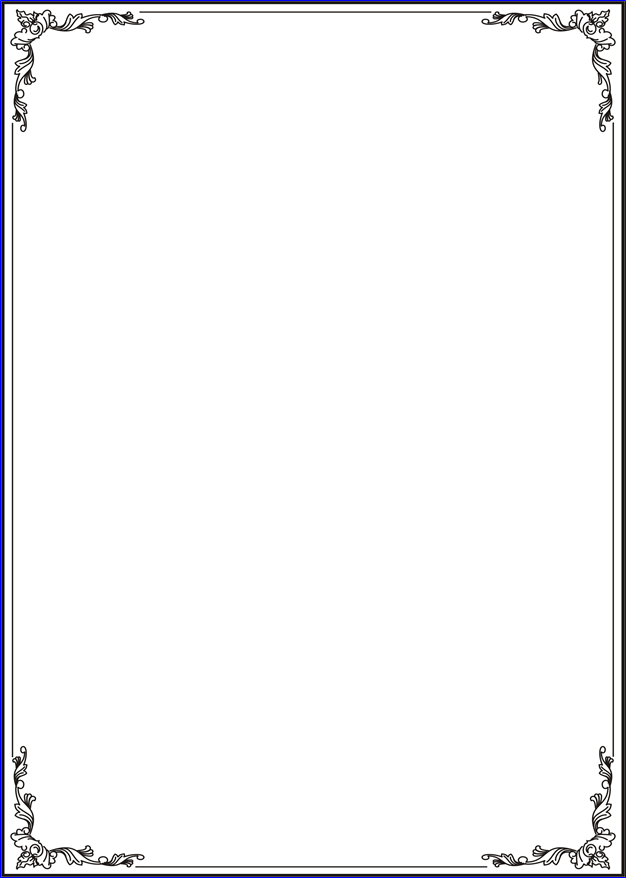
** BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**-----------------------**

A yellow and red logo

Description automatically generated

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH KHOA HỌC MÁY TÍNH**

**XÂY DỰNG WEBSITE QUẢN LÝ BÁN HÀNG LIPUMA**

**CBHD: Th.s Nguyễn Tuấn Tú**

**Sinh viên: Nguyễn Thị Huyền**

**Mã số sinh viên**: **2020603068**

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN 11](#_Toc167328966)

[1.1. Tổng quan 11](#_Toc167328967)

[1.2. Phân tích thiết kế hướng đội tượng bằng UML 11](#_Toc167328968)

[1.2.1. UML là gì 12](#_Toc167328969)

[1.2.2. UML là ngôn ngữ 12](#_Toc167328970)

[1.2.3. UML là ngôn ngữ hiển thị 12](#_Toc167328971)

[1.2.4. UML là ngôn ngữ đặc tả 13](#_Toc167328972)

[1.2.5. UML là ngôn ngữ xây dựng 13](#_Toc167328973)

[1.2.6. UML là ngôn ngữ tài liệu 13](#_Toc167328974)

[1.2.7. Một số biểu đồ 13](#_Toc167328975)

[1.3. Giới thiệu về HTML và CSS 14](#_Toc167328976)

[1.3.1. HTML 14](#_Toc167328977)

[1.3.2. CSS 20](#_Toc167328978)

[1.3.3. Php 23](#_Toc167328979)

[1.4. Kiến trúc MVC 26](#_Toc167328980)

[1.4.1. MVC là gì 26](#_Toc167328981)

[1.4.2. Đặc điểm của MVC 26](#_Toc167328982)

[1.4.3. Các thành phần trong mô hình mvc 27](#_Toc167328983)

[1.4.4. Chức năng của MVC 28](#_Toc167328984)

[1.4.5. MVC hoạt động như thế nào 29](#_Toc167328985)

[1.4.6. Vì sao nên sử dụng mô hình MVC? 30](#_Toc167328986)

[1.4.7. Nhược điểm của MVC 33](#_Toc167328987)

[1.4.8. Kết luận 37](#_Toc167328988)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 38](#_Toc167328989)

[2.1 Khảo sát hiện trạng 38](#_Toc167328990)

[2.1.1. Những câu hỏi khảo sát 38](#_Toc167328991)

[2.1.2. Tại sao phải xây dựng website bán giày 39](#_Toc167328992)

[2.2 Phân tích yêu cầu 39](#_Toc167328993)

[2.3.1. Yêu cầu hệ thống 40](#_Toc167328994)

[2.3.2. Phân tích hệ thống 41](#_Toc167328995)

[2.4.2. Biểu đồ lớp 54](#_Toc167328996)

[2.4.3. Biểu đồ tuần tự 54](#_Toc167328997)

[2.4.4. Biểu đồ triển khai 86](#_Toc167328998)

[CHƯƠNG 3: KẾT QUẢ XÂY DỰNG CHƯƠNG TRÌNH 87](#_Toc167328999)

[3.1. Màn hình trang chủ 87](#_Toc167329000)

[3.2. Màn hình giới thiệu 88](#_Toc167329001)

[3.3. Màn hình tất cả sản phẩm 88](#_Toc167329002)

[3.4. Màn hình tin tức 89](#_Toc167329003)

[3.5. Màn hình liên hệ hỗ trợ 90](#_Toc167329004)

[3.6. Màn hình giỏ hàng 90](#_Toc167329005)

[3.7. Màn hình đăng nhập người dùng 91](#_Toc167329006)

[3.8. Màn hình đăng nhập admin 92](#_Toc167329007)

[3.9. Màn hình thêm danh mục bài viết 93](#_Toc167329008)

[3.10. Màn hình thêm danh mục sản phẩm 94](#_Toc167329009)

[3.11. Màn hình liệt kê sản phẩm 95](#_Toc167329010)

[3.12. Màn hình thống kê doanh thu hàng tháng 95](#_Toc167329011)

[KẾT LUẬN 96](#_Toc167329012)

[TÀI LIỆU THAM KHẢO 98](#_Toc167329013)

**DANH MỤC HÌNH ẢNH**

[Hình 1.2: Các trình duyệt web được sử dụng 16](#_Toc167496732)

[Hình 1.3: Trang web đọc tài liệu HTML 17](#_Toc167496733)

[Hình 1.4: Hình ảnh trực quan của cấu trúc HTML 17](#_Toc167496734)

[Hình 1.5: Cú pháp CSS 22](#_Toc167496735)

[Hình 1.6: Mô hình kiến trúc MVC 27](#_Toc167496736)

[Hình 1.7: Mô hình khái niệm hóa MVC 29](#_Toc167496737)

[Hình 1.8: Mô hình tuyến tính MVC 30](#_Toc167496738)

[Hình 2.1: Biểu đồ UC tổng quát 50](#_Toc167496739)

[Hình 2.2: Biểu đồ UC phân rã cho chức năng quản lý giỏ hàng 50](#_Toc167496740)

[Hình 2.3: Biểu đồ UC phân rã cho chức năng quản lý danh mục bài viết 51](#_Toc167496741)

[Hình 2.4: Biểu đồ UC phân rã cho chức năng quản lý bài viết 51](#_Toc167496742)

[Hình 2.5: Biểu đồ UC phân rã cho chức năng quản lý danh mục sản phẩm 52](#_Toc167496743)

[Hình 2.6: Biểu đồ UC phân rã cho chức năng quản lý sản phẩm 53](#_Toc167496744)

[Hình 2.7: Biểu đồ UC phân rã cho chức năng quản lý đơn hàng 53](#_Toc167496745)

[Hình 2.8: Biểu đồ tuần tự cho chức năng đăng nhập 74](#_Toc167496746)

[Hình 2.9: Biểu đồ tuần tự cho chức năng xem danh mục bài viết 75](#_Toc167496747)

[Hình 2.10: Biểu đồ tuần tự cho chức năng thêm danh mục bài viết 75](#_Toc167496748)

[Hình 2.11: Biểu đồ tuần tự cho chức năng sửa danh mục bài viết 76](#_Toc167496749)

[Hình 2.12: Biểu đồ tuần tự cho chức năng xóa danh mục bài viết 76](#_Toc167496750)

[Hình 2.13: Biểu đồ tuần tự cho chức năng xem danh sách bài viết 77](#_Toc167496751)

[Hình 2.14: Biểu đồ tuần tự cho chức năng thêm bài viết 77](#_Toc167496752)

[Hình 2.15: Biểu đồ tuần tự cho chức năng sửa bài viết 78](#_Toc167496753)

[Hình 2.16: Biểu đồ tuần tự cho chức năng xóa bài viết 79](#_Toc167496754)

[Hình 2.17: Biểu đồ tuần tự cho chức năng xem danh mục sản phẩm 79](#_Toc167496755)

[Hình 2.18: Biểu đồ tuần tự cho chức năng thêm danh mục sản phẩm 80](#_Toc167496756)

[Hình 2.19: Biểu đồ tuần tự cho chức năng xóa danh mục sản phẩm 81](#_Toc167496757)

[Hình 2.20: Biểu đồ tuần tự cho chức năng xem danh sách sản phẩm 82](#_Toc167496758)

[Hình 2.21: Biểu đồ tuần tự cho chức năng thêm sản phẩm 82](#_Toc167496759)

[Hình 2.22: Biểu đồ trình tự cho chức năng sửa thông tin sản phẩm 83](#_Toc167496760)

[Hình 2.23: Biểu đồ tuần tự cho chức năng xóa sản phẩm 84](#_Toc167496761)

[Hình 2.24: Biểu đồ tuần tự cho chức năng xem đơn hàng 84](#_Toc167496762)

[Hình 2.25: Biểu đồ tuần tự cho chức năng quản lý tình trạng đơn hang 85](#_Toc167496763)

[Hình 2.26: Biểu đồ tuần tự cho chức năng gửi liên hệ 86](#_Toc167496764)

[Hình 2.27: Biểu đồ triển khai 86](#_Toc167496765)

[Hình 3.1: Màn hình trang chủ 87](#_Toc167496766)

[Hình 3.2: Màn hình trang chủ gồm các sản phẩm giày 87](#_Toc167496767)

[Hình 3.3: Màn hình giới thiệu 88](#_Toc167496768)

[Hình 3.4 :Màn hình tất cả sản phẩm 88](#_Toc167496769)

[Hình 3.5: Màn hình tin tức tất cả 89](#_Toc167496770)

[Hình 3.6: Màn hình tức tin cụ thể 89](#_Toc167496771)

[Hình 3.7: Màn hình liên hệ 90](#_Toc167496772)

[Hình 3.8: Màn hình giỏ hàng 91](#_Toc167496773)

[Hình 3.9: Màn hình giỏ hàng điền thông tin khách hàng 91](#_Toc167496774)

[Hình 3.11: Màn hình đăng nhập admin 92](#_Toc167496775)

[Hình 3.14: Màn hình thêm danh mục bài viết 93](#_Toc167496776)

[Hình 3.14: Màn hình thêm danh mục sản phẩm 94](#_Toc167496777)

[Hình 3.15: Màn hình thêm danh mục sản phẩm chi tiết 94](#_Toc167496778)

[Hình 3.16: Màn hình liệt kê sản phẩm 95](#_Toc167496779)

[Hình 3.17: Màn hình thống kê bán hàng 95](#_Toc167496780)

**DANH MỤC BẢNG BIỂU**

[Bảng 1.1: Bảng so sánh giữa MVC và MVVM 35](#_Toc167498024)

[Bảng 2.1*:* Kịch bản cho chức năng đăng nhập 54](#_Toc167498025)

[Bảng 2.2: Kịch bản cho chức năng xem danh mục bài viết 54](#_Toc167498026)

[Bảng 2.3: Kịch bản cho chức năng thêm danh mục bài viết 55](#_Toc167498027)

[Bảng 2.4: Kịch bản cho chức năng sửa danh mục bài viết 56](#_Toc167498028)

[Bảng 2.5: Kịch bản cho chức năng xóa danh mục bài viết 57](#_Toc167498029)

[Bảng 2.6: Kịch bản cho chức năng xem bài viết 57](#_Toc167498030)

[Bảng 2.7: Kịch bản cho chức năng thêm bài viết 58](#_Toc167498031)

[Bảng 2.8: Kịch bản cho chức năng sửa thông tin bài viết 59](#_Toc167498032)

[Bảng 2.9: Kịch bản cho chức năng xóa bài viết 60](#_Toc167498033)

[Bảng 2.10: Kịch bản cho chức năng xem danh mục sản phẩm 61](#_Toc167498034)

[Bảng 2.11: Kịch bản cho chức năng thêm danh mục sản phẩm 61](#_Toc167498035)

[Bảng 2.12: Kịch bản cho chức năng sửa danh mục sản phẩm 62](#_Toc167498036)

[Bảng 2.13: Kịch bản cho chức năng xóa danh mục sản phẩm 63](#_Toc167498037)

[Bảng 2.14: Kịch bản cho chức năng xem danh sách sản phẩm 64](#_Toc167498038)

[Bảng 2.15: Kịch bản cho chức năng thêm sản phẩm 65](#_Toc167498039)

[Bảng 2.16: Kịch bản cho chức năng sửa thông tin sản phẩm 65](#_Toc167498040)

[Bảng 2.17: Kịch bản cho chức năng xóa sản phẩm 66](#_Toc167498041)

[Bảng 2.18: Kịch bản cho chức năng xem đơn hàng 67](#_Toc167498042)

[Bảng 2.19: Kịch bản cho chức năng xem chi tiết đơn hàng và thay đổi trạng thái đơn hàng 68](#_Toc167498043)

[Bảng 2.20: Kịch bản cho chức năng thêm sản phẩm vào giỏ hàng 69](#_Toc167498044)

[Bảng 2.21: Kịch bản cho chức năng Sửa số lượng sản phẩm trong giỏ hàng 70](#_Toc167498045)

[Bảng 2.22: Kịch bản cho chức năng Xóa sản phẩm trong giỏ hàng 70](#_Toc167498046)

[Bảng 2.23: Kịch bản cho chức năng đặt hàng 71](#_Toc167498047)

[Bảng 2.24: Kịch bản cho chức năng gửi liên hệ 72](#_Toc167498048)

**LỜI CẢM ƠN**

Lời đầu tiên em xin chân thành cảm ơn các thầy, cô trong khoa Công nghệ thông tin, trường Đại Học Công Nghiệp Hà Nội đã tạo điều kiện thuận lợi cho em trong quá trình học tập tại trường cũng như trong thời gian thực hiện báo cáo đồ án tốt nghiệp.

Đặc biệt, em muốn gửi lời cảm ơn tới **Th.s Nguyễn Tuấn Tú** – giảng viên hướng dẫn đã chỉ bảo, giúp đỡ em khắc phục những khó khăn, thiếu sót để có thể hoàn thành bài báo cáo.

Dù đã rất cố gắng thực hiện báo cáo đồ án tốt nghiệp, nhưng do thời gian, kinh nghiệm của bản thân em còn hạn chế nên không thể tránh khỏi có thiếu sót. Kính mong nhận được sự đóng góp ý kiến từ phía thầy cô, bạn bè để em có thể nâng cao kiến thức của bản thân, phát triển tốt hơn.

Em xin chân thành cảm ơn!

*Hà Nội, năm 2024*

Sinh viên thực hiện

Huyền

Nguyễn Thị Huyền

**LỜI CAM ĐOAN**

Em xin cam đoan đề tài: *" Xây dựng Website quản lý bán hàng LIPUMA"* là một công trình nghiên cứu độc lập dưới sự hướng dẫn của giảng viên hướng dẫn: Th.s Nguyễn Tuấn Tú. Ngoài ra không có bất cứ sự sao chép của người khác. Mọi tài liệu tham khảo đều được trích dẫn cụ thể, rõ ràng.

Em cam đoan thời gian thực hiện đồ án là hoàn thành đúng những yêu cầu mà nhà trường và khoa đưa ra. Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá em xin chịu hoàn toàn trách nhiệm, kỷ luật của bộ môn và nhà trường đề ra nếu như có vấn đề xảy ra.

**LỜI NÓI ĐẦU**

Trong đồ án này, em đã nghiên cứu và triển khai một website quản lí bán hàng, một hệ thống quan trọng và hữu ích trong việc quản lí và vận hành doanh nghiệp.

Mục tiêu của đề tài này là tạo ra một website quản lí bán hàng toàn diện, giúp các doanh nghiệp có thể quản lí và điều hành hoạt động bán hàng một cách hiệu quả, tối ưu hóa quy trình kinh doanh và nâng cao sự phục vụ khách hàng. Trong quá trình nghiên cứu và thực hiện, em đã áp dụng các phương pháp phát triển phần mềm để tạo ra web bán hàng.

Trang web quản lí bán hàng mà em đã xây dựng bao gồm các tính năng tương thích hiển thị trên các thiết bị và có thể nâng cấp trong tương lai, hỗ trợ khách hàng một cách nhanh nhất khi được yêu cầu liên hệ lại.

Trong quá trình thực hiện đồ án, em đã đối mặt với những thách thức và khó khăn. Tuy nhiên, thông qua việc nghiên cứu sâu và sự nỗ lực không ngừng, tôi đã vượt qua các khó khăn đó và đạt được kết quả như mong đợi. Em hy vọng rằng đồ án này sẽ mang lại giá trị và đóng góp cho lĩnh vực quản lí bán hàng và phát triển doanh nghiệp.

# CHƯƠNG 1: TỔNG QUAN

## Tổng quan

Website (còn gọi là trang web) (có thể nhầm lẫn với "web page") hoặc trang mạng, là một tập hợp trang web, thường chỉ nằm trong một tên miền hoặc tên miền phụ trên World Wide Web của Internet. Một trang web là tập tin HTML hoặc XHTML có thể truy nhập dùng giao thức HTTP. Trang mạng có thể được xây dựng từ các tệp tin HTML (trang mạng tĩnh) hoặc vận hành bằng các CMS chạy trên máy chủ (trang mạng động).

Website tĩnh: là website mà người quản trị (những người không phải là lập trình viên) không thể tùy ý thay đổi nội dung và hình ảnh mà phải cần kiến thức về HTML cơ bản. Website tĩnh được viết hoàn toàn dựa trên nền tảng HTML CSS và thêm các hiệu ứng từ JavaScript nếu muốn.

Website động: là website được viết kèm theo một bộ công cụ quản trị để tùy biến nội dung thành cho webmaster (người quản trị) có thể dễ dàng thay đổi nội dung, hình ảnh. Website động được thiết kế bởi các lập trình viên để làm sao cho phép website có thể thay đổi được nội dung thường xuyên. Một số công nghệ, ngôn ngữ để xây dựng website động bao gồm PHP, ASP.NET, Java,…

## Phân tích thiết kế hướng đội tượng bằng UML

StarUML là một công cụ UML của MKLab. Phần mềm được cấp phép theo phiên bản sửa đổi của GNU GPL cho đến năm 2014, khi phiên bản viết lại 2.0.0 được phát hành để thử nghiệm beta theo giấy phép độc quyền. Sau khi bị bỏ rơi một thời gian, dự án đã hồi sinh để chuyển từ Delphi sang Java / Eclipse và sau đó dừng lại.

StarUML là một mô hình nền tảng, là phần mềm hỗ trợ UML (Unified Modeling Language). Nó hỗ trợ các phương pháp tiếp cận MDA (Model Driven Architecture) bằng cách hỗ trợ các khái niệm hồ sơ UML. ... Ngoài ra dùng StarUML sẽ đảm bảo tối đa hóa năng suất và chất lượng của các dự án phần mềm của bạn.

Vì nó cho phép mô hình hóa nên sẽ không phụ thuộc vào người code, ngôn ngữ code hay nền tảng sử dụng. Do mô hình hóa nên rất dễ đọc và dễ hiểu. Một điều mình thích nhất ở StarUML là khả năng tự sinh code từ những mô hình class.

### ***UML là gì***

UML là ngôn ngữ chuẩn để viết kế hoạch chi tiết phần mềm. Nó phù hợp cho việc mô hình hóa các hệ thống như hệ thông tin doanh nghiệp, các ứng dụng phân tán trên nền Web, hệ thống nhúng thời gian thực… UML là ngôn ngữ mô hình được cả con người và máy sử dụng. Phương pháp là cách cấu trúc rõ ràng suy nghĩ và hành động của ai đó.Cho người sử dụng biết làm cái gì, làm như thế nào, khi nào làm việc đó và tại sao lại làm như vậy. Phương pháp chứa mô hình và các mô hình này được sử dụng để mô tả cái gì đó.

### ***UML là ngôn ngữ***

Ngôn ngữ phải có từ vựng và quy tắc tổ hợp các từ trong từ vựng để giao tiếp. Ngôn ngữ mô hình là ngôn ngữ có từ vựng và quy tắc tập trung vào biểu diễn về mặt vật lý và khái niệm của hệ thống. UML là ngôn ngữ chuẩn công nghiệp để lập kế hoạch chi tiết phần mềm.

### ***UML là ngôn ngữ hiển thị***

Với nhiều lập trình viên, không có khoảng cách từ ý tưởng đến cài đặt mã trình thực tế là có thể trực tiếp viết mã trình cho một số việc, trực tiếp mô tả thuật toán và biểu thức bằng văn bản.UML là ngôn ngữ đồ họa. Với nhiều hệ thống, mô hình trong ngôn ngữ đồ họa dễ hiểu hơn so với ngôn ngữ lập trình. Sau mỗi biểu tượng đồ họa của ngôn ngữ UML là ngữ nghĩa. Vậy, khi xây dựng mô hình trong UML thì người phát triển khác hay các công cụ hỗ trợ mô hình hóa có thể hiểu mô hình một cách rõ ràng.

### 1***.2.4. UML là ngôn ngữ đặc tả***

Đặc tả là mô tả rõ ràng những điểm mấu chốt của vấn đề. UML cho phép mô tả mô hình chính xác, không nhập nhằng và hoàn thiện. UML hướng tới đặc tả thiết kế, phân tích và quyết định cái đặt trong quá trình phát triển và triển khai hệ thống phần mềm.

