MỤC LỤC

[CHƯƠNG 1: TỔNG QUAN 3](#_Toc519205321)

[1.1. ĐẶT VẤN ĐỀ 3](#_Toc519205322)

[1.2. GIẢI QUYẾT VẤN ĐỀ 4](#_Toc519205323)

[1.3. PHẠM VI ĐỀ TÀI 5](#_Toc519205324)

[1.4. KHẢO SÁT HIỆN TRẠNG 5](#_Toc519205325)

[1.6. PHƯƠNG PHÁP THỰC HIỆN 7](#_Toc519205326)

[1.6.1. Về lý thuyết: 7](#_Toc519205327)

[1.6.2. Về kỹ thuật: 7](#_Toc519205328)

[1.6.3. Các công cụ thực hiện: 8](#_Toc519205329)

[1.6.4. Cài đặt và kiểm tra lỗi: 8](#_Toc519205330)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 9](#_Toc519205331)

[2.1. HTML 9](#_Toc519205332)

[2.2. CSS 9](#_Toc519205333)

[2.3. AJAX & JQUERY 9](#_Toc519205334)

[2.3.1. Ajax 9](#_Toc519205335)

[2.3.2. jQuery 10](#_Toc519205336)

[2.4. HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL 10](#_Toc519205337)

[2.5. PHP VÀ PHP FRAMEWORK 11](#_Toc519205338)

[2.6. MÔ HÌNH MVC 12](#_Toc519205339)

[2.6.1. Khái niệm mô hình MVC 12](#_Toc519205340)

[2.6.2. Các thành phần trong mô hình MVC 13](#_Toc519205341)

[2.6.3. Quy trình hoạt động của mô hình MVC 13](#_Toc519205342)

[2.6.4. Ưu và nhược điểm của mô hình MVC 15](#_Toc519205343)

[2.7. LARAVEL FRAMEWORK 16](#_Toc519205344)

[2.7.1. Giới thiệu về Laravel 16](#_Toc519205345)

[2.7.2. Các tính năng của Laravel 17](#_Toc519205346)

[2.7.3. Biểu đồ các php framework sử dụng trong hai năm gần đây 18](#_Toc519205347)

[2.7.4. Tại sao nên chọn Laravel 18](#_Toc519205348)

[2.7.5. Các chặn đường phát triển 19](#_Toc519205349)

[2.7.6. Yêu cầu máy chủ để chạy Laravel 20](#_Toc519205350)

[2.7.7. Ưu và nhược điểm của Laravel 21](#_Toc519205351)

[2.7.8. Hướng dẫn cài đặt Laravel 21](#_Toc519205352)

[2.7.9. Một số lệnh console phổ biến được cung cấp của Laravel 22](#_Toc519205353)

[2.7.10. Cấu trúc thư mục của laravel 25](#_Toc519205354)

[2.7.11. Hướng dẫn tạo controller – model –view trong Laravel 27](#_Toc519205355)

[CHƯƠNG 3: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU 31](#_Toc519205356)

[3.1. ĐẶT TẢ HỆ THỐNG 31](#_Toc519205357)

[3.1.1. Giới thiệu hệ thống 31](#_Toc519205358)

[3.1.2. Mô tả hệ thống 31](#_Toc519205359)

[3.2. THIẾT KẾ CÁC MÔ HÌNH 33](#_Toc519205360)

[3.2.1. Mô hình phân rã chức năng người dùng 33](#_Toc519205361)

[3.2.2. Mô hình phân rã chức năng người quản lý 33](#_Toc519205362)

[3.2.3. Mô hình Use Case 34](#_Toc519205363)

[3.2.4. Flowchart 39](#_Toc519205364)

[3.3. THIẾT KẾ CƠ SỞ DỮ LIỆU 43](#_Toc519205365)

[3.3.1. Mô hình cơ sở dữ liệu 43](#_Toc519205366)

[3.3.2. Mô tả dữ liệu 43](#_Toc519205367)

[3.4. KẾT QUẢ XÂY DỰNG HỆ THỐNG 52](#_Toc519205368)

[3.4.1. Một số giao diện chính từ trang người dùng 52](#_Toc519205369)

[3.4.2. Một số giao diện chính từ trang quản lý 63](#_Toc519205370)

[CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 72](#_Toc519205371)

[4.1. KẾT QUẢ ĐẠT ĐƯỢC 72](#_Toc519205372)

[4.2. ƯU ĐIỂM VÀ HẠN CHẾ CỦA ĐỀ TÀI 72](#_Toc519205373)

[4.3. HƯỚNG PHÁT TRIỂN 73](#_Toc519205374)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1: Mô hình MVC 13](#_Toc518663311)

[Hình 2.2: Quy trình hoạt động của mô hình MVC 15](#_Toc518663312)

[Hình 2.3: Biểu đồ các framework sử dụng qua các năm 18](#_Toc518663313)

[Hình 2.4: Cấu trúc class controller của Laravel 28](#_Toc518663314)

[Hình 2.5: Cấu trúc class model của Laravel 29](#_Toc518663315)

[Hình 2.6: Cấu trúc file view của Laravel 29](#_Toc518663316)

[Hình 2.7: Cách gọi controller trong route của Laravel 30](#_Toc518663317)

[Hình 3.1: Mô hình phân rã chức năng người dùng 33](#_Toc518663318)

[Hình 3.2: Mô hình phân rã chức năng người quản lý 34](#_Toc518663319)

[Hình 3.3: Mô hình use case chức năng khách hàng chưa đăng nhập 35](#_Toc518663320)

[Hình 3.4: Mô hình use case khi khách hàng đăng nhập 36](#_Toc518663321)

[Hình 3.5: Mô hình use case chức năng nhân viên 37](#_Toc518663322)

[Hình 3.6: Mô hình use case chức năng người quản lý 38](#_Toc518663323)

[Hình 3.7: Sơ đồ flowchart chức năng thêm mới 39](#_Toc518663324)

[Hình 3.8: Sơ đồ flowchart chức năng chỉnh sửa 40](#_Toc518663325)

[Hình 3.9: Sơ đồ flowchart chức năng xóa 41](#_Toc518663326)

[Hình 3.10: Sơ đồ flowchart chức năng tìm kiếm 42](#_Toc518663327)

[Hình 3.11: Mô hình cơ sở dữ liệu 43](#_Toc518663328)

[Hình 3.12: Giao diện trang chủ của website 53](#_Toc518663329)

[Hình 3.13: Cửa sổ đăng nhập website 54](#_Toc518663330)

[Hình 3.14: Giao diện trang đăng ký tài khoản 55](#_Toc518663331)

[Hình 3.15: Giao diện trang chi tiết sản phẩm 56](#_Toc518663332)

[Hình 3.16: Giao diện trang thể loại hay nhóm sản phẩm 57](#_Toc518663333)

[Hình 3.17: Giao diện trang giỏ hàng 58](#_Toc518663334)

[Hình 3.18: Giao diện thanh toán đơn hàng (đã đăng nhập) 59](#_Toc518663335)

[Hình 3.19: Giao diện trang thanh toán đơn hàng (chưa đăng nhập) 60](#_Toc518663336)

[Hình 3.20: Giao diện trang tìm kiếm sản phẩm 61](#_Toc518663337)

[Hình 3.21: Giao diện trang chi tiết tin tức 62](#_Toc518663338)

[Hình 3.22: Giao diện trang sản phẩm yêu thích 63](#_Toc518663339)

[Hình 3.23: Giao diện trang đăng nhập quản lý 64](#_Toc518663340)

[Hình 3.24: Giao diện trang chủ trang admin 64](#_Toc518663341)

[Hình 3.25: Giao diện trang thêm sản phẩm 65](#_Toc518663342)

[Hình 3.26 Giao diện trang danh sách sản phẩm 66](#_Toc518663343)

[Hình 3.27: Giao diện trang chi tiết sản phẩm 67](#_Toc518663344)

[Hình 3.28: Giao diện trang thêm tin tức 68](#_Toc518663345)

[Hình 3.29: Giao diện trang nhân viên 69](#_Toc518663346)

[Hình 3.30: Giao diện đơn đặt hàng 70](#_Toc518663347)

[Hình 3.31: Giao diện chi tiết đơn hàng 70](#_Toc518663348)

[Hình 3.32: Giao diện mail thông báo đơn hàng 71](#_Toc518663349)

**DANH MỤC BẢNG**

[Bảng 1.1: So sánh một số website bán hàng online 7](#_Toc519262332)

[Bảng 1.2: Bảng các công cụ thực hiện 8](#_Toc519262333)

[Bảng 2.1: Một số lệnh console phổ biển được cung cấp của Laravel 23](#_Toc519262334)

[Bảng 3.1: Thực thể categories 43](#_Toc519262335)

[Bảng 3.2: Thực thể SubCategory 44](#_Toc519262336)

[Bảng 3.3: Thực thể users 44](#_Toc519262337)

[Bảng 3.4: Thực thể products 45](#_Toc519262338)

