TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ NGÀNH**

**THIẾT KẾ WEBSITE GIỚI THIỆU LỊCH SỬ VÀ SI SẢN VĂN HÓA TỈNH TRÀ VINH**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:**  ThS Phan Thị Phương Nam | **Sinh viên thực hiện:**  Mã số sinh viên: 110122081  Họ tên: Trần Tấn Hưng  Lớp: DA22TTD |

***Trà Vinh, tháng 12 năm 2024***

TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ NGÀNH**

**THIẾT KẾ WEBSITE GIỚI THIỆU LỊCH SỬ VÀ DI SẢN VĂN HÓA TỈNH TRÀ VINH**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:**  ThS Phan Thị Phương Nam | **Sinh viên thực hiện:**  Mã số sinh viên: 110122081  Họ tên: Trần Tấn Hưng  Lớp: DA22TTD |

***Trà Vinh, tháng 12 năm 2024***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Trước hết, tôi xin bày tỏ lòng biết ơn sâu sắc đến Thầy/Cô đã hướng dẫn, động viên và tạo điều kiện cho tôi hoàn thành báo cáo này. Nhờ có sự giúp đỡ và chỉ bảo tận tình của Thầy/Cô, tôi đã có thể hoàn thiện được các nội dung nghiên cứu và phát triển website về Lịch sử và Di sản văn hóa tỉnh Trà Vinh.

Tôi cũng xin gửi lời cảm ơn chân thành đến các chuyên gia, tổ chức, và cá nhân đã cung cấp những tài liệu quý báu, thông tin nghiên cứu, và sự hỗ trợ trong quá trình tìm hiểu và xây dựng nội dung. Những đóng góp này đã giúp tôi hiểu rõ hơn về giá trị lịch sử, văn hóa đặc sắc của tỉnh Trà Vinh.

Cuối cùng, tôi xin gửi lời cảm ơn đến gia đình và bạn bè đã luôn động viên và hỗ trợ tôi trong suốt quá trình thực hiện báo cáo này.

Chân thành cảm ơn!

**MỤC LỤC**

[1. Lý do chọn đề tài ix](#_Toc187194081)

[2. Mục tiêu ix](#_Toc187194082)

[3. Nội dung ix](#_Toc187194083)

[4. Đối tượng và phạm vi nghiên cứu x](#_Toc187194084)

[5. Phương pháp nghiên cứu x](#_Toc187194085)

[CHƯƠNG 1. TỔNG QUAN 1](#_Toc187194086)

[CHƯƠNG 2. Nghiên cứu lý thuyết 2](#_Toc187194087)

[2.1. Quy trình phát triển website 2](#_Toc187194088)

[2.1.1. Lên kế hoạch và thu thập yêu cầu: 2](#_Toc187194089)

[2.1.2. Thiết kế giao diện: 2](#_Toc187194090)

[2.1.3. Phát triển website: 2](#_Toc187194091)

[2.1.4. Thử nghiệm và triển khai: 2](#_Toc187194092)

[2.1.5. Bảo trì và cập nhật: 2](#_Toc187194093)

[2.1.6. Ngoài ra, quy trình phát triển website có thể bao gồm thêm các bước sau: 2](#_Toc187194094)

[2.2. Ngôn ngữ HTML 3](#_Toc187194095)

[2.2.1. Khái niệm về HTML 3](#_Toc187194096)

[2.2.2. HTML cơ bản 3](#_Toc187194097)

[2.2.3. Phần tử HTML 4](#_Toc187194098)

[2.2.4. Thuộc tính HTML 5](#_Toc187194099)

[2.2.5. Đoạn văn bản trong HTML 7](#_Toc187194100)

[2.2.6. Liên kết trong HTML 9](#_Toc187194101)

[2.2.7. Hình ảnh trong HTML 11](#_Toc187194102)

[2.2.8. Bảng trong HTML: 12](#_Toc187194103)

[2.2.9. Danh sách trong HTML 17](#_Toc187194104)

[2.2.10. Khung nội tuyến trong HTML 19](#_Toc187194105)

[2.2.11. Đường dẫn tệp trong HTML 20](#_Toc187194106)

[2.2.12. Biểu mẫu trong HTML 21](#_Toc187194107)

[2.3. Bảng định kiểu CSS 26](#_Toc187194108)

[2.3.1. CSS nội tuyến (Inline CSS) 26](#_Toc187194109)

[2.3.2. CSS nội bộ (Internal CSS) 26](#_Toc187194110)

[2.3.3. CSS bên ngoài (External CSS) 27](#_Toc187194111)

[2.4. Ngôn ngữ Javascript 27](#_Toc187194112)

[2.4.1. Giới thiệu về Javascript 27](#_Toc187194113)

[2.4.2. Thêm Javascript vào trang 28](#_Toc187194114)

[2.4.3. Javascript cơ bản 28](#_Toc187194115)

[2.4.4. Variables (Biến) 29](#_Toc187194116)

[2.4.5. Data types (Loại dữ liệu) 30](#_Toc187194117)

[2.4.6. Comparison Operators (Toán tử so sánh) 31](#_Toc187194118)

[2.4.7. Mathematical operators (Toán tử toán học) 32](#_Toc187194119)

[2.4.8. if/else statements (câu lệnh if/else) 32](#_Toc187194120)

[2.4.9. Loops (vòng lặp) 32](#_Toc187194121)

[2.4.10. Functions (Chức năng) 33](#_Toc187194122)

[2.4.11. Variable Scope and the var Keyword (Phạm vi biến và từ khóa var) 34](#_Toc187194123)

[2.4.12. The Browser Object (Đối tượng trình duyệt) 34](#_Toc187194124)

[2.4.13. Events 35](#_Toc187194125)

[2.4.14. DOM 36](#_Toc187194126)

[2.5. Ngôn ngữ PHP. 47](#_Toc187194127)

[2.5.1. Giới thiệu về PHP. 47](#_Toc187194128)

[2.5.2. Cú pháp PHP cơ bản 48](#_Toc187194129)

[2.5.3. Biểu mẫu PHP. 50](#_Toc187194130)

[2.6. Kết chương. 52](#_Toc187194131)

[CHƯƠNG 3. HIỆN THỰC HOÁ NGHIÊN CỨU 54](#_Toc187194132)

[3.1. Mô tả bài toán 54](#_Toc187194133)

[3.2. Đặc tả các yêu cầu 55](#_Toc187194134)

[3.2.1. Yêu cầu chức năng 55](#_Toc187194135)

[3.2.1.1 Trang chủ 55](#_Toc187194136)

[3.2.1.2 Thông tin lịch sử 55](#_Toc187194137)

[3.2.1.3 Di sản văn hóa 55](#_Toc187194138)

[3.2.1.4 Địa điểm du lịch 55](#_Toc187194139)

[3.2.1.5 Tương tác và đóng góp nội dung 55](#_Toc187194140)

[3.2.1.6 Tích hợp đa phương tiện 55](#_Toc187194141)

[3.2.1.7 Quản trị website 55](#_Toc187194142)

[3.2.2. Yêu cầu phi chức năng 56](#_Toc187194143)

[3.2.2.1 Hiệu năng (Performance) 56](#_Toc187194144)

[3.2.2.2 Tính tương thích (Compatibility) 56](#_Toc187194145)

[3.2.2.3 Bảo mật (Security) 56](#_Toc187194146)

[3.2.2.4 Khả năng mở rộng (Scalability) 56](#_Toc187194147)

[3.2.2.5 Tối ưu hóa công cụ tìm kiếm (SEO) 56](#_Toc187194148)

[3.2.3. Công cụ và công nghệ đề xuất 56](#_Toc187194149)

[3.2.3.1 Công nghệ phát triển 56](#_Toc187194150)

[3.2.4. Yêu cầu chức năng 56](#_Toc187194151)

[3.2.4.1 Trang chủ 56](#_Toc187194152)

[3.2.4.2 Lịch sử Trà Vinh 57](#_Toc187194153)

[3.2.4.3 Di sản văn hóa 57](#_Toc187194154)

[3.2.4.4 Địa điểm tham quan 57](#_Toc187194155)

[3.2.4.5 Đóng góp nội dung 57](#_Toc187194156)

[3.2.4.6 Liên hệ (Contact Section) 57](#_Toc187194157)

[3.3. Thiết kế dữ liệu 58](#_Toc187194158)

[3.3.1. Sơ đồ hệ thống 58](#_Toc187194159)

[Tên bảng: posts 58](#_Toc187194160)

[3.3.2. Mô hình dữ liệu 59](#_Toc187194161)

[3.4. Thiết kế giao diện 60](#_Toc187194162)

[CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU 62](#_Toc187194163)

[4.1. Giao diện trang chủ người dùng 62](#_Toc187194164)

[4.2. Giao diện chức năng tìm kiếm 63](#_Toc187194165)

[4.3. Giao diện chức năng dòng lịch sử 63](#_Toc187194166)

[4.4. Giao diện chức năng tư liệu 64](#_Toc187194167)

[4.5. Giao diện chức năng câu hỏi. 64](#_Toc187194168)

[4.6. Giao diện chức năng đăng ký 65](#_Toc187194169)

[4.7. Giao diện chức năng đăng tải bài viết 66](#_Toc187194170)

[4.8. Cơ sơ dữ liệu 66](#_Toc187194171)

[4.8.1. Cơ sở dữ liệu của hệ thống 66](#_Toc187194172)

[4.9. Kết chương 66](#_Toc187194173)

[CHƯƠNG 5. Kết luận và hướng phát triển 68](#_Toc187194174)

[5.1. Kết luận 68](#_Toc187194175)

[5.2. Hướng phát triển 68](#_Toc187194176)

**DANH MỤC HÌNH ẢNH**

[Hình 3.1 Sơ dồ hệ thống 58](#_Toc187268005)

[Hình 3.1 Mô hình dữ liệu 60](#_Toc187268006)

[Hình 4.1 Giao diện trang chủ của người dùng 62](#_Toc187268007)

[Hình 4.2 Giao diện chức năng tìm kiếm 63](#_Toc187268008)

[Hình 4.3 Hình giao diện chức năng dòng lịch sử 63](#_Toc187268009)

[Hình 4.4 Hình giao diện chức năng tư liệu 64](#_Toc187268010)

[Hình 4.5 Hình giao diện chức năng câu hỏi 64](#_Toc187268011)

[Hình 4.6 Hình giao diện người dùng 65](#_Toc187268012)

[Hình 4.7 Hình giao diện chức năng đăng nhập / đăng ký 65](#_Toc187268013)

[Hình 4.8 Hình giao diện chức năng đăng tải bài viết 66](#_Toc187268014)

[Hình 4.9 Hình giao diện chức năng quản trị được cung cấp bởi Xampp 66](#_Toc187268015)

**DANH MỤC BẢNG BIỂU**

[*Bảng 3.1 Bảng mô tả thuộc tính bảng post* 58](#_Toc187194486)

[*Bảng 3.2 Bảng mô tả thuộc tính bảng accouts* 59](#_Toc187194487)

**TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH**

Như chúng ta đã biết, di sản văn hóa và lịch sử là những tài sản vô giá của mỗi địa phương. Tuy nhiên, nhiều người vẫn chưa có đủ thông tin về các giá trị lịch sử và văn hóa đặc sắc của tỉnh Trà Vinh. Đây là một vấn đề cần được giải quyết để bảo tồn và phát huy giá trị di sản văn hóa của địa phương. Đặc biệt, việc thiếu một nền tảng giới thiệu đầy đủ và chi tiết về lịch sử và di sản văn hóa tỉnh Trà Vinh đang là một trở ngại lớn trong việc nâng cao nhận thức cộng đồng và du khách về giá trị này.

Hướng tiếp cận và giải pháp:

- Nghiên cứu và thu thập thông tin về lịch sử và các di sản văn hóa đặc trưng của tỉnh Trà Vinh.

- Xây dựng website giới thiệu thông tin lịch sử và di sản văn hóa tỉnh Trà Vinh bằng các ngôn ngữ cơ bản dễ tiếp cận đến người dùng như HTML, CSS, JavaScript và PHP.

- Cung cấp các bài viết, hình ảnh, và bản đồ về các địa danh lịch sử và di sản văn hóa nổi bật.

- Tích hợp tính năng tìm kiếm thông minh và bản đồ tương tác giúp người dùng dễ dàng khám phá các điểm di tích, di sản văn hóa và các sự kiện lịch sử.

Kết quả đạt được:

- Website hoạt động đầy đủ với giao diện thân thiện và dễ sử dụng.

- Cung cấp thông tin chi tiết về lịch sử, văn hóa và các di tích của tỉnh Trà Vinh, bao gồm ảnh minh họa, mô tả và các câu chuyện lịch sử.

- Phát triển tính năng tìm kiếm giúp người dùng dễ dàng tìm kiếm thông tin về các di sản văn hóa.

- Kiến thức lý thuyết về phát triển web hiện đại, thiết kế giao diện và tối ưu hóa trải nghiệm người dùng.

Chủ đề đã thành công trong việc giới thiệu đầy đủ và sinh động về lịch sử và di sản văn hóa tỉnh Trà Vinh, qua đó góp phần bảo tồn và phát huy những giá trị văn hóa truyền thống, đồng thời giúp du khách và cộng đồng có thể dễ dàng tiếp cận các thông tin quan trọng này.

**MỞ ĐẦU**

## Lý do chọn đề tài

Trà Vinh là một tỉnh miền Tây Nam Bộ, nổi bật với sự đa dạng về văn hóa, lịch sử và di sản truyền thống. Tuy nhiên, những giá trị này vẫn chưa được khai thác và giới thiệu rộng rãi, đặc biệt là trên các nền tảng số, mặc dù sự phát triển của công nghệ thông tin và Internet đã mang lại nhiều cơ hội để bảo tồn và phát huy di sản văn hóa. Các công cụ trực tuyến và website có thể đóng vai trò quan trọng trong việc quảng bá và nâng cao nhận thức cộng đồng về các giá trị văn hóa đặc sắc của địa phương.

Tuy nhiên, hiện nay, việc tìm kiếm thông tin về lịch sử và di sản văn hóa của Trà Vinh vẫn còn gặp nhiều khó khăn, do thiếu các nguồn tài liệu đầy đủ và dễ tiếp cận. Cũng đã có một số website chuyên biệt, cập nhật đầy đủ và phong phú về các di tích lịch sử, các lễ hội, cũng như các giá trị văn hóa độc đáo của Trà Vinh nhưng còn quá khó tiếp cận đến người dùng. Vì vậy, việc xây dựng một website giới thiệu lịch sử và di sản văn hóa của tỉnh Trà Vinh là rất cần thiết, giúp người dân địa phương và du khách dễ dàng tiếp cận thông tin, khám phá những nét đẹp văn hóa, cũng như góp phần bảo tồn và phát huy những giá trị quý báu này.

Với mục tiêu trên, tôi quyết định chọn đề tài "Xây dựng website giới thiệu về Lịch sử và Di sản Văn hóa tỉnh Trà Vinh" nhằm giúp cho việc quảng bá và bảo tồn di sản văn hóa trở nên hiệu quả hơn, đồng thời cung cấp một công cụ hữu ích cho cộng đồng trong việc tìm hiểu về quá trình hình thành, phát triển cũng như các nét đặc trưng của nền văn hóa Trà Vinh. Website này không chỉ đáp ứng nhu cầu tìm kiếm thông tin mà còn tạo cầu nối giữa quá khứ và hiện tại, giúp phát triển du lịch và nâng cao nhận thức cộng đồng về tầm quan trọng của việc bảo vệ di sản văn hóa.

## Mục tiêu

Phát triển trang web hoàn chỉnh: Xây dựng một website đầy đủ và dễ sử dụng, hỗ trợ người dùng trong việc tìm kiếm và khám phá các thông tin về lịch sử và di sản văn hoá của tỉnh Trà Vinh.

Cung cấp đầy đủ thông tin về lịch sử và di sản văn hoá Trà Vinh: Trang web sẽ tổng hợp và giới thiệu các thông tin phong phú về các di tích lịch sử, các lễ hội truyền thống, các sản phẩm văn hóa đặc trưng và những câu chuyện liên quan đến vùng đất Trà Vinh. Mục tiêu là giúp người sử dụng dễ dàng nắm bắt các thông tin quý báu, góp phần nâng cao nhận thức và ý thức bảo tồn di sản.

Áp dụng công nghệ hiện đại: Sử dụng các công nghệ web hiện đại trong việc thiết kế và phát triển trang web, đảm bảo tính năng động, hiệu quả và dễ sử dụng, từ đó giúp người dùng có trải nghiệm thuận tiện và trực quan nhất.

## Nội dung

Sử dụng HTML, CSS, PHP và JavaScript: Trang web sẽ được xây dựng chủ yếu dựa trên các công nghệ cơ bản như HTML, CSS, PHP và JavaScript. HTML sẽ được sử dụng để cấu trúc nội dung trang web, CSS giúp tạo ra giao diện đẹp mắt và dễ nhìn, trong khi PHP sẽ xử lý các tác vụ phía máy chủ, giúp trang web hoạt động linh hoạt và hiệu quả. JavaScript sẽ được áp dụng để xử lý các tương tác động, cải thiện trải nghiệm người dùng, như các hiệu ứng cuộn, tìm kiếm động và các tính năng tương tác khác.

Giao diện thân thiện, dễ sử dụng: Một trong những yếu tố quan trọng là thiết kế giao diện người dung đơn giản nhưng hiệu quả. Giao diện sẽ được tối ưu để người dùng dễ dàng truy cập và tìm kiếm thông tin về lịch sử, di sản văn hóa Trà Vinh. Các chức năng cơ bản như điều hướng, tìm kiếm và xem chi tiết các di tích sẽ được thiết kế dễ dàng sử dụng trên mọi thiết bị. Mục tiêu là mang đến một trải nghiệm trực quan và thân thiện, giúp người dùng từ mọi lứa tuổi đều có thể sử dụng một cách dễ dàng.

## Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- Các di sản và lịch sử văn hóa tỉnh Trà Vinh: Đây là đối tượng chính trong nghiên cứu, bao gồm các di tích lịch sử, các lễ hội, và những giá trị văn hóa đặc trưng của tỉnh Trà Vinh. Các di sản này không chỉ bao gồm các công trình kiến trúc, di tích lịch sử mà còn bao gồm các phong tục, tập quán, và những nét văn hóa đặc sắc của địa phương.

- Phát triển website giới thiệu lịch sử và di sản văn hóa: Nghiên cứu về các phương pháp và kỹ thuật xây dựng một website giới thiệu các di sản văn hóa và lịch sử của Trà Vinh, bao gồm cả việc thiết kế giao diện, phát triển tính năng, và tối ưu hóa trải nghiệm người dùng.

- Các công nghệ web cơ bản (HTML, CSS, PHP, JavaScript): Sử dụng các công nghệ này để xây dựng một nền tảng website hiệu quả và dễ sử dụng, giúp người dùng dễ dàng truy cập và tìm kiếm thông tin về các di sản văn hóa và lịch sử của tỉnh Trà Vinh.