### ***1.2.5.*** ***UML là ngôn ngữ xây dựng***

UML là không phải là ngôn ngữ lập trình trực quan, nhưng mô hình của nó có thể kết nối trực tiếp tới các ngôn ngữ lập trình khác nhau. Có nghĩa rằng có thể ánh xạ mô hình trong UML tới các ngôn ngữ lập trình khác nhau như Java, C++ hay các bảng CSDL quan hệ, CSDL hướng đối tượng. Ánh xạ này cho khả năng biến đổi thuận từ mô hình UML sang ngôn ngữ lập trình. Đồng thời, cho khả năng biến đổi ngược lại từ cài đặt về mô hình UML.

### ***1.2.6. UML là ngôn ngữ tài liệu***

UML hướng tới làm tài liệu kiến trúc hệ thống và các chi tiết của nó. UML cho khả năng biểu diễn yêu cầu, thử nghiệm, mô hình hóa các hoạt động lập kế hoạch và quản lý sản phẩm. UML cho biết giới hạn của hệ thống và các chức năng chính của nó thông qua UC và tác nhân.

### ***1.2.7. Một số biểu đồ***

Biểu đồ (diagram) là các hình vẽ bao gồm các ký hiệu phần tử mô hình hóa được sắp xếp để minh họa một thành phần cụ thể hay một khía cạnh cụ thể của hệ thống. Một mô hình hệ thống thường có nhiều loại biểu đồ, mỗi loại có nhiều biểu đồ khác nhau. Một biểu đồ là một thành phần của một hướng nhìn cụ thể; và khi được vẽ ra, nó thường thường cũng được xếp vào một hướng nhìn.

Biểu đồ Use case (Use Case Diagram): chỉ ra một số lượng các tác nhân ngoại cảnh và mối liên kết của chúng đối với Use case mà hệ thống cung cấp.

Biểu đồ trình tự (Sequence Diagram): chỉ ra trình tự các thông điệp (message) được gửi giữa các đối tượng. Nó cũng chỉ ra trình tự tương tác giữa các đối tượng, điều sẽ xảy ra tại một thời điểm cụ thể nào đó trong trình tự thực thi của hệ thống. Các biểu đồ trình tự chứa một loạt các đối tượng được biểu diễn bằng các đường thẳng đứng.

Biểu đồ hoạt động (Activity Diagram): Để nắm bắt công việc (hành động) sẽ phải được thực thi khi một thủ tục được thực hiện. Đây là tác dụng thường gặp nhất và quan trọng nhất của biểu đồ hoạt động. Để nắm bắt công việc nội bộ trong một đối tượng.Để chỉ ra một nhóm hành động liên quan có thể được thực thi ra sao, và chúng sẽ ảnh hưởng đến những đối tượng nằm xung quanh chúng như thế nào.Để chỉ ra một trường hợp sử dụng có thể được thực thể hóa như thế nào, theo khái niệm hành động và các sự biến đổi trạng thái của đối tượng.

## Giới thiệu về HTML và CSS

### ***HTML***

- HTML là gì?

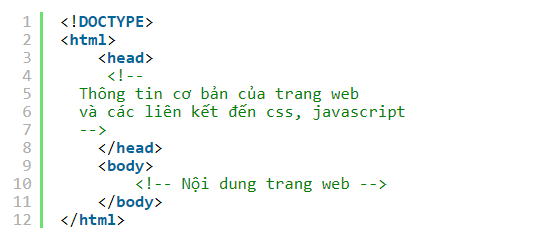
HTML là viết tắt của từ Hyper Text Markup Language, có nghĩa là ngôn ngữ đánh dấu siêu văn bản. HTML được sử dụng để định dạng và hiển thị văn bản trên trình duyệt tới người sử dụng. Nó là xương sống của một trang web. Vậy thì Hyper Text và Markup Language là gì?

Hypertext: siêu văn bản có nghĩa là "văn bản trong văn bản". Một văn bản có chứa một liên kết (link) là một siêu văn bản. Mỗi lần bản click vào một từ từ đó đưa bạn tới một trang web mới, đó là siêu văn bản.

Markup Language: ngôn ngữ đánh dấu là một ngôn ngữ lập trình được sử dụng để giúp văn bản dễ tương tác và linh động hơn. Nó có thể đặt văn bản vào trong các ảnh, bảng, liên kết...

Một tài liệu HTML chưa nhiều thẻ HTML và mỗi thẻ HTML chứa các nội dung khác nhau.

Ví dụ, một trang web chuẩn sẽ chứa các thẻ HTML như sau:



Hình 1.1: File HTML

- Các đặc điểm của HTML

* Đây là một ngôn ngữ rất dễ dàng và đơn giản.
* Rất dễ dàng để trình bày hiệu quả với HTML vì nó có nhiều thẻ định dạng.
* Đây là một ngôn ngữ đánh dấu vì vậy có thể sử dụng nó một cách linh hoạt để thiết kế trang web cùng với văn bản.
* Có thể liên kết đến các trang web khác.
* Là một nền tảng độc lập vì nó có thể hiển thị trên bất kỳ nền tảng nào khác như Windows, Linux và Max.
* Có thể thêm các hình ảnh, video, âm thanh vào các trang web khiến nó hấp dẫn và dễ tương tác hơn.

- Trình soạn thảo HTML

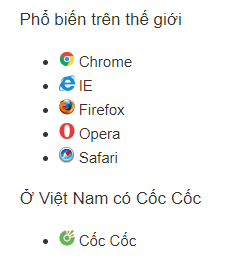
Visual Studio Code. Vậy Visual Studio Code là gì?

Visual Studio Code là một trong những trình soạn thảo mã nguồn mở rất phổ biến được các lập trình viên sử dụng. Với các ưu điểm nổi bật là sự nhanh chóng, nhẹ, hỗ trợ đa nền tảng cùng nhiều tính năng và là mã nguồn mở chính. Visual Studio Code ngày càng được ưa sử dụng, là lựa chọn hàng đầu của các lập trình viên

\* Những ưu điểm của Visual Studio Code:

* Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng HTML, CSS, JavaScript, C++,...
* Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung
* Các tiện ích mở rộng rất đa dạng và phong phú
* Tích hợp các tính năng quan trọng như tính năng bảo mật (Git), khả năng tăng tốc độ xử lý vòng lặp (Debug),…
* Đơn giản hóa việc tìm quản lý hết tất cả các Code trên hệ thống

\*Các trình duyệt web được sử dụng

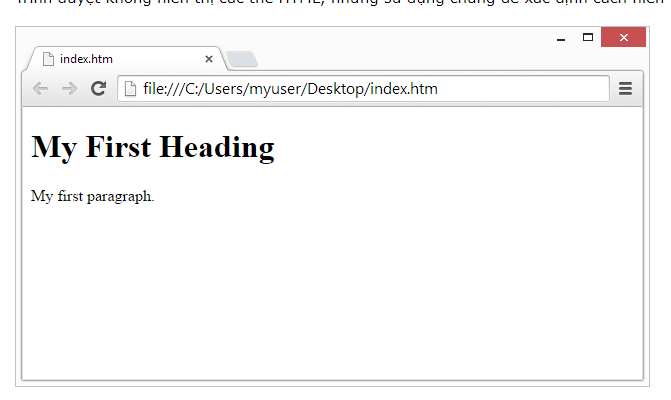


Hình 1.2: Các trình duyệt web được sử dụng

\*Trình duyệt web

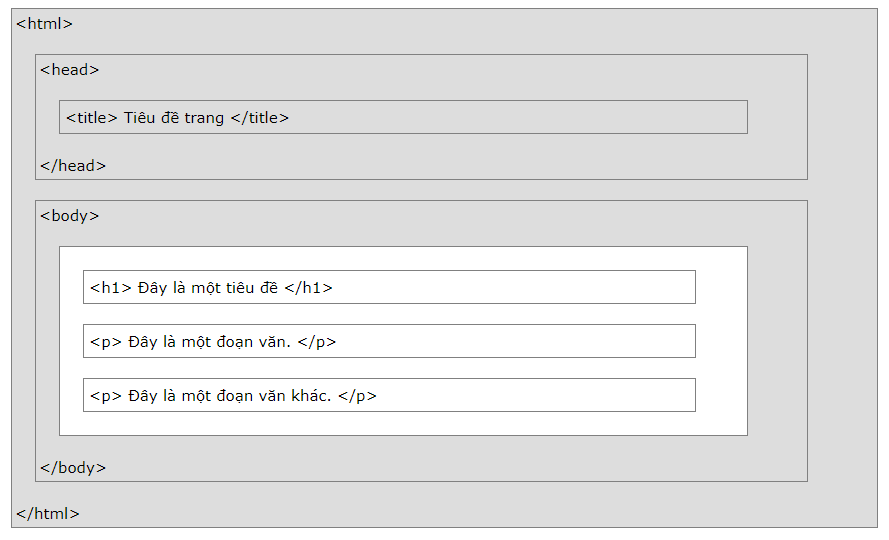
Mục đích của trình duyệt web (Chrome, Edge, Firefox, Safari) là đọc các tài liệu HTML và hiển thị chúng một cách chính xác.

Ví dụ:



Hình 1.3: Trang web đọc tài liệu HTML

\*Cấu trúc trang HTML



Hình 1.4: Hình ảnh trực quan của cấu trúc HTML

- HTML cơ bản

\*Tài liệu HTML

Tất cả các tài liệu HTML phải bắt đầu với một tuyên bố loại tài liệu: <!DOCTYPE html>.

Bản thân tài liệu HTML bắt đầu bằng <html>và kết thúc bằng </html>.

Phần hiển thị của tài liệu HTML nằm giữa <body>và </body>.

\*Khai báo <! DOCTYPE>

* Các <!DOCTYPE>tuyên bố đại diện cho các loại tài liệu, và giúp các trình duyệt để trang web hiển thị một cách chính xác.
* Nó chỉ được xuất hiện một lần, ở đầu trang (trước bất kỳ thẻ HTML nào).
* Các <!DOCTYPE>tuyên bố không phải là trường hợp nhạy cảm.
* Các <!DOCTYPE>tuyên bố cho HTML5 là:

\*Tiêu đề HTML

* Các tiêu đề HTML được xác định bằng các thẻ <h1>to <h6>.
* <h1>xác định tiêu đề quan trọng nhất. <h6>xác định tiêu đề ít quan trọng nhất:

\*Các đoạn văn của HTML được khai báo bằng thẻ <p>

\*Các liên kết trong HTML được khai báo bằng thẻ <a>

\* Hình ảnh HTML được xác định bằng thẻ <img>.

* Tệp nguồn ( src), văn bản thay thế ( alt) width và height được cung cấp dưới dạng các thuộc tính:

**Các thuộc tính của HTML**

*Thuộc tính href*

Các <a>thẻ định nghĩa một siêu liên kết. Các href thuộc tính xác định URL của trang liên kết đi vào:

*Thuộc tính src*

Các <img>thẻ được sử dụng để nhúng một hình ảnh trong một trang HTML. Các src quy định cụ thể thuộc tính đường dẫn đến hình ảnh sẽ được hiển thị:

*Thuộc tính chiều rộng và chiều cao*

Các <img>thẻ cũng nên chứa width và height các thuộc tính, xác định chiều rộng và chiều cao của hình ảnh (tính bằng pixel):

*Thuộc tính alt*

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

*Thuộc tính kiểu*

Các style thuộc tính được sử dụng để thêm phong cách cho một yếu tố, chẳng hạn như màu sắc, font chữ, kích thước, và nhiều hơn nữa.

*Thuộc tính lang*

Bạn phải luôn bao gồm lang thuộc tính bên trong <html>thẻ, để khai báo ngôn ngữ của trang Web. Điều này có nghĩa là để hỗ trợ các công cụ tìm kiếm và trình duyệt.

*Thuộc tính tiêu đề*

Các title thuộc tính định nghĩa một số thông tin thêm về một phần tử.

Giá trị của thuộc tính title sẽ được hiển thị dưới dạng chú giải công cụ khi bạn di chuột qua phần tử:

Báo giá đơn hay báo giá kép?

Dấu ngoặc kép xung quanh các giá trị thuộc tính là cách phổ biến nhất trong HTML, nhưng dấu ngoặc kép cũng có thể được sử dụng.

Trong một số trường hợp, khi bản thân giá trị thuộc tính chứa dấu ngoặc kép, cần sử dụng dấu ngoặc kép:

- Định dạng

* 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 chỉ số
* <sup> - Văn bản siêu cấp

- Màu: Màu HTML được chỉ định bằng tên màu xác định trước hoặc với các giá trị RGB, HEX, HSL, RGBA hoặc HSLA.

- Liên kết HTML

Các liên kết được tìm thấy trong gần như tất cả các trang web. Liên kết cho phép người dùng nhấp theo cách của họ từ trang này sang trang khác.

### ***CSS***

- CSS là gì?

CSS là ngôn ngữ chúng ta sử dụng để tạo kiểu cho một trang Web.

* CSS là viết tắt của Cascading Style Sheets
* CSS mô tả cách các phần tử HTML được hiển thị trên màn hình, giấy hoặc trong các phương tiện khác
* CSS tiết kiệm rất nhiều công việc. Nó có thể kiểm soát bố cục của nhiều trang web cùng một lúc
* Các bảng định kiểu bên ngoài được lưu trữ trong các tệp CSS

\*Tại sao sử dụng CSS?

CSS được sử dụng để xác định kiểu cho các trang web của bạn, bao gồm thiết kế, bố cục và các biến thể hiển thị cho các thiết bị và kích thước màn hình khác nhau.

\*CSS đã giải quyết một vấn đề lớn

HTML KHÔNG BAO GIỜ có ý định chứa các thẻ để định dạng một trang web!

HTML được tạo ra để mô tả nội dung của một trang web, như:

<h1> Đây là một tiêu đề </h1>

<p> Đây là một đoạn văn. </p>

Khi các thẻ như <font> và các thuộc tính màu được thêm vào đặc tả HTML, nó bắt đầu một cơn ác mộng đối với các nhà phát triển web. Việc phát triển các trang web lớn, nơi các phông chữ và thông tin màu sắc được thêm vào từng trang, đã trở thành một quá trình lâu dài và tốn kém. Để giải quyết vấn đề này, World Wide Web Consortium (W3C) đã tạo CSS.

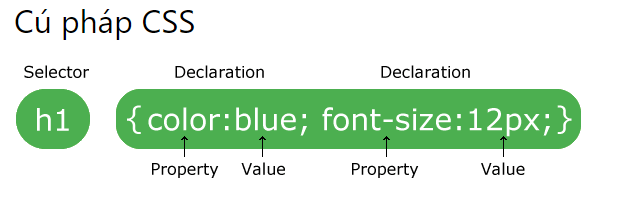
CSS đã xóa định dạng kiểu khỏi trang HTML!

\*CSS Tiết kiệm rất nhiều công việc!

Các định nghĩa kiểu thường được lưu trong các tệp .css bên ngoài.

Với tệp biểu định kiểu bên ngoài, bạn có thể thay đổi giao diện của toàn bộ trang web chỉ bằng cách thay đổi một tệp!

Cú pháp CSS



Hình 1.5: Cú pháp CSS

Bộ chọn trỏ đến phần tử HTML mà bạn muốn tạo kiểu.

Khối khai báo chứa một hoặc nhiều khai báo được phân tách bằng dấu chấm phẩy.

Mỗi khai báo bao gồm một tên thuộc tính CSS và một giá trị, được phân tách bằng dấu hai chấm.

Nhiều khai báo CSS được phân tách bằng dấu chấm phẩy và các khối khai báo được bao quanh bởi dấu ngoặc nhọn.

- Chèn CSS

Có 3 cách:

* CSS bên ngoài
* CSS nội bộ
* CSS nội tuyến

- Nhận xét CSS

* Nhận xét được sử dụng để giải thích mã và có thể hữu ích khi bạn chỉnh sửa mã nguồn vào một ngày nào đó.
* Các bình luận bị trình duyệt bỏ qua
* Nhận xét CSS được đặt bên trong <style> phần tử và bắt đầu bằng /\* và kết thúc bằng \*/

- Các cú pháp CSS cơ bản

* Màu nền:
* Màu văn bản
* Màu viền
* Đường viền
* Lề
* Khoảng cách
* Chiều rộng, chiều cao
* Biểu tượng
* Liên kết
* Lưới

### ***Php***

- PHP là gì?

PHP: Hypertext Preprocessor, thường được viết tắt thành PHP là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó rất thích hợp với web và có thể dễ dàng nhúng vào trang HTML. Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

- Lịch sử phát triển

PHP/FI

PHP được phát triển từ một sản phẩm có tên là  PHP/FI do  Rasmus Lerdorf tạo ra năm 1994 ban đầu được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của ông trên mạng. Ông đã đặt tên cho bộ mã kịch bản này là 'Personal Home Page Tools'. Khi cần đến các chức năng rộng hơn, Rasmus đã viết ra một bộ thực thi bằng C lớn hơn để có thể truy vấn tới các cơ sở dữ liệu và giúp cho người sử dụng phát triển các ứng dụng web đơn giản. Rasmus đã quyết định công bố mã nguồn của PHP/FI cho mọi người xem, sử dụng cũng như sửa các lỗi có trong nó đồng thời cải tiến mã nguồn.

PHP 7

Với việc sử dụng bộ nhân Zend Engine mới PHPNG cho tốc độ nhanh gấp 2 lần. Ngoài ra ở phiên bản này còn thêm vào rất nhiều cú pháp, tính năng mới giúp cho PHP trở nên mạnh mẽ hơn. Những tính năng mới quan trọng có thể kể đến như:

* Khai báo kiểu dữ liệu cho biến.
* Xác định kiểu dữ liệu sẽ trả về cho 1 hàm.
* Thêm các toán tử mới (??, <=>,...)

- Cú pháp

PHP chỉ phân tích các đoạn mã nằm trong những dấu giới hạn của nó. Bất cứ mã nào nằm ngoài những dấu giới hạn đều được xuất ra trực tiếp không thông qua xử lý bởi PHP. Các dấu giới hạn thường dùng nhất là <?php> và ?>, tương ứng với dấu giới hạn mở và đóng. Các dấu giới hạn <script language=”php”> và </script> cũng đôi khi được sử dụng. Cách viết dấu giới hạn dạng thẻ ngắn cũng có thể được dùng để thông báo bắt đầu đoạn mã PHP, là <? Hay <?=  (dấu này được sử dụng để in ra (echo) các xâu ký tự hay biến) với thẻ thông báo kết thúc đoạn mã PHP là ?> . Những thẻ này thường xuyên được sử dụng, tuy nhiên giống với những thẻ kiểu ASP  (<% hay <%= và %>), chúng không có tính di động cao bởi có thể bị vô hiệu khi cấu hình PHP. Bởi vậy, việc dùng các thẻ dạng ngăn hay các thẻ kiểu ASP không được khuyến khích. Mục đích của những dấu giới hạn này là ngăn cách mã PHP với những đoạn mã thuộc ngôn ngữ khác, gồm cả HTML. Mọi đoạn mã bên ngoài các dấu này đều bị hệ thống phân tích bỏ qua và được xuất ra một cách trực tiếp.

Các biến được xác định bằng cách thêm vào trước một dấu ($) và không cần xác định trước kiểu dữ liệu. Không giống với tên hàm và lớp, tên biến là trường hợp nhạy cảm. Cả dấu ngoặc kép (“”) và ký hiệu đánh dấu văn bản (<<EOF EOF;) đều có thể dùng để truyền xâu và giá trị biến. PHP coi xuống dòng như một khoảng trắng theo kiểu như một  ngôn ngữ dạng tự do (free-form language) (trừ khi nó nằm trong trích dẫn xâu), và các phát biểu được kết thúc bởi một dấu chấm phẩy. PHP có ba kiểu cú pháp chú thích:  /\* \*/ cho phép một đoạn chú thích tùy ý, trong khi đó // và # cho phép chú thích trong phạm vi một dòng. Phát biểu echo là một trong những lệnh của PHP cho phép xuất văn bản (vd. ra một trình duyệt web).

- Các hàm có sẵn thông dụng

* Hàm include(): đưa nội dung của một file chỉ định vào nội dung của file gọi nó.
* Hàm strlen(): Được sử dụng để trả lại chiều dài của một chuỗi.
* Hàm strpot(): được sử dụng để tìm kiếm một nhân vật / văn bản trong một chuỗi.
* Hàm phpinfo(): hiển thị chi tiết cấu hình PHP trên máy chủ
* Hàm date(): Hiển thị ngày tháng theo quy tắc đã thiết lập
* Hàm substr(): Tách một phần trong chuỗi.
* Hàm str\_word\_count(): Dùng để đếm có bao nhiêu từ trong chuỗi.
* Hàm str\_split(): Cắt các ký tự trong chuỗi và chuyển thành dạng mảng.

Hàm echo(): In dữ liệu chuỗi ra màn hình.

## Kiến trúc MVC

### ***MVC là gì***

MVC là viết tắt của Model-View-Controller. Cấu trúc Model-View-Controller (MVC) là một mẫu kiến trúc/mẫu thiết kế (design pattern) tách ứng dụng thành ba thành phần logic chính: Model, View và Controller. Mỗi thành phần kiến trúc được xây dựng để xử lý các khía cạnh phát triển cụ thể của một ứng dụng.

Cụ thể Model, View và Controller là gì trong MVC sẽ được giải thích rõ hơn ở mục tiếp theo trong bài viết này.