[Bảng 3.5: Thực thể product\_detail 46](#_Toc519262339)

[Bảng 3.6: Thực thể product\_colors 47](#_Toc519262340)

[Bảng 3.7: Thực thể order 48](#_Toc519262341)

[Bảng 3.8: Thực thể order\_detail 49](#_Toc519262342)

[Bảng 3.9: Thực thể staff 49](#_Toc519262343)

[Bảng 3.20: Thực thể wishlist 50](#_Toc519262344)

[Bảng 3.31: Thực thể promotion 51](#_Toc519262345)

[Bảng 3.42: Thực thể news 51](#_Toc519262346)

**DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| **Ký hiệu/ chữ viết tắt** | **Diễn giải** |
| DHTML | Là viết tắt của Dynamic HTML là một thể hiện của việc tạo ra một trang web bằng cách kết hợp các thành phần: ngôn ngữ đánh dấu HTML tĩnh, ngôn ngữ kịch bản máy khách (như là Javascript), và ngôn ngữ định dạng trình diễn Cascading Style Sheets và Document Object Model (DOM). |
| DOM | Là chữ viết tắt của Document Object Model ("Mô hình Đối tượng Tài liệu"), là một giao diện lập trình ứng dụng. Thường thường DOM, có dạng một cây cấu trúc dữ liệu, được dùng để truy xuất các tài liệu dạng HTML và XML. |
| ER | Chữ viết tắt của Entity Relationshop Model được CHEN giới thiệu vòa năm 1976. Đây là một mô hình được sử dụng rộng rãi trong các bản thiết kế cơ sở dữ liệu ở mức quan niệm. |
| HMVC | Viết tắt Hierarchical Model-View-Controller là một mô hình cấu trúc vô cùng mạnh mẽ. Được thiết kế tối ưu hóa để áp dụng cho các hệ thống lớn. Nó giúp khả năng mở rộng vả quản lý hệ thống rất đơn giản. |
| HTTP | Là chữ viết tắt của HyperText Transfer Protocol (giao thức truyền tải siêu văn bản). Đây là một giao thức ứng dụng trong bộ các giao thức TCP/IP (gồm một nhóm các giao thức nền tảng cho internet). HTTP hoạt động dựa trên mô hình Client |
| IP | Là viết tắt của địa chỉ Internet Protocol address (địa chỉ giao thức Internet). Mỗi thiết bị được kết nối vào mạng (như mạng Internet) cần có một địa chỉ. ... Tương tự, địa chỉ IP là một dãy số xác định máy tính để có thể gửi nhận dữ liệu đến các máy khác. |
| PHP | Viết tắt của Hypertext Preprocessor, là một ngôn ngữ lập trình để phát triển ứng dụng web. |
| MVC | Chữ viết tắt của Model – View – Controller là một kiến trúc phần mềm trong kỹ thuật phần mềm. |
| SGLM | Viết tắt của Standard Generalized Markup Language, là một hệ thống tổ chức và gắn thẻ yếu tố của một tài liệu. SGML được phát triển và tiêu chuẩn hóa bởi Tổ chức Tiêu chuẩn quốc tế (ISO) vào năm 1986 |
| XHTML | Viết tắt của Extensible HyperText Markup Language, "Ngôn ngữ Đánh dấu Siêu văn bản Mở rộng" là một ngôn ngữ đánh dấu có cùng các khả năng như HTML, nhưng có cú pháp chặt chẽ hơn. |

# 

**TÓM TẮT**

Công nghệ thông tin ngày càng phát triển mạnh mẽ và nhanh chóng, nó đã được ứng dụng rộng rãi trong nhiều lĩnh vực khoa học, kỹ thuật và đời sống. Phục vụ mạnh mẽ trong nhiều ngành nghề. Một trong số đó là lĩnh vực thời trang

Các cửa hàng thời trang mọc lên ngày càng nhiều nhưng vẫn tồn tại theo kiểu buôn bán truyền thống chưa tận dụng hết sức mạnh của công nghệ thông tin vào trong việc mua bán và quản lý hàng hóa

Do đó nhu cầu đặt ra là phải xây dựng một website bán quần áo online giúp cho người dùng không cần phải đến cửa hàng vẫn có thể mua hàng, giúp tiết kiệm thời gian chi phí cho cả người mua và bán. Giúp cửa hàng có thể giới thiệu sản phẩm đến khách hàng và quản lý sản phẩm của mình.

Từ những lý do cần thiết trên, sau quá trình học tập và nghiên cứu em quyết định chọn đề tài ***“Xây dựng website quần áo trực tuyến”*** trên nền tảng Laravel framework***,*** trong quá trình làm đề tài em sẽ nghiên cứu trình bày và tìm hiểu về Laravel framework vào thực tế.

Website bán quân áo trực tuyến được xây dựng nhằm đáp ứng các yêu cầu phục vụ các chức năng cơ bản về bán hàng như:

- Quản lý thông tin sản phẩm.

- Khách hàng mua hàng trực tuyến.

- Quản lý hóa đơn, lịch sử mua hàng của khách hàng.

- Khách hàng đánh giá, bình luận sản phẩm.

- Quản lý thống kê doanh thu và lợi nhuận.

- Và một số chức năng khác cần thiết cho khách hàng và quản lý.

Website bán hàng quần áo online được xây dựng trên nền tảng Laravel Framework 5.6 kết hợp các ngôn ngữ về web: PHP 7. HTML, Javascript, jQuery, Ajax… Kết hợp sử dụng hệ quản trị cơ sở dữ liệu MySQL và Apache Server.

Nội dung đề tài gồm 4 chương chính:

**Chương 1: Tổng quan**: Đặt vấn đề, giải quyết vấn đề, mục tiêu và phạm vi của đề tài.

**Chương 2: Cơ sở lý thuyết**: Giới thiệu về PHP framework, Ajax, jQuery, mô hình MVC và Laravel framework.

**Chương 3: Nội dung và kết quả nghiên cứu**: Thu thập thông tin và xây dựng các sơ đồ: ER, Use Case, cơ sở dữ liệu quan hệ… Trình bày kết quả nghiên cứu, nêu lên những kết quả đạt được, những phần còn hạn chế và hướng phát triển của đề tài.

**Chương 4: Kết luận và hướng phát triển**

# CHƯƠNG 1: TỔNG QUAN

## 1.1. ĐẶT VẤN ĐỀ

Sự phát triển bùng nổ của công nghệ thông tin và sự phát triển này ngày càng tăng mạnh, cuộc sống của mọi người cũng ngày càng được tiến bộ theo hướng công nghệ. Các sản phẩm về công nghệ thông tin ngày càng tăng mạnh, sự ra đời của hệ thống mua sắm trực tuyến – thương mại điện tử cũng từ đó tăng theo đánh dấu bước phát triển vượt bậc của website.

Internet ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực, vì internet có những đặc điểm nổi bậc như không giới hạn về không gian, thời gian hay khu vực các thông tin trên internet được cập nhật nhanh chóng, liên tục và chính xác. Mà trong thời đại phát triển của cuộc sống hiện nay rất cần thiết những yếu tố trên vì nó được ưu tiên cao nhất trong nền kinh tế hiện nay.

Ở thế giới nói chung và ở Việt Nam chúng ta nói riêng Internet đang có một sự phát triển khá nhanh chóng, các dịch vụ internet ngày càng phổ biến. Hầu hết người dùng sử dụng internet nhằm mục đích tra cứu thông tin, tìm kiếm tài liệu, mua sắm trực tuyến…

Các cửa hàng buôn bán quần áo xuất hiện ngày càng nhiều và cạnh tranh lẫn nhau, vì thế mà mỗi cửa hàng cần có một hệ thống bán hàng không chỉ tại cửa hàng (thủ công) mà phải theo hướng công nghệ (bán hàng trực tuyến). Bán hàng thủ công người mua hàng chỉ có thể mua hàng khi đến trực tiếp cửa hàng để có thể chọn sản phẩm rất bất tiện cho khách hàng ở xa muốn mua hàng. Tuy nhiên nếu cửa hàng đã áp dụng công nghệ thông tin vào việc bán hàng trực tuyến trên hệ thống website thì việc bán hàng ở những nơi khách hàng không thể đến cửa hàng sẽ mang lại lợi thế cho cửa hàng. Ngoài ra khách hàng còn có thể xem thông tin chi tiết từng sản phẩm một cách dễ dàng trước khi chọn cho mình một sản phẩm phù hợp.

Dựa trên những phân tích trên cùng những lợi ích mà hệ thống bán hàng online mang lại, em chọn đề tài **Xây dựng website bán quần áo online**

## 1.2. GIẢI QUYẾT VẤN ĐỀ

Trên thực tế việc bán hàng theo hình thức tại cửa hàng gặp rất nhiều khó khăn ngay cả đối với cửa hàng và khách hàng:

**Đối với khách hàng:**

- Khó khăn trong việc tìm kiếm sản phẩm.