- Tính năng tìm kiếm thông minh và tương tác người dùng: Phát triển các tính năng tìm kiếm, điều hướng thông minh và các công cụ giúp người dùng dễ dàng tiếp cận thông tin liên quan đến các di sản văn hóa, lịch sử Trà Vinh.

- Phạm vi nghiên cứu:

- Nghiên cứu và xây dựng hệ thống thông tin về các di sản văn hóa và lịch sử của Trà Vinh, bao gồm việc thu thập, phân loại và trình bày thông tin một cách có hệ thống và dễ dàng tiếp cận qua website.

- Phát triển và áp dụng các công nghệ web như HTML, CSS, PHP, JavaScript để thiết kế giao diện và chức năng website, đảm bảo người dùng có thể dễ dàng tìm kiếm và truy cập các thông tin về lịch sử, di sản và văn hóa.

## Phương pháp nghiên cứu

**Phương pháp nghiên cứu lý thuyết:**

Nghiên cứu các tài liệu lý thuyết về lịch sử và di sản văn hoá của tỉnh Trà Vinh, bao gồm các di tích lịch sử, các lễ hội, các phong tục tập quán và những giá trị văn hoá đặc trưng của địa phương.

Tìm hiểu các phương pháp trình bày thông tin lịch sử và di sản một cách rõ ràng và dễ hiểu trên các nền tảng trực tuyến, từ đó áp dụng vào việc thiết kế nội dung cho website.

**Phương pháp thực nghiệm:**

Dựa trên các kiến thức lý thuyết về lịch sử và di sản văn hoá, tiến hành xây dựng website giới thiệu về các giá trị này của tỉnh Trà Vinh, sử dụng các công nghệ cơ bản như HTML, CSS, PHP và JavaScript.

Thiết kế giao diện website sao cho dễ dàng truy cập và tìm kiếm thông tin về các di tích lịch sử, lễ hội, và các yếu tố văn hoá đặc trưng của Trà Vinh.

Thực hiện kiểm thử và điều chỉnh giao diện người dùng để đảm bảo tính thân thiện, dễ sử dụng và dễ dàng tương tác với các thông tin trên websit

# TỔNG QUAN

Trong những năm gần đây, việc bảo tồn và phát huy giá trị lịch sử, văn hoá di sản trở thành một nhiệm vụ quan trọng đối với nhiều địa phương tại Việt Nam. Tuy nhiên, việc giới thiệu và truyền tải các thông tin về lịch sử, di sản văn hoá qua các nền tảng số vẫn còn gặp không ít khó khăn. Một trong những vấn đề chính là thiếu các website tập trung và đầy đủ thông tin về di sản văn hoá, cũng như các công cụ tìm kiếm và chia sẻ thông tin dễ dàng.

Tại tỉnh Trà Vinh, một vùng đất có nền văn hoá đa dạng và phong phú, việc giới thiệu lịch sử và di sản văn hoá trên môi trường trực tuyến vẫn chưa thực sự được khai thác mạnh mẽ. Các di tích lịch sử, các lễ hội truyền thống và các giá trị văn hoá đặc sắc của địa phương vẫn chưa được người dân và du khách biết đến rộng rãi.

Nhận thức được thách thức này, đề tài này hướng đến việc phát triển một website chuyên biệt giới thiệu về lịch sử và di sản văn hoá tỉnh Trà Vinh. Website sẽ cung cấp đầy đủ thông tin về các di tích lịch sử, lễ hội, phong tục tập quán và những giá trị văn hoá đặc trưng của địa phương. Mục tiêu là giúp người dân địa phương và du khách dễ dàng tiếp cận các thông tin về di sản văn hoá, đồng thời tạo cầu nối giữa quá khứ và hiện tại, góp phần bảo tồn và phát huy những giá trị quý báu của tỉnh Trà Vinh.

Website sẽ được xây dựng bằng các công nghệ cơ bản như HTML, CSS, PHP và JavaScript, nhằm đảm bảo tính linh hoạt, hiệu suất và giao diện dễ sử dụng. Ngoài ra, các tính năng tìm kiếm và điều hướng thông minh sẽ được tích hợp để người dùng có thể dễ dàng tìm kiếm các di tích, lễ hội và các thông tin liên quan đến lịch sử và văn hoá Trà Vinh.

Mục tiêu của đề tài bao gồm:

1. Xây dựng website hoàn chỉnh với giao diện đẹp mắt, dễ sử dụng và phù hợp với nhu cầu của người dùng tại Việt Nam.

2. Cung cấp thông tin đầy đủ về các di sản lịch sử và văn hoá của Trà Vinh, giúp người dùng có thể tìm hiểu sâu về những giá trị này.

3. Tích hợp các tính năng tìm kiếm thông minh và hệ thống điều hướng dễ dàng, giúp người dùng nhanh chóng tiếp cận các thông tin về di tích lịch sử, lễ hội, và các giá trị văn hoá đặc trưng của tỉnh Trà Vinh.

# Nghiên cứu lý thuyết

## Quy trình phát triển website

### Lên kế hoạch và thu thập yêu cầu:

* Xác định mục tiêu của website.
* Phân tích đối tượng mục tiêu.
* Lập kế hoạch cho các chức năng và nội dung của website.
* Thu thập yêu cầu từ người dùng.

### Thiết kế giao diện:

* Tạo bản đồ trang web (sitemap) để xác định cấu trúc của website.
* Thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) của website.
* Tạo bản mẫu giao diện (wireframe) để phác thảo bố cục của website.

### Phát triển website:

* Lựa chọn công nghệ phù hợp để phát triển website, ví dụ như HTML, CSS, JavaScript, PHP, MySQL.
* Viết mã website theo các yêu cầu đã được xác định.
* Kiểm tra và sửa lỗi website.

### Thử nghiệm và triển khai:

* Thử nghiệm website trên các trình duyệt và thiết bị khác nhau.
* Sửa lỗi và hoàn thiện website.
* Triển khai website lên hosting.

### Bảo trì và cập nhật:

* Cập nhật nội dung website thường xuyên.
* Sửa lỗi website khi cần thiết.
* Cập nhật website để phù hợp với các công nghệ mới.

### Ngoài ra, quy trình phát triển website có thể bao gồm thêm các bước sau:

* Phân tích thông tin, tin tức chính thống từ các website khác.
* Tối ưu hóa công cụ tìm kiếm (SEO) cho website.
* Quảng bá website.

## Ngôn ngữ HTML

### Khái niệm về HTML

HTML là viết tắt của HyperText Markup Language: là ngôn ngữ cơ bản dùng để tạo ra các trang web. Nó cho phép chúng ta định dạng và tổ chức nội dung trên trang web bằng cách sử dụng các thẻ và các thuộc tính tương ứng. Nhờ vào HTML, chúng ta có thể hiển thị văn bản, hình ảnh, âm thanh, video và nhiều loại nội dung khác trên trình duyệt web [2].

### HTML cơ bản

Một tài liệu HTML phải có phần khai báo loại tài liệu bắt đầu bằng <!DOCTYPE html>.

- Phần chính của tài liệu HTML được đặt giữa thẻ <html> và </html>.

- Phần nội dung được hiển thị trên trình duyệt được đặt trong thẻ <body> và </body>.

- Ví dụ:

<!DOCTYPE html>

<html>

<body>

<h1>Heading</h1>

<p>My paragraph.</p>

</body>

</html>

\*Khai báo <!DOCTYPE>:

- Khai báo <!DOCTYPE> thể hiện loại tài liệu và giúp trình duyệt hiển thị trang web một cách chính xác.

- Nó chỉ được khai báo một lần, ở đầu trang (trước bất kỳ thẻ HTML nào).

- Khai báo <!DOCTYPE> không phân biệt chữ hoa chữ thường.

- Khai báo <!DOCTYPE> cho HTML5 là:

<!DOCTYPE html>

### Phần tử HTML

Một phần tử HTML được xác định bởi thẻ bắt đầu, một số nội dung và thẻ kết thúc.

Phần tử HTML là cặp thẻ bắt đầu và kết thúc để làm một nhiệm vụ nào đó:

<tên thẻ> Nội dung ở đây... </tên thẻ>

Ví dụ về một số thành phần HTML:

<h1>Animals</h1>

<p> Phần văn bản. </p>

\*Các phần tử HTML lồng nhau:

- Các phần tử HTML có thể được lồng vào nhau (điều này có nghĩa là các phần tử có thể chứa các phần tử khác).

- Tất cả các tài liệu HTML bao gồm các phần tử HTML lồng nhau.

- Ví dụ sau chứa bốn phần tử HTML ( <html>, <body>, <h1> và <p>):

<!DOCTYPE html>

<html>

<body>

<h1>Thẻ tiêu đề.</h1>

<p>Thẻ đoạn văn bản.</p>

</body>

</html>

Giải thích ví dụ:

- Phần tử <html> là phần tử gốc và nó xác định toàn bộ tài liệu HTML.

- Nó có thẻ bắt đầu <html>và thẻ kết thúc </html>.

- Sau đó, bên trong phần tử <html> có một phần tử <body>:

<body>

<h1>Thẻ tiêu đề.</h1>

<p>Thẻ đoạn văn bản.</p>

</body>

- Phần tử<body> xác định phần thân của tài liệu.

- Nó có thẻ bắt đầu <body>và thẻ kết thúc </body>.

Sau đó, bên trong phần tử <body> có hai phần tử khác: <h1>và <p>:

<h1>Thẻ tiêu đề</h1>

<p>Thẻ đoạn văn bản.</p>

- Phần tử <h1> xác định một tiêu đề.

- Nó có thẻ bắt đầu <h1>và thẻ kết thúc </h1>:

<h1>Thẻ tiêu đề</h1>

- Phần tử<p> xác định một đoạn văn.

- Nó có thẻ bắt đầu <p>và thẻ kết thúc </p>:

<p>Thẻ đoạn văn bản.</p>

- Không bao giờ được phép quên thẻ đóng vì nó sẽ gây ra lỗi nghiêm trọng!

- Các phần tử HTML trống là các phần tử không có nội dung và thẻ đóng.

- Ví dụ thẻ <br> xác định ngắt dòng và là phần tử trống không có thẻ đóng:

<p>Đây là thẻ đoạn văn bản <br> với thẻ ngắt dòng.</p>

- Thẻ HTML không phân biệt chữ hoa chữ thường: <P> và <p> là như nhau.

### Thuộc tính HTML

- Thuộc tính HTML cung cấp thông tin bổ sung về các phần tử HTML.

- Tất cả các phần tử HTML đều có thể có thuộc tính.

- Thuộc tính cung cấp thông tin bổ sung về các phần tử.

- Các thuộc tính luôn được chỉ định trong thẻ bắt đầu.

- Các thuộc tính thường có cặp tên/giá trị như: name="value".

\*Thuộc tính href:

- Thẻ <a>xác định một liên kết. Thuộc tính href chỉ định URL của trang web cần liên kết:

- Ví dụ:

<a href="https://ttsv.tvu.edu.vn"> Cổng thông tin sinh viên</a>

\*Thuộc tính src:

Thẻ <img>được sử dụng để nhúng hình ảnh vào trang HTML. Thuộc tính src chỉ định đường dẫn đến hình ảnh sẽ được hiển thị:

- Ví dụ:

<img src="Demo.jpg">

\*Thuộc tính chiều rộng và chiều cao:

- Thẻ <img> cũng có thể chứa các thuộc tính width và height giúp xác định chiều rộng và chiều cao của hình ảnh (tính bằng pixel):

- Ví dụ:

<img src="Demo.jpg" width="500" height="600">

\*Thuộc tính alt:

- Thuộc tính alt là bắt buộc cho thẻ <img> chỉ định văn bản thay thế cho hình ảnh nếu vì lý do nào đó mà hình ảnh không thể hiển thị. Điều này có thể do kết nối chậm hoặc lỗi thuộc tính src hoặc do người dùng sử dụng trình đọc màn hình.

- Ví dụ:

<img src="Demo.jpg" alt="Đây là bức ảnh!">

\*Thuộc tính style:

Thuộc tính style được sử dụng để thêm kiểu định dạng cho một phần tử, chẳng hạn như màu sắc, phông chữ, kích thước, v.v.

- Ví dụ:

<p style="color:red;">Đây là đoạn văn bản màu đỏ.</p>

\*Thuộc tính lang:

- Cần phải luôn luôn them thuộc tính lang vào bên trong thẻ <html> để khai báo ngôn ngữ của trang Web. Điều này nhằm hỗ trợ các công cụ tìm kiếm của trình duyệt.

- Ví dụ sau chỉ định tiếng Việt là ngôn ngữ:

<!DOCTYPE html>

<html lang="vi">

<body>

...

</body>

</html>

- Tiêu đề trong HTML là thẻ tiêu đề mà bạn muốn hiển thị trên trang web.

- Tiêu đề trong HTML được xác định bằng thẻ <h1> đến <h6>.

- <h1> giúp xác định tiêu đề quan trọng nhất và lớn nhất còn <h6> thì xác định tiêu đề ít quan trọng nhất và nhỏ nhất.

- Ví dụ:

<h1> Tiêu đề 1 </h1>

<h2> Tiêu đề 2 </h2>

<h3> Tiêu đề 3 </h3>

<h4> Tiêu đề 4 </h4>

<h5> Tiêu đề 5 </h5>

<h6> Tiêu đề 6 </h6>

\*Tiêu đề rất quan trọng:

- Công cụ tìm kiếm sử dụng các tiêu đề để tạo cấu trúc và nội dung trang web của bạn.

- Người dùng thường lướt qua một trang theo tiêu đề của nó. Điều quan trọng là sử dụng các tiêu đề để hiển thị cấu trúc tài liệu.

- <h1>nên sử dụng các tiêu đề cho các tiêu đề chính, tiếp theo là <h2>các tiêu đề ít quan trọng hơn <h3>, v.v.

### Đoạn văn bản trong HTML

Đoạn văn luôn bắt đầu trên một dòng mới và thường là ở dạng khối.

Phần tử <p> giúp xác định một đoạn văn bản.

Một đoạn văn luôn bắt đầu trên một dòng mới và trình duyệt sẽ tự động thêm một số khoảng trắng (lề) trước và sau một đoạn văn.

- Ví dụ:

<p>Đây là đoạn văn bản.</p>

<p> Đây là đoạn văn bản khác.</p>

\*Thẻ thanh ngang:

- Thẻ <hr> xác định điểm ngắt trong trang HTML và thường được hiển thị dưới dạng thanh ngang.

- Phần tử <hr> được sử dụng để phân tách nội dung (hoặc xác định thay đổi) trong trang HTML:

- Ví dụ:

<h1>Tiêu đề 1. </h1>

<p>Đoạn văn bản.</p>

<hr>

<h2>Tiêu đề 2.</h2>

<p>Đoạn văn bản.</p>

<hr>

- Lưu ý: Thẻ <hr> là thẻ trống, nghĩa là không có thẻ kết thúc.

\*Thẻ ngắt dòng:

- Phần tử <br> xác định ngắt dòng.

- Sử dụng thẻ <br> nếu bạn muốn ngắt dòng (một dòng mới) mà không bắt đầu một đoạn mới:

- Ví dụ:

<p>Đây là<br>một đoạn<br>văn bản.</p>

\*Phần tử <pre>:

- Phần tử <pre> xác định văn bản được định dạng sẵn.

- Văn bản bên trong một phần tử <pre> được hiển thị bằng phông chữ có chiều rộng cố định và nó giữ nguyên cả dấu cách và ngắt dòng:

- Ví dụ:

<pre>

Văn bản.

Sẽ hiển thị.

Giống như.

Thế này

</pre>

HTML chứa một số phần tử để xác định văn bản có ý nghĩa đặc biệt.

Các phần tử định dạng được thiết kế để hiển thị các loại văn bản đặc biệt:

- <b> Chữ in đậm.

- <strong> Văn bản quan trọng.

- <i> Văn bản in nghiêng.

- <em> Đoạn văn bản được nhấn mạnh.

- <mark> Văn bản được đánh dấu.

- <small> Văn bản nhỏ hơn.

- <del> Văn bản đã xóa.

- <ins> Đã chèn văn bản.

- <sub> Văn bản phía trên.

- <sup> Văn bản phía dưới.

### Liên kết trong HTML

- Thẻ <a> dùng để xác định một liên kết. Nó có cú pháp sau:

<a href="url">Văn bản liên kết</a>

Thuộc tính quan trọng nhất của phần tử <a> là thuộc tính href, cho biết đích đến của liên kết.

- Văn bản liên kết là phần mà người đọc sẽ nhìn thấy.

- Nhấp vào văn bản liên kết sẽ đưa người đọc đến địa chỉ URL được chỉ định.

- Ví dụ này cho thấy cách tạo liên kết đến cổng thông tin sinh viên:

<a href="https://www.ttsv.tvu.edu.vn/">Đến cổng thông tin sinh viên Đại học Trà Vinh</a>

- Theo mặc định, các liên kết sẽ xuất hiện như sau trong tất cả các trình duyệt:

- Một liên kết chưa được truy cập được gạch chân và có màu xanh lam.

- Một liên kết đã truy cập được gạch chân và có màu tím.

- Một liên kết đang hoạt động được gạch chân và có màu đỏ.

\*Thuộc tính đích:

- Theo mặc định, trang được liên kết sẽ được hiển thị trong cửa sổ trình duyệt hiện tại. Để thay đổi điều này, bạn phải chỉ định một mục tiêu khác cho liên kết.

- Thuộc tính target chỉ định nơi mở tài liệu được liên kết.

- Thuộc tính target có thể có một trong các giá trị sau:

\_self- Mặc định. Mở tài liệu trong cùng cửa sổ/tab khi tài liệu được nhấp vào.

\_blank- Mở tài liệu trong cửa sổ hoặc tab mới.

\_parent- Mở tài liệu trong khung cha.

\_top- Mở tài liệu ở chế độ toàn bộ cửa sổ.

- Ví dụ:

Sử dụng target="\_blank" để mở tài liệu được liên kết trong cửa sổ hoặc tab trình duyệt mới:

<a href="https:// https://ttsv.tvu.edu.vn//" target="\_blank">Đến cổng thông tin sinh viên Đại học Trà Vinh</a>

\*URL tuyệt đối so với URL tương đối:

- Cả hai ví dụ trên đều sử dụng URL tuyệt đối (địa chỉ web đầy đủ) trong thuộc tính href.

- Liên kết cục bộ (liên kết đến một trang trong cùng một trang web) được chỉ định bằng URL tương đối (không có phần "https://www"):

- Ví dụ:

<h2>URL tuyệt đối:</h2>

<p><a href="https://www.youtube.com /">YouTube</a></p>

<p><a href="https://www.google.com/">Google</a></p>

<h2>URL tương đối:</h2>

<p><a href="images.asp">Ảnh HTML</a></p>

<p><a href="/css/Demo.asp">CSS cơ bản</a></p>

\*Sử dụng hình ảnh làm liên kết:

- Để sử dụng hình ảnh làm liên kết, chỉ cần đặt thẻ <img> bên trong thẻ <a>:

- Ví dụ:

<a href="Demo.asp">

<img src="Meme.gif" alt="funny" style="width:42px;height:42px;">

</a>

Liên kết HTML có thể được sử dụng để tạo dấu trang để người đọc có thể chuyển đến các phần cụ thể của trang web.