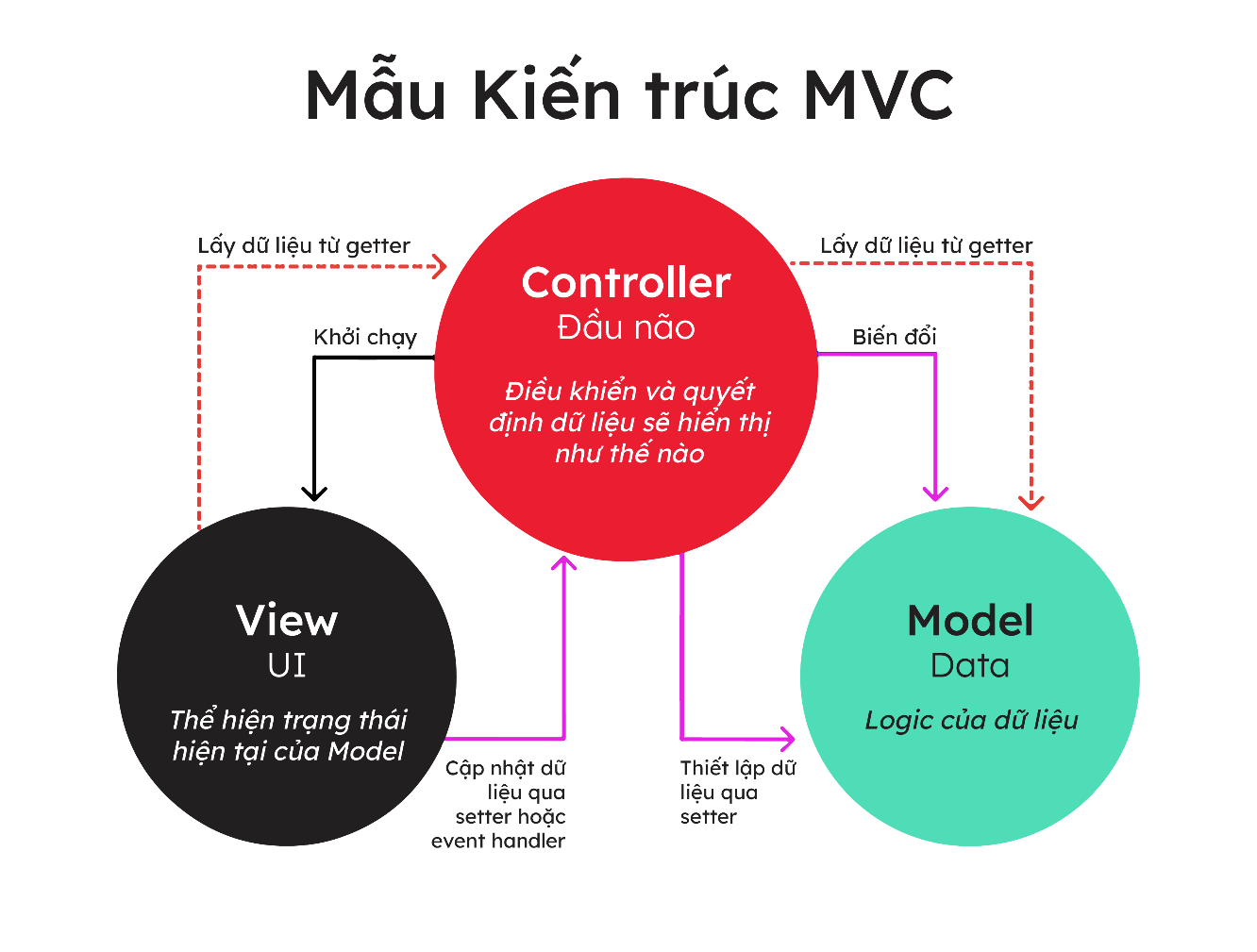
Mục tiêu chính của mẫu thiết kế này là giải quyết vấn đề người dùng kiểm soát một tập dữ liệu lớn và phức tạp bằng cách chia một ứng dụng lớn thành các phần cụ thể, tất cả đều có mục đích riêng.

### ***Đặc điểm của MVC***

* Cung cấp sự phân tách rõ ràng giữa logic nghiệp vụ, logic Ul và logic đầu vào.
* Cung cấp toàn quyền kiểm soát HTML và URL, giúp bạn dễ dàng thiết kế kiến trúc ứng dụng web.
* Có thể sử dụng để xây dựng các ứng dụng có URL dễ hiểu và có thể tìm kiếm được.
* Hỗ trợ Lập trình dựa trên thử nghiệm (Test-driven Development).

MVC là một mẫu thiết kế tiêu chuẩn được nhiều lập trình viên quen thuộc nhờ vào khả năng mở rộng và có thể mở rộng. MVC thường được sử dụng để làm framework phát triển web tiêu chuẩn cũng như các ứng dụng di động.

### ***Các thành phần trong mô hình MVC***



Hình 1.6: Mô hình kiến trúc MVC

* Model

Model là các thành phần của ứng dụng tương ứng với tất cả logic liên quan đến miền dữ liệu (data domain), hoặc nói ngắn gọn đây là phần back-end chứa tất cả logic dữ liệu của ứng dụng. Dữ liệu ở đây có thể là dữ liệu đang được truyền giữa các thành phần View và Controller hoặc bất kỳ dữ liệu nào khác liên quan đến logic của doanh nghiệp.

Nếu trạng thái của dữ liệu này thay đổi thì Model thường sẽ thông báo cho View (để màn hình có thể thay đổi khi cần) và đôi khi là Controller (nếu cần logic khác để cập nhật View).

Thông thường, các đối tượng Model có thể truy xuất từ cơ sở dữ liệu, thao tác và lưu trữ trạng thái Model trong cơ sở dữ liệu.

* View

View là các thành phần hiển thị giao diện người dùng (UI) của ứng dụng. Thông thường, giao diện người dùng này được tạo từ dữ liệu Model.

* Controller

Controller là các thành phần xử lý tương tác của người dùng để làm việc với Model (cập nhật logic dữ liệu) hoặc/ và với View (cập nhật hiển thị giao diện người dùng).

Trong ứng dụng MVC, Controller xử lý các giá trị chuỗi truy vấn và chuyển các giá trị này cho Model, từ đó Model sẽ truy vấn cơ sở dữ liệu bằng cách sử dụng các giá trị đó. View hiển thị thông tin do Controller xử lý và phản hồi đầu vào từ tương tác của người dùng.

### ***Chức năng của MVC***

MVC giúp bạn tạo các ứng dụng tách biệt cho các khía cạnh khác nhau của ứng dụng (logic đầu vào, logic nghiệp vụ và logic giao diện người dùng), đồng thời cung cấp sự kết nối giữa các thành phần này.

Mô hình MVC chỉ định vị trí của từng loại logic trong ứng dụng:

* Logic nghiệp vụ chính là Model.
* Logic giao diện người dùng thuộc về View.
* Logic đầu vào thuộc về Controller.

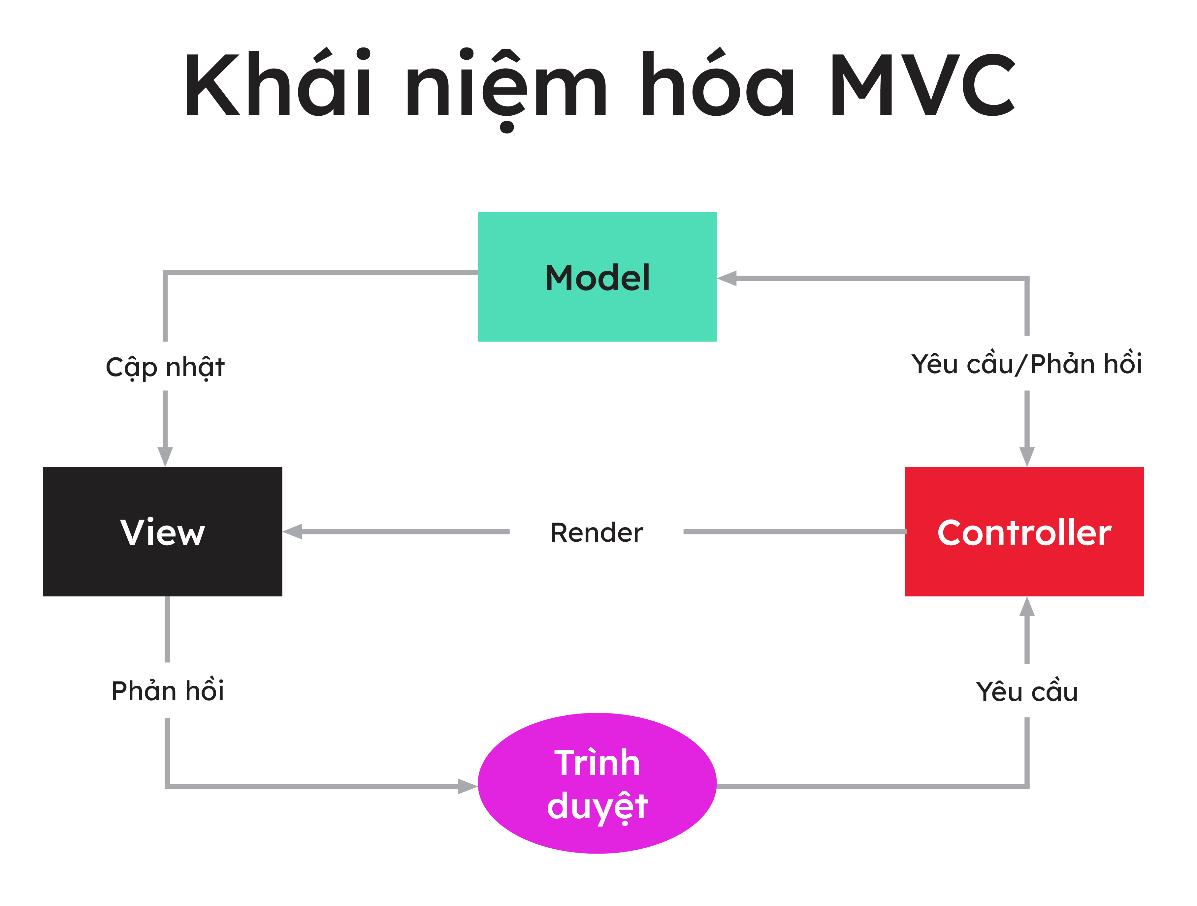
Sự tách biệt này giúp bạn quản lý sự phức tạp khi xây dựng một ứng dụng vì nó cho phép bạn tập trung vào một khía cạnh của việc triển khai tại một thời điểm. Ví dụ: Bạn có thể tập trung vào phần hiển thị giao diện người dùng mà không phụ thuộc vào logic nghiệp vụ.

Sự kết hợp giữa ba thành phần chính của ứng dụng MVC cũng thúc đẩy sự phát triển song song. Ví dụ một lập trình viên có thể làm việc trên View, lập trình viên thứ hai có thể làm việc trên logic của Controller và lập trình viên thứ ba có thể tập trung vào logic nghiệp vụ trong Model.

### ***MVC hoạt động như thế nào***

* Khái niệm hóa mô hình MVC

Về mặt khái niệm, mỗi cách tiếp cận để phát triển MVC đều giống nhau ở chỗ tất cả đều nỗ lực tuân theo nguyên tắc phân tách mối quan tâm (Separation of Concerns – SoC), một mô hình thiết kế chia ứng dụng thành các đơn vị riêng biệt với giảm thiểu ít nhất sự chồng chéo về chức năng.



Hình 1.7: Mô hình khái niệm hóa MVC

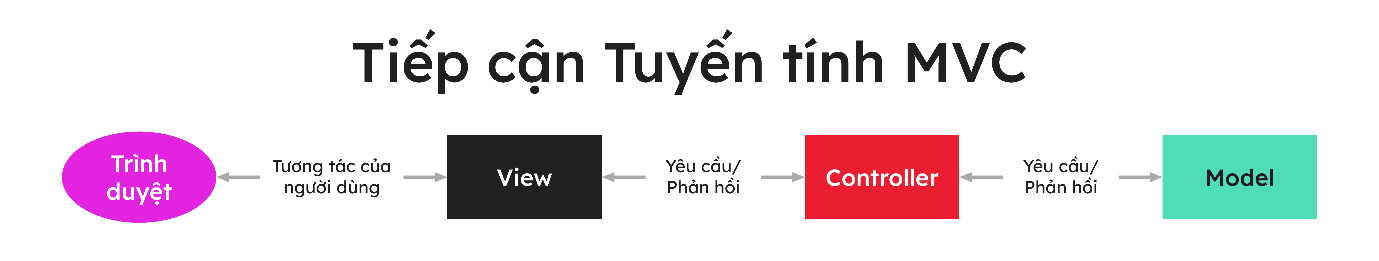
Hình minh họa trên cho thấy một cách để khái niệm hóa MVC. Theo cách tiếp cận này, Controller sẽ xử lý tất cả tương tác của người dùng, chẳng hạn như khi người dùng nhấp vào nút hoặc chọn một giá trị từ danh sách. Controller cũng cung cấp dữ liệu cho thành phần View để đáp ứng yêu cầu của người dùng.

Ngoài ra, Controller sẽ giao tiếp với thành phần Model, thành phần này sẽ gửi dữ liệu cập nhật đến thành phần View. Thành phần View chỉ liên quan đến việc hiển thị dữ liệu do Controller, Model hoặc cả hai cung cấp.

Mức độ trách nhiệm được giao cho từng thành phần và luồng liên lạc giữa chúng phụ thuộc vào việc triển khai riêng lẻ.

Các nhóm phát triển phải hiểu rõ về cách họ khái niệm hóa MVC trước khi bắt đầu thiết kế và xây dựng ứng dụng của mình. Ví dụ, trong hình minh họa, giao tiếp xảy ra giữa phần tử View và Model cũng như giữa thành phần View và Controller. Nhưng vẫn có những trường hợp khác.

* Tiếp cận tuyến tính



Hình 1.8: Mô hình tuyến tính MVC

Trong cách tiếp cận tuyến tính này, người dùng chỉ tương tác với phần tử View thông qua trình duyệt, View chỉ tương tác với Controller và Controller chỉ tương tác với Model.

Mỗi nhóm lập trình có thể có cách giải thích riêng về cách triển khai MVC. Nếu họ đang sử dụng MVC, thì họ cũng có thể cần phải giải quyết về cách mà framework đó cấu trúc một ứng dụng.

Điểm quan trọng là họ đều phải tuân thủ nguyên tắc SoC khi thiết kế và xây dựng các ứng dụng của mình, với mỗi thành phần chịu trách nhiệm về một nhóm nhiệm vụ riêng biệt.

### ***Vì sao nên sử dụng mô hình MVC?***

* Dễ dàng tổ chức các ứng dụng web quy mô lớn

Vì có sự phân tách code giữa ba cấp độ nên việc phân chia và tổ chức logic ứng dụng web thành các ứng dụng quy mô lớn trở nên cực kỳ dễ dàng (cần được quản lý bởi các nhóm lập trình viên lớn).

Ưu điểm chính của việc sử dụng các cách thực hành code như vậy là nó giúp tìm ra các phần code cụ thể một cách nhanh chóng và cho phép bổ sung chức năng mới một cách dễ dàng.

* Hỗ trợ gọi phương thức không đồng bộ (Asynchronous Method Invocation)

Vì kiến trúc MVC hoạt động tốt với [JavaScript](https://itviec.com/blog/javascript-la-gi/) và các framework JavaScript nên không có gì ngạc nhiên khi nó cũng hỗ trợ việc sử dụng Gọi phương thức không đồng bộ (AMI), cho phép các lập trình viên xây dựng các ứng dụng web tải nhanh hơn.

Điều đó có nghĩa là các ứng dụng MVC có thể được tạo ra để hoạt động ngay cả với các tệp PDF, trình duyệt dành riêng cho trang web và cả các tiện ích trên máy tính để bàn.

* Dễ dàng sửa đổi

Sử dụng phương pháp MVC cho phép dễ dàng sửa đổi toàn bộ ứng dụng. Việc thêm/cập nhật loại chế độ xem mới được đơn giản hóa trong mẫu MVC (vì một phần duy nhất độc lập với các phần khác).

Vì vậy, bất kỳ thay đổi nào trong một phần nhất định của ứng dụng sẽ không bao giờ ảnh hưởng đến toàn bộ kiến trúc. Ngược lại, điều này sẽ giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.

* Quá trình lập trình nhanh hơn

Vì có sự phân tách code giữa ba cấp độ, nên việc phát triển ứng dụng web bằng mô hình MVC cho phép một lập trình viên làm việc trên một phần cụ thể (giả sử là view) trong khi một lập trình viên khác có thể làm việc đồng thời trên bất kỳ phần nào khác (giả sử là controller).

Điều này cho phép dễ dàng thực hiện logic nghiệp vụ cũng như giúp đẩy nhanh quá trình lập trình gấp bốn lần. Người ta nhận thấy rằng khi so sánh với các mô hình phát triển khác, mô hình MVC cho thấy tốc độ phát triển cao hơn (lên đến ba lần).

* Lập kế hoạch và bảo trì dễ dàng

MVC rất hữu ích trong giai đoạn lập kế hoạch ban đầu của ứng dụng vì nó cung cấp cho lập trình viên một bản phác thảo về cách sắp xếp các ý tưởng của họ thành code thực tế.

MVC cũng là một công cụ tuyệt vời giúp hạn chế việc sao chép code và cho phép bảo trì ứng dụng dễ dàng.

* Trả về dữ liệu code không cần định dạng

Bằng cách trả về dữ liệu chưa được định dạng, MVC cho phép bạn tạo công cụ xem của riêng mình. Ví dụ: bất kỳ loại dữ liệu nào cũng có thể được định dạng bằng HTML, nhưng với MVC, bạn cũng có thể định dạng dữ liệu bằng Macromedia Flash hoặc Dream Viewer.

Ưu điểm này rất hữu ích cho các lập trình viên vì các thành phần tương tự có thể được sử dụng lại với bất kỳ giao diện nào.

* Hỗ trợ TTD (lập trình dựa trên thử nghiệm)

Ưu điểm chính của mẫu MVC là nó đơn giản hóa rất nhiều quá trình thử nghiệm. Nó giúp việc gỡ lỗi các ứng dụng quy mô lớn trở nên dễ dàng hơn vì nhiều cấp độ được xác định theo cấu trúc và được viết chính xác trong ứng dụng. Do đó, việc lập trình một ứng dụng bằng các bài kiểm thử đơn vị sẽ không gặp rắc rối.

* Nền tảng thân thiện với SEO

Nền tảng MVC hỗ trợ rất nhiều cho việc phát triển các ứng dụng web thân thiện với SEO. Để tạo ra nhiều lượt truy cập hơn từ một ứng dụng cụ thể, MVC cung cấp một cách dễ dàng để phát triển các URL RESTful thân thiện với SEO.

### ***Nhược điểm của MVC***

* Độ phức tạp

MVC có thể tăng độ phức tạp cho codebase vì MVC yêu cầu các lập trình viên tách code của họ thành ba thành phần riêng biệt: Model, View và Controller.

Sự tách biệt này có thể dẫn đến nhiều tệp, lớp và hướng dẫn hơn trong code, điều này có thể khiến các lập trình viên khó hiểu ứng dụng hơn, đặc biệt là đối với các dự án nhỏ hơn hoặc đơn giản. Nhu cầu quản lý tương tác giữa các thành phần này cũng có thể làm tăng thêm độ phức tạp.

* Đường cong học tập

Việc hiểu và triển khai MVC có thể là một thách thức, đặc biệt đối với các lập trình viên mới làm quen với mô hình MVC. Mô hình này đòi hỏi một sự thay đổi trong cách thiết kế và cấu trúc phần mềm. Các lập trình viên phải học cách suy nghĩ về Model, View và Controller cũng như hiểu cách các thành phần này tương tác với nhau.

Đường cong học tập này có thể làm chậm quá trình phát triển và dẫn đến sai lầm trong giai đoạn đầu áp dụng MVC.

* Duy trì khó khăn

Theo thời gian, việc duy trì ứng dụng MVC có thể trở nên khó khăn hơn. Nếu không có tài liệu phù hợp và tiêu chuẩn code rõ ràng, có thể khó theo dõi cách các thành phần khác nhau tương tác. Khi dự án phát triển, việc quản lý các thay đổi và cập nhật đối với MVC có thể trở nên cồng kềnh hơn. Điều này có thể dẫn đến tăng thời gian và chi phí phát triển.

* Tính không linh hoạt

MVC có thể hơi cứng nhắc trong một số trường hợp nhất định. Nó phù hợp nhất cho các ứng dụng có yêu cầu được xác định rõ ràng và hiểu biết rõ ràng về cách dữ liệu truyền qua hệ thống.

Trong các dự án có tính động (dynamic) hoặc thay đổi nhanh chóng, nơi các yêu cầu phát triển nhanh chóng, MVC có thể không phải là một mô hình linh hoạt hoặc dễ thích ứng nhất. Việc thực hiện những thay đổi đáng kể đối với kiến trúc có thể gặp khó khăn.

Số lượng tệp tăng lên

|  |  |  |
| --- | --- | --- |
|  | MVC (Model-View-Controller) | MVVM (Model-View-ViewModel) |
| Triết lý cốt lõi | Trong MVC, triết lý cốt lõi là tách ứng dụng thành ba thành phần chính: Model, View và Controller.  Model đại diện cho dữ liệu và logic nghiệp vụ, View xử lý bản trình bày và giao diện người dùng, còn Controller quản lý sự tương tác của người dùng và đóng vai trò trung gian giữa Model và View. | MVVM mở rộng khái niệm phân tách hơn nữa. MVVC giới thiệu ViewModel, hoạt động như một trung gian giữa Model và View.  ViewModel chịu trách nhiệm hiển thị dữ liệu và lệnh mà View có thể liên kết. |
| Ràng buộc dữ liệu | MVC thường không cung cấp hỗ trợ tích hợp cho liên kết dữ liệu hai chiều giữa Model và View. Trong MVC, các bản cập nhật cho Model hoặc View yêu cầu đồng bộ hóa thủ công. | MVVM nhấn mạnh đến liên kết dữ liệu hai chiều. Các thay đổi trong ViewModel được tự động phản ánh trong View và các thay đổi có thể cập nhật ViewModel.  Điều này giúp đơn giản hóa việc cập nhật giao diện người dùng và giảm nhu cầu đồng bộ hóa thủ công. |
| Hướng phụ thuộc | Trong MVC, Controller thường có ảnh hưởng mạnh mẽ đến cả Model và View.  Controller xử lý thông tin đầu vào của người dùng và đưa ra quyết định về cách cập nhật Model cũng như cách trình bày View. | Trong MVVM, ViewModel có vai trò thụ động hơn.  ViewModel không tương tác trực tiếp với View và không cần biết về View. Thay vào đó, View quan sát và liên kết với ViewModel.  Điều này thúc đẩy sự phân tách mối quan tâm mạnh mẽ hơn. |
| Khả năng kiểm tra | MVC có thể khó khăn hơn khi kiểm thử đơn vị, đặc biệt khi Controller kết hợp chặt chẽ giữa Model và View. | MVVM thường mang lại nhiều code, dễ kiểm tra hơn. ViewModel có thể được kiểm thử một cách độc lập vì nó không phụ thuộc vào các chi tiết cụ thể của View. |
| Framework và hệ sinh thái | MVC là một mẫu kiến trúc chung và có thể được triển khai bằng nhiều ngôn ngữ và framework lập trình khác nhau. Nó không bị ràng buộc với một công nghệ cụ thể. | MVVM được liên kết chặt chẽ với các nền tảng và công nghệ cụ thể, chẳng hạn như WPF và Xamarin cho các ứng dụng trên máy tính để bàn và thiết bị di động, cung cấp hỗ trợ tích hợp cho MVVM. |

Bảng 1.1: Bảng so sánh giữa MVC và MVVM

Việc triển khai mô hình MVC thường dẫn đến số lượng tệp trong dự án tăng lên. Mỗi thành phần thường được biểu diễn trong các tệp hoặc lớp riêng biệt.