- Biết rất ít thông tin về sản phẩm tại cửa hàng.

- Thiếu thông tin nhận xét từ những người đã sử dụng sản phẩm.

**Đối với cửa hàng:**

- Khó khăn trong việc giới thiệu sản phẩm đến với khách hàng ở xa.

- Khó khăn trong việc tìm kiếm và quản lý đơn hàng.

- Việc thống kê báo cáo gặp nhiều khó khăn.

- Quản lý sản phẩm trong kho và số lượng bán rất bất tiện.

🡺 Vì những lý do trên, việc xây dựng một website bán quần áo trực tuyến là rất cần thiết cho cửa hàng trong việc giới thiệu sản phẩm tại cửa hàng cũng như giúp khách hàng gần xa có thể có đầy đủ thông tin và giúp khách hàng có thể lựa chọn sản phẩm hợp lý mà không nhất thiết phải tự mình đến cửa hàng. Website còn cung cấp hệ thống đánh giá và bình chọn nhằm giúp khách hàng có thể biết được những đánh giá và nhận xét của khách hàng mua trước đó. Ngoài ra đối với cửa hàng website còn cung cấp các chức năng quan trọng cho việc quản lý, buôn bán sản phẩm và thống kê báo cáo.

## 1.3. PHẠM VI ĐỀ TÀI

Đề tài “Xây dựng website quần áo trực tuyến” Phải đảm bảo các chức năng sau:

* **Đối với người quản lý cửa hàng:**
* Tạo mới thêm sửa xóa các danh mục như: sản phẩm, thể loại, nhóm sản phẩm, Quản lý thông tin nhập hàng, tiếp nhận và xử lý đơn hàng.
* Quản lý thêm xóa sửa nhân viên nhập và duyệt đơn hàng.
* Thống kê sản phẩm đã bán, sản phẩm tồn kho, sản phẩm bán chạy.
* Quản lý thêm xóa sửa tin tức cho cửa hàng.
* Quản lý sliders quảng cáo sản phẩm cho cửa hàng.
* **Đối với khách hàng:**
* Mua sản phẩm trực tuyến trên website.
* Đăng nhập xem sửa thông tin cá nhân và đơn hàng đã đặt.
* Góp ý, nhận xét, chia sẻ sản phẩm.
* Tìm kiếm sản phẩm theo thể loại, nhóm sản phẩm hoặc nhà sản xuất.
* Có chức năng lọc và sắp xếp sản phẩm theo giá sản phẩm.
* Tạo tài khoản trên website – Liên kết facebook.
* Chức thêm sản phẩm vào ưa thích.

## 1.4. KHẢO SÁT HIỆN TRẠNG

Trên mạng internet hiện nay của thế giới nói chung và của nước ta nói riêng việc buôn bán sản phẩm qua website khá nhiều và tương đối mạnh mẽ. Một số website buôn bán sản phẩm online phát triển mạnh ở nước ta điển hình như: **lazada.vn, tiki.vn, Canifa** sau đây là một vài ưu điểm nhược của những website trên so với website bán quần áo trực tuyến Atshop.

|  |  |  |
| --- | --- | --- |
| Tên website | Ưu điểm | Nhược điểm |
| Lazada, Tiki, Canifa | * Hệ thống sản phẩm đa dạng * Giao diện đẹp thu hút người dùng * Các chức năng của web đáp ứng nhu cầu thực tế của người dùng | * Chi phí triển khai và bảo trì lớn khiến các công ty hay cửa hàng mới hoạt đông gặp khó khăn trong vấn đề triển khai, xây dựng * Các cửa hàng muốn kinh doanh trên những website trên cần trả chi phí hoa hồng |
| Website ATshop | * Đáp ứng được yêu cầu của người dùng * Tập trung vào một mặt hàng đó là quần áo nên có thể dễ quản lý mà cơ sở dữ liệu không cần quá phức tạp * Phù hợp để triển khai cho những cửa hàng thời trang mới   thành lập   * Chi phí triển khai bảo trì và nâng cấp rẻ * Việc xây dựng những website có quy mô vừa và nhỏ tập trung vào mặt hàng đang kinh doanh sẽ sẽ giúp các cừa hàng tiết kiệt chi phí mà vẫn có thể bán và quản lý sản phẩm như những website lớn mà không phải thông qua những website đó. Chính vì sự tiện lợi đó mà ngày càng nhiều cửa hàng muốn có một website buôn bán sản phẩm cho mình. | * Hệ thống quản lý sản phẩm chưa được tối ưu |

Bảng 1.1: So sánh một số website bán hàng online

## 1.6. PHƯƠNG PHÁP THỰC HIỆN

1.6.1. Về lý thuyết:

Để phục vụ cho quá trình phát triển và hoàn chỉnh website ta cần phải áp dụng các mảng kiến thức sau vào đề tài:

* Nắm vững kỹ thuật phân tích thiết kế hệ thống thông tin.
* Tìm hiểu tài liệu về hệ quản trị cơ sở dữ liệu MYSQL.
* Có kiến thức cơ bản về ngôn ngữ lập trình web như: PHP, CSS, Javascript, jQuery, Ajax...
* Ứng dụng công nghệ Laravel framework trong ngôn ngữ lập trình PHP để áp dụng vào việc xậy dựng đề tài.
* Nắm vững mô hình MVC trong lập trình website.
* Tham khảo các đề tài website bán hàng liên quan.

1.6.2. Về kỹ thuật:

Áp dụng những kiến thức đã học ở trường, kết hợp việc nghiên cứu công nghệ Laravel framework tích hợp vào website phải đạt được ứng yêu cầu kỹ thuật sau:

* Giao diện phải thân thiện với người dùng – dễ dàng thao tác.
* Bảo đảm độ tin cậy và nhất quán về dữ liệu.
* Chạy tương thích trên hầu hết các trình duyệt web từ máy tính đến điện thoại.
* Khả năng vận hành và nâng cấp dễ dàng.

1.6.3. Các công cụ thực hiện:

Bảng 1.2: Bảng các công cụ thực hiện

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công cụ** | **Chức năng** | **Giai đoạn** |
| **1** | PhpStorm | Viết mã PHP | Lập trình |
| **2** | Power Designer | Thiết kế các mô hình | Phân tích hệ thống |
| **3** | Microsoft Word | Soạn thảo văn bản | Viết báo cáo |
| **4** | Microsoft Power Point | Soạn thảo trình chiếu | Báo cáo |
| **5** | Xampp | Tạo server chạy ứng dụng | Chạy ứng dụng |
| **6** | PhpMyAdmin | Quản lý database | Tất cả |

1.6.4. Cài đặt và kiểm tra lỗi:

* Chạy thử nghiệm trên localhost và hosting.
* Kiểm tra lỗi trong quá trình vận hành

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. HTML

HTML (viết tắt cho Hypertext Markup Language) là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho Website.

## 2.2. CSS

CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Ta có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng…thì CSS sẽ giúp chúng ta có thể thêm một chút “phong cách” vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc…

## 2.3. AJAX & JQUERY

2.3.1. Ajax

AJAX (viết tắt: "Asynchronous JavaScript and XML" - nghĩa là "JavaScript và XML không đồng bộ") là một nhóm các công nghệ phát triển web được sử dụng để tạo ra các ứng dụng web động hay các ứng dụng giàu tính Internet (rich Internet application). Từ ngữ Ajax được ông Jesse James Garrett đưa ra và dùng lần đầu tiên vào tháng 2 năm 2005 để định nghĩa cho kỹ thuật này, mặc dù các hỗ trợ cho Ajax đã có mặt trên các chương trình duyệt từ 10 năm trước.

Giống như DHTML, LAMP hay SPA, Ajax tự nó không phải là một công nghệ mà là một thuật ngữ mô tả việc sử dụng kết hợp một nhóm nhiều công nghệ với nhau. Trong đó, HTML và CSS được kết hợp với nhau để đánh dấu và định kiểu thông tin. DOM và JavaScript kết hợp lại để hiển thị thông tin động và cho phép người dùng tương tác với các thông tin này. JavaScript cùng với đối tượng XMLHttpRequest hỗ trợ việc trao đổi dữ liệu bất đồng bộ giữa trình duyệt và máy chủ nhằm hạn chế việc tải lại nguyên trang website.

**Ưu điểm của Ajax:**

* Dễ học, dễ sử dụng.
* Dễ tiếp cận.
* Giảm băng thông.
* Thời gian hiển thị trang web nhanh hơn.
* Tăng tính tương tác giữa trang web với người dùng.

2.3.2. jQuery

jQuery [1]chính là một thư viện kiểu mới của JavaScript giúp đơn giản hóa cách viết JavaScript và tăng tốc độ xử lý các sự kiện trên trang web. jQuery thêm tính tương tác Ajax vào trong trang web. jQuery được thiết kế để thay đổi cách viết mới cho JavaScript.