\*Tạo dấu trang trong HTML:

- Dấu trang có thể hữu ích nếu một trang web rất dài.

- Để tạo dấu trang - trước tiên hãy tạo dấu trang rồi thêm liên kết vào dấu trang đó.

- Khi nhấp vào liên kết, trang sẽ cuộn xuống hoặc cuộn lên vị trí có dấu trang.

- Ví dụ:

- Đầu tiên, sử dụng thuộc tính id để tạo dấu trang:

<h2 id="demo">Animals</h2>

- Sau đó, thêm liên kết vào dấu trang ("Chuyển tới Chương 4"), từ trong cùng một trang:

<a href="#demo">Chuyển đến Animals</a>

- Bạn cũng có thể thêm liên kết vào dấu trang trên một trang khác:

<a href="demo.html#demo"> Chuyển tới Animals</a>

### Hình ảnh trong HTML

Hình ảnh có thể cải thiện thiết kế và sự xuất hiện của một trang web.

\*Cú pháp thêm hình ảnh vào HTML:

- Thẻ <img> được sử dụng để nhúng hình ảnh vào trang web.

- Hình ảnh không được chèn vào trang web về mặt kỹ thuật; hình ảnh được liên kết đến các trang web. Thẻ <img> tạo một không gian lưu giữ cho hình ảnh được tham chiếu.

- Thẻ <img> là thẻ trống, chỉ chứa các thuộc tính và không có thẻ đóng.

- Thẻ <img>có hai thuộc tính bắt buộc:

src - Chỉ định đường dẫn đến hình ảnh.

alt - Chỉ định văn bản thay thế cho hình ảnh.

- Cú pháp:

<img src="url" alt="vanbanthaythechoanh">

\*Thuộc tính src:

- Thuộc tính src là bắt buộc chỉ định đường dẫn (URL) tới hình ảnh.

- Lưu ý: Khi một trang web tải, tại thời điểm đó, trình duyệt sẽ lấy hình ảnh từ máy chủ web và chèn nó vào trang. Do đó, hãy đảm bảo rằng hình ảnh thực sự ở cùng một vị trí so với trang web, nếu không khách truy cập của bạn sẽ nhận được biểu tượng liên kết bị hỏng. Biểu tượng liên kết bị hỏng và alt văn bản được hiển thị nếu trình duyệt không tìm thấy hình ảnh.

- Ví dụ:

<img src="Animals\_Dog.jpg" alt="Hình ảnh chú chó">

\*Hình ảnh trong thư mục khác:

- Nếu bạn có hình ảnh của mình trong thư mục con, bạn phải đưa tên thư mục vào thuộc tính src:

- Ví dụ:

<img src="/images/Meme.gif" alt="Icon funny" style="width:128px;height:128px;">

\*Hình ảnh trên Máy chủ/Trang web khác:

- Một số trang web trỏ đến một hình ảnh trên một máy chủ khác.

- Để trỏ đến hình ảnh trên máy chủ khác, bạn phải chỉ định URL tuyệt đối (đầy đủ) trong thuộc tính src:

- Ví dụ:

<img src="https://www.w3schools.com/images/w3schools\_green.jpg" alt="W3Schools.com">

### Bảng trong HTML:

Bảng HTML cho phép các nhà phát triển web sắp xếp dữ liệu thành hàng và cột.

\*Xác định bảng HTML:

- Một bảng trong HTML bao gồm các ô bảng bên trong các hàng và cột.

- Ví dụ một bảng HTML đơn giản:

<table>

<tr>

<th>Họ và Tên</th>

<th>MSSV</th>

<th>Đoàn khoa</th>

</tr>

<tr>

<td>Nguyễn Văn A</td>

<td>Da2xxxx</td>

<td>Kỹ thuật và Công nghệ</td>

</tr>

</table>

\*Ô bảng:

Mỗi ô của bảng được xác định bởi thẻ mở <td>và một thẻ đóng</td>.

Mọi thứ ở giữa thẻ <td>và thẻ </td>là nội dung của ô bảng.

- Ví dụ:

<table>

<tr>

<td>Windown</td>

<td>Linux</td>

<td>Unbutu</td>

</tr>

</table>

\*Hàng bảng:

Mỗi hàng của bảng bắt đầu bằng thẻ mở <tr>và kết thúc bằng một thẻ đóng </tr>.

- Ví dụ:

<table>

<tr>

<td>Alex</td>

<td>John</td>

<td>David</td>

</tr>

<tr>

<td>16</td>

<td>14</td>

<td>10</td>

</tr>

</table>

Bạn có thể có bao nhiêu hàng tùy thích trong một bảng; chỉ cần đảm bảo rằng số lượng ô trong mỗi hàng giống nhau.

\*Tiêu đề bảng:

- Đôi khi bạn muốn các ô của mình là ô tiêu đề bảng. Trong những trường hợp đó, hãy sử dụng thẻ <th> thay vì thẻ <td>:

- Ví dụ đặt hàng đầu tiên là ô tiêu đề bảng:

<table>

<tr>

<th>Tiêu đề 1</th>

<th>Tiêu đề 2</th>

<th>Tiêu đề 3</th>

</tr>

<tr>

<td>Alex</td>

<td>John</td>

<td>David</td>

</tr>

<tr>

<td>16</td>

<td>14</td>

<td>10</td>

</tr>

</table>

- Theo mặc định, các văn bản trong <th> được in đậm và căn giữa nhưng bạn có thể thay đổi điều đó bằng CSS.

\*Bảng chú thích:

- Bạn có thể thêm chú thích làm tiêu đề cho toàn bộ bảng.

- Để thêm chú thích vào bảng, hãy sử dụng thẻ <caption>:

<table style="width:100%">

<caption>Monthly savings</caption>

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$50</td>

</tr>

</table>

\*Colspan:

- Để tạo một ô trải dài trên nhiều cột, hãy sử dụng thuộc tính colspan:

- Ví dụ:

<table>

<tr>

<th colspan="2">Họ và Tên</th>

<th>Tuổi</th>

</tr>

<tr>

<td>Jully</td>

<td>43</td>

</tr>

<tr>

<td>Eve</td>

<td>57</td>

</tr>

</table>

\*Rowspan:

- Để tạo một ô trải dài trên nhiều hàng, hãy sử dụng thuộc tính rowspan:

- Ví dụ:

<table>

<tr>

<th>Name</th>

<td>Jill</td>

</tr>

<tr>

<th rowspan="2">Phone</th>

<td>1234-xxxx</td>

</tr>

</table>

\*Nhóm bảng trong HTML:

- Nếu bạn muốn tạo kiểu cho hai cột đầu tiên của bảng, hãy sử dụng các phần tử <colgroup> và <col> .

- Phần tử <colgroup> này nên được sử dụng làm nơi chứa các thông số kỹ thuật của cột.

- Mỗi nhóm được chỉ định bằng một phần tử <col>.

- Thuộc tính span chỉ định số lượng cột có kiểu.

- Thuộc tính style chỉ định kiểu cho các cột.

- Ví dụ:

<table>

<colgroup>

<col span="2" style="background-color: #ff0000">

</colgroup>

<tr>

<th>MON</th>

<th>TUE</th>

<th>WED</th>

<th>THU</th>

### Danh sách trong HTML

Danh sách HTML cho phép các nhà phát triển web nhóm một tập hợp các mục liên quan vào danh sách.

\*Danh sách HTML không có thứ tự:

- Danh sách không có thứ tự bắt đầu bằng thẻ <ul>. Mỗi mục danh sách bắt đầu bằng thẻ <li>.

- Các mục trong danh sách sẽ được đánh dấu bằng dấu đầu dòng (vòng tròn nhỏ màu đen) theo mặc định.

- Ví dụ:

<ul>

<li>Coca-cola</li>

<li>Tea</li>

<li>Milk</li>

</ul>

\*Danh sách HTML có thứ tự:

- Một danh sách có thứ tự bắt đầu bằng thẻ <ol>. Mỗi mục danh sách bắt đầu bằng thẻ <li>.

- Các mục trong danh sách sẽ được đánh dấu bằng số theo mặc định.

- Ví dụ:

<ol>

<li>Coca-cola</li>

<li>Tea</li>

<li>Milk</li>

</ol>

\*Danh sách mô tả HTML:

- HTML cũng hỗ trợ danh sách mô tả.

- Danh sách mô tả là danh sách các thuật ngữ, kèm theo mô tả cho từng thuật ngữ.

- Thẻ <dl> xác định danh sách mô tả, thẻ <dt> xác định thuật ngữ (tên) và thẻ <dd> mô tả từng thuật ngữ:

- Ví dụ:

<dl>

<dt>Coca-cola</dt>

<dd>- Mô tả</dd>

<dt>Milk</dt>

<dd>- white</dd>

</dl>

\*Danh sách HTML lồng nhau:

- Danh sách có thể được lồng vào nhau (danh sách bên trong danh sách)

- Ví dụ:

<ul>

<li>Coca-cola</li>

<li>Tea

<ul>

<li>Milk tea</li>

<li>Green tea</li>

</ul>

</li>

<li>Milk</li>

</ul>

\*Danh sách HTML có thứ tự - Thuộc tính loại:

- Thuộc tính type của thẻ <ol>, xác định loại điểm đánh dấu mục danh sách:

Loại Mô tả

type="1" Các mục trong danh sách sẽ được đánh số bằng số (mặc định).

type="A" Các mục trong danh sách sẽ được đánh số bằng chữ cái in hoa.

type="a" Các mục trong danh sách sẽ được đánh số bằng chữ cái thường.

type="I" Các mục trong danh sách sẽ được đánh số bằng số La Mã in hoa.

type="i" Các mục trong danh sách sẽ được đánh số bằng số La Mã thường.

\*Thay đổi thứ tự danh sách:

- Theo mặc định, danh sách có thứ tự sẽ bắt đầu đếm từ 1. Nếu muốn bắt đầu đếm từ một số được chỉ định, bạn có thể sử dụng thuộc tính start:

- Ví dụ:

<ol start="50">

<li>Coca-cola</li>

<li>Tea</li>

<li>Milk</li>

</ol>

### Khung nội tuyến trong HTML

Khung nội tuyến HTML được sử dụng để hiển thị một trang web trong một trang web.

- Thẻ <iframe> chỉ định khung nội tuyến.

- Khung nội tuyến được sử dụng để nhúng tài liệu khác trong tài liệu HTML hiện tại.

- Cú pháp:

<iframe src="url" title="description"></iframe>

\*Đặt chiều cao và chiều rộng:

- Sử dụng các thuộc tính height và width để chỉ định kích thước của iframe.

- Chiều cao và chiều rộng được chỉ định bằng pixel theo mặc định

- Ví dụ:

<iframe src="demo.htm" height="250" width="350" title="Iframe Example"></iframe>

\*Mục tiêu cho một liên kết:

- Iframe có thể được sử dụng làm khung đích cho một liên kết.

- Thuộc tính target của liên kết phải tham chiếu đến thuộc tính name của iframe

- Ví dụ:

<iframe src="demo.htm" name="iframe\_a" title="Iframe Example"></iframe>

<p><a href="https://www.w3schools.com" target="iframe\_a">W3Schools.com</a></p>

### Đường dẫn tệp trong HTML

Đường dẫn tệp mô tả vị trí của tệp trong cấu trúc thư mục của trang web.

- Ví dụ về đường dẫn tệp:

Đường dẫn Miêu tả

<img src="picture.jpg"> Tệp "picture.jpg" nằm trong cùng thư mục với trang hiện tại

<img src="images/picture.jpg"> Tệp "picture.jpg" nằm trong thư mục images trong thư mục hiện tại

<img src="/images/picture.jpg"> Tệp "picture.jpg" nằm trong thư mục images ở thư mục gốc của trang web hiện tại

<img src="../picture.jpg"> Tệp "picture.jpg" nằm trong thư mục cao hơn một cấp so với thư mục hiện tại

\*Đường dẫn tệp tuyệt đối:

- Đường dẫn tệp tuyệt đối là URL đầy đủ của tệp:

Ví dụ

<img src="https://www.w3schools.com/images/picture.jpg" alt="Mountain">

\*Đường dẫn tệp tương đối:

- Đường dẫn tệp tương đối trỏ đến tệp tương ứng với trang hiện tại.

- Trong ví dụ sau, đường dẫn tệp trỏ đến một tệp trong thư mục images nằm ở thư mục gốc của trang web hiện tại:

- Ví dụ:

<img src="/images/picture.jpg" alt="Mountain">

### Biểu mẫu trong HTML

Một biểu mẫu HTML được sử dụng để thu thập thông tin đầu vào của người dùng.

- Đầu vào của người dùng thường được gửi đến máy chủ để xử lý.

\*Phần tử <form>:

- Phần tử <form> được sử dụng để tạo biểu mẫu HTML cho người dùng nhập:

<form>

form elements

</form>

- Phần tử <form> là nơi chứa các loại phần tử đầu vào khác nhau, chẳng hạn như: trường văn bản, hộp kiểm, nút radio, nút gửi, v.v.

\*Phần tử <input>:

- Phần tử HTML <input> là phần tử biểu mẫu được sử dụng nhiều nhất.

- Một phần tử <input> có thể được hiển thị theo nhiều cách, tùy thuộc vào thuộc tính type.

\*Trường văn bản:

- <input type="text"> giúp xác định trường nhập một dòng để nhập văn bản.

- Ví dụ:

- Một biểu mẫu có các trường nhập văn bản:

<form>

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname">

</form>

\*Phần tử <label>:

- Lưu ý việc sử dụng phần tử <label> trong ví dụ trên.

- Thẻ <label>xác định nhãn cho nhiều thành phần biểu mẫu.

- Phần tử <label> này hữu ích cho người dùng trình đọc màn hình vì trình đọc màn hình sẽ đọc to nhãn khi người dùng tập trung vào phần tử đầu vào.

- Phần tử <label> này cũng giúp người dùng gặp khó khăn khi nhấp vào các vùng rất nhỏ (chẳng hạn như nút radio hoặc hộp kiểm) - bởi vì khi người dùng nhấp vào văn bản trong phần <label>tử, nó sẽ chuyển đổi nút radio/hộp kiểm.

- Thuộc tính for của thẻ <label> phải giống thuộc tính id của phần tử <input> để liên kết chúng lại với nhau.

\*Nút chọn:

- <input type="radio"> xác định một nút radio.

- Nút radio cho phép người dùng chọn MỘT trong số các lựa chọn có giới hạn.

- Ví dụ:

Một biểu mẫu có các nút radio:

<p>Choose your favorite Web language:</p>

<form>

<input type="radio" id="html" name="fav\_language" value="HTML">

<label for="html">HTML</label><br>

<input type="radio" id="css" name="fav\_language" value="CSS">

<label for="css">CSS</label><br>

<input type="radio" id="javascript" name="fav\_language" value="JavaScript">

<label for="javascript">JavaScript</label>

</form>

\*Hộp kiểm:

- <input type="checkbox"> xác định một hộp kiểm .

- Các hộp kiểm cho phép người dùng chọn các tùy chọn KHÔNG hoặc NHIỀU tùy chọn trong một số lựa chọn giới hạn.

- Ví dụ:

Một biểu mẫu có các hộp kiểm:

<form>

<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">

<label for="vehicle1"> I have a bike</label><br>

<input type="checkbox" id="vehicle2" name="vehicle2" value="Car">

<label for="vehicle2"> I have a car</label><br>

<input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">

<label for="vehicle3"> I have a boat</label>

</form>

\*Nút Gửi:

- <input type="submit"> xác định nút để gửi dữ liệu biểu mẫu tới trình xử lý biểu mẫu.

- Trình xử lý biểu mẫu thường là một tệp trên máy chủ có tập lệnh để xử lý dữ liệu đầu vào.

- Trình xử lý biểu mẫu được chỉ định trong thuộc tính action của biểu mẫu.

- Ví dụ:

Một biểu mẫu có nút gửi:

<form action="/action\_page.php">

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

<input type="submit" value="Submit">

</form>

\*Thuộc tính tên cho <input>:

- Lưu ý rằng mỗi trường đầu vào phải có một thuộc tính name để được gửi.

- Nếu thuộc tính name bị bỏ qua, giá trị của trường đầu vào sẽ không được gửi đi.

- Ví dụ:

Ví dụ này sẽ không gửi giá trị của trường nhập "Tên":

<form action="/action\_page.php">

<label for="fname">First name:</label><br>

<input type="text" id="fname" value="John"><br><br>

<input type="submit" value="Submit">

</form>

\*Thuộc tính action:

- Thuộc tính action xác định hành động sẽ được thực hiện khi biểu mẫu được gửi.

- Thông thường, dữ liệu biểu mẫu được gửi đến một tệp trên máy chủ khi người dùng nhấp vào nút gửi.

- Trong ví dụ bên dưới, dữ liệu biểu mẫu được gửi đến tệp có tên "action\_page.php".

- Tệp này chứa tập lệnh phía máy chủ xử lý dữ liệu biểu mẫu:

- Ví dụ

Khi gửi, hãy gửi dữ liệu biểu mẫu tới "action\_page.php":

<form action="/action\_page.php">

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

<input type="submit" value="Submit">

</form>

\*Thuộc tính method:

- Thuộc tính method chỉ định phương thức HTTP sẽ được sử dụng khi gửi dữ liệu biểu mẫu.

- Dữ liệu biểu mẫu có thể được gửi dưới dạng biến URL (với method="get") hoặc dưới dạng giao dịch bài đăng HTTP (với method="post").

- Phương thức HTTP mặc định khi gửi dữ liệu biểu mẫu là GET.

- Ví dụ:

Ví dụ này sử dụng phương thức GET khi gửi dữ liệu biểu mẫu:

<form action="/action\_page.php" method="get">

\*Các phần tử biểu mẫu HTML:

Phần tử HTML <form>có thể chứa một hoặc nhiều phần tử biểu mẫu sau:

<input><label><select><textarea><button><fieldset><legend><datalist><output>

<option><optgroup>

\*Phần tử <select>:

- Phần tử <select> giúp xác định danh sách thả xuống:

- Ví dụ:

<label for="cars">Choose a car:</label>

<select id="cars" name="cars">

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="fiat">Fiat</option>

<option value="audi">Audi</option>

</select>

- Phần tử <option> xác định một tùy chọn có thể được chọn.

- Theo mặc định, mục đầu tiên trong danh sách thả xuống được chọn.

- Để xác định tùy chọn được chọn trước, hãy thêm thuộc tính selected vào <option>:

Ví dụ

<option value="fiat" selected>Fiat</option>

\*Phần tử <textarea>:

- Phần tử <textarea> xác định trường nhập nhiều dòng (vùng văn bản):

- Ví dụ:

<textarea name="message" rows="10" cols="30">

The cat was playing in the garden.

</textarea>

- Thuộc tính rows chỉ định số dòng hiển thị trong vùng văn bản.

- Thuộc tính cols chỉ định chiều rộng hiển thị của vùng văn bản.

\*Các phần tử <fieldset> và <legend>:

- Phần tử <fieldset> được sử dụng để nhóm dữ liệu liên quan trong một biểu mẫu.