Điều này có thể có lợi cho các ứng dụng lớn, phức tạp, nhưng đối với các dự án nhỏ hơn, nó có thể dẫn đến số lượng tệp lớn không cần thiết, khiến dự án trở nên khó tổ chức và quản lý hơn.

### ***Kết luận***

* Tóm lại, MVC là một mẫu kiến trúc tách ứng dụng thành ba thành phần chính: Model, View và Controller. Sự tách biệt này mang lại một số lợi ích, bao gồm cải thiện tổ chức code, khả năng bảo trì và khả năng mở rộng. MVC rất linh hoạt, có thể áp dụng trên các nền tảng web, máy tính để bàn và thiết bị di động.
* Ưu điểm của MVC bao gồm khả năng sử dụng lại nâng cao, kiểm tra dễ dàng hơn và khả năng thực hiện các thay đổi trong một thành phần mà không ảnh hưởng đến các thành phần khác. Điều này giúp các nhóm lập trình làm việc hiệu quả hơn và quản lý các dự án phức tạp một cách dễ dàng.

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## . Khảo sát hiện trạng

### ***2.1.1. Những câu hỏi khảo sát***

Ta cần sử dụng phương pháp phỏng vấn và nghiệp vụ phỏng vấn như sau:

Người quản lý:

* Bạn đang gặp khó khăn gì khi quản lý gian hàng của mình?

Lợi nhuận chưa cao do phải tốn nhiều thời gian để kiểm kê các vấn đề liên quan, lượng hàng và khách hàng không đều đặn.

* Bạn mong muốn điều gì khi quản lý bán hàng trong thời gian tới?

Lợi nhuận cao hơn, tăng số lượng khách hàng và chất lượng phục vụ, có một hệ thống thay đổi quản lý bán hàng và khách hàng

* Bạn muốn hệ thống sẽ cung cấp thông tin gì?

Thông tin sản phẩm giày dép, thông tin nhà cung cấp, bảo hành, quản lý khách hàng, thống kê lượng hàng

* Ai có thể sử dụng hệ thống này?

Quản lý bán hàng, nhân viên bán hàng, khách mua hàng

* Muốn thêm các chức năng gì cho hệ thống mang lại hiệu quả cao và dễ sử dụng?

Tìm kiếm thông tin sản phẩm, quản lý mặt hàng, quản lý nhãn hàng, thống kê doanh số…

Khách hàng:

* Bạn gặp những khó khăn gì khi mua hàng trực tiếp

Mất thời gian đi lại, phục vụ không tốt

* Bạn mong muốn gì khi mua hàng qua mạng

Mua sắm nhanh, xem nhanh được các sản phẩm, giá thành sản phẩm tốt hơn, nhận nhiều ưu đãi

* Bạn có muốn chọn ra những sản phẩm vừa với nhu cầu của mình?

Chọn lựa sản phẩm phù hợp với túi tiền, phù hợp với mục đích đi lại

### ***Tại sao phải xây dựng website bán giày***

* *Cung cấp đầy đủ thông tin*: Internet là mạng kết nối toàn cầu để chia sẻ dữ liệu, vì thế website cũng là nơi mà khách hàng tìm kiếm thông tin và sản phẩm nhiều nhất. Nếu đã có website, thì việc quản lý sẽ dễ dàng hơn trong việc đưa thông tin về sản phẩm/ dịch vụ/ hình ảnh thương hiệu của mình tiếp cận tới khách hàng một cách nhanh chóng, đồng bộ, đầy đủ và hiệu quả.
* *Nâng cao chất lượng phục vụ khách hàng*: Các website bán hàng có thể tích hợp chức năng chat trực tuyến để giải đáp những thắc mắc, khiếu nại của khách hàng. Ngoài ra, trên các website thương mại điện tử bán hàng tiết kiệm được chi phí đi lại cũng như thời gian mua sắm. Những điều này thực sự quan trọng nếu đối tượng khách hàng của doanh nghiệp chủ yếu ở xa hay không thuận tiện trong việc di chuyển.
* *Dễ dàng tiếp cận và mở rộng thị trường:*  lượng người dùng internet rất lớn và không ngừng tăng lên, đây cũng là một lợi thế mới cho việc phát triển website quảng bá cho cửa hàng.

## . Phân tích yêu cầu

* **Chức năng**

Đây là một website bán và giới thiệu sản phẩm của cửa hàng đến người tiêu dùng với các chi tiết về mặt hàng và giá cả chính xác. Có các chức năng sau:

* Cho phép cập nhật hàng vào CSDL
* Hiển thị danh sách các mặt hàng
* Hiển thị hàng hóa khách hàng đã chọn mua
* Xử lý đơn hàng
* **Yêu cầu đặt ra**

- Thiết bị phần mềm

* Máy tính có thể thiết kế web
* Phần mềm thiết kế web: Visual Studio Code
* Phần mềm hỗ trợ: Xampp

- Yêu cầu trang web

Hệ thống có 2 phần:

Phần khách hàng:

Khách hàng là những người có nhu cầu mua sản phẩm, họ sẽ tìm kiếm các mặt hàng cần thiết từ hệ thống và đặt mua. Vì thế phải có các chức năng sau:

+ Hiển thị danh sách các mặt hàng của cửa hàng để khách hàng xem, lựa chọn và đặt mua.

+ Khách hàng xem các thông tin khác của cửa hàng

Dành cho người quản trị: Admin có quyền kiểm soát mọi hoạt động của hệ thống. Người này được cấp username và password để đăng nhập hệ thống thực hiện các chức năng của mình

+ Chức năng cập nhật, sửa, xóa các mặt hàng

+ Tiếp nhận kiểm tra đơn đặt hàng của khách hàng. Hiển thị đơn đặt hàng

Ngoài các chức năng trên thì trang web phải được thiết kế sao cho dễ hiểu, giao diện mang tính dễ dùng đẹp mắt và làm sao cho khách hàng thấy được thông tin cần tìm, cung cấp các thông tin quảng cáo hấp dẫn, các tin tức khuyến mãi để thu hút khách hàng trong quá trình đặt mua qua mạng. Đồng thời trang Web phải luôn đổi mới, hấp dẫn.

* 1. **. Phân tích hệ thống**

### ***2.3.1. Yêu cầu hệ thống***

* Yêu cầu chung:

- Đối với hệ thống:Hệ thống được xây dựng nhằm tăng khả năng quảng bá sản phẩm, thương hiệu đến với khách hàng. Từ đó thúc đẩy công việc của cửa hàng nhằm đem lại lợi ích cho cửa hàng và khách hàng.

* Yêu cầu chi tiết:
* Xác định tác nhân (gồm 2 tác nhân):

+ Khách hàng: khách hàng có thể xem tất cả và tìm kiếm các thông tin mà quản trị viên đưa lên website, bên cạnh đó khách hàng còn có thể đặt hàng trực tiếp trên website và gửi những phản ánh của mình đến với cửa hàng.

+ Người quản trị: đăng nhập vào hệ thống để quản lý tất cả các thông tin như các bài viết (thêm, cập nhật, xóa), danh mục sản phẩm (thêm, cập nhật, xóa), sản phẩm (thêm, cập nhật, xóa), các đơn hàng của khách hàng (xử lý) hay kiểm tra email phản ánh của khách hàng tới cửa hàng, cuối cùng sẽ thống kê các đơn hàng online trong từng tháng.

### ***2.3.2. Phân tích hệ thống***

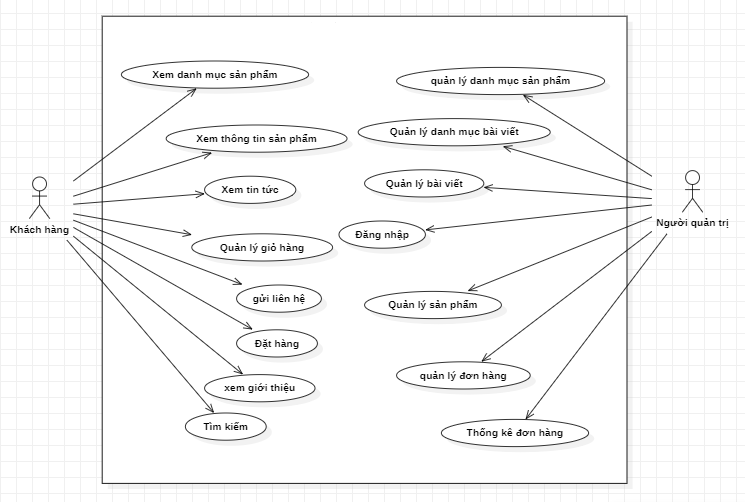
***2.3.2.1. Xác định mục tiêu***

* Người quản trị
* Đăng nhập
* Xem và cập nhật các thông tin của website
* Quản lý đơn hàng
* Thống kê đơn hàng
* Khách hàng
* Xem thông tin trên website
* Tìm kiếm sản phẩm
* Gửi liên hệ
* Đặt hàng

***2.3.2.2. Đặc tả các Use Case***

* Đăng nhập vào hệ thống
* Tác nhân: người quản trị
* Mô tả: UC cho phép đăng nhập vào hệ thống
* Luồng sự kiện chính:
* Người quản trị vào trang đăng nhập hệ thống
* Giao diện đăng nhập hiển thị
* Nhập tên đăng nhập và mật khẩu
* Hệ thống kiểm tra tên đăng nhập và mật khẩu
* Nếu đăng nhập thành công thì chuyển sang bước 6. Nếu nhập sai tên đăng nhập hoặc mật khẩu hoặc sai cả tên đăng nhập và mật khẩu thì chuyển sang luồng rẽ nhánh A1
* Cho phép người quản trị đăng nhập vào trang chủ trang quản trị
* UC kết thúc
* Luồng sự kiện rẽ nhánh A1: người quản trị đăng nhập không thành công
* Hệ thống thông báo quá trình đăng nhập không thành công
* Hệ thống yêu cầu nhập lại tên đăng nhập và mật khẩu
* Nếu đồng ý thì quay lại bước 2 của luồng sự kiện chính, nếu không đồng ý thì UC kết thúc
* Hậu điều kiện: người quản trị đã đăng nhập thành công và có thể sử dụng các chức năng mà hệ thống cung cấp
* Quản lý danh mục bài viết
* Tác nhân: Người quản trị
* Mô tả: UC cho phép thêm, sửa, xóa danh mục bài viết
* Tiền điều kiện: Người quản trị đã đăng nhập được vào hệ thống
* Luồng sự kiện chính:
* Thêm danh mục bài viết
* Người quản lý chọn chức năng thêm danh mục bài viết
* Giao diện thêm danh mục bài viết hiển thị
* Nhập thông tin vào các trường
* Hệ thống kiểm tra thông tin nhập vào
* Thông báo thêm danh mục bài viết vào CSDL thành công
* UC kết thúc
* Sửa danh mục bài viết
* Người quản trị chọn chức năng sửa danh mục bài viết
* Giao diện sửa thông tin của danh mục bài viết hiển thị
* Nhập thông tin cần sửa
* Hệ thống kiểm tra thông tin đã sửa
* Hệ thống thông báo đã sửa thành công
* UC kết thúc
* Xóa danh mục bài viết
* Người quản trị chọn chức năng xóa danh mục bài viết
* Giao diện thông tin danh mục bài viết cần xóa hiển thị
* Người quản trị nhấn xóa
* Nếu chọn xóa thì thực hiện xóa danh mục bài viết.
* Hậu điều kiện: Danh mục bài viết được cập nhật trong hệ thống
* Quản lý bài viết
* Tác nhân: Người quản trị
* Mô tả: UC cho phép thêm, sửa, xóa bài viết
* Tiền điều kiện: Người quản trị đã đăng nhập được vào hệ thống
* Luồng sự kiện chính:
* Thêm bài viết:
* Người quản lý chọn chức năng thêm bài viết
* Giao diện thêm bài viết hiển thị
* Nhập thông tin vào các trường
* Hệ thống kiểm tra thông tin nhập vào
* Thông báo thêm bài viết vào CSDL thành công
* UC kết thúc
* Sửa bài viết
* Người quản trị chọn chức năng sửa bài viết
* Giao diện sửa thông tin của bài viết hiển thị
* Nhập thông tin cần sửa
* Hệ thống kiểm tra thông tin đã sửa
* Hệ thống thông báo đã sửa thành công
* UC kết thúc
* Xóa bài viết
* Người quản trị chọn chức năng xóa bài viết
* Giao diện thông tin bài viết cần xóa hiển thị
* Người quản trị nhấn xóa
* Nếu chọn xóa thì thực hiện xóa bài viết.
* Hậu điều kiện: Bài viết được cập nhật trong hệ thống
* Quản lý danh mục sản phẩm
* Tác nhân: Người quản trị
* Mô tả: UC cho phép thêm, sửa, xóa danh mục sản phẩm
* Tiền điều kiện: Người quản trị đã đăng nhập được vào hệ thống
* Luồng sự kiện chính:
* Thêm danh mục sản phẩm
  + Người quản lý chọn chức năng thêm danh mục sản phẩm
  + Giao diện thêm danh mục sản phẩm hiển thị
  + Nhập thông tin vào các trường
  + Hệ thống kiểm tra thông tin nhập vào
  + Thông báo thêm danh mục sản phẩm vào CSDL thành công
  + UC kết thúc
* Sửa danh mục sản phẩm
  + Người quản trị chọn chức năng sửa danh mục sản phẩm
  + Giao diện sửa thông tin của danh mục sản phẩm hiển thị
  + Nhập thông tin cần sửa
  + Hệ thống kiểm tra thông tin đã sửa
  + Hệ thống thông báo đã sửa thành công
  + UC kết thúc
* Xóa danh mục sản phẩm
* Người quản trị chọn chức năng xóa danh mục sản phẩm
* Giao diện thông tin danh mục sản phẩm cần xóa hiển thị
* Người quản trị nhấn xóa
* Nếu chọn xóa thì thực hiện xóa danh mục sản phẩm.
* Hậu điều kiện: Danh mục sản phẩm được cập nhật trong hệ thống
* Quản lý sản phẩm
* Tác nhân: Người quản trị
* Mô tả: UC cho phép thêm, sửa, xóa sản phẩm
* Tiền điều kiện: Người quản trị đã đăng nhập được vào hệ thống
* Luồng sự kiện chính:
* Thêm sản phẩm
  + Người quản lý chọn chức năng thêm sản phẩm
  + Giao diện thêm sản phẩm hiển thị
  + Nhập thông tin vào các trường
  + Hệ thống kiểm tra thông tin nhập vào
  + Thông báo thêm sản phẩm vào CSDL thành công
  + UC kết thúc
* Sửa danh mục sản phẩm
  + Người quản trị chọn chức năng sửa sản phẩm
  + Giao diện sửa thông tin của sản phẩm hiển thị
  + Nhập thông tin cần sửa
  + Hệ thống kiểm tra thông tin đã sửa
  + Hệ thống thông báo đã sửa thành công
  + UC kết thúc
* Xóa sản phẩm
  + Người quản trị chọn chức năng xóa sản phẩm
  + Giao diện thông tin sản phẩm cần xóa hiển thị
  + Người quản trị nhấn xóa
  + Nếu chọn xóa thì thực hiện xóa sản phẩm
* Hậu điều kiện: sản phẩm được cập nhật trong hệ thống
* Quản lý đơn hàng
* Tác nhân: Người quản trị
* Mô tả: UC cho phép cập nhật tình trạng đơn hàng
* Tiền điều kiện: Người quản trị đã đăng nhập được vào hệ thống
* Luồng sự kiện chính:
* Người quản lý chọn chức năng xem chi tiết đơn hàng
  + Giao diện chi tiết đơn hàng hiển thị
  + Nhấn nút “đã xử lý”
  + Hệ thống kiểm tra t
  + Cập nhật đơn hàng vào CSDL thành công
  + UC kết thúc
* Hậu điều kiện: đơn hàng được cập nhật trong hệ thống
* Thống kê doanh thu
* Tác nhân: Người quản trị
* Mô tả: UC cho phép thống kê doanh thu
* Tiền điều kiện: Người quản trị đã đăng nhập được vào hệ thống
* Luồng sự kiện chính:
* Người quản lý chọn chức năng thống kê doanh thu
* Giao diện thống kê doanh thu hiển thị
* Hậu điều kiện: thống kê doanh thu thành công
* Đặt hàng
* Tác nhân: Khách hàng
* Mô tả: UC cho phép đặt hàng trên hệ thống
* Tiền điều kiện: Khách hàng đã thêm sản phẩm vào giỏ hàng
* Luồng sự kiện chính:
* Khách hàng chọn chức năng đặt hàng
  + Giao diện đặt hàng hiển thị
  + Nhập thông tin vào các trường
  + Hệ thống kiểm tra thông tin
  + Nếu đặt hàng thành công thì chuyển sang bước 6. Nếu nhập không đủ thông tin thì chuyển sang luồng rẽ nhánh A1
  + Cho phép khách hàng đặt hàng
  + Thông báo đặt hàng thành công
  + UC kết thúc
* Luồng sự kiện rẽ nhánh A1: khách hàng đặt hàng không thành công
* Hệ thống thông báo quá trình đặt hàng không thành công
* Hệ thống yêu cầu nhập đủ thông tin
* Nếu đồng ý thì quay lại bước 2 của luồng sự kiện chính, nếu không đồng ý thì UC kết thúc
* Hậu điều kiện: khách hàng đặt hàng thành công, chờ xác nhận đơn hàng và giao hàng
* Quản lý giỏ hàng
* Tác nhân: Khách hàng
* Mô tả: UC cho phép sử dụng các chức năng của giỏ hàng trên hệ thống
* Tiền điều kiện:
* Luồng sự kiện chính:
* Thêm sản phẩm vào giỏ hàng
* Khách hàng chọn chức năng thêm sản phẩm vào giỏ hàng
* Thông báo thêm sản phẩm thành công
* Giao diện giỏ hàng hiển thị
* UC kết thúc
* Cập nhật giỏ hàng
  + Khách hàng chọn chức năng cập nhật giỏ hàng
  + Giao diện giỏ hàng hiển thị
  + Nhập thông tin cần sửa
  + Hệ thống kiểm tra thông tin đã sửa
  + Hệ thống thông báo đã sửa thành công
  + UC kết thúc
* Xóa sản phẩm khỏi giỏ hàng
  + Khách hàng chọn chức năng xóa sản phẩm
  + Giao diện lựa chọn hiển thị
  + Khách hàng nhấn xóa
  + Nếu chọn xóa thì thực hiện xóa sản phẩm khỏi giỏ hàng
* Hậu điều kiện: khách hàng quản lý giỏ hàng thành công
* Tìm kiếm sản phẩm
* Tác nhân: Khách hàng
* Mô tả: UC cho phép sử dụng chức năng tìm kiếm trên hệ thống
* Tiền điều kiện:
* Luồng sự kiện chính:
* Tìm kiếm
* Khách hàng chọn chức năng tìm kiếm
* Khách hàng nhập thông tin muốn tìm kiếm
* Hệ thống kiểm tra có tồn tại thông tin đó không
* Hệ thống thông báo kết quả
* Hệ thống hiển thị kết quả
* UC kết thúc
* Hậu điều kiện: khách hàng tìm kiếm thành công
* Gửi liên hệ
* Tác nhân: Khách hàng
* Mô tả: UC cho phép sử dụng chức năng gửi liên hệ trên hệ thống
* Tiền điều kiện:
* Luồng sự kiện chính:
* Gửi liên hệ
* Khách hàng chọn chức năng gửi liên hệ trên hệ thống
* Khách hàng nhập thông tin
* Hệ thống kiểm tra thông tin đã được nhập đầy đủ vào các trường
* Hệ thống thông báo kết quả
* UC kết thúc
* Hậu điều kiện: khách hàng gửi liên hệ thành công
  1. **. Phân tích thiết kế với UML**
     1. ***Biểu đồ use case***

Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ UC tổng quát

Biểu đồ UC phân rã cho chức năng quản lý giỏ hàng của tác nhân khách hàng

A diagram of a diagram

Description automatically generated

Hình 2.2: Biểu đồ UC phân rã cho chức năng quản lý giỏ hàng

Biểu đồ UC phân rã cho chức năng quản lý danh mục bài viết

A diagram of a diagram

Description automatically generated

Hình 2.3: Biểu đồ UC phân rã cho chức năng quản lý danh mục bài viết

Biểu đồ UC phân rã cho chức năng quản lý bài viết

A diagram of a diagram

Description automatically generated

Hình 2.4: Biểu đồ UC phân rã cho chức năng quản lý bài viết

Biểu đồ UC phân rã cho chức năng quản lý danh mục sản phẩm

A diagram of a diagram

Description automatically generated

Hình 2.5: Biểu đồ UC phân rã cho chức năng quản lý danh mục sản phẩm

Biểu đồ UC phân rã cho chức năng quản lý sản phẩm

A diagram of a diagram

Description automatically generated

Hình 2.6: Biểu đồ UC phân rã cho chức năng quản lý sản phẩm

Biểu đồ UC phân rã cho chức năng quản lý đơn hàng

A diagram of a diagram

Description automatically generated

Hình 2.7: Biểu đồ UC phân rã cho chức năng quản lý đơn hàng

### ***Biểu đồ lớp***

A diagram of a computer

Description automatically generatedHình 2.8: Biểu đồ lớp

* + 1. Biểu đồ tuần tự
       1. **Kịch bản**

1. Kịch bản các chức năng cho người quản trị

|  |  |
| --- | --- |
| Tên UC | Đăng nhập |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị đăng nhập vào hệ thống |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Đăng nhập thành công |
| Kích hoạt | Người quản trị chọn “login” trên hệ thống |

