phù hợp và xây dựng nội dung chất lượng cao. Đồng thời, cần thường xuyên cập nhật thông tin và quảng bá trang web để thu hút du khách truy cập.

Với sự nỗ lực, hy vọng trang web giới thiệu địa điểm du lịch Đà Lạt sẽ trở thành một kênh thông tin hữu ích cho du khách và góp phần thúc đẩy ngành du lịch Đà Lạt phát triển.

## Hướng phát triển

### ****Mở rộng nội dung**:**

Cung cấp thông tin chi tiết và đầy đủ hơn về các địa điểm du lịch ở Đà Lạt và các tỉnh khác, bao gồm lịch sử, văn hóa, ẩm thực, mua sắm,...

Giới thiệu các hoạt động du lịch mới và độc đáo ở Đà Lạt.

Chia sẻ kinh nghiệm du lịch Đà Lạt từ những du khách experienced.

Cập nhật thường xuyên tin tức và sự kiện về du lịch Đà Lạt.

### ****Nâng cao chất lượng hình ảnh và video**:**

Sử dụng hình ảnh và video chất lượng cao, sắc nét để thu hút du khách.

Đa dạng hóa hình thức thể hiện hình ảnh và video, bao gồm ảnh chụp, video flycam, video 360 độ,...

Cho phép du khách tự đăng tải hình ảnh và video của họ lên trang web.

### ****Tăng cường tính tương tác**:**

Tích hợp các tính năng tương tác như bình luận, đánh giá, chia sẻ trên mạng xã hội.

Tạo diễn đàn để du khách có thể trao đổi thông tin và kinh nghiệm du lịch Đà Lạt.

Tổ chức các cuộc thi ảnh, viết bài về du lịch Đà Lạt.

Trả lời trực tiếp các câu hỏi và thắc mắc của du khách.

### ****Phát triển các tính năng hữu ích**:**

Tích hợp tính năng đặt phòng khách sạn, vé máy bay, tour du lịch.

Cung cấp bản đồ du lịch Đà Lạt chi tiết và tiện ích.

Cho phép du khách so sánh giá cả và đặt dịch vụ du lịch trực tiếp trên trang web.

Hỗ trợ thanh toán trực tuyến an toàn và tiện lợi.

### ****Quảng bá trang web**:**

Thực hiện chiến dịch quảng cáo trên mạng xã hội, Google Ads,...

Hợp tác với các công ty du lịch và influencer để quảng bá trang web.

Tham gia các hội chợ du lịch và sự kiện liên quan đến du lịch.

Tối ưu hóa trang web cho công cụ tìm kiếm (SEO) để thu hút du khách truy cập thông qua tìm kiếm Google.

### ****Phân tích dữ liệu và theo dõi hiệu quả**:**

Sử dụng các công cụ phân tích dữ liệu web để theo dõi lưu lượng truy cập, hành vi người dùng và hiệu quả của các chiến dịch quảng cáo.

Dựa trên dữ liệu thu thập được, điều chỉnh nội dung và tính năng của trang web để đáp ứng tốt hơn nhu cầu của du khách.

Cập nhật xu hướng du lịch mới nhất để luôn mang đến những trải nghiệm tốt nhất cho du khách.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] “HTML là gì? CSS là gì? Tổng quan HTML CSS - NordicCoder.” Accessed: Jun. 13, 2023. [Online]. Available: https://nordiccoder.com/blog/tong-quan-html-css-ban-can-biet/

[2] N. R. Jennifer, *Learning Web Design*.

[3] Static, “Thẻ <link /> trong HTML,” Thẻ <link /> trong HTML. Accessed: Apr. 17, 2024. [Online]. Available: https://hoclaptrinh.vn/tutorial/tag-html/the-link-trong-html

[4] T. Blog, “CSS là gì?,” TopDev. Accessed: Apr. 15, 2024. [Online]. Available: https://topdev.vn/blog/css-la-gi/

[5] “JavaScript là gì? - Giải thích về JavaScript (JS) - AWS,” Amazon Web Services, Inc. Accessed: Apr. 15, 2024. [Online]. Available: https://aws.amazon.com/vi/what-is/javascript/

[6] Hưng N., “Bootstrap là gì? Giới thiệu chi tiết về Bootstrap nhất 2024.” Accessed: Apr. 15, 2024. [Online]. Available: https://vietnix.vn/bootstrap-la-gi/

**PHỤ LỤC**

**Cách nhúng bootstrap vào trang HTML**

Có hai cách:

Cách 1: Nhúng bằng liên kết của Bootstrap

Nhúng Bootstrap vào HTML thông qua liên kết của Bootstrap (CDN). Để làm điều này, bạn sẽ cần truy cập vào trang chủ của Bootstrap, và sao chép đoạn mã CDN của Bootstrap vào thẻ <head> của trang HTML

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">  
  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

Cách 2: Nhúng bằng cách tự Host

Để nhúng Bootstrap vào HTML bằng cách tự host, bạn cần tải Bootstrap từ trang chủ của nó. Sau đó, giải nén ra thư mục được đặt tại đường dẫn chuyển đến các tệp CSS và JS trong thư mục tải xuống và sao chép chúng vào thư mục của dự án của bạn.