- Phần tử <legend> xác định chú thích cho phần tử <fieldset>.

Ví dụ

<form action="/action\_page.php">

<fieldset>

<legend>Personalia:</legend>

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

<input type="submit" value="Submit">

</fieldset>

</form>

## Bảng định kiểu CSS

### CSS nội tuyến (Inline CSS)

CSS nội tuyến được viết trực tiếp trong thẻ HTML, sử dụng thuộc tính style. Loại CSS này được sử dụng để định dạng các phần tử HTML cụ thể. Ví dụ:

<p style="color: red; font-size: 20px;">Đây là một đoạn văn bản được định dạng bằng CSS nội tuyến.</p> [2].

### CSS nội bộ (Internal CSS)

CSS nội bộ được viết trong thẻ <style> nằm trong phần <head> của tài liệu HTML. Loại CSS này được sử dụng để định dạng nhiều phần tử HTML trong cùng một trang web.

Ví dụ:

<head>

<style>

p {

color: red;

font-size: 20px;

}

</style>

</head>

### CSS bên ngoài (External CSS)

CSS bên ngoài được viết trong một tệp riêng có phần mở rộng .css. Tệp này được liên kết với tài liệu HTML bằng thẻ <link> trong phần <head> của tài liệu HTML. Loại CSS này được sử dụng để định dạng nhiều trang web trong cùng một dự án. Ví dụ:

<head>

<link rel="stylesheet" href="style.css">

</head>

## Ngôn ngữ Javascript

### Giới thiệu về Javascript

JavaScript là ngôn ngữ lập trình bổ sung tính tương tác và hành vi tùy chỉnh cho các trang web của chúng tôi. Nó là ngôn ngữ kịch bản phía máy khách, có nghĩa là nó chạy trên máy của người dùng chứ không phải trên máy chủ, như các ngôn ngữ lập trình web khác như PHP và Ruby.

Điều đó có nghĩa là JavaScript, JavaScript cũng được biết đến như một ngôn ngữ lập trình động và được gõ lỏng lẻo .JavaScript là một ngôn ngữ kịch bản nhẹ nhưng cực kỳ mạnh mẽ. Chúng ta thường gặp nó nhất thông qua trình duyệt của mình, nhưng JavaScript đã xâm nhập vào mọi thứ, từ ứng dụng gốc, tệp PDF đến sách điện tử.

Thông thường chúng ta sẽ gặp JavaScript như một cách để thêm tính tương tác vào một trang.

Tất cả các thành phần, thuộc tính và văn bản trên một trang web đều có thể được truy cập bằng các tập lệnh sử dụng DOM.

JavaScript cho phép bạn tạo các giao diện có độ phản hồi cao để cải thiện trải nghiệm người dùng và cung cấp chức năng động mà không cần đợi máy chủ tải trang mới [1].

### Thêm Javascript vào trang

Giống như CSS, bạn có thể nhúng tập lệnh ngay trong tài liệu hoặc giữ nó ở tệp bên ngoài và liên kết nó với trang. Cả hai phương pháp đều sử dụng phần tử script .

\*Tập lệnh nhúng

-Để nhúng tập lệnh vào một trang, chỉ cần thêm mã làm nội dung của tập lệnh

**\*Tập lệnh ngoài**

***Nhược điểm:***

- Phương thức khác sử dụng thuộc tính src để trỏ đến tệp tập lệnh (có phần mở rộng .js hậu tố) theo URL của nó. Trong trường hợp này, phần tử script không có nội dung:

<script src="my\_script.js"></script>

***Ưu điểm:***

- Tập lệnh bên ngoài là bạn có thể áp dụng cùng một tập lệnh cho nhiều trang (cùng lợi ích mà biểu định kiểu bên ngoài mang lại). Tất nhiên, nhược điểm là mỗi tập lệnh bên ngoài yêu cầu một yêu cầu HTTP bổ sung của máy chủ, điều này làm chậm hiệu suất.

**\* Vị trí tập lệnh:**

- Phần tử tập lệnh có thể xuất hiện ở bất kỳ đâu trong tài liệu, nhưng vị trí phổ biến nhất dành cho tập lệnh là ở phần đầu của tài liệu và ở cuối phần nội dung. Bạn không nên rải chúng khắp tài liệu vì chúng sẽ khó tìm và bảo trì.

- Đối với hầu hết các tập lệnh, phần cuối của tài liệu, ngay trước thẻ </body> , là vị trí ưu tiên vì trình duyệt sẽ hoàn tất việc phân tích tài liệu và cấu trúc DOM của nó:

### Javascript cơ bản

Có một số quy tắc cú pháp phổ biến áp dụng cho tất cả JavaScript. Điều quan trọng cần biết là JavaScript phân biệt chữ hoa chữ thường. Một biến có tên myVariable, một biến có tên myvariable và một biến có tên MYVariable sẽ được coi là ba đối tượng khác nhau

**Statements (Các câu lệnh)**

- Một câu lệnh là một lệnh cho trình duyệt biết phải làm gì. Đây là một câu lệnh đơn giản giúp trình duyệt hiển thị cảnh báo với cụm từ “Thank you”:

alert(“Thanks you.”);

- Dấu chấm phẩy ở cuối câu lệnh cho JavaScript biết rằng đó là phần cuối của lệnh, giống như dấu chấm kết thúc một câu.

**Comments (Bình luận)**

- Có hai phương pháp sử dụng chú thích. Đối với các chú thích một dòng, hãy sử dụng hai ký tự gạch chéo **(//)** ở đầu dòng

**//** Đây là chú thích một dòng.

- Mọi thứ trong ký tự **/\* \*/** đều bị trình duyệt bỏ qua. Bạn có thể sử dụng cú pháp này để **“nhận xét”** các ghi chú và thậm chí cả các đoạn tập lệnh khi khắc phục sự cố.

**/\***

Đây là chú thích nhiều dòng.

\*/

### Variables (Biến)

- Một biến giống như một thùng chứa thông tin. Bạn đặt tên cho nó rồi gán cho nó một giá trị, giá trị này có thể là số, chuỗi văn bản, một phần tử trong DOM hoặc một hàm—thực ra là bất kỳ thứ gì. Điều này cho chúng ta một cách thuận tiện để tham chiếu giá trị đó sau này theo tên. Bản thân giá trị có thể được sửa đổi và gán lại theo bất kỳ cách nào mà logic của tập lệnh của chúng tôi yêu cầu.

- Khai báo sau đây tạo một biến có tên foo và gán cho nó giá trị 5:

var foo = 5;

- Một số quy tắc đặt tên biến:

• Nó phải bắt đầu bằng một chữ cái hoặc dấu gạch dưới.

• Nó có thể chứa các chữ cái, chữ số và dấu gạch dưới dưới bất kỳ hình thức kết hợp nào.

• Nó có thể không chứa khoảng trắng ký tự. Ngoài ra, hãy sử dụng dấu gạch dưới thay cho dấu cách hoặc đóng khoảng trắng và sử dụng kiểu chữ lạc đà thay thế (ví dụ: my\_variable hoặc myVariable).

• Nó không được chứa các ký tự đặc biệt (ví dụ: ! /\ + \* =).

### Data types (Loại dữ liệu)

**Undefined (Không xác định)**

**-** Loại dữ liệu đơn giản nhất trong số này có thể không được xác định. Nếu chúng ta khai báo một biến bằng cách đặt tên nhưng không có giá trị thì biến đó chứa giá trị không xác định.

Var foo;

Alert(foo);

**Null (Vô giá trị)**

- Tương tự như không xác định, việc gán một biến null (một lần nữa, phân biệt chữ hoa chữ thường)

Var foo = null;

Alert(foo);

**Numbers (số)**

- Biến này hoạt động giống như chính số đó, vì vậy bạn có thể thực hiện các thao tác với nó giống như bạn sẽ làm với bất kỳ số nào khác bằng cách sử dụng ký hiệu toán học cổ điển: +, -, \* và / tương ứng cho cộng, trừ, nhân và chia. Trong ví dụ này, chúng ta sử dụng dấu cộng (+) để thêm foo vào chính nó (foo + foo).

Var foo = 5;

Alert(foo+foo); //10

**Strings (dây)**

- Một loại dữ liệu khác có thể được lưu vào biến là chuỗi, về cơ bản là một dòng văn bản. Việc đặt các ký tự trong một tập hợp dấu ngoặc đơn hoặc dấu ngoặc kép cho biết đó là một chuỗi.

Var foo = “number”;

Alert(foo); //number

Trước đó chúng ta đã thấy dấu cộng (+) dùng để cộng số. Khi dấu cộng được sử dụng với các chuỗi, nó sẽ nối các chuỗi lại với nhau (gọi là nối) thành một chuỗi dài:

Var foo = “1”;

Alert(foo+foo); //11

**Boolean**

- Chúng ta cũng có thể gán cho một biến một giá trị đúng hoặc sai. Đây được gọi là giá trị Boolean và nó là cơ sở cho tất cả các loại logic nâng cao. Các giá trị Boolean sử dụng các từ khóa đúng và sai được tích hợp trong JavaScript, vì vậy dấu ngoặc kép là không cần thiết.

Var foo = true;

**Arrays (Mảng)**

- Mảng là một nhóm gồm nhiều giá trị (được gọi là thành viên) có thể được gán cho một biến duy nhất. Các giá trị trong một mảng được cho là được lập chỉ mục, nghĩa là bạn có thể tham chiếu chúng theo số theo thứ tự chúng xuất hiện trong danh sách.

- Bây giờ, bất cứ khi nào bạn cần truy cập bất kỳ giá trị nào trong số đó, bạn có thể lấy chúng từ một mảng foo bằng cách tham chiếu số chỉ mục của chúng:

Alert(foo[0]);

Alert(foo[1]);

Alert(foo[2]);

### Comparison Operators (Toán tử so sánh)

- Có một tập hợp các ký tự đặc biệt được gọi là toán tử so sánh để đánh giá và so sánh các giá trị theo nhiều cách khác nhau:

Toán tử Mô tả

== bằng

=== bằng giá trị và loại dữ liệu

!= không bằng

!== không bằng giá trị hoặc loại dữ liệu

> lớn hơn

< nhỏ hơn

>= lớn hơn hoặc bằng

<= nhỏ hơn hoặc bằng

? toán tử ba ngôi

- Khi chúng ta so sánh hai giá trị, **JavaScript** sẽ đánh giá câu lệnh và trả về cho chúng ta giá trị Boolean tùy thuộc vào câu lệnh đó là đúng hay sai.

Alert(5==5); //true

Alert(5!=5); //false

Alert(5<5);//false

### Mathematical operators (Toán tử toán học)

- Loại toán tử khác là toán tử toán học, thực hiện các hàm toán học trên các giá trị số (và tất nhiên, các biến chứa giá trị số). Chúng ta đã đề cập sơ qua về các toán tử đơn giản để cộng (+), trừ (-), nhân (\*) và chia (/). Ngoài ra còn có một số phím tắt hữu ích bạn nên biết:

### if/else statements (câu lệnh if/else)

- Các câu lệnh if/else là cách chúng ta khiến JavaScript tự hỏi một câu hỏi đúng/sai. Chúng ít nhiều là nền tảng cho tất cả logic nâng cao có thể được viết bằng JavaScript và chúng cũng đơn giản như việc lập trình. Trên thực tế, chúng gần như được viết bằng tiếng Anh đơn giản. Cấu trúc của câu lệnh điều kiện như sau:

If(true){

// lệnh

}

- Chúng ta cũng có thể kiểm tra rõ ràng xem có điều gì đó sai hay không bằng cách sử dụng toán tử so sánh != , có nghĩa là “không bằng”.

Var two = 2;

If(1 == 2){

two = 1;

}

### Loops (vòng lặp)

- Vòng lặp cho phép chúng ta dễ dàng xử lý các tập dữ liệu khổng lồ.

- Có một số cách để viết vòng lặp, nhưng phương thức for là một trong những cách phổ biến nhất. Cấu trúc cơ bản của vòng lặp for như sau:

- Các nhà phát triển sử dụng vòng lặp để thực hiện một số tác vụ, chẳng hạn như sau:

• Lặp qua danh sách các phần tử trên trang và kiểm tra giá trị của từng phần tử, áp dụng một kiểu cho từng phần tử hoặc thêm/xóa/thay đổi thuộc tính trên mỗi phần tử.

• Tạo một mảng các phần tử mới trong mảng ban đầu có giá trị nhất định.

### Functions (Chức năng)

- Các hàm cho phép sử dụng lại mã bất cứ khi nào nó được tham chiếu, do đó bạn không cần phải viết đi viết lại.

- Có hai loại hàm: những loại hàm “có sẵn” (hàm JavaScript gốc) và những loại do bạn tự tạo ra (hàm tùy chỉnh). Chúng ta hãy nhìn vào từng cái.

**Hàm gốc**

- Hàng trăm hàm được xác định trước được tích hợp vào JavaScript, bao gồm:

alert(), confirm(), and prompt()

- Các chức năng này kích hoạt hộp thoại cấp trình duyệt

Date()

- Trả về ngày và giờ hiện tại.

parseInt("123")

- Hàm này, trong số những thứ khác, sẽ lấy kiểu dữ liệu chuỗi chứa số và biến nó thành kiểu dữ liệu số. Chuỗi được truyền cho hàm dưới dạng đối số.

setTimeout(functionName, 5000)

- Thực thi một chức năng sau một khoảng thời gian trì hoãn. Hàm được chỉ định trong đối số đầu tiên và độ trễ được chỉ định bằng mili giây trong đối số thứ hai (trong ví dụ: 5.000 mili giây, tương đương 5 giây).

**Chức năng tùy chỉnh**

- Để tạo một hàm tùy chỉnh, chúng ta nhập từ khóa hàm , theo sau là tên của hàm, tiếp theo là mở và đóng dấu ngoặc đơn, sau đó là mở và đóng dấu ngoặc nhọn:

**Arguments (Tranh Luận)**

- Việc có một hàm thực thi cùng một mã trong toàn bộ tập lệnh của bạn dường như không hữu ích lắm. Chúng ta có thể “truyền đối số” (cung cấp dữ liệu) cho các hàm gốc và hàm tùy chỉnh để áp dụng logic của hàm cho các tập hợp dữ liệu khác nhau tại các thời điểm khác nhau. Để giữ một vị trí cho các đối số, hãy tạo một tên có thể thay đổi (hoặc một chuỗi tên được phân tách bằng dấu phẩy) trong dấu ngoặc đơn sau tên của hàm tại thời điểm hàm được xác định.

**Trả về một giá trị**

- Việc sử dụng một hàm để tính toán một cái gì đó rồi trả lại cho bạn một giá trị mà bạn có thể sử dụng ở nơi khác trong tập lệnh của mình là điều khá phổ biến. Chúng ta có thể thực hiện điều này bằng cách sử dụng những gì chúng ta biết hiện nay, thông qua việc áp dụng thông minh các biến số, nhưng có một cách dễ dàng hơn nhiều.

### Variable Scope and the var Keyword (Phạm vi biến và từ khóa var)

- Khái niệm về tính sẵn có của biến này được gọi là phạm vi của nó. Một biến có thể được bất kỳ tập lệnh nào trên trang của bạn sử dụng đều có phạm vi toàn cầu và một biến chỉ có sẵn trong hàm gốc của nó là có phạm vi cục bộ.

- Các biến JavaScript sử dụng các hàm để quản lý phạm vi của chúng. Nếu một biến được xác định bên ngoài một hàm, biến đó sẽ có phạm vi toàn cầu và có sẵn cho tất cả các tập lệnh. Khi bạn xác định một biến trong một hàm và bạn muốn nó chỉ được sử dụng bởi hàm đó, bạn có thể gắn cờ biến đó là nằm trong phạm vi cục bộ bằng cách đặt trước tên biến bằng từ khóa var :

- Hãy nhớ rằng không phải lúc nào chúng ta cũng có thể kiểm soát tất cả mã đang hoạt động trên trang của mình. Việc các trang bao gồm mã do bên thứ ba viết là rất phổ biến, ví dụ:

• ***Tập lệnh hiển thị quảng cáo***

***• Tập lệnh phân tích và theo dõi người dùng***

***• Nút “chia sẻ” trên mạng xã hội***

### The Browser Object (Đối tượng trình duyệt)

-Ngoài khả năng kiểm soát các thành phần trên trang web, JavaScript còn cho phép bạn truy cập và khả năng thao tác các phần của trình duyệt.

-Trong JavaScript, trình duyệt được gọi là đối tượng cửa sổ . Đối tượng window có một số thuộc tính và phương thức mà chúng ta có thể sử dụng để tương tác với nó

+Bảng dưới đây liệt kê một vài thuộc tính và phương thức có thể được sử dụng với window,có thể truy cập đường link sao để tham khảo đầy đủ (developer.mozilla.org/en-US/docs/Web/API/Window).

### Events

- Sự kiện là một hành động có thể được phát hiện bằng JavaScript, chẳng hạn như khi tải tài liệu hoặc khi người dùng nhấp vào một phần tử hoặc chỉ di chuyển chuột qua phần tử đó. HTML 4.0 giúp tập lệnh có thể được gắn với các sự kiện trên trang, cho dù do người dùng, chính trình duyệt hay các tập lệnh khác khởi tạo. Điều này được gọi là ràng buộc sự kiện.

- Bạn có thể chỉ định hàm sẽ được chạy trong một thuộc tính trong phần đánh dấu.ví dụ:

/\* myFunction will now run when the user clicks anything within 'body' \*/

- Mặc dù vẫn hoạt động nhưng đây là một cách lỗi thời để gắn các sự kiện vào các thành phần trong trang. Nên tránh điều này vì lý do tương tự như việc chúng ta tránh sử dụng thuộc tính kiểu trong đánh dấu của mình để áp dụng kiểu cho các thành phần riêng lẻ.

- Trong trường hợp này, nó làm mờ ranh giới giữa lớp ngữ nghĩa và lớp hành vi trên các trang của chúng tôi và có thể nhanh chóng dẫn đến cơn ác mộng về bảo trì.

**Là một phương pháp**

- Là một cách tiếp cận có phần lỗi thời khác để đính kèm các sự kiện, mặc dù nó giữ mọi thứ nghiêm ngặt trong tập lệnh của chúng tôi.

window.onclick = myFunction; /\* myFunction sẽ chạy khi người dùng nhấp vào bất cứ thứ gì trong cửa sổ trình duyệt \*/

- Chúng ta cũng có thể sử dụng hàm ẩn danh thay vì hàm được xác định trước:

window.onclick = function() {

/\* Bất kỳ mã nào được đặt ở đây sẽ chạy khi người dùng nhấp vào bất cứ thứ gìtrong cửa sổ trình duyệt \*/

};

- Cách tiếp cận này có lợi ích là vừa đơn giản vừa dễ bảo trì, nhưng có một nhược điểm khá lớn.

window.onclick = myFunction;

window.addEventListener

**addEventListener**

- Mặc dù thoạt nhìn phức tạp hơn một chút, nhưng cách tiếp cận này cho phép chúng ta giữ logic trong tập lệnh của mình và cho phép chúng ta thực hiện nhiều liên kết trên một đối tượng. Cú pháp dài dòng hơn một chút.

window.addEventListener("click", myFunction);

- Lưu ý rằng chúng ta bỏ qua từ “on” trước đó trong trình xử lý sự kiện bằng cú pháp này.