## 2.4. HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

MySQL [2]là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập cơ sở dự liệu trên internet. MySQL miễn phí hoàn toàn cho nên ta có thể tải về MySQL từ trang chủ (http://mysql.com). Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix …

**Một số đặc điểm của MYSQL:**

MySQL là một phần mềm quản trị cơ sở dữ liệu dạng server-based (gần tương đương với SQL Server của Microsoft).

MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu, mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu.

MySQL có cơ chế phân quyền người sử dụng riêng, mỗi người dùng có thể được quản lý một hoặc nhiều cơ sở dữ liệu khác nhau, mỗi người dùng có một tên truy cập (username) và mật khẩu (password) tương ứng để truy xuất đến cơ sở dữ liệu.

## 2.5. PHP VÀ PHP FRAMEWORK

PHP[3] (PHP: Hypertext Preprocessor) là một ngôn ngữ lập trình kịch bản[2] 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[4] đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

Ngôn ngữ PHP là một ngôn ngữ lập trình web rất phổ biến vì tính linh hoạt dễ sử dụng, dễ học. Bên cạnh đó PHP còn hỗ trợ lập trình hướng đối tượng (OPP)[5] thì việc dùng nó để phát triển các ứng dụng là rất dễ dàng. Tuy nhiên, với tính mềm dẻo, linh hoạt của nó đã làm cho người lập trình phát triển ứng dụng với nó không có một quy định nào về cấu trúc của tập tin, việc quản lý các mã lệnh trong khi phát triển ứng dụng cũng gặp không ít khó khắn. Từ đó, đã tạo nên một vấn đề rất khó giải quyết là làm sao nâng cấp và sữa chữa các ứng dụng khi nó xảy ra sự cố hay phát triển ứng dụng cho phù hợp hơn với nhu cầu của người sử dụng, trong các thời điểm khác nhau.

PHP FRAMEWORK là cấu trúc mới của ngôn ngữ PHP giúp phát triển các phần mềm ứng dụng. Nói cách khác PHP FRAMEWORK là một thư viện được lập trình sẵn dựa trên nền của ngôn ngữ web mới để hỗ trợ các lớp, các chức năng cho người sử dụng.

PHP FRAMEWORK làm cho sự phát triển của những ứng dụng web viết bằng ngôn ngữ PHP trở nên trôi chảy hơn, bằng cách cung cấp 1 cấu trúc cơ bản để xây dựng những ứng dụng đó. Hay nói cách khác, PHP Framework giúp đỡ chúng ta thúc đẩy nhanh chóng quá trình phát triển ứng dụng, giúp ta tiết kiệm được thời gian, tăng sự ổn định cho ứng dụng, và giảm thiểu số lần phải viết lại mã cho lập trình viên. Ngoài ra PHP Framework còn giúp những người mới bắt đầu có thể xây dựng các ứng dụng ổn định hơn nhờ việc tương tác chính xác giữa các Database, mã (PHP) và giao diện (HTML) một cách riêng biệt. Điều này cho phép ta dành nhiều thời gian để tạo ra các ứng dụng web, hơn là dành nhiều thời gian để viết các đoạn mã lặp lại trong một dự án.

## 2.6. MÔ HÌNH MVC

2.6.1. Khái niệm mô hình MVC

MVC là chữ viết tắt của Model - View – Controller, là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Mô hình này giúp các nhà phát triển tách ứng dụng ra làm ba thành phần khác nhau Model, View, Controller. Mỗi thành phần có nhiệm vụ riêng và độc lập với các thành phần khác. Mô hình này được sử dụng khá rộng rãi. đặc biệt là trong lập trình website.



Hình 2.1: Mô hình MVC

(Nguồn: Teachlead.vn)

2.6.2. Các thành phần trong mô hình MVC

**Model:** Đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class hay hàm xử lý database.

**View:** Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng giao diện như textbox, images, input, Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML.

**Controller:** Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng. Đảm nhận trách nhiệm xử lý các thao tác từ người dùng sau đó thực hiện thao tác với Model tương ứng nếu cần và gửi dữ liệu sang View thích hợp rồi trả kết quả về người dùng.

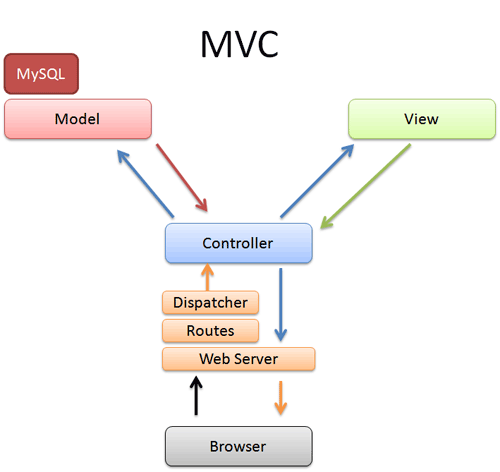
2.6.3. Quy trình hoạt động của mô hình MVC

Người dùng sử dụng một trình duyệt web bất kỳ để có thể gửi những yêu cầu (HTTP Request) có thể kèm theo những dữ liệu nhập tới những Controller xử lý tương ứng. Việc xác định Controller xử lý sẽ dựa vào một bộ Routing điều hướng.

Khi Controller nhận được yêu cầu gửi tới, nó sẽ chịu trách nhiệm kiểm tra yêu cầu đó có cần dữ liệu từ Model hay không? Nếu có, nó sẽ sử dụng các class/function cần thiết trong Model và nó sẽ trả ra kết quả (Resulting Arrays), khi đó Controller sẽ xử lý giá trị đó và trả ra View để hiển thị. Controller sẽ xác định các View tương ứng để hiển thị đúng với yêu cầu.

Khi nhận được dữ liệu từ Controller, view sẽ chịu trách nhiệm xây dựng các thành phẩn hiển thị như hình ảnh, thông tin dữ liệu… và trả về GUI Content để Controller đưa ra kết quả lên màn hình người dùng.

Trình duyệt sẽ nhận giá trị trả về (HTTP Response) và sẽ hiển thị với người dùng. Và sẽ kết thúc một quy trình hoạt động.



Hình 2.2: Quy trình hoạt động của mô hình MVC

2.6.4. Ưu và nhược điểm của mô hình MVC

* **Ưu điểm**
* Các dự án có thể áp dụng ngay mô hình MVC mà không phụ thuộc vào môi trường, nền tảng xây dựng hay ngôn ngữ lập trình phát triển.
* Quy hoạch các class/ function vào các thành phần riêng biệt Controller – Model – View, khi đó sẽ dễ dàng xây dựng – phát triển – quản lý – vận hành và bảo trì một dự án, tạo sự rõ ràng, trong sáng trong quá trình phát triển dự án, kiểm soát được các luồng xử lý và tạo ra các thành phần xử lý nghiệp vụ chuyên biệt hóa.
* Tạo thành mô hình chuẩn cho nhiều dự án, các chuyên gia sẽ tiếp cận – tìm hiểu những dự án đó một cách nhanh chóng và hiệu quả. Nếu ta nắm rõ mô hình MVC của một dự án nào đó, thì khi tiếp cận với một dự án khác mà ta chưa từng biết hoặc tiếp xúc, nhưng nó lại được xây dựng với mô hình MVC thì sẽ không khó khăn gì mà cực kỳ dễ dàng. Học một nhưng có thể hiểu và sử dụng được mười.
* Giúp các chuyên gia lập trình, nhà quản lý, nhà đầu tư, PM… có thể hiểu được dự án hoạt động ra sao hoặc giúp các lập trình viên dễ dàng quản lý – phát triển dự án. Nó không phải ngôn ngữ, nhưng khi họ cùng nhìn vào nó thì sẽ tự hiểu nó là gì, khi đó họ có thể trao đổi các yêu cầu và bàn bạc công việc.
* Đây là một mô hình chuẩn, nó tối ưu nhất hiện nay so với nhiều mô hình khác và được sử dụng trong nhiều dự án và nhiều lĩnh vực, đặc biệt trong công nghệ sản xuất ứng dụng – phần mềm. Các lập trình viên sử dụng mô hình chuẩn MVC để có thể dễ dàng phân phối và chuyển giao công nghệ.
* Đây là mô hình đơn giản, xử lý những nghiệp vụ đơn giản, và dễ dàng triển khai với các dự án nhỏ.
* **Nhược điểm**

Yêu cầu về chuyên môn khá cao, có kiến thức vững về các mô hình chuẩn

Khó triển khai với những dự án yêu cầu phức tạp. Hiện nay đang có một khái niệm mô hình mới đó là HMVC đang dần thay thế cho MVC