**Các class bootstrap được sử dụng trong dự án:**

Class trong phần Navigation:

navbar-expand-lg: Giúp thanh điều hướng mở rộng khi màn hình lớn hơn 992px.

navbar-light: Giúp thanh điều hướng có màu sáng.

fixed-top: Giúp thanh điều hướng cố định ở đầu trang khi cuộn trang.

container px-4 px-lg-5: Giúp thanh điều hướng có kích thước phù hợp với màn hình và thêm khoảng đệm bên trái và bên phải.

navbar-toggler: Giúp hiển thị nút hamburger để thu gọn thanh điều hướng khi màn hình nhỏ.

navbar-toggler-right: Giúp nút hamburger hiển thị ở bên phải thanh điều hướng.

collapse navbar-collapse: Giúp thu gọn phần nội dung của thanh điều hướng khi màn hình nhỏ.

nav-item: Giúp tạo các mục trong thanh điều hướng.

nav-link: Giúp tạo liên kết đến các trang khác trong website.

Class trong phần Masthead:

masthead: Giúp tạo phần tiêu đề trang web.

container px-4 px-lg-5: Giúp phần tiêu đề có kích thước phù hợp với màn hình và thêm khoảng đệm bên trái và bên phải.

d-flex h-100 align-items-center justify-content-center: Giúp nội dung trong phần tiêu đề hiển thị theo chiều dọc, cao 100%, các mục được căn chỉnh theo chiều dọc và chiều ngang.

text-center: Giúp căn chỉnh nội dung trong phần tiêu đề theo chiều ngang.

Class trong phần About:

about-section text-center: Giúp tạo phần giới thiệu về website với nội dung được căn chỉnh theo chiều ngang.

container px-4 px-lg-5: Giúp phần giới thiệu có kích thước phù hợp với màn hình và thêm khoảng đệm bên trái và bên phải.

row gx-4 gx-lg-5 justify-content-center: Giúp tạo các hàng trong phần giới thiệu với khoảng cách giữa các cột là 4px trên màn hình nhỏ và 5px trên màn hình lớn.

col-lg-8: Giúp cột chứa nội dung giới thiệu có độ rộng 8 trên 12 cột trên màn hình lớn.

text-white-50: Giúp nội dung có màu trắng và độ mờ 50%.

Class trong phần Phong cảnh:

projects-section bg-light: Giúp tạo phần giới thiệu về các địa điểm du lịch với nền màu sáng.

container px-4 px-lg-5: Giúp phần giới thiệu có kích thước phù hợp với màn hình và thêm khoảng đệm bên trái và bên phải.

row gx-0 mb-4 mb-lg-5 align-items-center: Giúp tạo các hàng trong phần giới thiệu với khoảng cách giữa các cột là 0px, thêm khoảng cách dưới 4px trên màn hình nhỏ và 5px trên màn hình lớn, các mục được căn chỉnh theo chiều dọc.

col-xl-8 col-lg-7: Giúp cột chứa ảnh có độ rộng 8 trên 12 cột trên màn hình lớn và 7 trên 12 cột trên màn hình nhỏ.

col-xl-4 col-lg-5: Giúp cột chứa nội dung có độ rộng 4 trên 12 cột trên màn hình lớn và 5 trên 12 cột trên màn hình nhỏ.

featured-text text-center text-lg-left: Giúp nội dung được căn chỉnh theo chiều ngang trên màn hình nhỏ và căn chỉnh trái trên màn hình lớn.

bg-black text-center h-100 project: Giúp tạo phần giới thiệu về địa điểm du lịch với nền màu đen, nội dung được căn chỉnh theo chiều ngang và cao 100%.

d-flex h-100: Giúp nội dung trong phần giới thiệu địa điểm du lịch hiển thị theo chiều dọc, cao 100%.

project-text w-100 my-auto text-center text-lg-left: Giúp nội dung trong phần giới thiệu địa điểm du lịch có độ rộng 100%, được căn chỉnh theo chiều dọc và căn chỉnh trái trên màn hình lớn.

Class trong phần Signup:

signup-section: Giúp tạo phần đăng ký nhận thông tin.

container px-4 px-lg-5: Giúp phần đăng ký có kích thước phù hợp với màn hình và thêm khoảng đệm bên trái và bên phải.

row gx-4 gx-lg-5: Giúp tạo các hàng trong phần đăng ký với khoảng cách giữa các cột là 4px trên màn hình nhỏ và 5px trên màn hình lớn.

col-md-10 col-lg-8: Giúp cột chứa nội dung đăng ký có độ rộng 10 trên 12 cột trên màn hình nhỏ và 8 trên 12 cột trên màn hình lớn.

text-center: Giúp căn chỉnh nội dung trong phần đăng ký theo chiều ngang.

mb-5 mb-md-0: Giúp thêm khoảng cách dưới 5px trên màn hình nhỏ và 0px trên màn hình lớn.

Class trong phần Footer:

footer py-5 bg-black text-white: Giúp tạo phần chân trang với nền màu đen, nội dung màu trắng và có khoảng đệm trên và dưới là 5px.

container px-4 px-lg-5: Giúp phần chân trang có kích thước phù hợp với màn hình và thêm khoảng đệm bên trái và bên phải.

small text-center: Giúp nội dung trong phần chân trang có kích thước nhỏ và được căn chỉnh theo chiều ngang.