- Giống như phương thức trước, addEventListener() cũng có thể được sử dụng với hàm ẩn danh:

window.addEventListener("click", function(e) { });

- Đây chỉ là phần giới thiệu ngắn gọn nên tôi khuyên bạn nên tìm hiểu thêm thông tin về addEventListener() tại trang “eventTarget.addEventListener” trên Tài liệu WebMDN (developer.mozilla.org/en/DOM/element.addEventListener).

### DOM

**DOM là gì?**

DOM là một tiêu chuẩn W3C (World Wide Web Consortium).

DOM xác định một tiêu chuẩn để truy cập tài liệu:

"Mô hình đối tượng tài liệu W3C (DOM) là một nền tảng và giao diện trung lập về ngôn ngữ cho phép các chương trình và tập lệnh truy cập và cập nhật động nội dung, cấu trúc và kiểu dáng của tài liệu."

Tiêu chuẩn W3C DOM được chia thành 3 phần khác nhau:

Core DOM - mô hình chuẩn cho mọi loại tài liệu

XML DOM - mô hình chuẩn cho tài liệu XML

HTML DOM - mô hình chuẩn cho tài liệu HTML

**DOM HTML là gì?**

HTML DOM là một mô hình đối tượng và giao diện lập trình tiêu chuẩn cho HTML. Nó định nghĩa:

·Các phần tử HTML dưới dạng đối tượng

·Thuộc tính của tất cả các phần tử HTML

·Các phương pháp truy cập tất cả các phần tử HTML

·Các sự kiện cho tất cả các phần tử HTML

Nói cách khác: HTML DOM là một tiêu chuẩn về cách lấy, thay đổi, thêm hoặc xóa các phần tử HTML.

**Giao diện lập trình DOM**

HTML DOM có thể được truy cập bằng JavaScript (và bằng các ngôn ngữ lập trình khác).

Trong DOM, tất cả các phần tử HTML được định nghĩa là đối tượng .

Giao diện lập trình là các thuộc tính và phương thức của từng đối tượng.

Thuộc tính là một giá trị mà bạn có thể nhận hoặc đặt (như thay đổi nội dung của phần tử HTML).

Phương thức là một hành động bạn có thể thực hiện (như thêm hoặc xóa phần tử HTML).

**Đối tượng tài liệu HTML DOM**

Đối tượng tài liệu đại diện cho trang web của bạn.

Nếu bạn muốn truy cập bất kỳ phần tử nào trong trang HTML, bạn luôn bắt đầu bằng việc truy cập đối tượng tài liệu.

Dưới đây là một số ví dụ về cách bạn có thể sử dụng đối tượng tài liệu để truy cập và thao tác HTML.

**Tìm các phần tử HTML**

document.getElementById(id)

document.getElementsByTagName(name)

document.getElementsByClassName(name)

**Thay đổi các phần tử HTML**

element.innerHTML =  new html content

element.attribute = new value

element.style.property = new style

element.setAttribute(attribute, value)

**Thêm và xóa các phần tử**

document.createElement(element)

document.removeChild(element)

document.appendChild(element)

document.replaceChild(new, old)

document.write(text)

**Thêm trình xử lý sự kiện**

document.getElementById(id).onclick = function(){code}

Tìm đối tượng HTML

document.anchors

document.applets

document.baseURI

document.body

document.cookie

document.doctype

document.documentElement

document.documentMode

document.documentURI

document.domain

document.domConfig

document.embeds

document.forms

document.head

document.images

document.implementation

document.inputEncoding

document.lastModified

document.links

document.readyState

document.referrer

document.scripts

document.strictErrorChecking

document.title

document.URL

**Tìm các phần tử HTML**

Thông thường, với JavaScript, bạn muốn thao tác với các phần tử HTML.

**Để làm như vậy, trước tiên bạn phải tìm các phần tử. Có nhiều hướng khác nhau để làm điều đó:**

·Tìm các phần tử HTML theo id

·Tìm các phần tử HTML theo tên thẻ

·Tìm các phần tử HTML theo tên lớp

·Tìm các phần tử HTML bằng bộ chọn CSS

·Tìm các phần tử HTML bằng bộ sưu tập đối tượng HTML

**Tìm phần tử HTML theo Id**

Cách dễ nhất để tìm một phần tử HTML trong DOM là sử dụng id phần tử.

Vd:

<!DOCTYPE html>

<html>

<body>

<p id="intro">hello</p>

<p>zhongli<b>venti</b>raiden</p>

<p id="demo"></p>

<script>

const element = document.getElementById("intro");

document.getElementById("demo").innerHTML =

"kq: " + element.innerHTML;

</script>

</body>

</html>

**Tìm các phần tử HTML theo tên lớp**

Nếu bạn muốn tìm tất cả các thành phần HTML có cùng tên lớp, hãy sử dụng getElementsByClassName().

Để thay đổi nội dung của phần tử HTML, hãy sử dụng cú pháp sau:

document.getElementById(id).innerHTML = new HTML

Thay đổi giá trị của một thuộc tính

Để thay đổi giá trị của thuộc tính HTML, hãy sử dụng cú pháp sau:

document.getElem Nội dung HTML động

JavaScript có thể tạo nội dung HTML động:

Ngày : Thứ Hai, 11/03/2024 20:55:46 GMT+0700 ( Giờ Đông Dương)entById(id).attribute = new value;

Không bao giờ sử dụng document.write()sau khi tài liệu được tải. Nó sẽ ghi đè lên tài liệu.

**Ví dụ về JavaScript**

function validateForm() {

  let x = document.forms["myForm"]["fname"].value;

  if (x == "") {

    alert("Name must be filled out");

    return false;

  }

}

JavaScript có thể xác thực đầu vào số

JavaScript thường được sử dụng để xác thực dữ liệu đầu vào dạng số:

Nếu bạn muốn tìm tất cả các phần tử HTML khớp với một bộ chọn CSS được chỉ định (id, tên lớp, loại, thuộc tính, giá trị của thuộc tính, v.v.), hãy sử dụng querySelectorAll()phương thức này.

Thay đổi nội dung HTML

Cách dễ nhất để sửa đổi nội dung của phần tử HTML là sử dụng thuộc tính innerHTML.

Để thay đổi nội dung của phần tử HTML, hãy sử dụng cú pháp sau:

document.getElementById(id).innerHTML = new HTML

Để thay đổi giá trị của thuộc tính HTML, hãy sử dụng cú pháp sau:

document.getE JavaScript có thể tạo nội dung HTML động:

Ngày : Thứ Hai, 11/03/2024 20:55:46 GMT+0700 ( Giờ Đông Dương)

lementById(id).attribute = new value

Trong JavaScript, document.write()có thể được sử dụng để ghi trực tiếp vào luồng đầu ra HTML:

Xác thực mẫu HTML có thể được thực hiện bằng JavaScript.

Nếu trường biểu mẫu (fname) trống, hàm này sẽ cảnh báo một thông báo và trả về sai để ngăn việc gửi biểu mẫu:

JavaScript thường được sử dụng để xác thực dữ liệu đầu vào dạng số:

Trình duyệt có thể thực hiện xác thực biểu mẫu HTML một cách tự động:

Nếu trường biểu mẫu (fname) trống, requiredthuộc tính sẽ ngăn việc gửi biểu mẫu này:

Để thay đổi kiểu của phần tử HTML, hãy sử dụng cú pháp sau:

document.getElementById(id).style.property = new style

HTML DOM cho phép bạn thực thi mã khi có sự kiện xảy ra.

Các sự kiện được trình duyệt tạo ra khi "sự việc xảy ra" với các phần tử HTML:

·Một phần tử được nhấp vào

·Trang đã được tải

·Các trường đầu vào được thay đổi

Bạn sẽ tìm hiểu thêm về các sự kiện trong chương tiếp theo của hướng dẫn này.

Một trang web cơ bản

Để trình bày cách tạo hoạt ảnh HTML bằng JavaScript, chúng tôi sẽ sử dụng một trang web đơn giản

Ví dụ về các sự kiện HTML:

·Khi người dùng click chuột

·Khi một trang web được tải

·Khi một hình ảnh đã được tải

·Khi chuột di chuyển qua một phần tử

·Khi một trường đầu vào được thay đổi

·Khi một biểu mẫu HTML được gửi

Khi người dùng nhấn một phím

Để gán sự kiện cho các phần tử HTML, bạn có thể sử dụng thuộc tính sự kiện.

HTML DOM cho phép bạn gán các sự kiện cho các thành phần HTML bằng JavaScript:

Sự kiện onloadvà onunloadđược kích hoạt khi người dùng vào hoặc rời khỏi trang.

Sự kiện này onloadcó thể được sử dụng để kiểm tra loại trình duyệt và phiên bản trình duyệt của khách truy cập, đồng thời tải phiên bản phù hợp của trang web dựa trên thông tin.

Các sự kiện onloadvà onunloadcó thể được sử dụng để xử lý cookie.

Theo tiêu chuẩn W3C HTML DOM, mọi thứ trong tài liệu HTML đều là một nút:

·Toàn bộ tài liệu là một nút tài liệu

·Mỗi phần tử HTML là một nút phần tử

·Văn bản bên trong các phần tử HTML là các nút văn bản

·Mỗi thuộc tính HTML là một nút thuộc tính (không được dùng nữa)

·Tất cả các bình luận đều là nút bình luận

·Với HTML DOM, tất cả các nút trong cây nút có thể được truy cập bằng JavaScript.

·Các nút mới có thể được tạo và tất cả các nút có thể được sửa đổi hoặc xóa.

Các nút trong cây nút có mối quan hệ phân cấp với nhau.

Các thuật ngữ cha mẹ, con cái và anh chị em được sử dụng để mô tả các mối quan hệ.

·Trong cây nút, nút trên cùng được gọi là nút gốc (hoặc nút gốc)

·Mỗi nút có chính xác một nút cha, ngoại trừ nút gốc (không có nút cha)

·Một nút có thể có nhiều nút con.

·Anh chị em ruột (anh chị em) là các nút có cùng cha mẹ

Từ HTML ở trên bạn có thể đọc:

·<html>là nút gốc

·<html>không có cha mẹ

·<html>là cha mẹ của <head>và<body>

·<head>là đứa con đầu lòng của<html>

·<body>là đứa con cuối cùng của<html>

Và:

·<head>có một đứa con:<title>

·<title>có một con (nút văn bản): "Hướng dẫn DOM"

·<body>có hai con: <h1>và<p>

·<h1>có một đứa con: "DOM Bài học một"

·<p>có một đứa con: "Xin chào thế giới!"

·<h1>và <p>là anh em ruột

Bạn có thể sử dụng các thuộc tính nút sau để điều hướng giữa các nút bằng JavaScript:

·-*parentNode*

*·-childNodes[nodenumber]*

*·-firstChild*

*·-lastChild*

*·-nextSibling*

*·-previousSibling*

Một lỗi phổ biến trong quá trình xử lý DOM là mong muốn một nút phần tử chứa văn bản.

Ví dụ:

<title id="demo">DOM Tutorial</title>

Nút phần tử <title>(trong ví dụ trên) không chứa văn bản.

Nó chứa một nút văn bản có giá trị "Hướng dẫn DOM".

Giá trị của nút văn bản có thể được truy cập bằng innerHTMLthuộc tính của nút:

myTitle = document.getElementById("demo").innerHTML;

Truy cập thuộc tính bên trongHTML cũng giống như truy cập vào thuộc nodeValue tính con đầu tiên:

myTitle = document.getElementById("demo").firstChild.nodeValue;

Việc truy cập đứa trẻ đầu tiên cũng có thể được thực hiện như thế này:

myTitle = document.getElementById("demo").childNodes[0].nodeValue;

Tất cả (3) ví dụ sau đây truy xuất văn bản của một <h1>phần tử và sao chép nó vào một <p>phần tử:

Trong hướng dẫn này, chúng tôi sử dụng thuộc tính InnerHTML để truy xuất nội dung của một phần tử HTML.

Tuy nhiên, việc tìm hiểu các phương pháp khác ở trên rất hữu ích để hiểu cấu trúc cây và cách điều hướng của DOM.

Có hai thuộc tính đặc biệt cho phép truy cập vào toàn bộ tài liệu:

·document.body- Phần thân của tài liệu

·document.documentElement- Giấy tờ đầy đủ

Thuộc nodeNametính chỉ định tên của một nút.

·nodeName ở chế độ chỉ đọc

·nodeName của một nút phần tử giống với tên thẻ

·nodeName của một nút thuộc tính là tên thuộc tính

·nodeName của nút văn bản luôn là #text

-nodeName của nút tài liệu luôn là #documen

Thuộc nodeValuetính chỉ định giá trị của một nút.

·nodeValue cho các nút phần tử lànull

·nodeValue cho các nút văn bản chính là văn bản

·nodeValue cho các nút thuộc tính là giá trị thuộc tính

Thuộc nodeTypetính chỉ được đọc. Nó trả về loại nút.

document.getElementById("id02").innerHTML = document.getElementById("id01").nodeTy<!DOCTYPE html>

Các thuộc tính nodeType quan trọng nhất là:

DOCUMENT\_TYPE\_NODE

COMMENT\_NODE

TEXT\_NODE

ATTRIBUTE\_NODE

ELEMENT\_NODE

Loại 2 không được dùng nữa trong HTML DOM (nhưng vẫn hoạt động). Nó không được dùng nữa trong XML DOM.

Để thêm một phần tử mới vào HTML DOM, trước tiên bạn phải tạo phần tử (nút phần tử), sau đó nối nó vào phần tử hiện có

Phương appendChild()thức trong ví dụ trước đã thêm phần tử mới làm phần tử con cuối cùng của phần tử cha.

Nếu không muốn, bạn có thể sử dụng insertBefore()phương pháp:

Để xóa phần tử HTML, hãy sử dụng remove() phương thức:

Đối với các trình duyệt không hỗ trợ *remove()*phương thức, bạn phải tìm nút cha để xóa một phần tử:

Để thay thế một phần tử vào HTML DOM, hãy sử dụng replaceChild()phương thức:

Phương *getElementsByTagName()*thức trả về một*HTMLCollection*đối tượng.

Một *HTMLCollection*đối tượng là một danh sách (bộ sưu tập) các phần tử HTML dạng mảng.

Đoạn mã sau chọn tất cả *<p>*các thành phần trong tài liệu:

Ví dụ

*const myCollection = document.getElementsByTagName("p");*

Các phần tử trong bộ sưu tập có thể được truy cập bằng số chỉ mục.

Để truy cập phần tử <p> thứ hai, bạn có thể viết:

myCollection[1]

Lưu ý: Chỉ số bắt đầu từ 0.

Thuộc*lengthtính* xác định số HTMLCollection có thể trông giống như một mảng nhưng thực tế không phải vậy.

Bạn có thể lặp qua danh sách và tham chiếu đến các phần tử bằng một số (giống như một mảng).

Tuy nhiên, bạn không thể sử dụng các phương thức mảng như valueOf(), pop(), push() hoặc join() trên HTMLCollection.

phần tử trong một *HTMLCollection:*

Một NodeListđối tượng là một danh sách (tập hợp) các nút được trích xuất từ ​​một tài liệu.

Một NodeListđối tượng gần giống như một HTMLCollectionđối tượng.

Một số trình duyệt (cũ hơn) trả về đối tượng NodeList thay vì HTMLCollection cho các phương thức như getElementsByClassName().

Tất cả các trình duyệt đều trả về một đối tượng NodeList cho thuộc tính này childNodes.

Hầu hết các trình duyệt đều trả về đối tượng NodeList cho phương thức này querySelectorAll().

Đoạn mã sau chọn tất cả <p>các nút trong tài liệu:

const myNodeList = document.querySelectorAll("p");

Các phần tử trong NodeList có thể được truy cập bằng số chỉ mục.

Để truy cập nút <p> thứ hai, bạn có thể viết:

myNodeList[1]

Polyfill là một thuật ngữ do Remy Sharp đặt ra để mô tả một “shim” JavaScript giúp bình thường hóa các hành vi khác nhau giữa các trình duyệt (remysharp. com/2010/10/08/what-is-a-polyfill).

Shim HTML5 (hoặc shiv)

Picturefill

Để sử dụng Picturefill, hãy tải xuống tập lệnh và thêm nó vào phần đầu tài liệu. Tập lệnh đầu tiên tạo thành phần hình ảnh cho các trình duyệt không nhận ra nó. Tập lệnh thứ hai gọi chính tập lệnh Picturefill và thuộc tính async cho trình duyệt biết nó có thể tải Picturefill một cách không đồng bộ—nghĩa là không cần đợi tập lệnh hoàn tất trước khi tải phần còn lại của tài liệu.

*<head>*

*document.createElement("picture" );*

*<script>*

*<script src=”picturefill.js” async>*

*</script>*

*</head>*

Mặt khác, các trình duyệt không có JavaScript cũng không hỗ trợ thành phần hình ảnh sẽ chỉ nhìn thấy văn bản thay thế cho hình ảnh. Tải xuống Picturefill và nhận thông tin về cách sử dụng nó tại scottjehl.github.io/picturefill/.

## Ngôn ngữ PHP.

### Giới thiệu về PHP.

PHP là ngôn ngữ lập trình máy chủ và là công cụ mạnh mẽ để tạo ra các trang web động và tương tác.

PHP là một giải pháp thay thế miễn phí, hiệu quả và được sử dụng rộng rãi cho các đối thủ cạnh tranh như ASP của Microsoft.

PHP là từ viết tắt của "PHP: Hypertext Preprocessor"

PHP là một ngôn ngữ kịch bản mã nguồn mở được sử dụng rộng rãi

Các tập lệnh PHP được thực thi trên máy chủ

PHP có thể tải xuống và sử dụng miễn phí

**PHP là một ngôn ngữ tuyệt vời và phổ biến!**

Nó đủ mạnh để trở thành cốt lõi của hệ thống blog lớn nhất trên web (WordPress)!  
Nó đủ sâu để chạy các mạng xã hội lớn!  
Nó cũng đủ dễ để trở thành ngôn ngữ máy chủ đầu tiên cho người mới bắt đầu!

Các tệp PHP có thể chứa văn bản, HTML, CSS, JavaScript và mã PHP

Mã PHP được thực thi trên máy chủ và kết quả được trả về trình duyệt dưới dạng HTML thuần túy

Các tập tin PHP có phần mở rộng " .php"

PHP có thể tạo ra nội dung trang động

PHP có thể tạo, mở, đọc, ghi, xóa và đóng các tệp trên máy chủ

PHP có thể thu thập dữ liệu biểu mẫu

PHP có thể gửi và nhận cookie

PHP có thể thêm, xóa, sửa đổi dữ liệu trong cơ sở dữ liệu của bạn

PHP có thể được sử dụng để kiểm soát quyền truy cập của người dùng

PHP có thể mã hóa dữ liệu

Với PHP, bạn không bị giới hạn ở đầu ra HTML. Bạn có thể đầu ra hình ảnh hoặc tệp PDF. Bạn cũng có thể đầu ra bất kỳ văn bản nào, chẳng hạn như XHTML và XML.