Bảng 2.1*:* Kịch bản cho chức năng đăng nhập

Chuỗi sự kiện chính :

* Người quản trị nhập tên đăng nhập và mật khẩu rồi bấm “login”
* Hệ thống hiển thị thông báo
* Hệ thống hiển thị giao diện trang chủ
* Người quản trị đăng nhập thành công

- Kịch bản cho chức năng xem danh mục bài viết

|  |  |
| --- | --- |
| Tên UC | Xem danh mục bài viết |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn liệt kê danh mục bài viết |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xem danh mục bài viết thành công |
| Kích hoạt | Người quản trị chọn liệt kê danh mục trên hệ thống |

Bảng 2.2: Kịch bản cho chức năng xem danh mục bài viết

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xem danh mục bài viết
* Hệ thống lấy dữ liệu và hiển thị giao diện tất cả danh mục bài viết
* Người quản trị thoát khỏi chức năng xem danh mục bài viết

- Kịch bản cho chức năng thêm danh mục bài viết

|  |  |
| --- | --- |
| Tên UC | Thêm danh mục bài viết |
| Tác nhân chính | Người quản trị |
| Tiền điều kiện | Người quản trị chọn danh mục bài viết |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Thêm danh mục bài viết thành công |
| Kích hoạt | Người quản trị chọn thêm danh mục bài viết trên hệ thống |

Bảng 2.3: Kịch bản cho chức năng thêm danh mục bài viết

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu thêm danh mục bài viết
* Hệ thống hiển thị giao diện thêm và yêu cầu người dùng điền thông tin
* Người dùng nhập thông tin sau đó nhấn Thêm
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng thêm danh mục bài viết

- Kịch bản cho chức năng sửa danh mục bài viết

|  |  |
| --- | --- |
| Tên UC | Sửa danh mục bài viết |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn sửa danh mục bài viết |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Sửa danh mục bài viết thành công |
| Kích hoạt | Người quản trị chọn Sửa danh mục bài viết trên hệ thống |

Bảng 2.4: Kịch bản cho chức năng sửa danh mục bài viết

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu sửa danh mục bài viết
* Hệ thống hiển thị giao diện sửa và yêu cầu người dùng điền thông tin
* Người quản trị nhập thông tin sau đó nhấn Sửa
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng sửa danh mục bài viết

- Kịch bản cho chức năng xóa danh mục bài viết

|  |  |
| --- | --- |
| Tên UC | Xóa danh mục bài viết |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn xóa danh mục bài viết |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xóa danh mục bài viết thành công |
| Kích hoạt | Người quản trị chọn xóa danh mục bài viết trên hệ thống |

Bảng 2.5: Kịch bản cho chức năng xóa danh mục bài viết

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xóa danh mục bài viết
* Hệ thống hiển thị giao diện xóa và yêu cầu người dùng chọn phương thức xóa
* Người dùng nhấn Xóa
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng xóa danh mục bài viết

-Kịch bản cho chức năng xem bài viết

|  |  |
| --- | --- |
| Tên UC | Xem bài viết |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn liệt kê bài viết |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xem bài viết thành công |
| Kích hoạt | Người quản trị chọn liệt kê trên hệ thống |

Bảng 2.6: Kịch bản cho chức năng xem bài viết

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xem bài viết
* Hệ thống lấy dữ liệu và hiển thị giao diện tất cả bài viết
* Người quản trị thoát khỏi chức năng xem bài viết

- Kịch bản cho chức năng thêm bài viết

|  |  |
| --- | --- |
| Tên UC | Thêm bài viết |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn bài viết |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Thêm bài viết thành công |
| Kích hoạt | Người quản trị chọn thêm bài viết trên hệ thống |

Bảng 2.7: Kịch bản cho chức năng thêm bài viết

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu thêm bài viết
* Hệ thống hiển thị giao diện thêm và yêu cầu người dùng điền thông tin
* Người dùng nhập thông tin sau đó nhấn Thêm
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng thêm bài viết

-Kịch bản cho chức năng sửa thông tin bài viết

|  |  |
| --- | --- |
| Tên UC | Sửa bài viết |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn sửa bài viết |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Sửa bài viết thành công |
| Kích hoạt | Người quản trị chọn Sửa bài viết trên hệ thống |

Bảng 2.8: Kịch bản cho chức năng sửa thông tin bài viết

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu sửa bài viết
* Hệ thống hiển thị giao diện sửa và yêu cầu người dùng điền thông tin
* Người quản trị nhập thông tin sau đó nhấn Sửa
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng sửa bài viết

- Kịch bản cho chức năng xóa bài viết

|  |  |
| --- | --- |
| Tên UC | Xóa bài viết |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn xóa bài viết |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xóa bài viết thành công |
| Kích hoạt | Người quản trị chọn xóa bài viết trên hệ thống |

Bảng 2.9: Kịch bản cho chức năng xóa bài viết

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xóa bài viết
* Hệ thống hiển thị giao diện xóa và yêu cầu người dùng nhấn nút
* Người dùng nhấn Xóa
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng xóa bài viết

- Kịch bản cho chức năng xem danh mục sản phẩm

|  |  |
| --- | --- |
| Tên UC | Xem danh mục sản phẩm |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn liệt kê danh mục sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xem danh mục sản phẩm thành công |
| Kích hoạt | Người quản trị chọn liệt kê trên hệ thống |

Bảng 2.10: Kịch bản cho chức năng xem danh mục sản phẩm

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xem danh mục sản phẩm
* Hệ thống lấy dữ liệu và hiển thị giao diện tất cả danh mục sản phẩm
* Người quản trị thoát khỏi chức năng xem danh mục sản phẩm

- Kịch bản cho chức năng thêm danh mục sản phẩm

|  |  |
| --- | --- |
| Tên UC | Thêm danh mục sản phẩm |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn danh mục sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Thêm danh mục sản phẩm thành công |
| Kích hoạt | Người quản trị chọn thêm danh mục sản phẩm trên hệ thống |

Bảng 2.11: Kịch bản cho chức năng thêm danh mục sản phẩm

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu thêm danh mục sản phẩm
* Hệ thống hiển thị giao diện thêm và yêu cầu người dùng điền thông tin
* Người dùng nhập thông tin sau đó nhấn Thêm
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng thêm danh mục sản phẩm

-Kịch bản cho chức năng sửa danh mục sản phẩm

|  |  |
| --- | --- |
| Tên UC | Sửa danh mục sản phẩm |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn sửa danh mục sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Sửa danh mục sản phẩm thành công |
| Kích hoạt | Người quản trị chọn Sửa danh mục sản phẩm trên hệ thống |

Bảng 2.12: Kịch bản cho chức năng sửa danh mục sản phẩm

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu sửa danh mục sản phẩm
* Hệ thống hiển thị giao diện sửa và yêu cầu người dùng điền thông tin
* Người quản trị nhập thông tin sau đó nhấn Sửa
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng sửa danh mục sản phẩm

- Kịch bản cho chức năng xóa danh mục sản phẩm

|  |  |
| --- | --- |
| Tên UC | Xóa danh mục sản phẩm |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn xóa danh mục sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xóa danh mục sản phẩm thành công |
| Kích hoạt | Người quản trị chọn xóa danh mục sản phẩm trên hệ thống |

Bảng 2.13: Kịch bản cho chức năng xóa danh mục sản phẩm

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xóa danh mục sản phẩm
* Hệ thống hiển thị giao diện xóa và yêu cầu người dùng nhấn nút
* Người dùng nhấn Xóa
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng xóa danh mục sản phẩm

- Kịch bản cho chức năng xem danh sách sản phẩm

|  |  |
| --- | --- |
| Tên UC | Xem danh sách sản phẩm |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn liệt kê sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xem sản phẩm thành công |
| Kích hoạt | Người quản trị chọn liệt kê trên hệ thống |

Bảng 2.14: Kịch bản cho chức năng xem danh sách sản phẩm

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xem sản phẩm
* Hệ thống lấy dữ liệu và hiển thị giao diện tất cả sản phẩm
* Người quản trị thoát khỏi chức năng xem danh sách sản phẩm

- Kịch bản cho chức năng thêm sản phẩm

|  |  |
| --- | --- |
| Tên UC | Thêm sản phẩm |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Thêm sản phẩm thành công |
| Kích hoạt | Người quản trị chọn thêm sản phẩm trên hệ thống |

Bảng 2.15: Kịch bản cho chức năng thêm sản phẩm

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu thêm sản phẩm
* Hệ thống hiển thị giao diện thêm và yêu cầu người dùng điền thông tin
* Người dùng nhập thông tin sau đó nhấn Thêm
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng thêm sản phẩm

- Kịch bản cho chức năng sửa thông tin sản phẩm

|  |  |
| --- | --- |
| Tên UC | Sửa thông tin sản phẩm |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn sửa sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Sửa sản phẩm thành công |
| Kích hoạt | Người quản trị chọn Sửa sản phẩm trên hệ thống |

Bảng 2.16: Kịch bản cho chức năng sửa thông tin sản phẩm

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu sửa thông tin sản phẩm
* Hệ thống hiển thị giao diện sửa và yêu cầu người dùng điền thông tin
* Người quản trị nhập thông tin sau đó nhấn Sửa
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng sửa thông tin sản phẩm

- Kịch bản cho chức năng xóa sản phẩm

|  |  |
| --- | --- |
| Tên UC | Xóa sản phẩm |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn xóa sản phẩm |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xóa danh mục thành công |
| Kích hoạt | Người quản trị chọn xóa sản phẩm trên hệ thống |

Bảng 2.17: Kịch bản cho chức năng xóa sản phẩm

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xóa sản phẩm
* Hệ thống hiển thị giao diện xóa và yêu cầu người dùng nhấn nút
* Người dùng nhấn Xóa
* Hệ thống kiểm tra thông tin mới và xác định tính hợp lệ
* Hệ thống hiển thị cập nhật
* Người quản trị thoát khỏi chức năng xóa sản phẩm

- Kịch bản cho chức năng xem đơn hàng

|  |  |
| --- | --- |
| Tên UC | Xem đơn hàng |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn liệt kê đơn hàng |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xem đơn hàng thành công |
| Kích hoạt | Người quản trị chọn liệt kê trên hệ thống |

Bảng 2.18: Kịch bản cho chức năng xem đơn hàng

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xem đơn hàng
* Hệ thống lấy dữ liệu và hiển thị giao diện tất cả đơn hàng
* Người quản trị thoát khỏi chức năng xem đơn hàng

- Kịch bản cho chức năng xem chi tiết đơn hàng và thay đổi trạng thái đơn hàng

|  |  |
| --- | --- |
| Tên UC | Xem chi tiết đơn hàng và thay đổi trạng thái đơn hàng |
| Tác nhân chính | Người quản trị |
| Tên điều kiện | Người quản trị chọn chi tiết đơn hàng |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xem chi tiết đơn hàng và thay đổi trạng thái đơn hàng thành công |
| Kích hoạt | Người quản trị chọn xem trên hệ thống |

Bảng 2.19: Kịch bản cho chức năng xem chi tiết đơn hàng và thay đổi trạng thái đơn hàng

Chuỗi sự kiện chính :

* Người quản trị kích hoạt yêu cầu xem chi tiết đơn hàng
* Hệ thống lấy dữ liệu và hiển thị giao diện chi tiết về đơn hàng
* Hệ thống yêu cầu xử lý đơn hàng
* Người quản trị kích hoạt Xử lý đơn hàng
* Hệ thống hiển thị cập nhật về tình trạng đơn hàng
* Người quản trị thoát khỏi chức năng xem chi tiết đơn hàng

1. Kịch bản cho khách hàng

Kịch bản cho chức năng thêm sản phẩm vào giỏ hàng

|  |  |
| --- | --- |
| Tên UC | Thêm sản phẩm vào giỏ hàng |
| Tác nhân chính | Khách hàng |
| Tên điều kiện | Khách hàng chọn thêm sản phẩm vào giỏ hàng |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Thêm sản phẩm vào giỏ hàng thành công |
| Kích hoạt | Khách hàng chọn thêm trên hệ thống |

Bảng 2.20: Kịch bản cho chức năng thêm sản phẩm vào giỏ hàng

Chuỗi sự kiện chính :

* Khách hàng kích hoạt yêu cầu thêm sản phẩm vào giỏ hàng
* Hệ thống lấy dữ liệu sản phẩm và hiển thị giao diện các sản phẩm
* Hệ thống yêu cầu thêm sản phẩm vào giỏ hàng
* Khách hàng kích hoạt Thêm sản phẩm
* Hệ thống hiển thị cập nhật về giỏ hàng
* Khách hàng thoát khỏi chức năng thêm sản phẩm vào giỏ hàng

- Kịch bản cho chức năng Sửa số lượng sản phẩm trong giỏ hàng

|  |  |
| --- | --- |
| Tên UC | Sửa số lượng sản phẩm trong giỏ hàng |
| Tác nhân chính | Khách hàng |
| Tên điều kiện | Khách hàng chọn sửa số lượng sản phẩm trong giỏ hàng |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Sửa số lượng sản phẩm trong giỏ hàng thành công |
| Kích hoạt | Khách hàng chọn sửa số lượng trên hệ thống |

Bảng 2.21: Kịch bản cho chức năng Sửa số lượng sản phẩm trong giỏ hàng

Chuỗi sự kiện chính :

* Khách hàng kích hoạt yêu cầu sửa số lượng sản phẩm trong giỏ hàng
* Hệ thống lấy dữ liệu sản phẩm và hiển thị giao diện giỏ hàng
* Hệ thống yêu cầu sửa số lượng sản phẩm vào giỏ hàng
* Khách hàng thay đổi số lượng và nhấn nút cập nhật lại
* Hệ thống hiển thị cập nhật về giỏ hàng
* Khách hàng thoát khỏi chức năng sửa số lượng sản phẩm trong giỏ hàng

- Kịch bản cho chức năng Xóa sản phẩm trong giỏ hàng

|  |  |
| --- | --- |
| Tên UC | Xóa sản phẩm trong giỏ hàng |
| Tác nhân chính | Khách hàng |
| Tên điều kiện | Khách hàng chọn xóa sản phẩm trong giỏ hàng |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Xóa sản phẩm trong giỏ hàng thành công |
| Kích hoạt | Khách hàng chọn xóa sản phẩm trên hệ thống |

Bảng 2.22: Kịch bản cho chức năng Xóa sản phẩm trong giỏ hàng

Chuỗi sự kiện chính :

* Khách hàng kích hoạt yêu cầu xóa sản phẩm trong giỏ hàng
* Hệ thống lấy dữ liệu sản phẩm và hiển thị giao diện giỏ hàng
* Hệ thống yêu cầu xóa sản phẩm vào giỏ hàng
* Khách nhấn nút cập nhật xóa
* Hệ thống hiển thị cập nhật về giỏ hàng
* Khách hàng thoát khỏi chức năng xóa sản phẩm trong giỏ hàng

- Kịch bản cho chức năng đặt hàng

|  |  |
| --- | --- |
| Tên UC | Đặt hàng |
| Tác nhân chính | Khách hàng |
| Tên điều kiện | Khách hàng chọn đặt hàng |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Đặt hàng thành công |
| Kích hoạt | Khách hàng chọn đặt hàng trên hệ thống |

Bảng 2.23: Kịch bản cho chức năng đặt hàng

Chuỗi sự kiện chính :

* Khách hàng kích hoạt yêu cầu đặt hàng
* Hệ thống hiển thị giao diện đặt hàng
* Hệ thống yêu cầu nhập thông tin
* Khách hàng nhập thông tin và nhấn nút cập nhật đặt hàng
* Hệ thống hiển thị thông báo
* Khách hàng thoát khỏi chức năng đặt hàng

- Kịch bản cho chức năng gửi liên hệ

|  |  |
| --- | --- |
| Tên UC | Gửi liên hệ |
| Tác nhân chính | Khách hàng |
| Tên điều kiện | Khách hàng chọn gửi liên hệ |
| Đảm bảo tối thiểu | Hệ thống loại bỏ và quay về bước trước đó |
| Đảm bảo thành công | Gửi liên hệ thành công |
| Kích hoạt | Khách hàng chọn gửi liên hệ trên hệ thống |

Bảng 2.24: Kịch bản cho chức năng gửi liên hệ

Chuỗi sự kiện chính :

* Khách hàng kích hoạt yêu cầu gửi liên hệ
* Hệ thống hiển thị giao diện gửi liên hệ
* Hệ thống yêu cầu nhập thông tin
* Khách hàng nhập thông tin và nhấn nút gửi liên hệ
* Hệ thống hiển thị thông báo
* Khách hàng thoát khỏi chức năng gửi liên hệ

#### ***2.4.3.2. Biểu đồ***

1. Biểu đồ tuần tự cho tác nhân Người quản trị

- Biểu đồ tuần tự cho chức năng đăng nhập 

Hình 2.9: Biểu đồ tuần tự cho chức năng đăng nhập

- Biểu đồ tuần tự cho chức năng xem danh mục bài viết

A graph with black text

Description automatically generated with medium confidence

Hình 2.10: Biểu đồ tuần tự cho chức năng xem danh mục bài viết

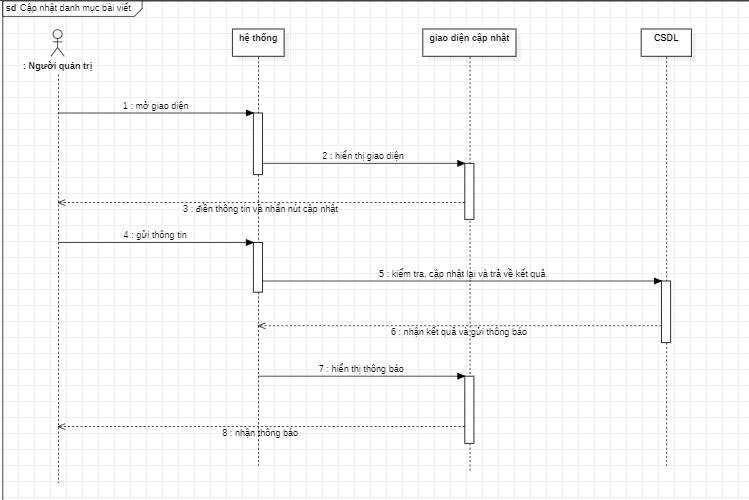
-Biểu đồ tuần tự cho chức năng thêm danh mục bài viết

A diagram of a project

Description automatically generated

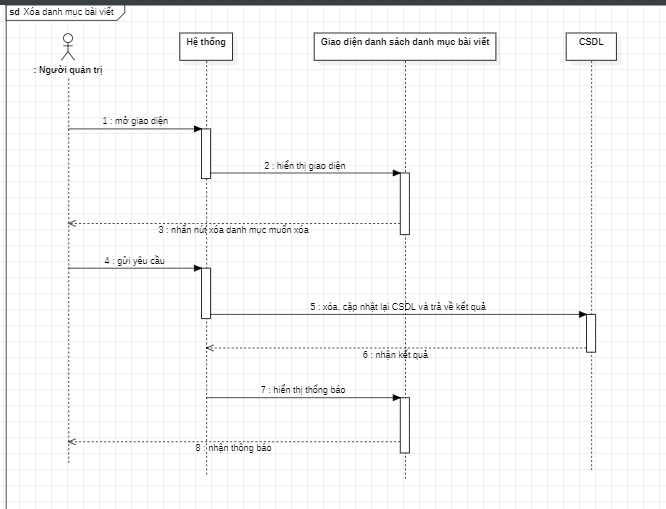
Hình 2.11: Biểu đồ tuần tự cho chức năng thêm danh mục bài viết

- Biểu đồ tuần tự cho chức năng sửa danh mục bài viết



Hình 2.12: Biểu đồ tuần tự cho chức năng sửa danh mục bài viết

- Biểu đồ tuần tự cho chức năng xóa danh mục bài viết



Hình 2.13: Biểu đồ tuần tự cho chức năng xóa danh mục bài viết

- Biểu đồ tuần tự cho chức năng xem danh sách bài viết

A diagram of a diagram

Description automatically generated

Hình 2.14: Biểu đồ tuần tự cho chức năng xem danh sách bài viết

- Biểu đồ tuần tự cho chức năng thêm bài viết

A screenshot of a computer screen

Description automatically generated

Hình 2.15: Biểu đồ tuần tự cho chức năng thêm bài viết

- Biểu đồ tuần tự cho chức năng sửa bài viết

A diagram of a project

Description automatically generated

Hình 2.16: Biểu đồ tuần tự cho chức năng sửa bài viết

- Biểu đồ tuần tự cho chức năng xóa bài viết

A diagram of a project

Description automatically generated

Hình 2.17: Biểu đồ tuần tự cho chức năng xóa bài viết

- Biểu đồ tuần tự cho chức năng xem danh mục sản phẩm

A screen shot of a diagram

Description automatically generated

Hình 2.18: Biểu đồ tuần tự cho chức năng xem danh mục sản phẩm

- Biểu đồ tuần tự cho chức năng thêm danh mục sản phẩm

A diagram of a project

Description automatically generated

Hình 2.19: Biểu đồ tuần tự cho chức năng thêm danh mục sản phẩm

- Biểu đồ tuần tự cho chức năng sửa danh mục sản phẩm

A diagram of a project

Description automatically generatedHình 2.20: Biểu đồ tuần tự cho chức năng sửa danh mục sản phẩm