## 2.7. LARAVEL FRAMEWORK

2.7.1. Giới thiệu về Laravel

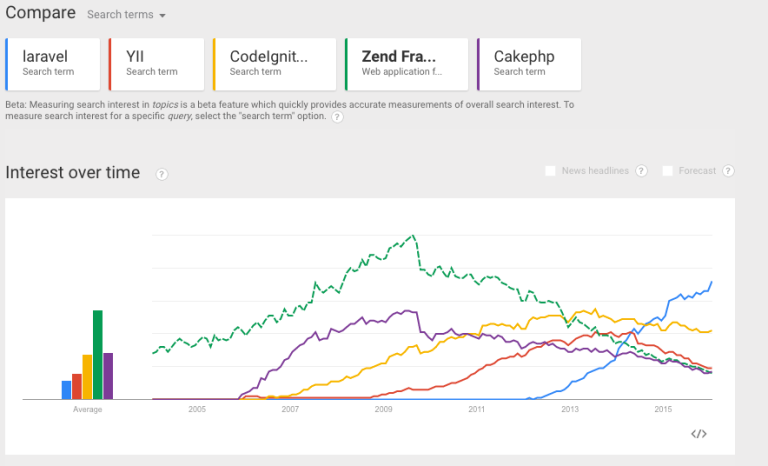
Laravel là một PHP framework [6]mã nguồn mở và miễn phí, được phát triển bởi Taylor Otwell và nhắm vào mục tiêu hỗ trợ phát triển các ứng dụng web theo kiến trúc MVC. Những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu – rõ ràng, một hệ thống đóng gói modular và quản lý gói phụ thuộc, nhiều cách khác nhau để truy cập vào các cơ sở dữ liệu quan hệ, nhiều tiện ích khác nhau hỗ trợ việc triển khai vào bảo trì ứng dụng. Laravel là một framework khá mới mẻ cho lập trình viên, phiên bản đầu tiên được phát hành vào tháng 6/2011.

Ngay từ khi ra mắt, Laravel đã được chú ý bởi nhiều đặc điểm hay như Eloquent ORM, model and relationships, routing, caching, session, blade template, …

2.7.2. Các tính năng của Laravel

* Route trong Laravel thật sự khác biệt, mới và đầy mạnh mẽ. Mọi url điều hướng đều có thể quản lý trong route của laravel.
* Master layout được tích hợp sẵn cùng Blade template giúp code của chúng ta trở nên gọn gàng và tiện dụng. Các file layout có thể dễ dàng extends của nhau giúp code ngắn gọn và dễ dàng quản lý.
* Migration quản lý database thật dễ dàng khi làm việc với đội nhóm.
* Eloquent class đầy mạnh mẽ nổi bật khi xử lý cơ sở dữ liệu quan hệ 1 – N và N – N, 1-1 tối ưu tất cả các câu truy vấn cơ sở dữ liệu.
* Composer quản lý và tích hợp các thư viện khác thật nhanh và không lo lắng khi thư viện đó bị thay đổi, laravel có đầy đủ các thư viện cơ bản đủ để thực hiện mọi yêu cầu của chúng ta.
* Document dễ đọc, dễ hiểu và có đầy đủ các ví dụ. Tuy ra đời muộn hơn các framework khác nhưng laravel lại có hướng dẫn chi tiết và đầy đủ ví dụ ngay tại trang chủ, các ví vụ dễ đọc dễ hiểu, cộng đồng phát triển rộng lớn và luôn luôn được update kịp thời khi có lỗi.
* Eloquent ORM: đây là một ORM tuyệt vời với khả năng migration data và làm việc tốt với MySQL, SQL Server, SQLite, Các câu truy vấn database dễ hiểu, lấy dữ liệu nhanh chóng.
* Package-Library phong phú, đa dạng, đáp ứng được hầu hết các nhu cầu cơ bản của chúng ta.
* User authentication được tích hợp sẵn, lập trình viên chỉ cần gọi class là có thể sử dụng theo ý muốn.
* Và một số tính năng khác.

2.7.3. Biểu đồ các php framework sử dụng trong hai năm gần đây



Hình 2.3: Biểu đồ các framework sử dụng qua các năm

(Nguồn: Teachlead.vn)

2.7.4. Tại sao nên chọn Laravel

* Với những ưu điểm vượt trội và thừa hưởng ưu điểm và thế mạnh từ những framework đi trước, có cộng đồng sử dụng rộng lớn, sử dụng ORM để thao tác với database dễ dàng, tích hợp composer làm công cụ quản lý code.
* Có số lượng người sử dụng nhiều tại thời điểm hiện tại.
* Hỗ trợ autoload theo namespace.
* Document được laravel hỗ trợ khá dễ dàng tìm hiểu cho người mới bắt đầu.
* Các lênh tương tác với cơ sỡ dự liệu ngắn gọn và thân thiện.
* Quản lý layout thật sự đơn giản với Blade Template.
* Dễ dàng tích hợp các thư viện khác vào dự án, và được quản lý dễ dàng với Composer.
* Bộ điều khiển định tuyến (routers) khá mạnh mẽ.

2.7.5. Các chặn đường phát triển

Vào tháng 6/2011 Laravel [7]được phát hành phiên bản beta nhằm tạo nên một lựa chọn mới thay thế cho Codeiginter. Phiên bản 1.0 được phát hành vào cùng tháng với phiên bản beta này, Laravel cung cấp hàng tá các tính năng mà Codeiginter đang thiếu như: xác thực người dùng, đa ngôn ngữ, MVC. Tuy nhiên tại phiên bản 1.0 Laravel vẫn chưa đưa vào controllers điều này khiến nó chưa thực sự chuẩn MVC

Laravel 2.0 được phát hành vào tháng 9/2011 với việc chuẩn hóa mô hình MVC, controllers được đưa vào, hỗ trợ thiết kế IOC, và giới thiệu một tính năng mới là template engine Blade. Một thiếu sót và nhược điểm của phiên bản này là loại bỏ các thư viện bên ngoài, điều này khiến Laravel 2.0 chưa thực sự hấp dẫn lắm so với CodeIginer.

Laravel 3.0 được ra mắt vào tháng 2/2012 với một loạt tính năng hấp dẫn như việc thêm vào môi trường code dòng lệnh CLI với tên artisan. Bổ sung tính năng kết nối với các hệ quản trị cơ sở dữ liệu khác. Tạo xóa sửa tổ chức dữ liệu bằng các phiên bản trong hệ thống migrate. Tại phiên bản này tính năng hấp dẫn Events (tạo sự kiện) cũng là một điểm mạnh khiến Laravel phát triển được một cộng đồng người dùng.

Laravel 4.0 với tên mã lluminate được phát hành vào tháng 5/2013 đây là một cải tiến vượt bậc, loại bỏ toàn bộ những khuyết điểm của các phiên bản trước. Được viết lại hoàn toàn theo chuẩn MVC chặt chẻ Laravel 4.0 cung cấp 1 dàn khung cơ bản mạnh mẽ cho các ứng dụng được viết trên nó. Có thể nói đây là phiên bản mới hoàn toàn, tất cả các thành phần tổ chức thư viện đều được đóng gói và hỗ trợ qua composer, chính điều này khiến cho áp dụng các thư viện bên ngoài dễ dàng hơn bao giờ khác, nó tách biệt rõ ràng từng phần thư viện. Hỗ trợ hàng đợi, nâng cao tính năng Migrate, hỗ trợ nhiều phương thức gửi mail…

Laravel 5.0 phát hành tháng 2/2015 đã khiến nó vươn lên vượt bậc, và lọt vào top các framework được lựa chon ưu tiên khi phát triển các ứng dụng doanh nghiệp bằng PHP. Các tính năng mới trong bản phát hành Laravel 5 bao gồm hỗ trợ lập kế hoạch các tác vụ được thực hiện định kỳ thông qua gói có tên Scheduler, một lớp trừu tượng có tên Flysystem cho phép lưu trữ từ xa theo cách tương tự như các hệ thống tệp cục bộ, cải thiện việc xử lý assets thông qua Elixir. xác thực được xử lý bên ngoài đơn giản thông qua gói Socialite tùy chọn. Laravel 5 cũng giới thiệu cấu trúc cây thư mục nội bộ mới cho các ứng dụng đã phát triển. Vào tháng 3 năm 2015, một khảo sát SitePoint đã liệt kê Laravel là khung công tác PHP phổ biến nhất.

Laravel 5.3, được phát hành vào ngày 23 tháng 8 năm 2016. Các tính năng mới trong 5.3 tập trung vào việc cải thiện tốc độ phát triển bằng cách bổ sung thêm các cải tiến cho các tác vụ phổ biến. Phiên bản này có nhiều tính năng mới, như Laravel Dusk, Laravel Mix, Blade Components và Slots, Markdown Emails, Automatic Facades, Route Improvements, Higher Order Messaging cho Collections, và nhiều thứ khác nữa.

Laravel 5.5, phát hành vào ngày 30 tháng 8 năm 2017.