PHP chạy trên nhiều nền tảng khác nhau (Windows, Linux, Unix, Mac OS X, v.v.)

PHP tương thích với hầu hết các máy chủ hiện nay (Apache, IIS, v.v.)

PHP hỗ trợ nhiều loại cơ sở dữ liệu

PHP miễn phí. Tải xuống từ nguồn PHP chính thức: [www.php.net](http://www.php.net/)

PHP dễ học và chạy hiệu quả trên phía máy chủ

### Cú pháp PHP cơ bản

Có thể đặt tập lệnh PHP ở bất kỳ vị trí nào trong tài liệu.

Một tập lệnh PHP bắt đầu <?phpvà kết thúc bằng ?>:

**<?php**  
// PHP code goes here  
**?>**

Phần mở rộng tệp mặc định cho tệp PHP là " .php".

Một tệp PHP thường chứa các thẻ HTML và một số mã lệnh PHP.

Dưới đây là ví dụ về một tệp PHP đơn giản, với một tập lệnh PHP sử dụng hàm PHP tích hợp " " echođể xuất văn bản "Hello World!" trên một trang web:

<!DOCTYPE html>  
<html>  
<body>  
  
<h1>My first PHP page</h1>  
  
**<?php**  
echo "Hello World!";  
**?>**  
  
</body>   
</html>

**Phân biệt chữ hoa và chữ thường trong PHP.**

Trong PHP, các từ khóa (ví dụ: if, else, while, echo, v.v.), lớp, hàm và hàm do người dùng định nghĩa không phân biệt chữ hoa chữ thường.

Trong ví dụ dưới đây, cả ba câu lệnh echo đều bình đẳng và hợp lệ:

Ví dụ:

ECHOgiống như echo:

<!DOCTYPE html>  
<html>  
<body>  
  
**<?php**  
ECHO "Hello World!<br>";  
echo "Hello World!<br>";  
EcHo "Hello World!<br>";  
**?>**  
  
</body>  
</html>

**Lưu ý:** Tuy nhiên, tất cả tên biến đều phân biệt chữ hoa chữ thường!

Hãy xem ví dụ bên dưới; chỉ có câu lệnh đầu tiên sẽ hiển thị giá trị của $colorbiến! Điều này là do $color, $COLOR, và $coLORđược coi là ba biến khác nhau:

Ví dụ:

$COLORkhông giống *như*$color :

<!DOCTYPE html>  
<html>  
<body>  
  
**<?**php  
$color = "red";  
echo "My car is " . $color . "<br>";  
echo "My house is " . $COLOR . "<br>";  
echo "My boat is " . $coLOR . "<br>";  
**?>**  
  
</body>  
</html>

Biến PHP:

Trong PHP, một biến bắt đầu bằng $dấu, theo sau là tên của biến:

Ví dụ:

$x = 5;  
$y = "John";

Trong ví dụ trên, biến $xsẽ giữ giá trị 5, và biến $ysẽ giữ giá trị "John".

**Lưu ý:** Khi bạn gán giá trị văn bản cho một biến, hãy đặt giá trị trong dấu ngoặc kép.

**Lưu ý:** Không giống như các ngôn ngữ lập trình khác, PHP không có lệnh để khai báo biến. Biến được tạo ra ngay khi bạn gán giá trị cho nó lần đầu tiên.

Một biến có thể có tên ngắn (như $xvà $y) hoặc tên mô tả chi tiết hơn ( $age, $carname, $total\_volume).

Quy tắc cho biến PHP:

* Một biến bắt đầu bằng $dấu, theo sau là tên của biến
* Tên biến phải bắt đầu bằng một chữ cái hoặc ký tự gạch dưới
* Tên biến không thể bắt đầu bằng số
* Tên biến chỉ có thể chứa các ký tự chữ và số và dấu gạch dưới (Az, 0-9 và)
* Tên biến phân biệt chữ hoa chữ thường ( $agevà $AGElà hai biến khác nhau)

**Biến đầu ra.**

Câu lệnh PHP echothường được sử dụng để xuất dữ liệu ra màn hình.

Ví dụ sau đây sẽ chỉ cách xuất văn bản và biến:

$txt = "W3Schools.com";  
echo "I love $txt!";

Ví dụ sau đây sẽ tạo ra kết quả tương tự như ví dụ trên:

$txt = "W3Schools.com";  
echo "I love " . $txt . "!";

Ví dụ sau sẽ đưa ra tổng của hai biến:

$x = 5;  
$y = 4;  
echo $x + $y;

**Kiểu dữ liệu PHP.**

Biến có thể lưu trữ dữ liệu ở nhiều kiểu khác nhau và mỗi kiểu dữ liệu khác nhau có thể thực hiện những chức năng khác nhau.

* PHP hỗ trợ các kiểu dữ liệu sau:
* Sợi dây
* Số nguyên
* Float (số dấu phẩy động - còn gọi là double)
* Boolean
* Mảng
* Sự vật
* Vô giá trị
* Tài nguyên

### Biểu mẫu PHP.

**PHP - Biểu mẫu HTML đơn giản.**

Ví dụ dưới đây hiển thị một biểu mẫu HTML đơn giản với hai trường nhập liệu và một nút gửi:

<html>  
<body>  
  
<form action="welcome.php" method="POST">  
Name: <input type="text" name="name"><br>  
E-mail: <input type="text" name="email"><br>  
<input type="submit">  
</form>  
  
</body>  
</html>

Khi người dùng điền vào biểu mẫu ở trên và nhấp vào nút gửi, dữ liệu biểu mẫu sẽ được gửi để xử lý đến tệp PHP có tên "welcome.php". Dữ liệu biểu mẫu được gửi bằng phương thức HTTP POST.

Để hiển thị dữ liệu đã gửi, bạn chỉ cần lặp lại tất cả các biến.

"welcome.php" trông như thế này:

<html>  
<body>  
  
Welcome **<?php** echo $\_POST["name"]; **?>**<br>  
Your email address is: **<?php** echo $\_POST["email"]; **?>**  
  
</body>  
</html>

Kết quả đầu ra có thể giống như thế này:

Welcome John  
Your email address is [**john.doe@example.com**](mailto:john.doe@example.com)

Kết quả tương tự cũng có thể đạt được khi sử dụng phương thức HTTP GET:

Ví dụ tương tự, nhưng phương thức được đặt thành GET thay vì POST:

<html>  
<body>  
  
<form action="welcome\_get.php" method="GET">  
Name: <input type="text" name="name"><br>  
E-mail: <input type="text" name="email"><br>  
<input type="submit">  
</form>  
  
</body>  
</html>

và "welcome\_get.php" trông như thế này:

<html>  
<body>  
  
Welcome **<?php** echo $\_GET["name"]; **?>**<br>  
Your email address is: **<?php** echo $\_GET["email"]; **?>**  
  
</body>

</html>

Đoạn mã trên khá đơn giản và không bao gồm bất kỳ xác thực nào.

Bạn cần xác thực dữ liệu biểu mẫu để bảo vệ tập lệnh của mình khỏi mã độc hại.

**Hãy nghĩ đến BẢO MẬT khi xử lý biểu mẫu PHP!**

Trang này không chứa bất kỳ xác thực biểu mẫu nào, nó chỉ hiển thị cách bạn có thể gửi và lấy dữ liệu biểu mẫu.

Tuy nhiên, các trang tiếp theo sẽ chỉ cách xử lý biểu mẫu PHP với tính bảo mật! Xác thực đúng dữ liệu biểu mẫu là điều quan trọng để bảo vệ biểu mẫu của bạn khỏi tin tặc và kẻ gửi thư rác!

GET so với POST:

Cả GET và POST đều tạo ra một mảng (ví dụ: array( key1 => value1, key2 => value2, key3 => value3, ...)). Mảng này chứa các cặp khóa/giá trị, trong đó khóa là tên của các điều khiển biểu mẫu và giá trị là dữ liệu đầu vào từ người dùng.

Cả GET và POST đều được coi là $\_GET và $\_POST. Đây là các siêu toàn cục, nghĩa là chúng luôn có thể truy cập được, bất kể phạm vi - và bạn có thể truy cập chúng từ bất kỳ hàm, lớp hoặc tệp nào mà không cần phải làm bất cứ điều gì đặc biệt.

$\_GET là một mảng các biến được truyền vào tập lệnh hiện tại thông qua các tham số URL.

$\_POST là một mảng các biến được truyền tới tập lệnh hiện tại thông qua phương thức HTTP POST.

Thông tin được gửi từ biểu mẫu bằng phương thức GET có thể được **mọi người nhìn thấy** (tất cả tên biến và giá trị đều được hiển thị trong URL). GET cũng có giới hạn về lượng thông tin cần gửi. Giới hạn là khoảng 2000 ký tự. Tuy nhiên, vì các biến được hiển thị trong URL nên có thể đánh dấu trang. Điều này có thể hữu ích trong một số trường hợp.

GET có thể được sử dụng để gửi dữ liệu không nhạy cảm.

**Lưu ý:** KHÔNG BAO GIỜ được sử dụng GET để gửi mật khẩu hoặc thông tin nhạy cảm khác!

Thông tin được gửi từ biểu mẫu bằng phương thức POST sẽ **không hiển thị với người khác** (tất cả tên/giá trị đều được nhúng trong nội dung của yêu cầu HTTP) và **không giới hạn** lượng thông tin được gửi.

Hơn nữa, POST hỗ trợ các chức năng nâng cao như hỗ trợ nhập nhị phân nhiều phần trong khi tải tệp lên máy chủ.

Tuy nhiên, vì các biến không được hiển thị trong URL nên không thể đánh dấu trang.

**Các nhà phát triển thích POST để gửi dữ liệu biểu mẫu.**

## Kết chương.

Trong chương này, chúng ta đã tìm hiểu về ba thành phần cơ bản nhưng cực kỳ quan trọng của lập trình web: HTML, CSS và JavaScript.

HTML, ngôn ngữ đánh dấu siêu văn bản, là cốt lõi của mọi trang web. Nó giúp chúng ta xây dựng cấu trúc của trang web bằng cách sử dụng các thẻ và phần tử. Chúng ta đã thảo luận về cách sử dụng các thẻ HTML để tạo ra các phần tử như tiêu đề, đoạn văn bản, hình ảnh và liên kết, cũng như cách chúng tương tác với nhau.

CSS, ngôn ngữ kiểu trang, là công cụ mạnh mẽ để tùy chỉnh giao diện của trang web. Chúng ta đã tìm hiểu về cách sử dụng CSS để thay đổi màu sắc, font chữ, kích thước và vị trí của các phần tử HTML, tạo ra giao diện trực quan và thú vị cho người dùng.

PHP là ngôn ngữ lập trình máy chủ và là công cụ mạnh mẽ để tạo ra các trang web động và tương tác.

PHP là một giải pháp thay thế miễn phí, hiệu quả và được sử dụng rộng rãi cho các đối thủ cạnh tranh như ASP của Microsoft.

Cuối cùng, JavaScript là ngôn ngữ lập trình cho phía client, cho phép chúng ta thêm tính năng động và tương tác vào trang web. Chúng ta đã thảo luận về cách sử dụng JavaScript để thực hiện các hành động như xử lý sự kiện, thay đổi nội dung và tương tác với người dùng.

Bằng cách hiểu và vận dụng HTML, CSS, PHP và JavaScript một cách hiệu quả, chúng ta có thể xây dựng những trang web đẹp và chức năng, mang lại trải nghiệm tốt nhất cho người dùng.

# HIỆN THỰC HOÁ NGHIÊN CỨU

## Mô tả bài toán

**Bối cảnh:**

Trà Vinh là một tỉnh thuộc vùng Đồng bằng sông Cửu Long, nổi tiếng với lịch sử lâu đời và sự đa dạng văn hóa, đặc biệt là cộng đồng người Khmer. Với sự phát triển của công nghệ thông tin, việc xây dựng một nền tảng trực tuyến để giới thiệu lịch sử, di sản văn hóa và những nét đặc sắc của Trà Vinh không chỉ giúp bảo tồn giá trị văn hóa mà còn góp phần thúc đẩy du lịch và phát triển kinh tế địa phương.

**Vấn đề:**

Hiện nay, thông tin về lịch sử và di sản văn hóa Trà Vinh vẫn chưa được hệ thống hóa và phổ biến một cách rộng rãi qua môi trường trực tuyến. Việc thiếu một nền tảng tập trung có thể dẫn đến sự khó khăn trong việc tiếp cận thông tin của cả người dân địa phương lẫn du khách.

**Mục tiêu:**

Phát triển một website giới thiệu lịch sử và di sản văn hóa Trà Vinh với các tính năng:

1. **Cung cấp thông tin chi tiết:** Giới thiệu về lịch sử, các di tích, di sản văn hóa vật thể và phi vật thể.
2. **Tăng cường trải nghiệm người dùng:** Thiết kế giao diện thân thiện, dễ sử dụng, phù hợp với đối tượng người dùng đa dạng.
3. **Tương tác trực tuyến:** Cho phép người dùng đóng góp nội dung, chia sẻ trải nghiệm, hoặc đặt câu hỏi liên quan.
4. **Phát triển nội dung đa phương tiện:** Cung cấp bài viết, hình ảnh, video để giúp nội dung thêm sinh động và hấp dẫn.
5. **Hỗ trợ quảng bá du lịch:** Tích hợp bản đồ địa điểm tham quan, các tour gợi ý và thông tin liên lạc cần thiết.

**Các thách thức chính:**

1. **Thu thập và tổ chức dữ liệu:** Tìm kiếm và xác thực thông tin lịch sử, văn hóa và hình ảnh liên quan để đảm bảo tính chính xác và toàn diện.
2. **Thiết kế giao diện và trải nghiệm người dùng (UI/UX):** Tối ưu hóa trang web để thân thiện với người dùng, bao gồm cả người dùng thiết bị di động.
3. **Tương thích kỹ thuật:** Đảm bảo website hoạt động ổn định trên nhiều nền tảng và trình duyệt khác nhau.
4. **Bảo mật và quản lý nội dung:** Đảm bảo an toàn cho dữ liệu và hệ thống, đồng thời xây dựng cơ chế quản lý, cập nhật nội dung hiệu quả.

**Giải pháp đề xuất:**

* **Công nghệ:** Sử dụng các nền tảng phát triển web phổ biến như HTML, CSS, JavaScript (frontend) và các framework như Django, Flask hoặc WordPress (backend).
* **Tích hợp bản đồ:** Sử dụng API bản đồ (như Google Maps API) để hiển thị vị trí các di tích và danh lam thắng cảnh.
* **Quản lý nội dung:** Xây dựng hệ thống quản trị nội dung (CMS) để dễ dàng thêm, sửa và xóa thông tin.
* **Marketing:** Tối ưu hóa công cụ tìm kiếm (SEO) để tăng khả năng tiếp cận của website trên các nền tảng tìm kiếm.

## Đặc tả các yêu cầu

### Yêu cầu chức năng

#### Trang chủ

* Hiển thị các thông tin nổi bật về lịch sử, văn hóa, và danh lam thắng cảnh của Trà Vinh.
* Cung cấp thanh điều hướng đến các mục nội dung chính: Lịch sử, Di sản văn hóa, Địa điểm du lịch, và Liên hệ.
* Tích hợp thanh tìm kiếm để người dùng dễ dàng tra cứu thông tin.

#### Thông tin lịch sử

* Trình bày các mốc thời gian quan trọng trong lịch sử hình thành và phát triển của Trà Vinh.
* Có chức năng xem thông tin theo thời kỳ hoặc sự kiện lịch sử.

#### Di sản văn hóa

* Danh sách các di sản văn hóa vật thể và phi vật thể (chùa chiền, lễ hội, nghề - truyền thống, nghệ thuật biểu diễn).
* Chi tiết mỗi di sản bao gồm: mô tả, ý nghĩa, hình ảnh và video minh họa.

Tích hợp bản đồ hiển thị vị trí của các di sản.

#### Địa điểm du lịch

* Danh mục các địa điểm tham quan nổi bật của Trà Vinh (chùa Âng, ao Bà Om, Cồn Chim, v.v.).
* Thông tin bao gồm: mô tả ngắn, địa chỉ, hướng dẫn di chuyển và hình ảnh thực tế.
* Tính năng đánh giá (rating) và nhận xét (review) từ người dùng.

#### Tương tác và đóng góp nội dung

* Cho phép người dùng gửi bài viết, hình ảnh hoặc video để đóng góp nội dung cho website.
* Hệ thống kiểm duyệt nội dung trước khi đăng.

#### Tích hợp đa phương tiện

* Hiển thị video, hình ảnh chất lượng cao.
* Chức năng slideshow hoặc gallery để tăng trải nghiệm người dùng.

#### Quản trị website

* Quản lý nội dung: thêm, sửa, xóa bài viết và tài nguyên.
* Quản lý đóng góp của người dùng: duyệt hoặc từ chối nội dung.
* Xem báo cáo lưu lượng truy cập và hoạt động trên website.

### Yêu cầu phi chức năng

#### Hiệu năng (Performance)

* Thời gian tải trang không vượt quá 3 giây trên mạng thông thường.
* Hỗ trợ số lượng truy cập đồng thời tối đa 500 người dùng.

#### Tính tương thích (Compatibility)

* Website hoạt động tốt trên các trình duyệt phổ biến (Chrome, Firefox, Safari, Edge).
* Tương thích với các thiết bị: máy tính, điện thoại thông minh, và máy tính bảng.

#### Bảo mật (Security)

* Đảm bảo an toàn dữ liệu người dùng và ngăn chặn tấn công từ bên ngoài (SQL Injection, XSS, CSRF).
* Hỗ trợ mã hóa dữ liệu và giao thức HTTPS.

#### Khả năng mở rộng (Scalability)

* Thiết kế cơ sở dữ liệu và hệ thống để có thể mở rộng nội dung trong tương lai.
* Dễ dàng tích hợp thêm các tính năng như ngôn ngữ đa dạng hoặc thương mại điện tử.

#### Tối ưu hóa công cụ tìm kiếm (SEO)

* Thiết lập meta tags, từ khóa và URL thân thiện để tăng khả năng xuất hiện trên các công cụ tìm kiếm như Google.

### Công cụ và công nghệ đề xuất

#### Công nghệ phát triển

* Frontend:
  + **HTML & CSS:** Để xây dựng giao diện đơn giản và trực quan.
  + **JavaScript cơ bản:** Để thêm tính năng tương tác nhẹ nhàng (nếu cần).
  + Sử dụng thư viện **Bootstrap** (nếu cần) để hỗ trợ thiết kế giao diện đẹp mà không cần viết quá nhiều CSS.
* **Backend:**
  + **Python với Flask:** Đây là framework dễ học, phù hợp cho các dự án nhỏ và trung bình. Flask có cú pháp gọn gàng, dễ tiếp cận, và có nhiều tài liệu hướng dẫn.
  + **SQLite:** Cơ sở dữ liệu tích hợp sẵn trong Python, dễ sử dụng cho các dự án vừa và nhỏ, không yêu cầu cài đặt phức tạp.
* **Hosting:**
  + **PythonAnywhere hoặc Replit:** Cung cấp nền tảng miễn phí (hoặc giá rẻ) để bạn triển khai website lên mạng. Đây là lựa chọn lý tưởng cho người mới bắt đầu.