* Biểu đồ tuần tự cho chức năng xóa danh mục sản phẩm

A diagram of a project

Description automatically generated

Hình 2.21: Biểu đồ tuần tự cho chức năng xóa danh mục sản phẩm

* Biểu đồ tuần tự cho chức năng xem danh sách sản phẩm

A diagram of a diagram

Description automatically generated

Hình 2.22: Biểu đồ tuần tự cho chức năng xem danh sách sản phẩm

- Biểu đồ tuần tự cho chức năng thêm sản phẩm

A diagram of a project

Description automatically generated

Hình 2.23: Biểu đồ tuần tự cho chức năng thêm sản phẩm

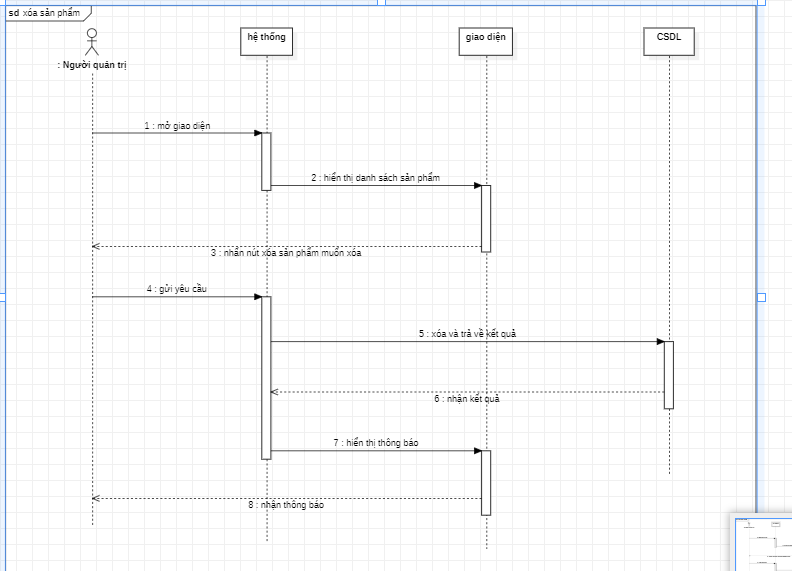
* Biểu đồ trình tự cho chức năng sửa thông tin sản phẩm

A screenshot of a computer screen

Description automatically generated

Hình 2.24: Biểu đồ trình tự cho chức năng sửa thông tin sản phẩm

- Biểu đồ tuần tự cho chức năng xóa sản phẩm



Hình 2.25: Biểu đồ tuần tự cho chức năng xóa sản phẩm

- Biểu đồ tuần tự cho chức năng xem đơn hàng

A screen shot of a diagram

Description automatically generated

Hình 2.26: Biểu đồ tuần tự cho chức năng xem đơn hàng

- Biểu đồ tuần tự cho chức năng xem chi tiết và quản lý tình trạng đơn hàng

A screenshot of a graph

Description automatically generated

Hình 2.27: Biểu đồ tuần tự cho chức năng quản lý tình trạng đơn hàng

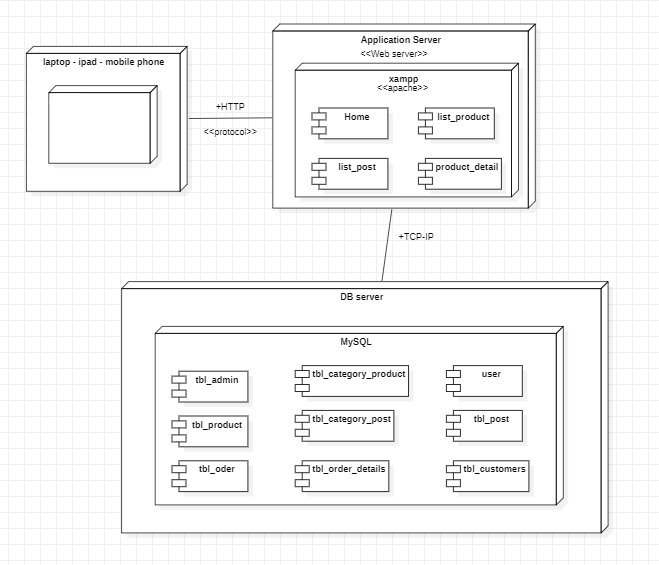
* Biểu đồ tuần tự cho chức năng Gửi liên hệ

*A diagram with text on it

Description automatically generated with medium confidence*

Hình 2.28: Biểu đồ tuần tự cho chức năng gửi liên hệ

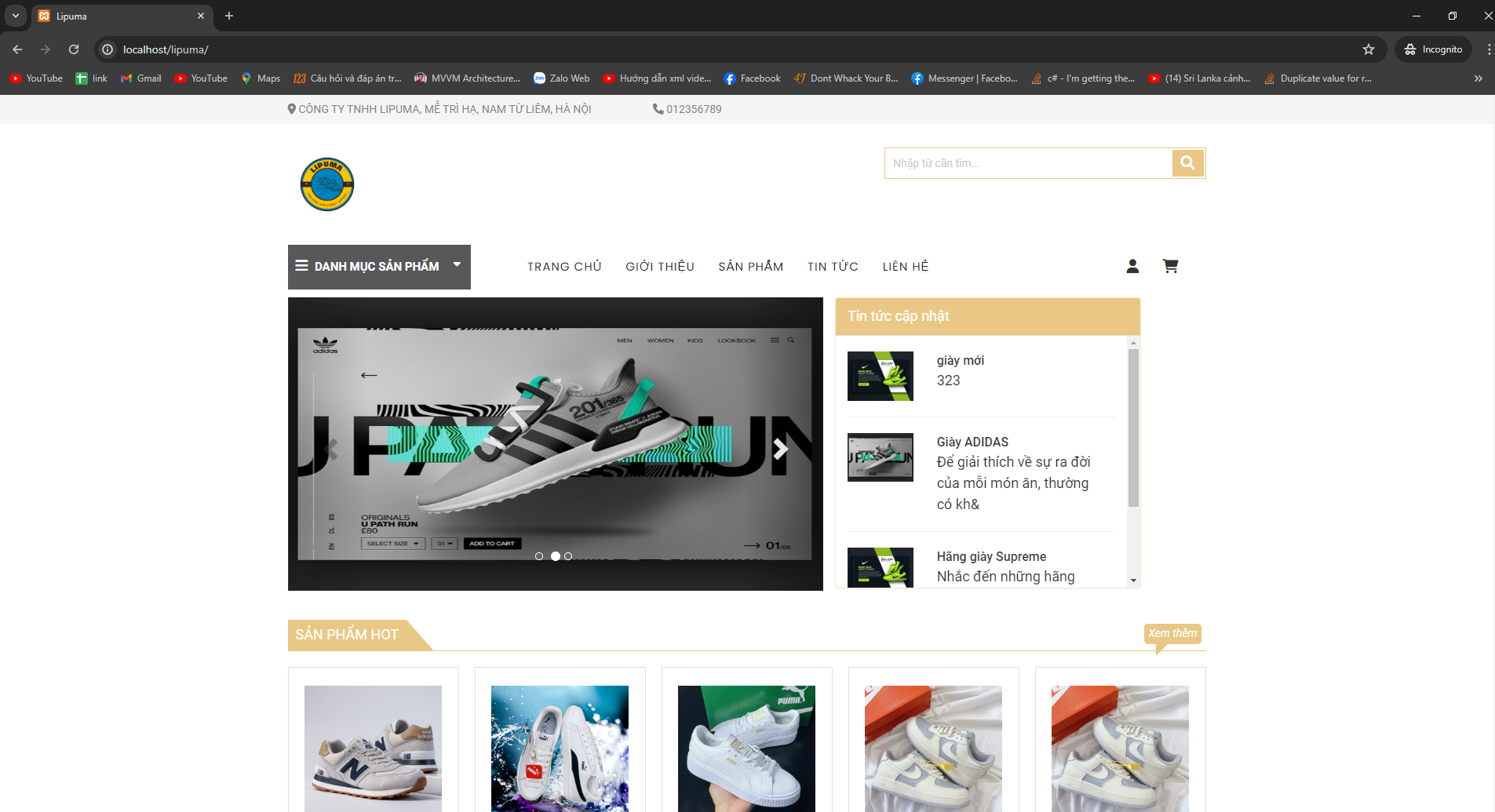
### ***Biểu đồ triển khai***



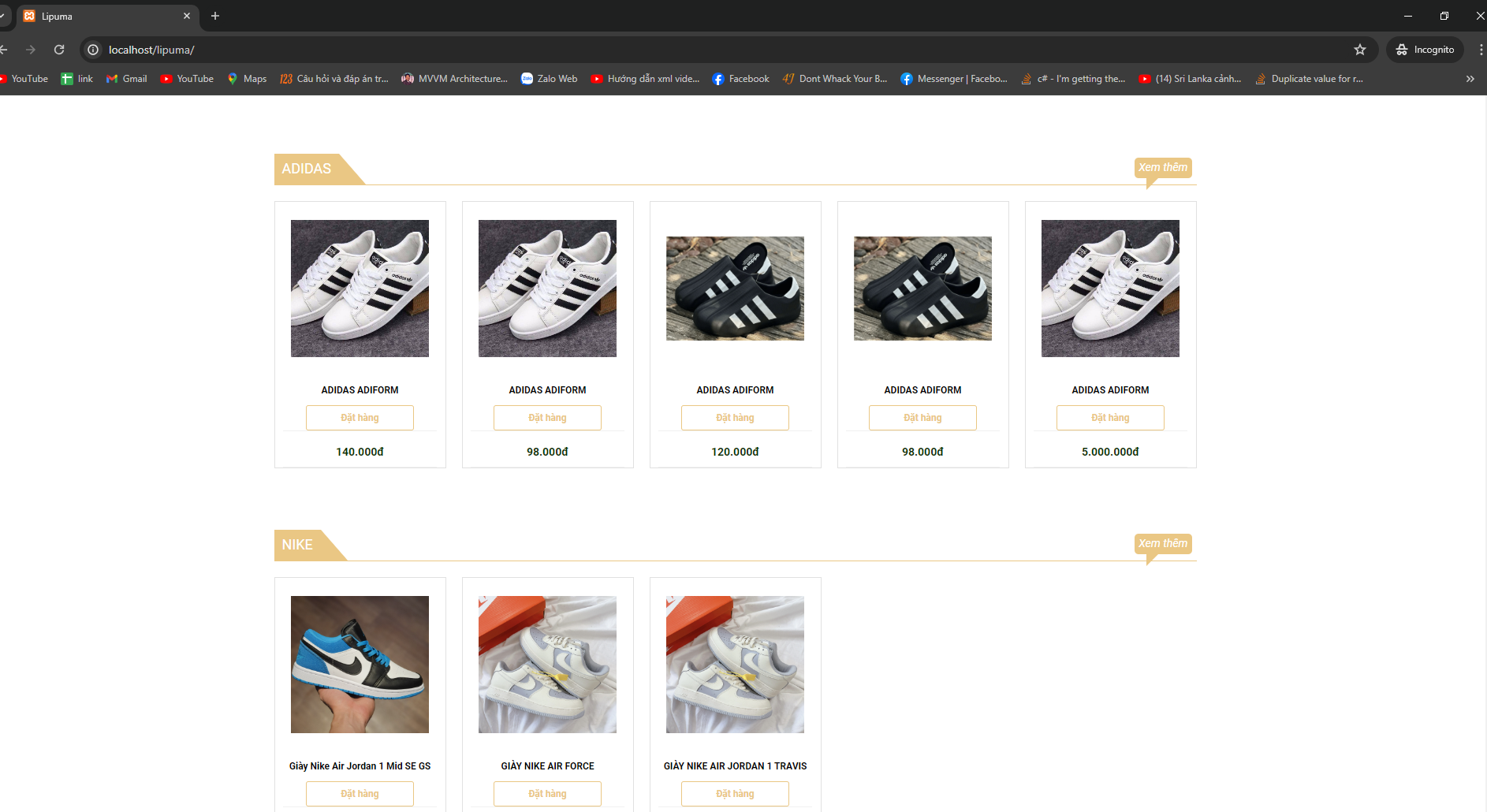
Hình 2.29: Biểu đồ triển khai

# CHƯƠNG 3: KẾT QUẢ XÂY DỰNG CHƯƠNG TRÌNH

## ***3.1. Màn hình trang chủ***

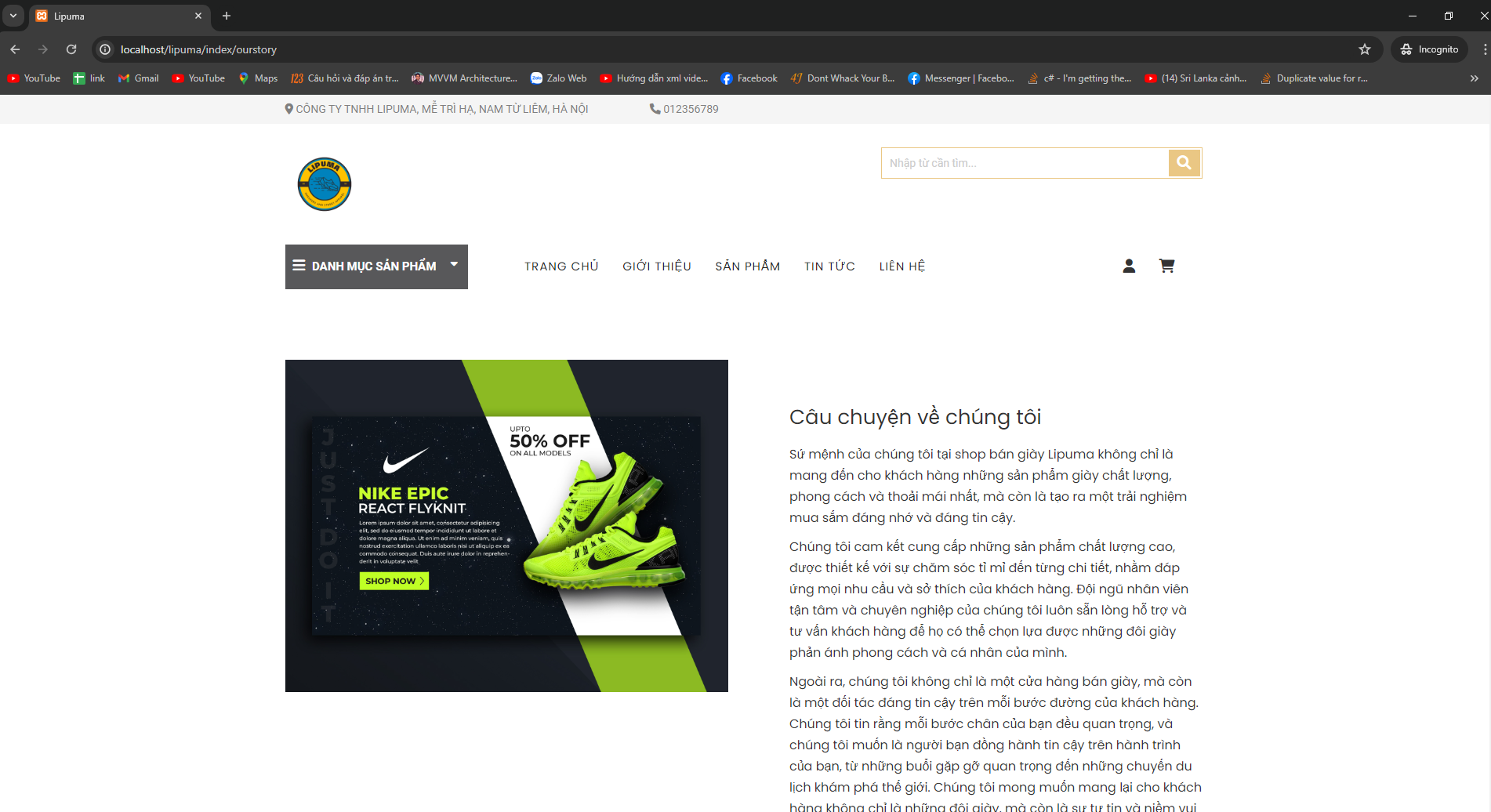


Hình 3.1: Màn hình trang chủ



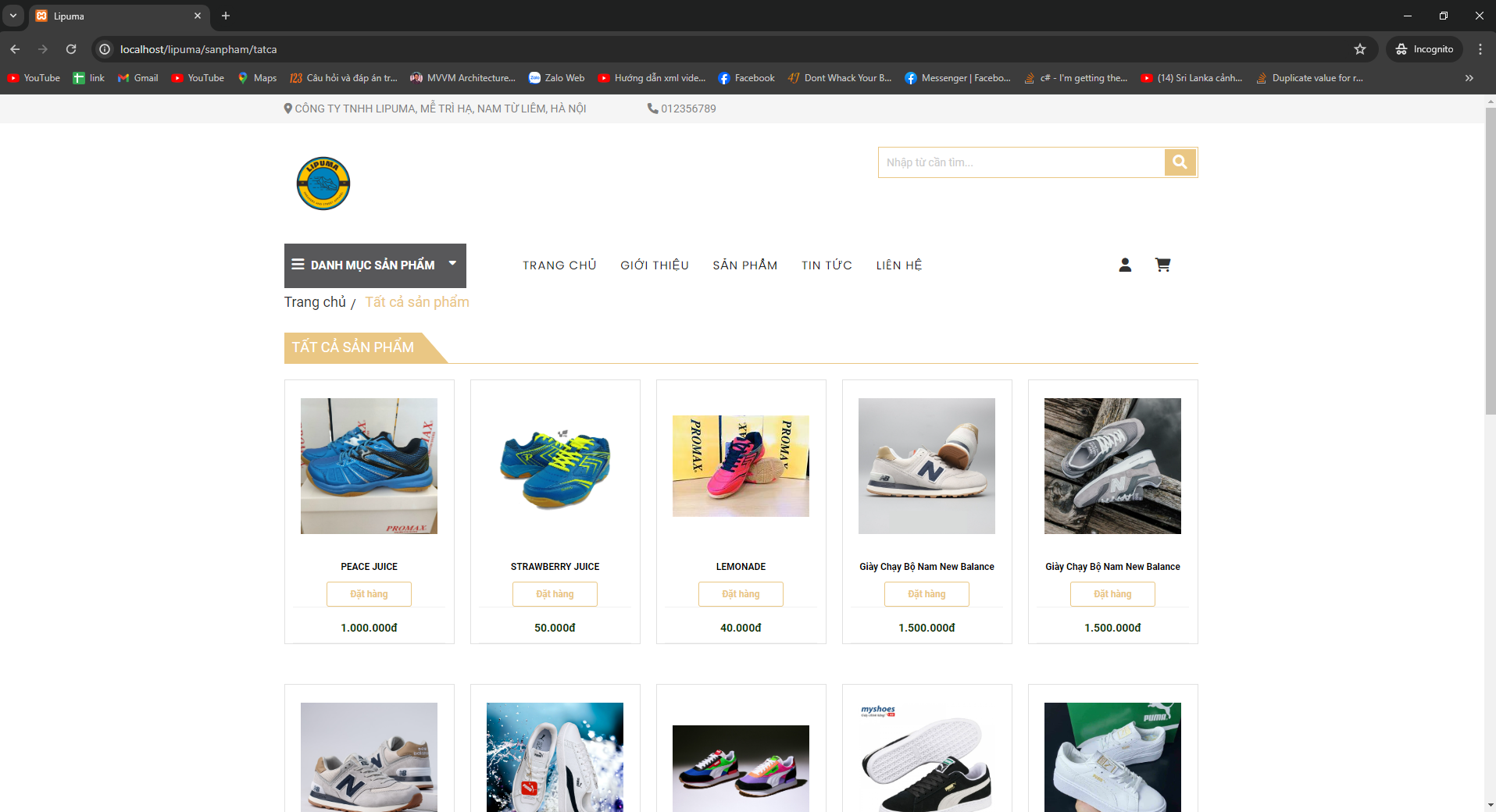
Hình 3.2: Màn hình trang chủ gồm các sản phẩm giày