Laravel 5.6, phát hành vào ngày 7 tháng 2 năm 2018…

2.7.6. Yêu cầu máy chủ để chạy Laravel

* PHP >= 5.6.4.
* Hỗ trợ OpenSSL PHP Extension.
* Hỗ trợ PDO PHP Extension.
* Hỗ trợ Mbstring PHP Extension.
* Hỗ trợ Tokenizer PHP Extension.
* Hỗ trợ XML PHP Extension

2.7.7. Ưu và nhược điểm của Laravel

* **Ưu điểm:**
* Autoload theo namespace.
* Hỗ trợ Template Engine nên việc thiết kế giao diện rất đơn giản.
* Hỗ trợ Eloquent, Fluent Query Builder trong việc truy xuất dữ liệu từ Database một cách dễ dàng và nhanh chóng.
* Hỗ trợ Route mạnh mẽ.
* Tài liệu từ trang chủ của Laravel dễ đọc và dễ hiểu.
* Migration quản lý Database dễ dạng khi làm việc với nhóm.
* Được đông đảo cộng đồng sử dụng và được hỗ trợ lâu dài.
* **Nhược điểm:**
* Tài liệu tiếng việt chưa được nhiều để tìm hiểu.
* Khó khăn cho người mới bắt đầu tìm hiểu nếu như chưa nắm bắt được mô hình MVC và hướng đối tượng.

2.7.8. Hướng dẫn cài đặt Laravel

Cách 1: Thông qua Laravel Installer

Đây là một thư viện của Laravel viết ra giúp bạn cài đặt Laravel bởi 1 câu lệnh duy nhất, các bạn cài đặt thông qua Composer nhé. Bạn mở Terminal (CMD hoặc Git Bash) ở bất kỳ vị trí nào và gõ dòng lệnh sau:

-composer global require "laravel/installer"

Khi cài đặt xong bạn cần chắc chắn rằng đường dẫn tập tin thực thi của Composer đã được thêm vào Windows Enviroment Variables Path (nếu bạn không biết thêm như thế nào có thể xem video này).

Đối với Windows, đường dẫn đó là "%appdata%\Composer\vendor\bin" và đối với macOS và Linux thì nó ở "~/.composer/vendor/bin".

Sau khi cài đặt xong, chúng ta sẽ di chuyển vào thư mục htdocs của XAMPP[8], tại đây các bạn mở cửa sổ lệnh (đối với windows thì nhấp Shift + chuột phải và chọn Commad Window Here hoặc Git Bash Here) và gõ dòng lệnh sau:

- **laravel tên\_project**

Trong đó tên\_project chính là tên thư mục laravel project của bạn.

Cách 2: Thông qua Composer[9]

Chúng ta sẽ di chuyển thẳng vào thư mục htdocs của XAMPP, tại đây các bạn mở cửa sổ lệnh (như trên) và gõ dòng lệnh sau:

* **composer create-project --prefer-dist laravel/laravel blog**

Sau khi cài đặt hoàn tất, bạn chỉ cần mở WebServer của bạn lên và chạy đến thưc mục public trong thư mục Laravel project của các bạn hoặc thực thi lệnh sau từ thư mục Laravel project của các bạn.

- **php artisan serve**

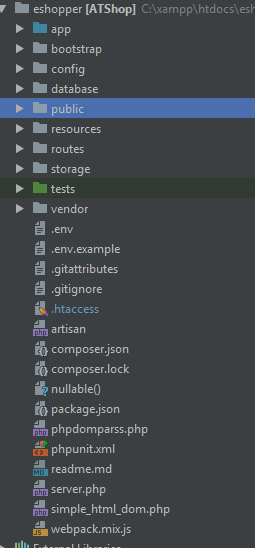
2.7.9. Một số lệnh console phổ biến được cung cấp của Laravel

Laravel có chức năng tự động tạo file và sinh cấu trúc cơ bản cực kỳ hữu ích đối với lập trình viên, giúp chúng ta có thể tạo nhanh chóng cấu trúc của một controller, model, middelware … Dưới đây là các lệnh phổ biến thường được sử dụng. (Chúng ta sử dụng lệnh bằng cách truy cập vào thư mục chứa code của ứng dụng nhấn phím shift trên bàn phím đồng thời click chuột phải ta sẽ thấy mục “Open command window here” và chúng ta chọn nó và thực hiện một số lệnh được laravel cung cấp bên dưới).

Bảng 2.1: Một số lệnh console phổ biển được cung cấp của Laravel

|  |  |  |
| --- | --- | --- |
| **STT** | **Lệnh** | **Mô tả chức năng** |
| 1 | clear-compiled | Hủy bỏ các tập tin lớp đã biên dịch |
| 2 | down | Đặt ứng dụng vào chế độ bảo trì |
| 3 | env | Hiện thị chi tiết file cấu hình |
| 4 | app:name | Đặt namespace cho ứng dụng |
| 5 | auth: clear-resets | Xóa token khôi phục mật khẩu hết hạn |
| 6 | cache: clear | Xóa toàn bộ cache ứng dụng |
| 7 | cache: table | Tạo bảng cache trong cơ sở dữ liệu |
| 8 | Config: cache | Tạo tập tin cấu hình dạng cache |
| 9 | config: clear | Xóa bỏ tập tin cấu hình |
| 10 | key: generate | Tạo khóa ngẫu nhiên cho ứng dụng |
| 11 | make: auth | Tạo cơ chế xác thực người dùng đăng ký, đăng nhập |
| 12 | make: controller | Tạo một lớp controller mới |
| 13 | make: event | Tạo một lớp sự kiện |
| 14 | make: job | Tạo một lớp công việc |
| 15 | make: listener | Tạo một lớp lắng nghe |
| 16 | make: mail | Tạo một lớp quản lý mail |
| 17 | make: middleware | Tạo một lớp bộ lọc xác thực |
| 18 | make: migration | Tạo một trình quản lý phiên bản dữ liệu trong database |
| 19 | make: model | Tạo một lớp model |
| 20 | make: notification | Tạo một lớp thông báo cho ứng dụng |
| 21 | make: policy | Tạo một lớp chính sách cho ứng dụng |
| 22 | make: provider | Tạo một lớp cung cấp dịch vụ |
| 23 | make: request | Tạo một lớp quản lý kiểm tra dữ liệu gửi hợp lệ từ người dùng |
| 24 | migrate: install | Cài đặt một migration |
| 25 | migrate: refresh | Làm mới lại migration |
| 26 | migrate: reset | Khôi phục lại migration |
| 27 | migrate: rollback | Quay lại migration trước đó |
| 28 | migrate: status | Xem trạng thái migration hiện tại |
| 29 | route: cache | Tạo cache đường dẫn liên kết ứng dụng |
| 30 | route: clear | Xóa cache đường dẫn ứng dụng |
| 31 | route: list | Danh sách đường dẫn ứng dụng đã tạo |
| 32 | schedule: run | Chạy một lịch trình đã lên sẵn |

2.7.10. Cấu trúc thư mục của laravel



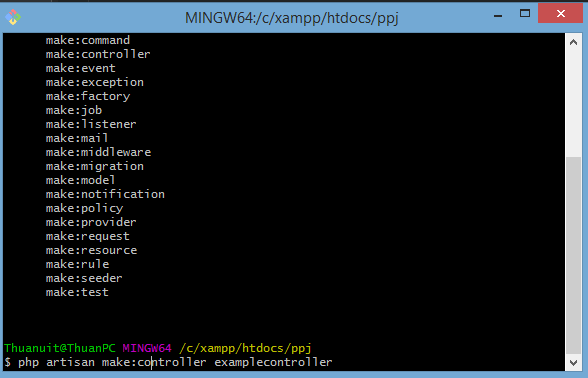
Hình 2.4: Cấu trúc thư mục của Laravel

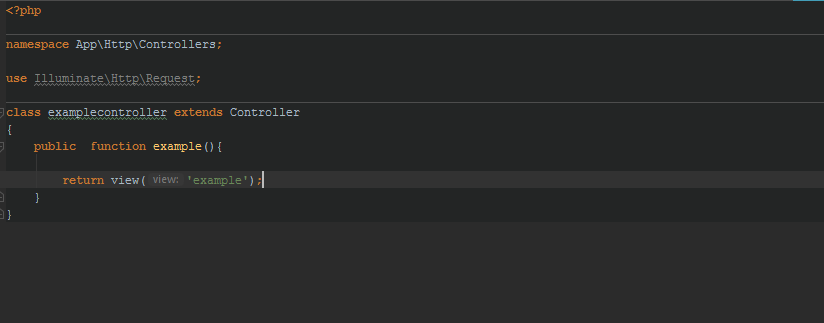
* **app** là thư mục chứa các file cấu hình, lưu trữ, tập lệnh của laravel, trong đó gồm có:
* **Console** là thư mục chưa các lệnh console cần thực thi.
* **Exceptions** là thư mục chứa các ngoại lệ trong ứng dụng**.**
* **Http** là thư mục chứa các điều khiển của ứng dụng bao gồm các phần quan trong như controllers, middleware, request.
* **Providers** là thư mục chứa các dịch vụ cần tạo**.**
* **bootstrap** là thư mục chứa các tập tin khởi động và thục thi ứng dụng.
* **config** là thư mục chứa các file cấu hình ứng dụng: mail, app, auth,…
* **database** Là thư mụcchứa các file xây dựng và khởi tạo dữ liệu.
* **resource** là thư mục chứa các file giao diện, ngôn ngữ cho ứng dụng.
* **routes** là thư mục chứa các file cấu hình đường dẫn định tuyến cho ứng dụng.
* **storare** là thư mục chứa các file logs và cache của ứng dụng.
* **public** là thư mục chứa các file css, js, hình ảnh, file chạy ứng dụng index.php…
* **tests** là thư mục chứa các trình kiểm thử tự động.
* **vendor** là thư mục chứa bộ mã nguồn và thư viện đi kèm của laravel.
* **composer.json** à file cấu hình việc thao tác với composer như install hay update Laravel, thêm và xóa các thư viền hỗ trợ.
* **artisan** là file mà laravel tạo ra để hỗ trợ chạy lệnh: php artisan.
* **server.php** cần có để chạy lệnh: php artisan serve.
* **.env** là file cấu hình ứng dụng**.**
* **Và một số file khác như:** .gitattributes, gitignore, composer.lock, gulpfile.js, phpunit.xml, readme.md.