### Yêu cầu chức năng

#### Trang chủ

* + Hiển thị thông tin giới thiệu ngắn gọn về Trà Vinh (bao gồm văn hóa, lịch sử, và các điểm nổi bật).
  + Có thanh điều hướng đến các mục chính: Lịch sử, Di sản văn hóa, Địa điểm tham quan, và Liên hệ.
  + Tích hợp thanh tìm kiếm để người dùng dễ dàng tra cứu nội dung trên website.

#### Lịch sử Trà Vinh

* + Mục cung cấp thông tin chi tiết về lịch sử hình thành và phát triển của Trà Vinh.
  + Hiển thị theo dạng danh sách sự kiện hoặc mốc thời gian, giúp người dùng dễ theo dõi.
  + Cho phép hiển thị hình ảnh hoặc tài liệu lịch sử (nếu có).

#### Di sản văn hóa

* Danh sách các di sản văn hóa của tỉnh, chia thành hai loại:
* Di sản vật thể: Ví dụ: các công trình kiến trúc (chùa Khmer, ao Bà Om).
* Di sản phi vật thể: Ví dụ: lễ hội, nghệ thuật truyền thống (dàn nhạc ngũ âm, múa rô-băm).
* Thông tin chi tiết bao gồm: mô tả, ý nghĩa, hình ảnh, video minh họa (nếu có).
* Chức năng lọc hoặc phân loại di sản theo loại hình hoặc địa điểm.

#### Địa điểm tham quan

* Hiển thị danh sách các địa điểm tham quan nổi tiếng tại Trà Vinh.
* Mỗi địa điểm cung cấp các thông tin:
* Mô tả ngắn gọn.
* Địa chỉ hoặc hướng dẫn di chuyển.
* Hình ảnh minh họa.
* Tích hợp bản đồ (sử dụng Google Maps iframe) để hiển thị vị trí của các địa điểm.

#### Đóng góp nội dung

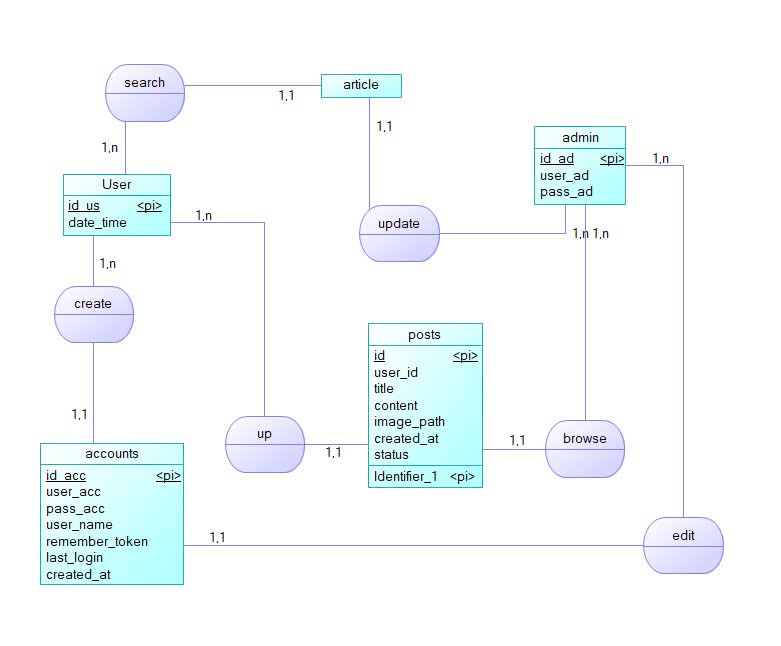
* Người dùng có thể gửi bài viết, hình ảnh hoặc tài liệu liên quan đến văn hóa và lịch sử Trà Vinh.
* Cung cấp form gửi nội dung bao gồm các trường: Tên, Email, Tiêu đề bài viết, Nội dung, và File đính kèm (nếu có).
* Hệ thống thông báo nội dung cần được quản trị viên kiểm duyệt trước khi hiển thị trên website.

#### Liên hệ (Contact Section)

* Cung cấp thông tin liên lạc, bao gồm:
* Địa chỉ email và số điện thoại hỗ trợ.
* Liên kết đến các mạng xã hội (nếu có).
* Form liên hệ để người dùng gửi ý kiến phản hồi hoặc thắc mắc, với các trường: Tên, Email, Nội dung phản hồi.

## Thiết kế dữ liệu

### Sơ đồ hệ thống



Hình 3.1 Sơ dồ hệ thống

## Tên bảng: posts

*Bảng 3.1 Bảng mô tả thuộc tính bảng post*

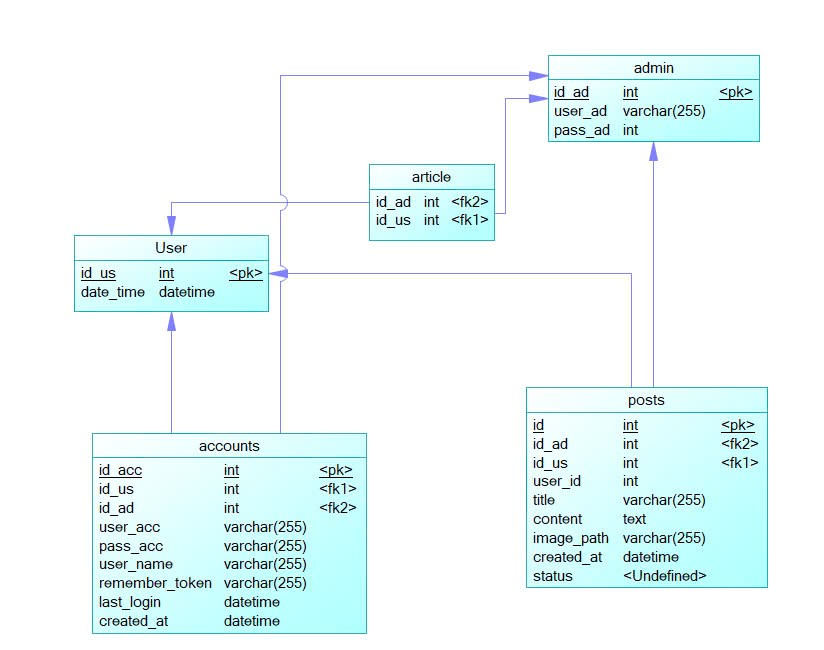
|  |  |  |
| --- | --- | --- |
| Tên cột | Loại dữ liệu | Mô tả |
| id | INT (AUTO\_INCREMENT) | Mã bài viết duy nhất (Primary Key). |
| user\_id | INT | ID của người dùng đã tạo bài viết (Foreign Key, liên kết với bảng users). |
| title | VARCHAR(255) | Tiêu đề bài viết. |
| content | TEXT | Nội dung chi tiết của bài viết. |
| image\_path | VARCHAR(255) | Đường dẫn tới hình ảnh minh họa của bài viết (nếu có). |
| created\_at | DATETIME | Thời gian bài viết được tạo. |
| status | ENUM('draft', 'published', 'archived') | Trạng thái bài viết: nháp, đã xuất bản, hoặc đã lưu trữ. |

**Tên bảng**: accounts

*Bảng 3.2 Bảng mô tả thuộc tính bảng accouts*

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Đặc điểm | Mô tả |
| id\_acc | INTEGER | |  | | --- | |  | | **Primary Key**, Auto Increment | | ID duy nhất đại diện cho mỗi tài khoản. |
| user\_acc | VARCHAR(255) | **NOT NULL**, Unique | Tên đăng nhập của người dùng (username) |
| pass\_acc | VARCHAR(255) | NOT NULL | Mật khẩu của tài khoản (nên mã hóa bằng bcrypt). |
| user\_name | VARCHAR(255) |  | |  | | --- | |  | | Tên hiển thị hoặc tên đầy đủ của người dùng. | |
| remember\_token | VARCHAR(255) |  | |  | | --- | |  | | Token để ghi nhớ phiên đăng nhập (cho tính năng "Nhớ tài khoản"). | |
| last\_login | DATETIME |  | Thời điểm lần đăng nhập gần nhất. |
| created\_at | DATETIME | **NOT NULL**, Default: NOW() | Ngày giờ tạo tài khoản. |

### Mô hình dữ liệu



Hình 3.1 Mô hình dữ liệu

## Thiết kế giao diện

**Mục tiêu:**

Thiết kế một giao diện website về lịch sử và di sản văn hóa tỉnh Trà Vinh đảm bảo tính trực quan, dễ sử dụng và phù hợp với nội dung giới thiệu.

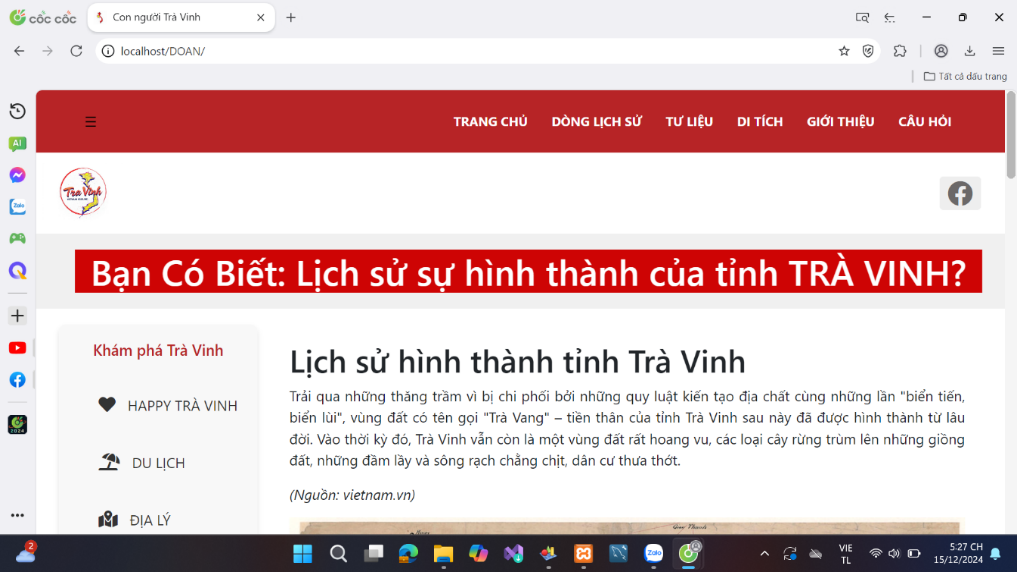
**Cấu trúc cơ bản của giao diện:**

* + **Thanh điều hướng:**
* Hiển thị các mục chính: Trang Chủ, Dòng Lịch Sử, Tư Liệu, Di Tích, Giới Thiệu, Câu Hỏi.
* Màu sắc: Nền màu đỏ đậm (như hình), chữ màu trắng để nổi bật.
* Thêm biểu tượng menu (3 gạch ngang) để thuận tiện khi xem trên thiết bị di động (Responsive Design).
  + **Nội dung chính (Content Area):**
* Tùy theo mục bạn chọn trên thanh điều hướng, khu vực này sẽ thay đổi nội dung.
* Các thành phần quan trọng:
* Trang chủ: Tổng quan giới thiệu về tỉnh Trà Vinh.
* Dòng lịch sử: Dòng thời gian (Timeline) về các sự kiện lịch sử.
* Tư liệu: Thư viện tư liệu (hình ảnh, video, tài liệu PDF).
* Di tích: Danh sách và thông tin các di tích nổi bật.
* Giới thiệu: Giới thiệu dự án hoặc thông tin nhóm thực hiện.
* Câu hỏi: Phần hỏi đáp hoặc các câu đố trắc nghiệm để tương tác với người dùng.
  + **Chân trang (Footer):**

Thông tin liên hệ, bản quyền, hoặc liên kết đến các trang mạng xã hội.

# KẾT QUẢ NGHIÊN CỨU

## Giao diện trang chủ người dùng



Hình 4.1 Giao diện trang chủ của người dùng

**Thanh điều hướng chính**

Thanh điều hướng chính sẽ nằm ở đầu trang website, với các mục lựa chọn như: Trang Chủ, Dòng Lịch Sử, Tư Liệu, Di Tích, Giới Thiệu, Câu Hỏi. Mục đích của thanh điều hướng là giúp người dùng dễ dàng chuyển qua lại giữa các trang của website. Các mục này sẽ được phân loại hợp lý để tạo sự mạch lạc và dễ tìm kiếm.

* **Trang Chủ:** Giới thiệu tổng quan về Trà Vinh, các thông tin cơ bản và hình ảnh nổi bật.
* **Dòng Lịch Sử:** Hiển thị một dòng thời gian (Timeline) với các sự kiện quan trọng trong lịch sử tỉnh Trà Vinh.
* **Tư Liệu:** Thư viện tư liệu, bao gồm hình ảnh, video, tài liệu PDF liên quan đến di sản văn hóa.
* **Di Tích:** Cung cấp thông tin về các di tích văn hóa, lịch sử nổi bật ở Trà Vinh, kèm theo hình ảnh và mô tả chi tiết.
* **Giới Thiệu:** Thông tin về nhóm phát triển website, mục đích của dự án.
* **Câu Hỏi:** Một phần tương tác cho phép người dùng tham gia trả lời câu hỏi hoặc tham gia các bài kiểm tra liên quan đến lịch sử và văn hóa Trà Vinh.

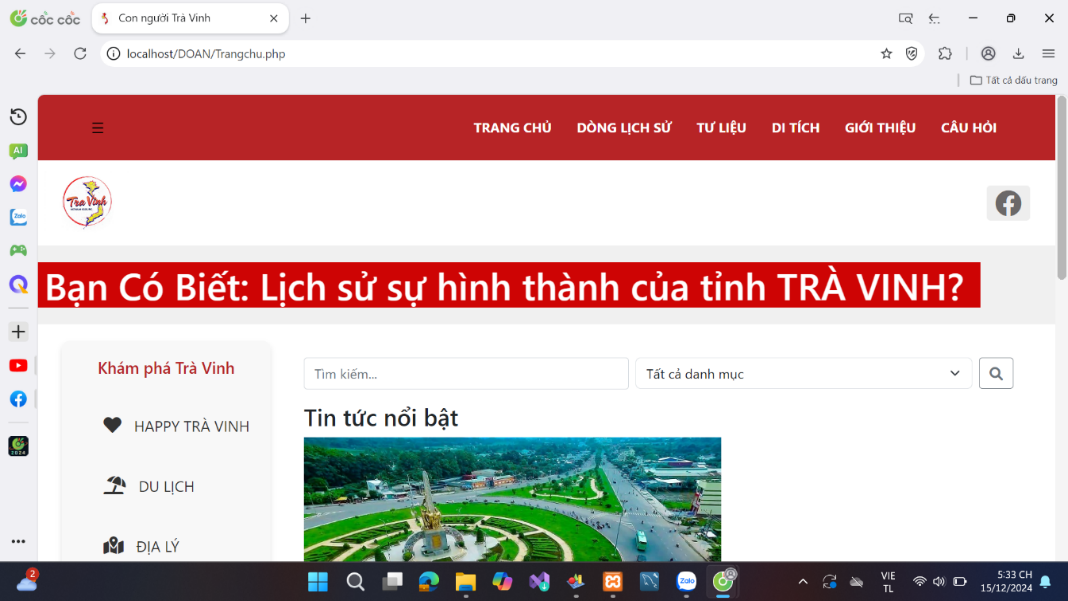
**Menu bên**

Thanh menu bên (sidebar) sẽ xuất hiện khi người dùng muốn tìm kiếm các thông tin cụ thể về Trà Vinh. Các mục trong thanh menu bao gồm:

* **Happy Trà Vinh**
* **Du Lịch**
* **Địa Lý**
* **Văn Hóa**

Mỗi mục sẽ đi kèm với một biểu tượng dễ hiểu (ví dụ: biểu tượng trái tim cho "Happy Trà Vinh", biểu tượng bãi biển cho "Du Lịch", biểu tượng bản đồ cho "Địa Lý", và mặt nạ cho "Văn Hóa"). Mục đích của menu này là giúp người dùng có thể khám phá các khía cạnh khác nhau của Trà Vinh một cách trực quan và dễ dàng.

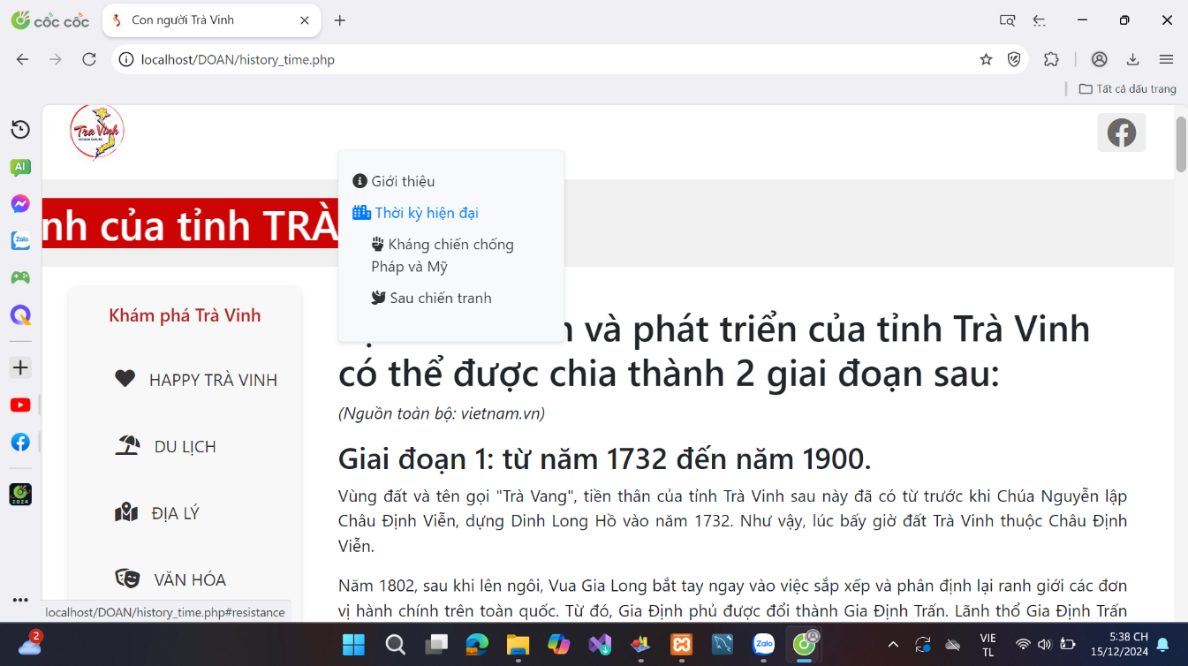
## Giao diện chức năng tìm kiếm



Hình 4.2 Giao diện chức năng tìm kiếm

Khi người dùng nhấn vào logo của website sẽ được chuyển đến trang tìm kiếm giúp người dùng có thể thao tác nhanh hơn trong việc tìm kiếm thông tin của website.