## 3.2. Màn hình giới thiệu



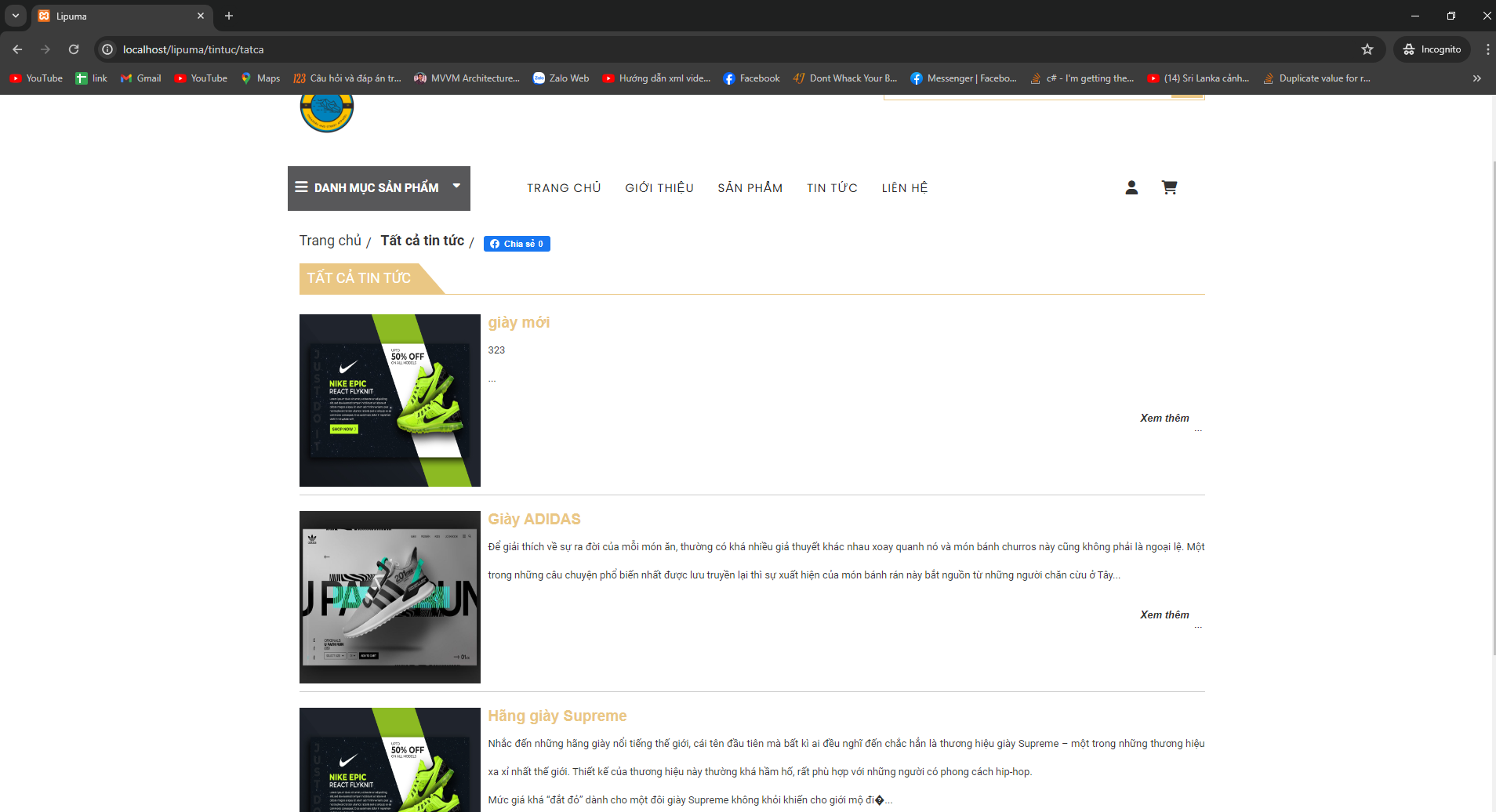
Hình 3.3: Màn hình giới thiệu

## 3.3. Màn hình tất cả sản phẩm

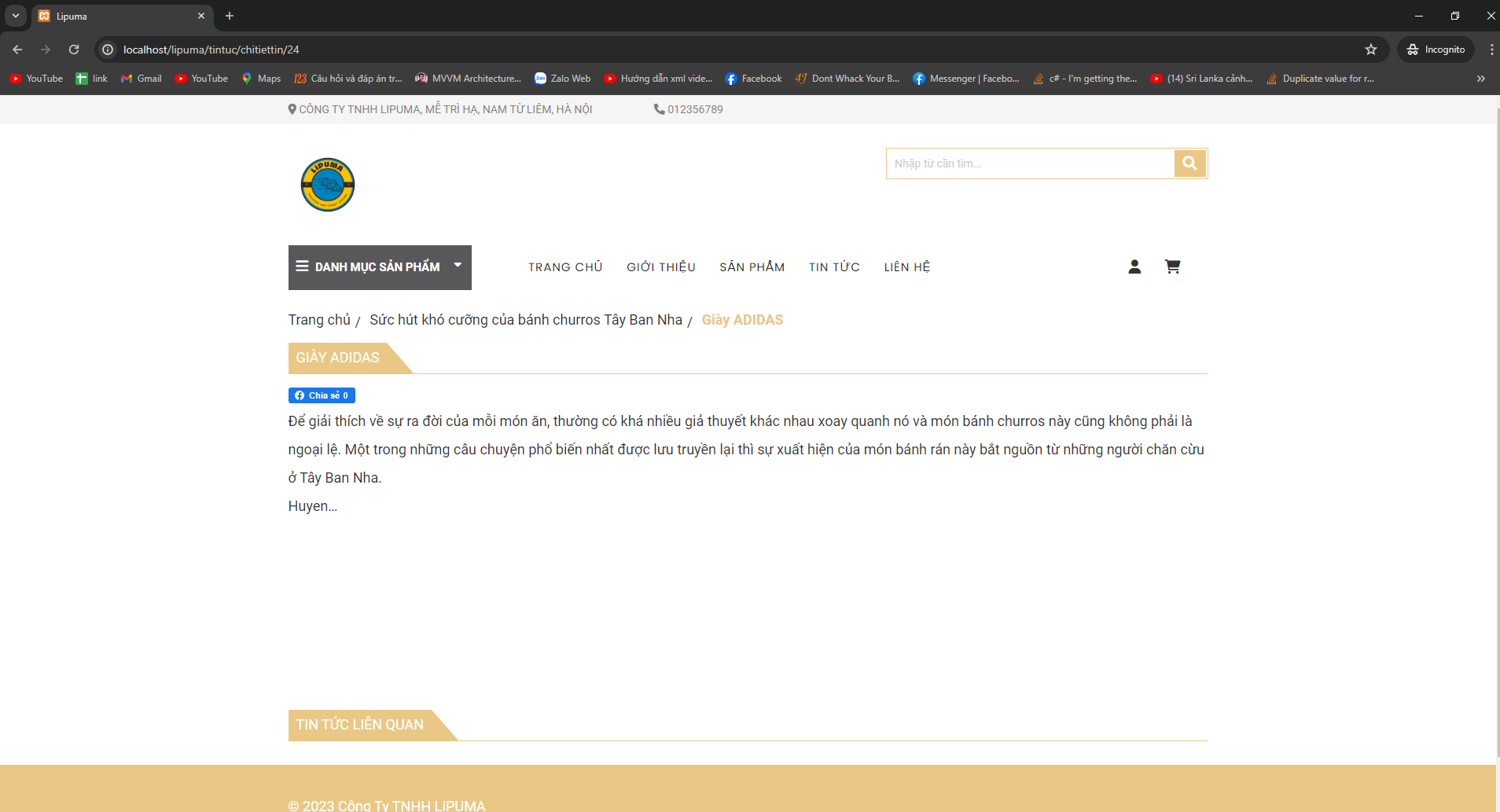


Hình 3.4 :Màn hình tất cả sản phẩm

## 3.4. Màn hình tin tức

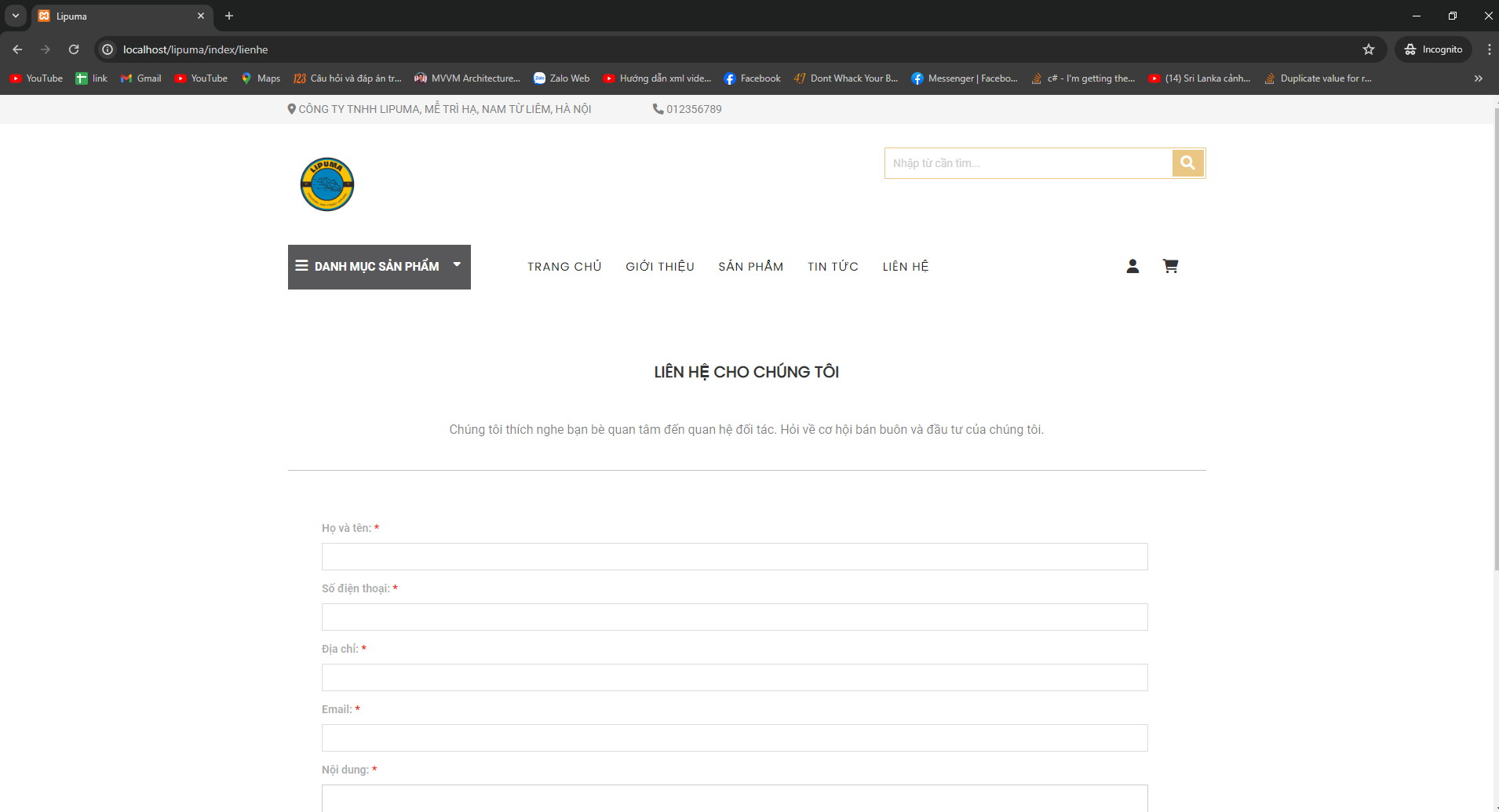


Hình 3.5: Màn hình tin tức tất cả



Hình 3.6: Màn hình tức tin cụ thể

## 3.5. Màn hình liên hệ hỗ trợ

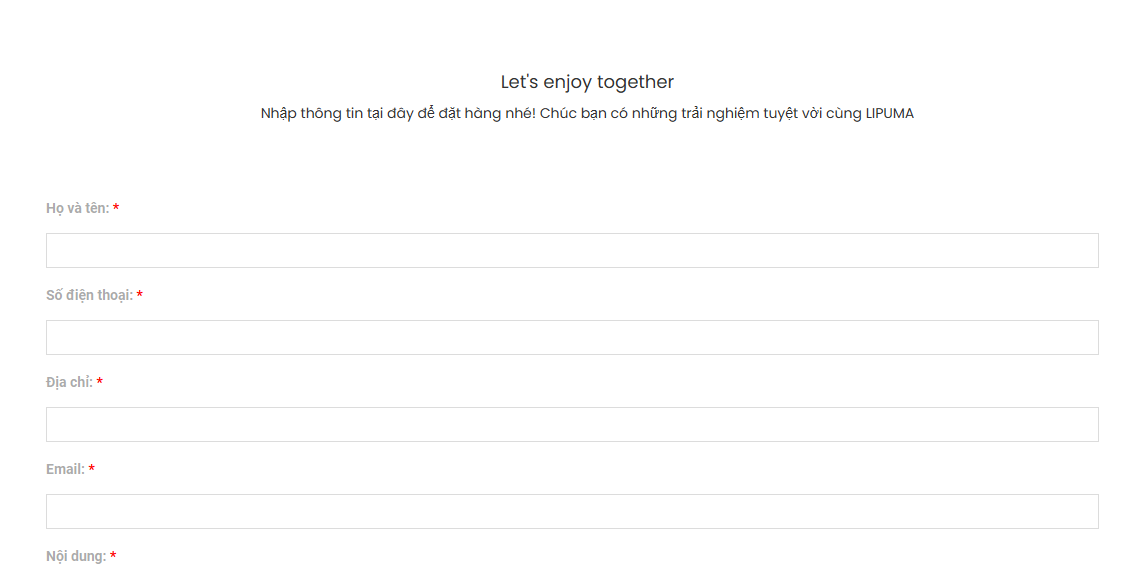


Hình 3.7: Màn hình liên hệ

## 3.6. Màn hình giỏ hàng

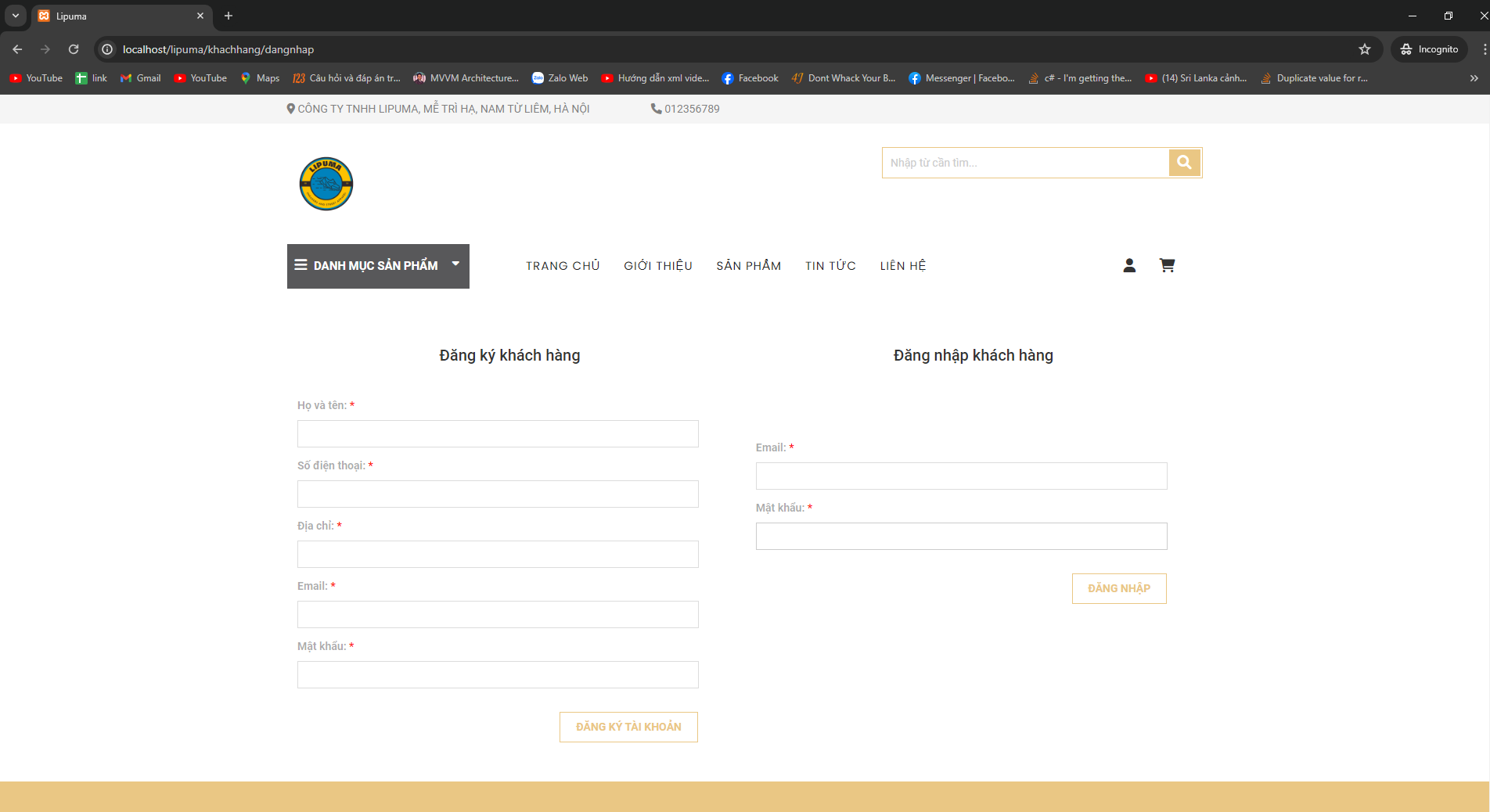


Hình 3.8: Màn hình giỏ hàng

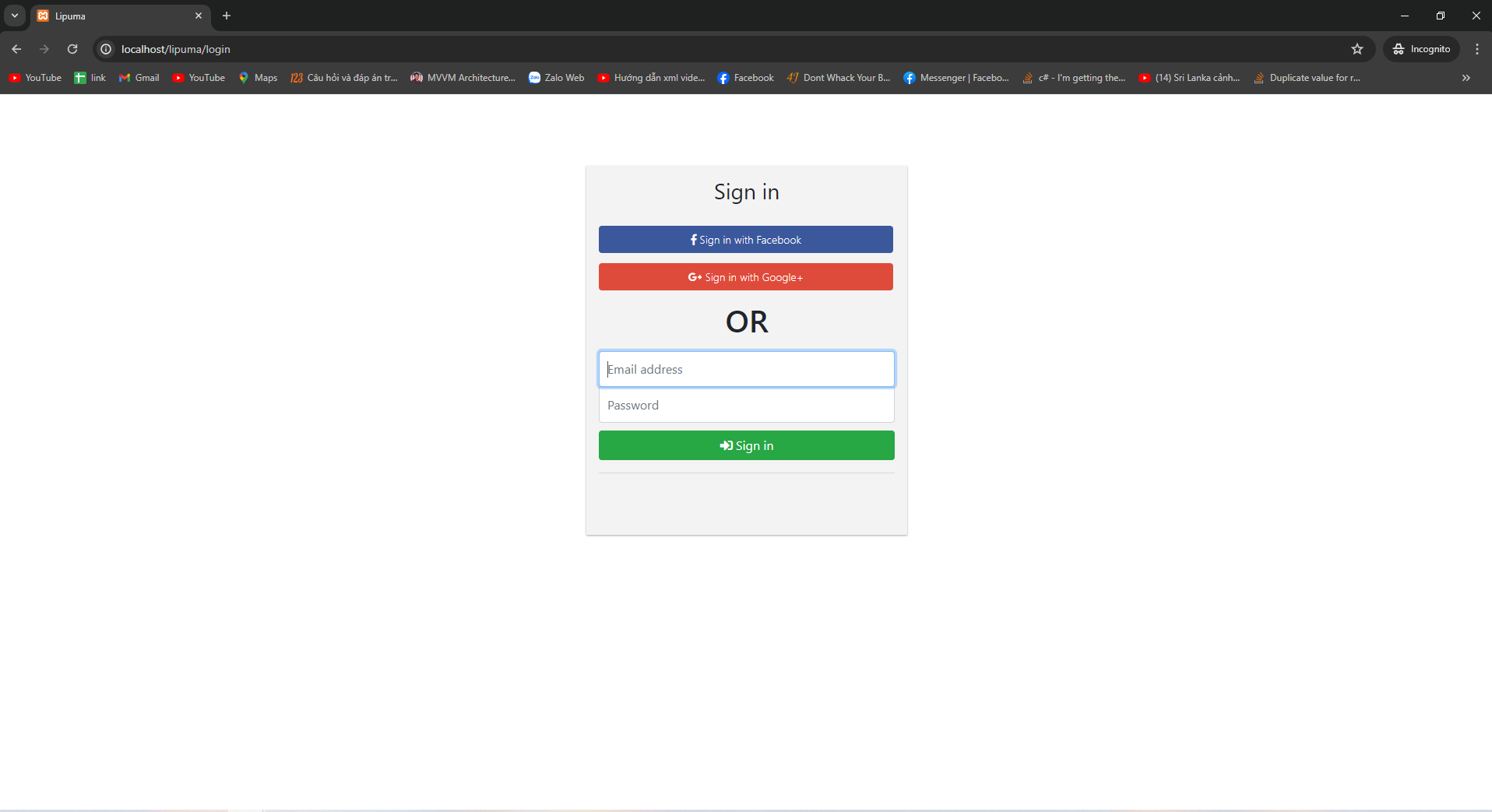


Hình 3.9: Màn hình giỏ hàng điền thông tin khách hàng

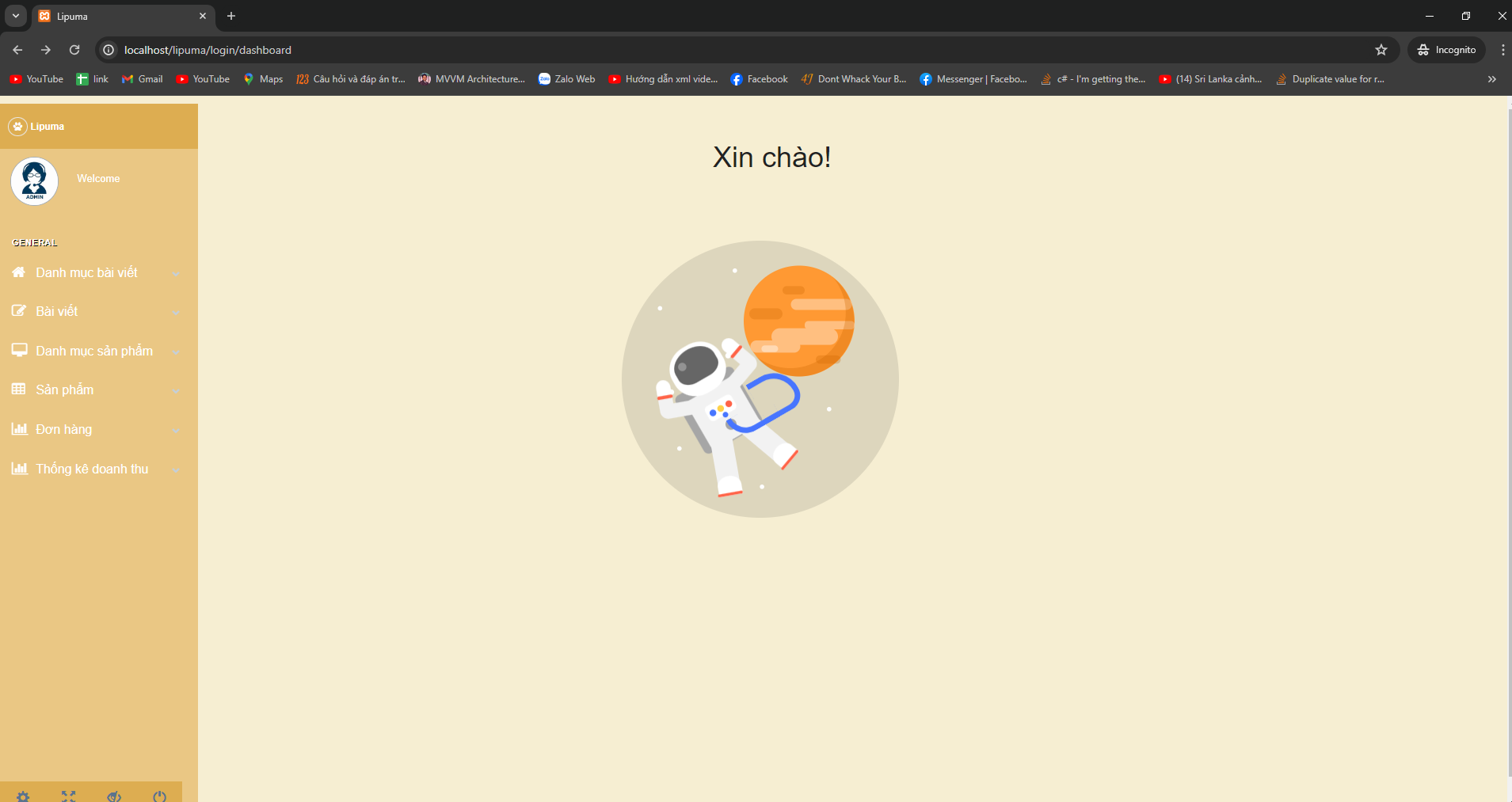
## 3.7. Màn hình đăng nhập người dùng

Hình 3.10: Màn hình đăng nhập khách hàng

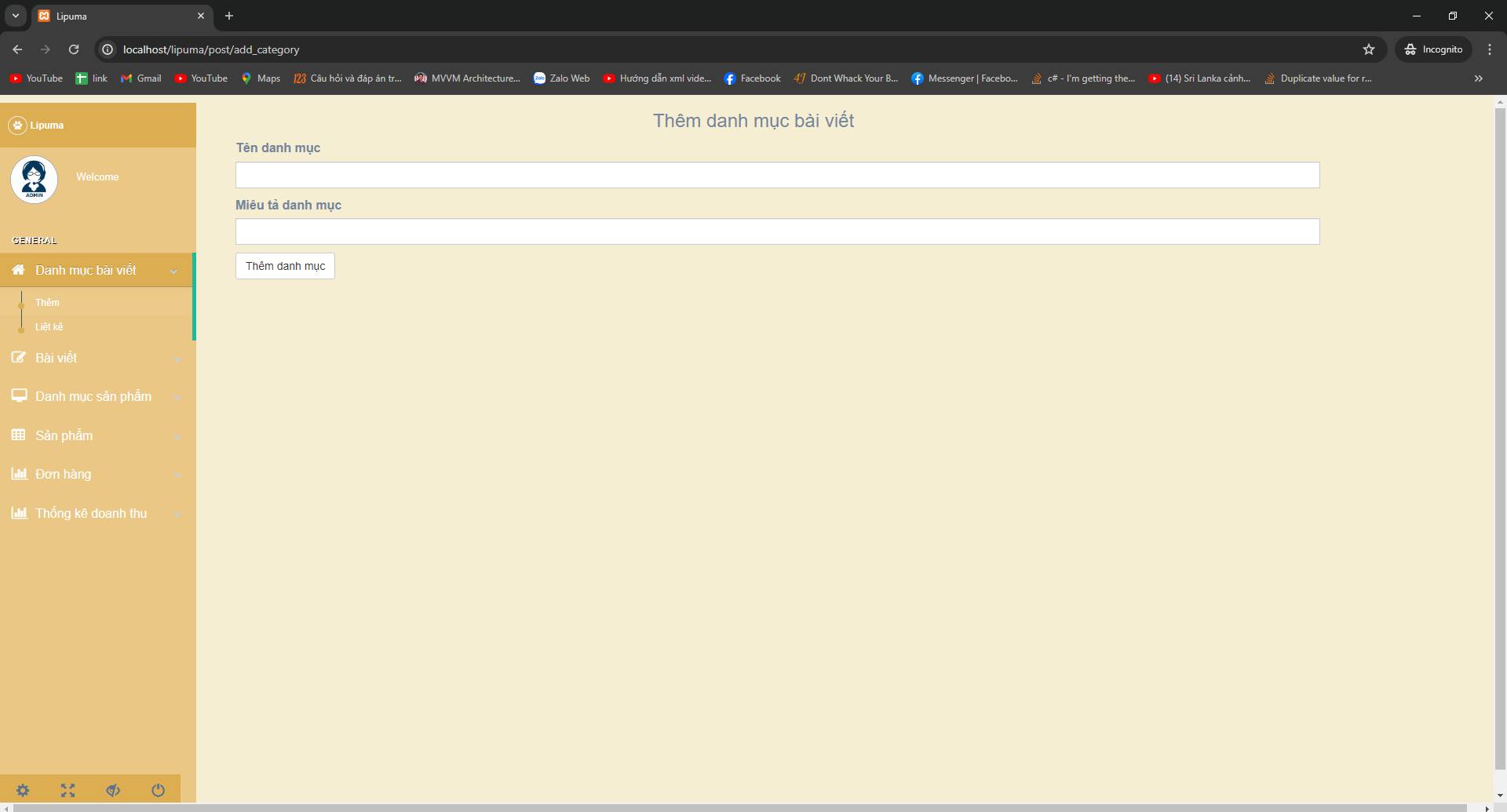
## 3.8. Màn hình đăng nhập admin



Hình 3.11: Màn hình đăng nhập admin

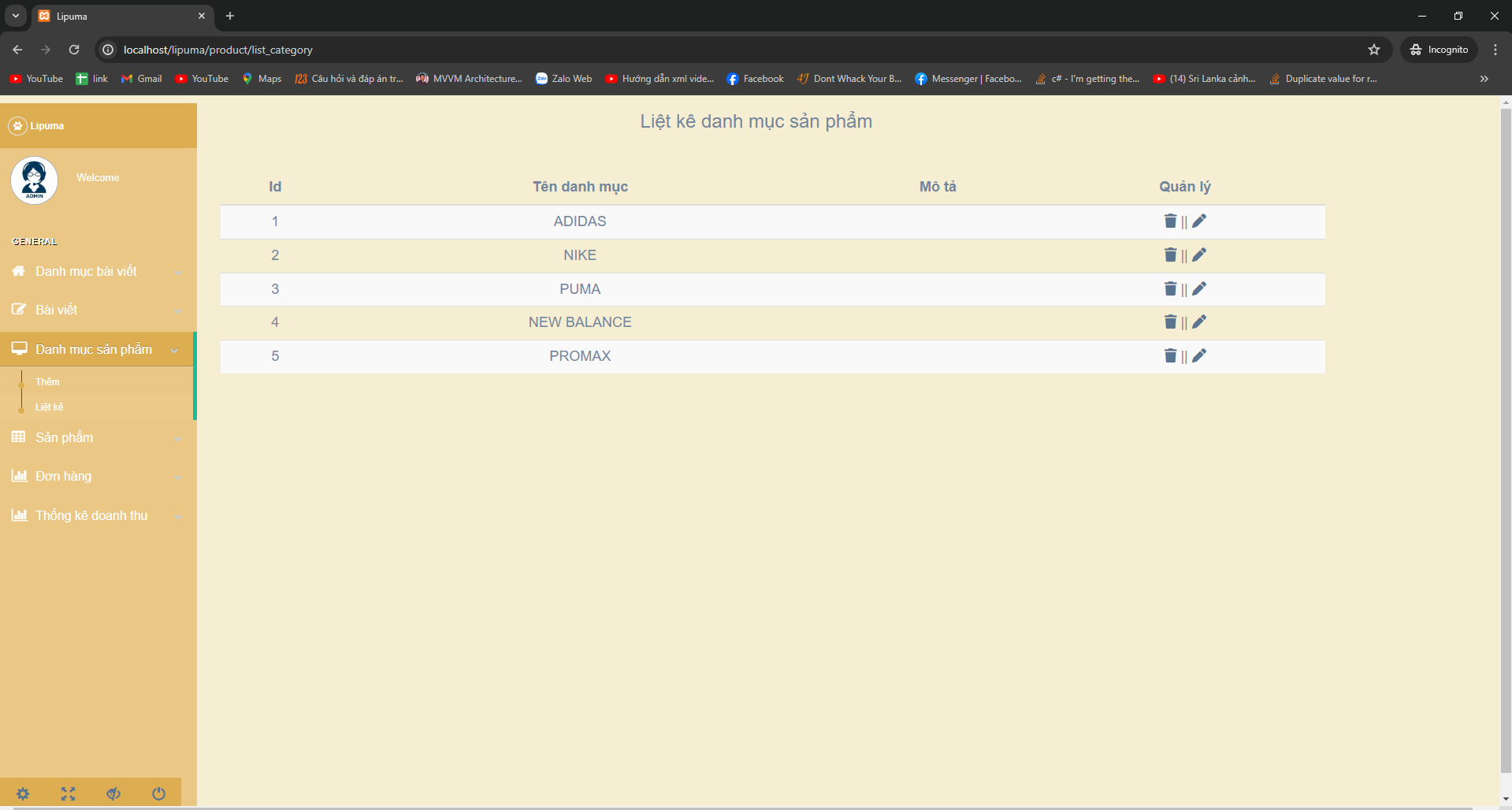
Hình 3.12: Giao diện màn hình đăng nhập admin

## 3.9. Màn hình thêm danh mục bài viết

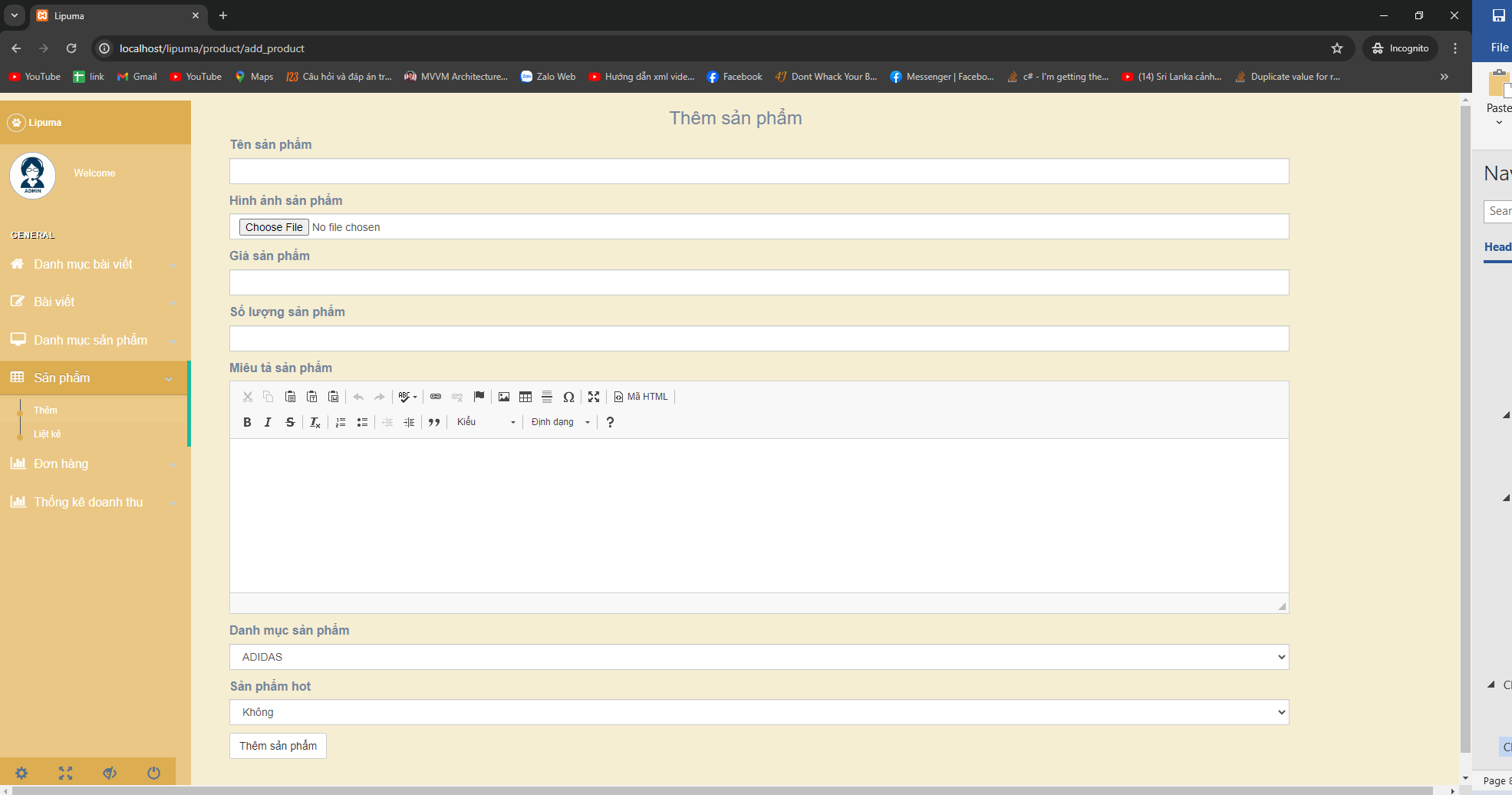


Hình 3.13: Màn hình thêm danh mục bài viết

## 3.10. Màn hình thêm danh mục sản phẩm

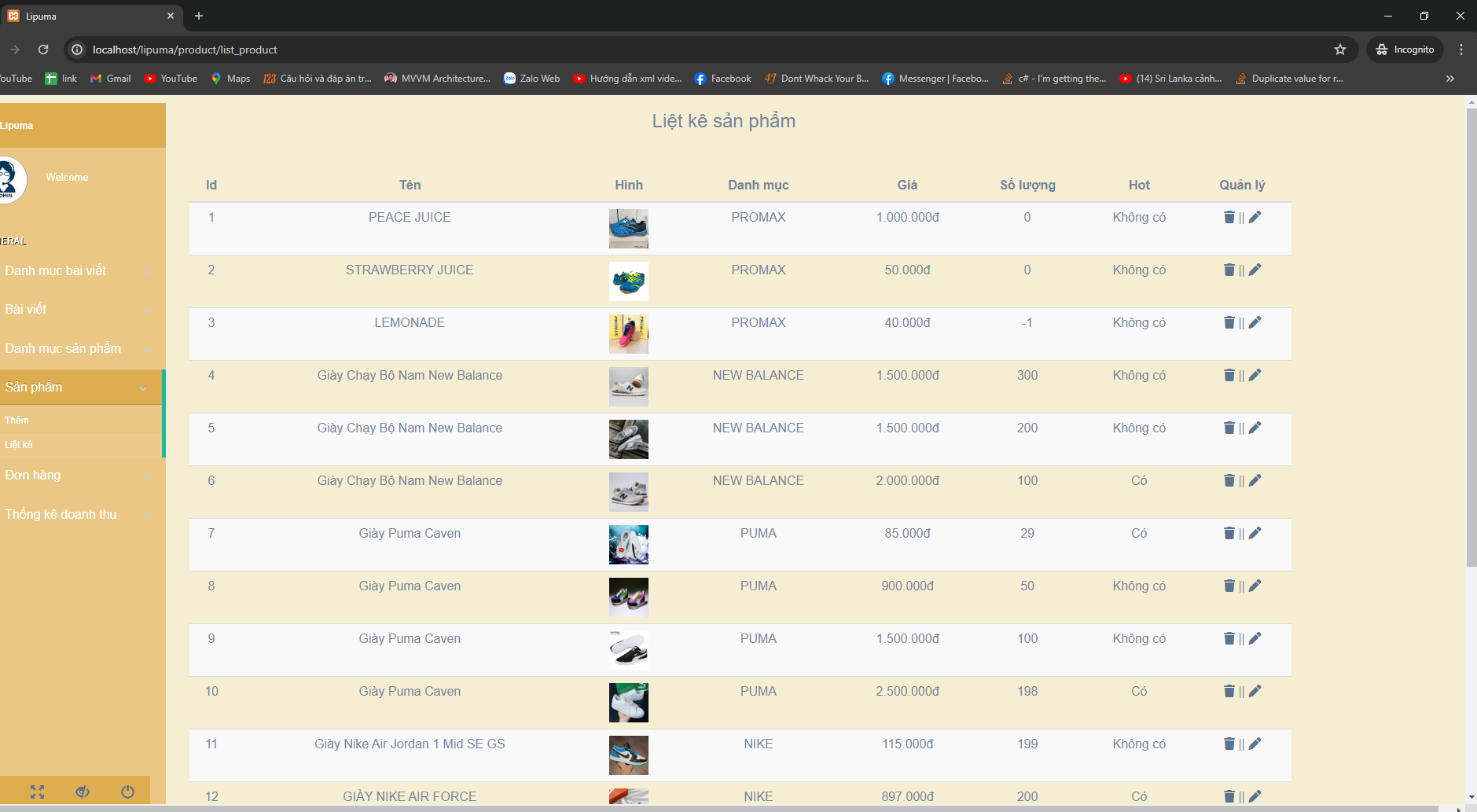


Hình 3.14: Màn hình thêm danh mục sản phẩm



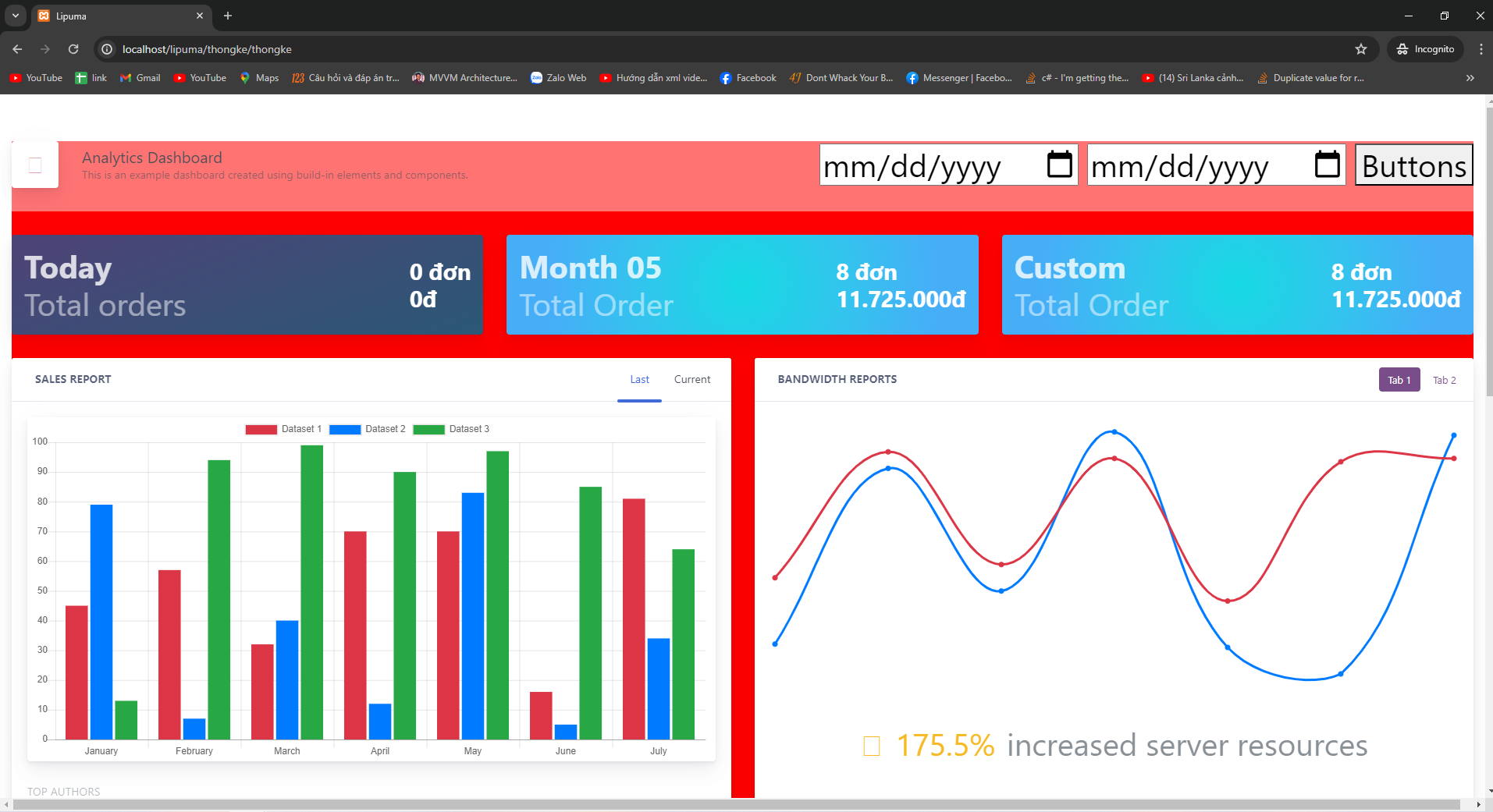
Hình 3.15: Màn hình thêm danh mục sản phẩm chi tiết

## 3.11. Màn hình liệt kê sản phẩm



Hình 3.16: Màn hình liệt kê sản phẩm