2.7.11. Hướng dẫn tạo controller – model –view trong Laravel

* Hướng dẫn tạo controller

Để tạo controller trong laravel ta sử dụng lệnh console sau: **php artisan make: controller TenController** sau khi sử dụng lệnh này laravel sẽ tự động sinh ra cấu trúc của một file controller hay ta có thể tạo bằng tay trong thư mục app/Http/Controllers và nhập code của cấu trúc controller như sau:

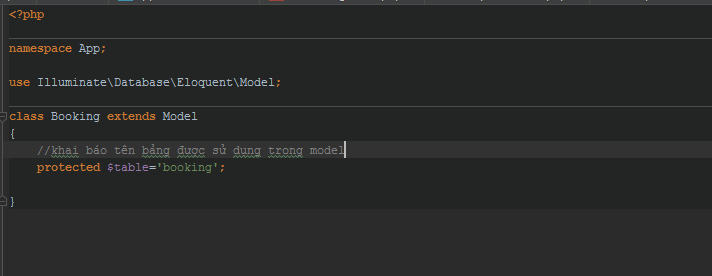




Hình 2.4: Cấu trúc class controller của Laravel

* Hướng dẫn tạo model

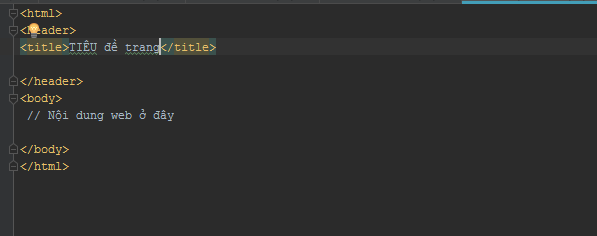
Để tạo model trong laravel ta sử dụng lệnh console sau: **php artisan make: model Tenmodel** sau khi sử dụng lệnh này laravel sẽ tự động sinh ra cấu trúc của một file model hay ta có thể tạo bằng tay trong thư mục app và nhập code của cấu trúc model như sau:



Hình 2.5: Cấu trúc class model của Laravel

* Hướng dẫn tạo view

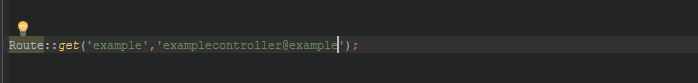
Để tạo view trong laravel ta vào thư mục resources/views của laravel và tạo file view cần sử dụng, vì laravel sử dụng blade template nên để tạo một file view ta cần thêm đuôi.blade trước file .php, ví dụ ta tạo một file **example.balde.php** có cấu trúc như sau:



Hình 2.6: Cấu trúc file view của Laravel

**🡺** để có thể truy cập vào một controller ta cần khai báo trong file cấu hình route của laravel cách khai báo đơn giản như sau:

Trong đó example là đường dẫn route, examplecontroller là controller sẽ được gọi đến và example là hàm sẽ được gọi



Hình 2.7: Cách gọi controller trong route của Laravel

# CHƯƠNG 3: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU

## 3.1. ĐẶT TẢ HỆ THỐNG

3.1.1. Giới thiệu hệ thống

Website bán quần áo online là hệ thống buôn bán sản phẩm trực tuyến thông qua internet nhằm phục vụ khách hàng mua sắm dễ dàng và thuận tiện, khách hàng có thể mua sắm ở bất cứ nơi đâu bắt cứ nơi nào có kết nối internet trên máy tính và điện thoại di động của mình.

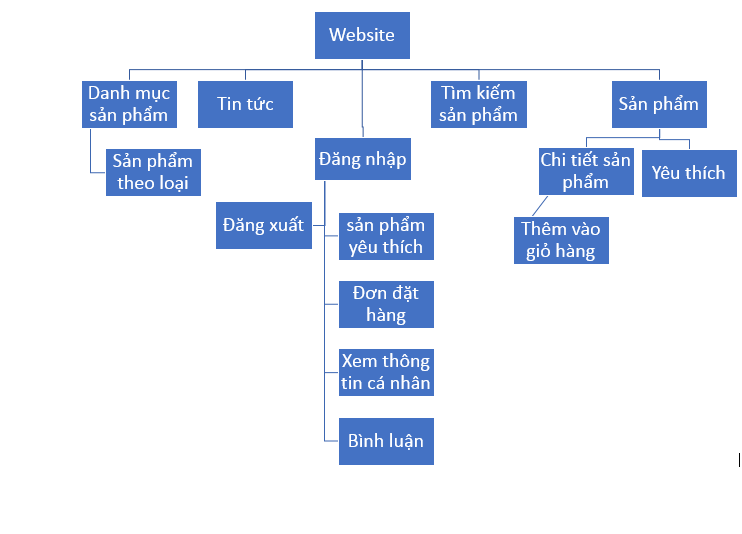
3.1.2. Mô tả hệ thống

Cửa hàng chuyên bán các loại quần áo được cung cấp từ các nhà sản xuất khác nhau, cửa hàng cần một hệ thống quản lý các chức năng như:

* Quản lý sản phẩm:
  + Thông tin sản phẩm bao gồm: mã sản phẩm, tên sản phẩm, giá bán, giá giảm nếu có và một số thông tin cần thiết khác. Một sản phẩm sẽ thuộc về môt thể loại duy nhất. Mỗi thể loại sẽ thuộc về một nhóm sản phẩm để dễ dàng quản lý. Mỗi một sản phẩm cần lưu lại danh sách hình nhằm mục đích giới thiệu đến khách hàng.
  + Thông tin thể loại gồm có: tên thể loại, đường dẫn liên kết. Một thể loại có thể có không hoặc nhiều thể loại con.
  + Thông tin nhóm sản phẩm gồm có: tên nhóm sản phẩm, đường dẫn liên kết. Một nhóm sản phẩm có thể có không hoặc nhiều nhóm con.
* Quản lý phương thức thanh toán:
* Phương thức thanh toán cần biết được hình thức thanh toán mà khách hàng chọn.
* Quản lý khách hàng:
* Khi khách hàng truy cập website sẽ có các chức năng cơ bản như: mua hàng trực tuyến, tìm kiếm sản phẩm, so sánh sản phẩm, đánh giá ngôi sao cho sản phẩm, đăng ký tài khoản trên hệ thống…
* Thông tin khách hàng được lưu trên hệ thống bao gồm: họ và tên, giới tính, địa chỉ, số điện thoại, email…
* Khi khách hàng đăng nhập sẽ có thêm các chức năng như: bình luận, thêm sản phẩm ưa thích, xem đơn hàng đã đặt trước đó, cập nhật thông tin cá nhân.
* Khi khách hàng quên mật khẩu có thể nhập email mà khách hàng đăng ký để thực hiện chức năng khôi phục lại mật khẩu.
* Quản lý đơn hàng:
* Mỗi đơn hàng khi khách hàng đặt trên website sẽ được lưu vào hệ thống và khi nhân viên đã tiếp nhận được đơn hàng thì sẽ liên hệ với khách hàng thực hiện bước xác nhận với khách hàng và sẽ giao hàng đến địa chỉ mà khách hàng đã cung cấp trên hệ thống.
* Khi khách hàng đặt hàng, hệ thống sẽ nhận được yêu cầu đặt hàng và gửi đến email người mua thông tin chi tiết đơn hàng đã đặt. Tuy nhiên, yêu cầu đặt hàng cần thông qua một bước xác nhận đơn hàng, cửa hàng chỉ xác nhận đơn hàng nếu yêu cầu đặt hàng của người mua thỏa mãn các tiêu chí đơn hàng tại cửa hàng.
* Thông tin đơn hàng cần biết những thông tin cơ bản sau: mã số đơn hàng (tự động), thuộc khách hàng nào, ngày đặt, ngày nhận, tình trạng đơn hàng và ghi chú nếu có.
* Quản lý nhân viên:
* Người quản lý tại của hàng có thể thêm sửa xóa nhân viên trên hệ thống
* Thông tin nhân viên gồm có: họ và tên, giới tính, địa chỉ, số điện thoại, email.
* Ngoài ra hệ thống cần biết được ai là người quản lý, người quản lý sẽ có đầy đủ các chức năng trên hệ thống.
* Quản lý các danh mục khác:
* Website còn cung cấp các chức năng về tin tức, sliders, trang viết bài