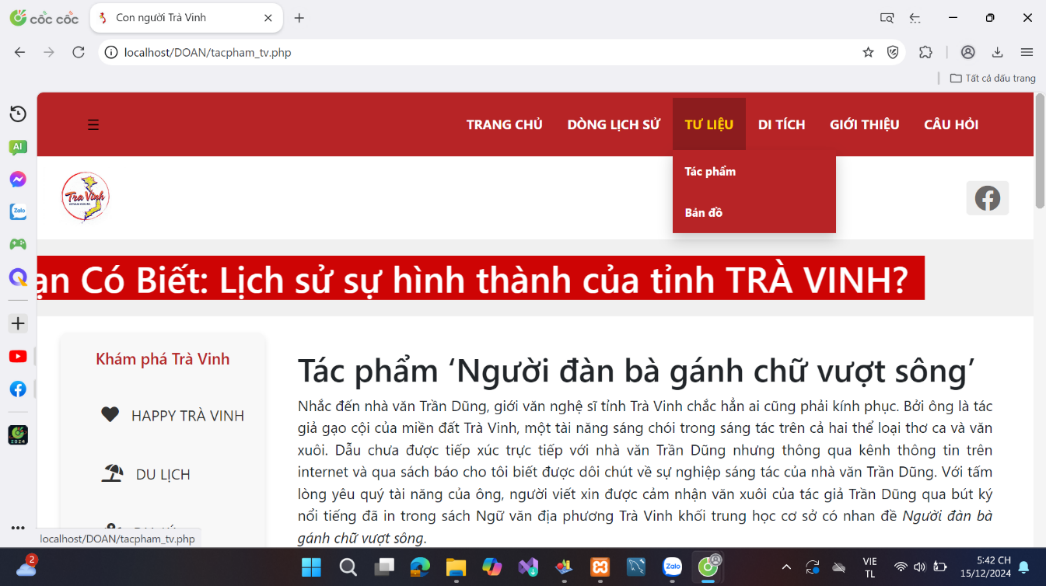
## Giao diện chức năng dòng lịch sử



Hình 4.3 Hình giao diện chức năng dòng lịch sử

Chức năng dòng lịch sử sẽ giúp cho người dùng khám phá được từ sự hình thành và quá trình đấu tranh chống giặc, phát triển cho đến hiện tại vô cùng chi tiết. Tích hợp thêm mục lục những mốc thời gian có thể giúp người dùng hiểu được những mốc thời gian quan trọng và dễ dàng trong việc tìm kiếm thông tin.

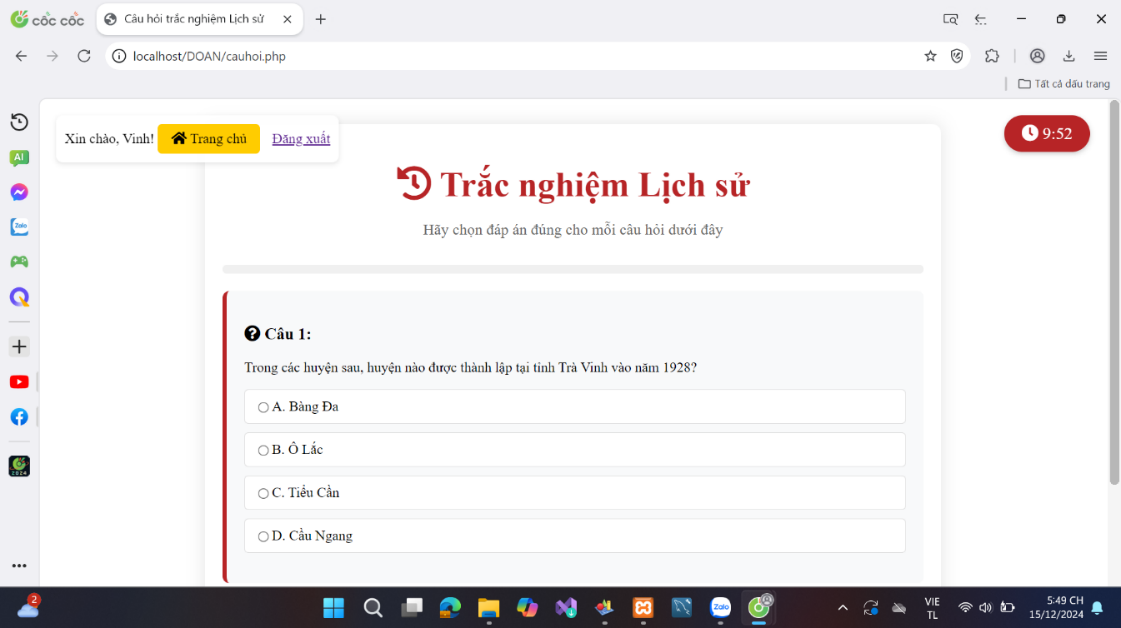
## Giao diện chức năng tư liệu



Hình 4.4 Hình giao diện chức năng tư liệu

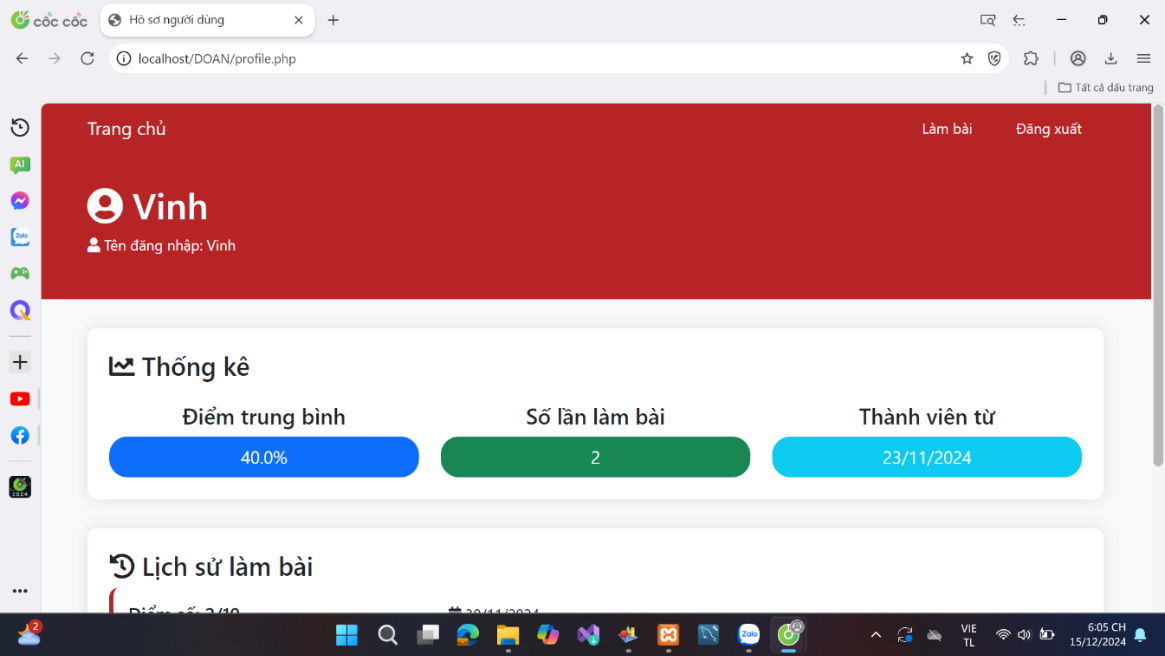
Chức năng tư liệu là chức năng khá thú vị, chức năng sẽ giúp cho người dùng biết thêm về các tư liệu văn học, lịch sử, v.v. Có cả bản đồ toàn bộ tỉnh Trà Vinh có thể giúp người dùng thuận tiện trong việc đi lại và tham quan các điểm du lịch hoặc di sản văn hóa.

## Giao diện chức năng câu hỏi.



Hình 4.5 Hình giao diện chức năng câu hỏi

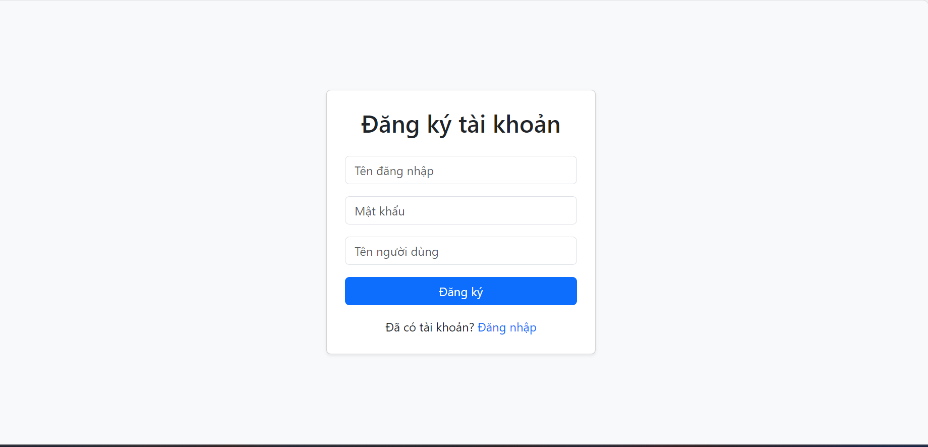
Chức năng câu hỏi là phần quan trọng của website, chức năng giúp cho người dùng có thể kiểm tra kiến thức thông qua các câu hỏi trắc nghiệm về lịch sử, văn hóa, du lịch, của tỉnh Trà Vinh. Ngoài ra, chức năng có thể cũng cố thêm kiến thức cho người dùng thông qua các câu hỏi để người dùng có thể biết thêm thông tin về tỉnh Trà Vinh hơn.



Hình 4.6 Hình giao diện người dùng

Điểm số của toàn bộ câu hỏi mà người dùng thực hiện sẽ được lưu tại đây kèm với thời gian của bài làm.

## Giao diện chức năng đăng ký



Hình 4.7 Hình giao diện chức năng đăng nhập / đăng ký

Giao diện được thiết kế đơn giản, thân thiện với người. Khi người dùng đăng ký thành công trang web sẽ tự động chuyển đến trang đăng nhặp, khi người dùng đã có tại khoản có thể sử dụng các chức năng như trả lời câu hỏi và tải bài viết lên trang web.

## Giao diện chức năng đăng tải bài viết

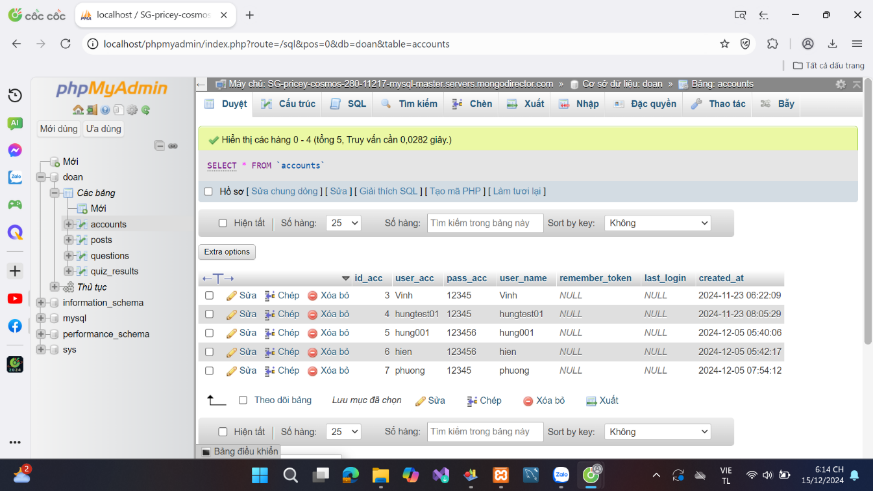


Hình 4.8 Hình giao diện chức năng đăng tải bài viết

Ở đây, người dùng có thể đăng tải các bài viết về Trà Vinh với phần tiêu đề, nội dung bài viết, và có cả phần tải ảnh để minh họa nội dung. Giúp cho các người dùng có thể tương tác với nhau.

## Cơ sơ dữ liệu

### Cơ sở dữ liệu của hệ thống

Giao diện được sử dụng bởi MySQL trong Xampp được nhà cung cấp tích hợp sẵn, được cung cấp đầy đủ các chức năng từ truy suất dữ liệu đến các chức năng cở bản của một quản trị viên như thêm, xóa. sửa, duyệt các bài viết của người dùng.

Hình 4.9 Hình giao diện chức năng quản trị được cung cấp bởi Xampp

## Kết chương

Qua việc triển khai và phân tích các chức năng của website **Giới thiệu Lịch sử và Di sản Văn hóa Tỉnh Trà Vinh**, chúng ta có thể nhận thấy rằng mỗi chức năng đều đóng vai trò quan trọng trong việc tạo ra một trải nghiệm người dùng mượt mà và trực quan. Các chức năng này không chỉ giúp người dùng dễ dàng khám phá thông tin về di tích, lịch sử, văn hóa, mà còn tối ưu hóa sự tương tác với nội dung của website.

Nhờ vào các chức năng được thiết kế một cách logic và thuận tiện, website sẽ mang đến cho người dùng một trải nghiệm tìm hiểu về Trà Vinh thú vị và dễ dàng. Các chức năng này không chỉ giúp người dùng tiếp cận thông tin một cách nhanh chóng, mà còn tạo ra sự tương tác trực quan và sinh động. Đồng thời, việc thiết kế giao diện responsive đảm bảo rằng người dùng trên mọi nền tảng đều có thể truy cập và sử dụng website một cách tối ưu.

Chúng ta có thể thấy rằng mỗi chức năng đều hỗ trợ mục tiêu chính của website, là giới thiệu và bảo tồn các giá trị lịch sử, văn hóa, và di sản của tỉnh Trà Vinh, góp phần nâng cao nhận thức cộng đồng và thúc đẩy ngành du lịch của tỉnh. Các chức năng này cũng sẽ tiếp tục được cải tiến và phát triển trong tương lai để đáp ứng nhu cầu ngày càng cao của người dùng.

# Kết luận và hướng phát triển

## Kết luận

Website Giới thiệu Lịch sử và Di sản Văn hóa Tỉnh Trà Vinh được thiết kế và phát triển với mục tiêu chính là cung cấp một nền tảng trực tuyến để giới thiệu về lịch sử, văn hóa, và các di tích quan trọng của tỉnh Trà Vinh. Với các chức năng chủ yếu như thanh điều hướng đơn giản, menu bên dễ sử dụng, các hiệu ứng động sinh động, và tính năng tìm kiếm linh hoạt, website đã đạt được những mục tiêu ban đầu về việc truyền tải thông tin một cách mạch lạc và dễ tiếp cận.

Các chức năng tương tác của website không chỉ giúp người dùng nhanh chóng tìm kiếm thông tin về các di sản văn hóa, lịch sử mà còn tạo ra một trải nghiệm người dùng thú vị, trực quan. Ngoài ra, thiết kế giao diện tối giản nhưng hiện đại, kết hợp với các hiệu ứng động và tương thích trên nhiều thiết bị, giúp website thu hút người dùng, từ các du khách đến các nhà nghiên cứu, học giả.

Tuy nhiên, việc phát triển một website như vậy là một quá trình liên tục, và mặc dù đã có sự tích hợp tốt về các chức năng cơ bản, vẫn còn những cơ hội để nâng cấp và cải thiện để website trở nên toàn diện hơn nữa.

## Hướng phát triển

**Mở rộng và cập nhật nội dung**

* + Thêm các chuyên mục mới: Có thể mở rộng các chuyên mục về văn hóa địa phương, các lễ hội truyền thống, ẩm thực đặc trưng, hoặc các chương trình sự kiện đặc biệt của Trà Vinh để thu hút thêm người dùng.
  + Cập nhật thường xuyên: Đảm bảo rằng các thông tin về di sản, di tích lịch sử và các địa điểm du lịch luôn được cập nhật kịp thời, chính xác. Việc này không chỉ cung cấp giá trị cho người dùng mà còn duy trì sự tương tác của họ với website.

**Tích hợp công nghệ mới**

* + Hệ thống quản lý nội dung (CMS) mạnh mẽ hơn: Một hệ thống CMS nâng cao sẽ giúp quản trị viên dễ dàng hơn trong việc cập nhật và quản lý nội dung, đồng thời cải thiện khả năng tối ưu hóa SEO để website có thể tiếp cận được nhiều người dùng hơn.

**Phát triển tính năng tương tác**

* Hệ thống đánh giá và phản hồi: Thêm tính năng cho phép người dùng để lại đánh giá, phản hồi về các địa điểm du lịch, các di tích, giúp người dùng khác có thêm thông tin và lựa chọn khi đến Trà Vinh.
* Cộng đồng người dùng: Xây dựng một diễn đàn hoặc một hệ thống blog nơi người dùng có thể chia sẻ kinh nghiệm, thông tin về các chuyến du lịch, các hoạt động văn hóa tại Trà Vinh.

**Tối ưu hóa trải nghiệm người dùng (UX/UI)**

* Tối ưu hóa tốc độ tải trang: Dù website đã được tối ưu hóa cho các thiết bị di động, nhưng tốc độ tải trang luôn là yếu tố quan trọng. Việc cải thiện tốc độ tải trang sẽ nâng cao trải nghiệm người dùng và làm cho website trở nên hấp dẫn hơn.
* Tính năng cá nhân hóa: Website có thể áp dụng các tính năng cá nhân hóa, như đề xuất các bài viết hoặc địa điểm du lịch dựa trên sở thích của người dùng, giúp người dùng cảm thấy website đáp ứng nhu cầu của họ tốt hơn.

Website Giới thiệu Lịch sử và Di sản Văn hóa Tỉnh Trà Vinh đã hoàn thành mục tiêu cung cấp một nền tảng trực tuyến để giới thiệu và bảo tồn các giá trị văn hóa và lịch sử quan trọng của tỉnh. Tuy nhiên, để website phát triển và duy trì sự hấp dẫn lâu dài, việc tiếp tục cải tiến các chức năng, tích hợp công nghệ mới và mở rộng nội dung là rất quan trọng. Các hướng phát triển trên sẽ giúp website không chỉ phục vụ nhu cầu thông tin mà còn thúc đẩy du lịch, giáo dục và cộng đồng nghiên cứu về Trà Vinh trong tương lai.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1]Crockford, D. (2008). JavaScript: The Good Parts. O'Reilly Media.

[2]Duckett, J. (2011). HTML and CSS: Design and Build Websites. John Wiley & Sons.

[3]Duckett, J. (2011). PHP & MySQL: Server-side Web Development. John Wiley & Sons.

[4]Haverbeke, M. (2018). Eloquent JavaScript (3rd ed.). No Starch Press

.

[5]Hogan, B. P. (2011). HTML5 and CSS3: Level Up with Today's Web Technologies. Pragmatic Bookshelf.

[6]Lockhart, J. (2015). Modern PHP: New Features and Good Practices. O'Reilly Media.

[7]Meyer, E. A. (2006). CSS: The Definitive Guide. O'Reilly Media.

[8]Simpson, K. (2014-2016). You Don’t Know JS (series). O'Reilly Media.

[9]W3Schools. (n.d.). Retrieved from <https://www.w3schools.com/> ngày truy cập 21/12/2024