## 3.12. Màn hình thống kê doanh thu hàng tháng



Hình 3.17: Màn hình thống kê bán hàng

# CHƯƠNG 4: KẾT LUẬN

**Kết quả đạt được**

* Sau thời gian thực hiện đề tài, chương trình đã hoàn thành và đạt được một số kết quả sau:
* Tìm hiểu và nắm khá rõ về các công cụ để xây dựng website
* Biết cách thiết kế website
* Giao diện của chương trình thân thiện dễ sử dụng
* Giúp người dùng sử dụng website hiệu quả
* Tăng hiệu suất công việc
* Người dùng dễ dàng xem và đặt hàng mà không cần di chuyển tận nơi

**Hạn chế**

* Trong quá trình xây dựng website gặp nhiều lỗi phát sinh
* Các chức năng còn rất hạn chế

**Hướng phát triển**

* Phát triển website trở thành website quảng bá và bán hàng hàng đầu
* Phát triển website với các yêu cầu chức năng cao hơn, quy mô lớn hơn để nhanh chóng tiếp cận và thu hút người dùng
* Không ngừng cập nhật và nâng cấp cho website

**Đề nghị ý kiến**

* Trong thời gian thực hiện đề tài này không thể tránh khỏi những thiếu sót, em rất mong nhận được những đóng góp ý kiến từ phía quý thầy cô và các bạn để hoàn thiện hơn website này.
* Em xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

**1. Tài liệu Tiếng Việt**

[1]. https://www.mypage.vn/thiet-ke-giao-dien-web-bang-html.html

[2]. https://toidicode.com/php-co-ban

**2. Tài liệu tiếng Anh**

[1]. https://fullstack.edu.vn/

[2]. <https://www.php.net/>

[3]. https://www.w3schools.com/php/

[4]. https://stackoverflow.com/