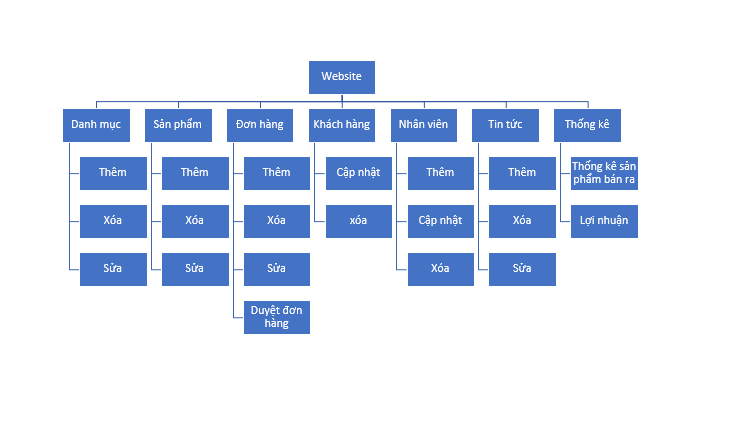
## 3.2. THIẾT KẾ CÁC MÔ HÌNH

3.2.1. Mô hình phân rã chức năng người dùng



Hình 3.1: Mô hình phân rã chức năng người dùng

3.2.2. Mô hình phân rã chức năng người quản lý

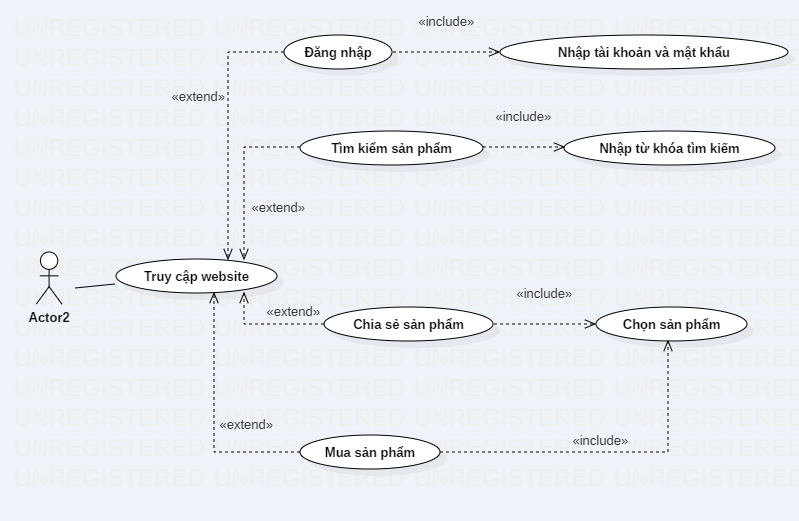


Hình 3.2: Mô hình phân rã chức năng người quản lý

3.2.3. Mô hình Use Case

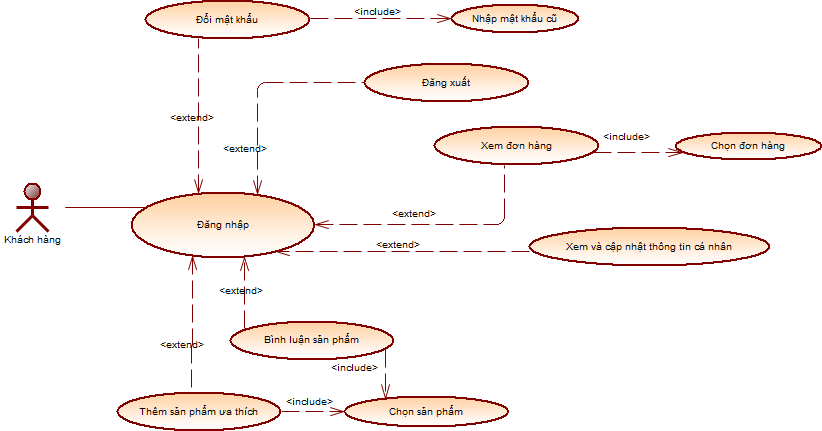
* Mô hình use case chức năng của khách hàng

+/ Đối với khách hàng chưa đăng nhập: Khi khách hàng truy cập website chưa đăng nhập thì được sử dụng các chức năng như: tìm kiếm sản phẩm, chia sẻ sản phẩm, mua sản phẩm



Hình 3.3: Mô hình use case chức năng khách hàng chưa đăng nhập

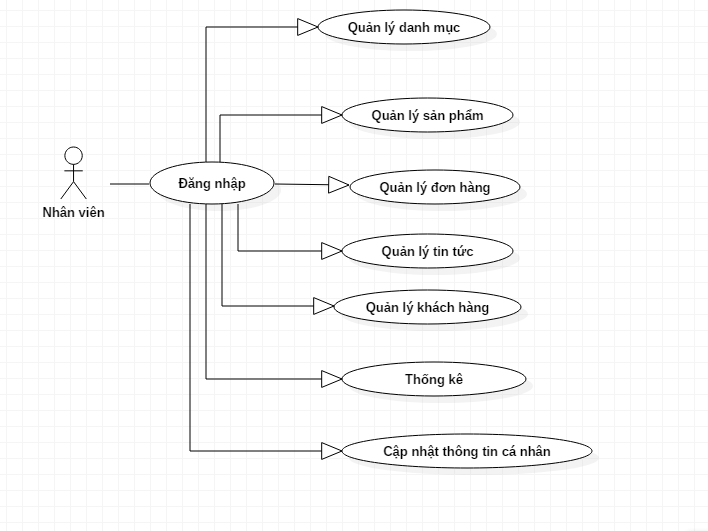
+/Khi khách hàng đã đăng nhập vào website ngoài các chức năng mà khách hàng chưa đăng nhập có thể sử dụng thì khách hàng có thể sử dụng thêm các chức năng như: đổi mật khẩu, xem đơn hàng, xem và cập nhật thộng tin cá nhân, bình luận sản phẩm, thêm sản phẩm ưa thích, đăng xuất.



Hình 3.4: Mô hình use case khi khách hàng đăng nhập

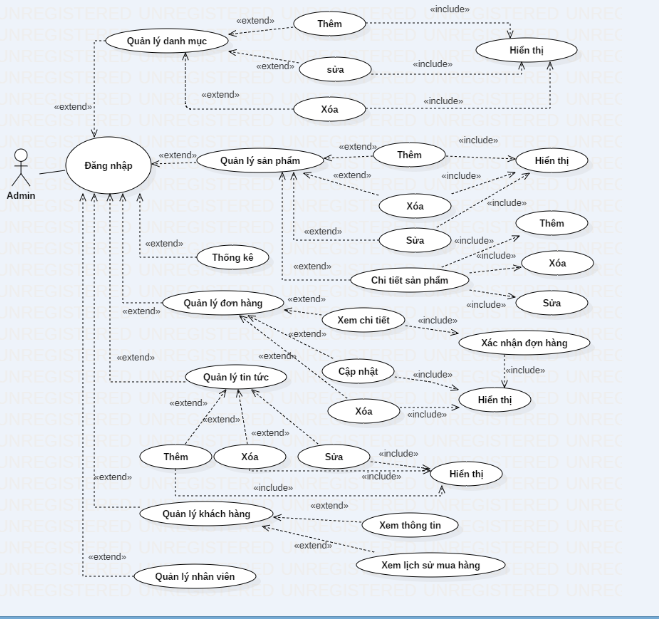
* Mô hình use case chức năng của nhân viên

Khi nhân viên đăng nhập vào hệ thống quản lý sẽ có các chức năng sau: quản lý danh mục, quản lý đơn hàng, sản phẩm, thống kê…



Hình 3.5: Mô hình use case chức năng nhân viên

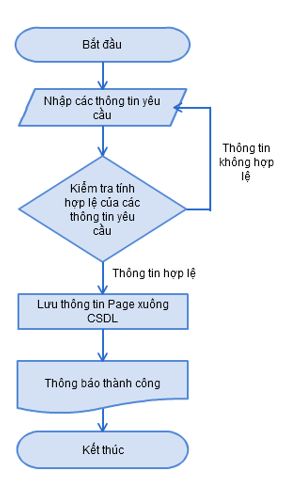
Đối với người quản lý (administrator) sẽ có đầy đủ các chức năng trên hệ thống như: có chức năng của thống kê, quản lý danh mục, quản lý khách hàng, đơn hàng, nhân viên…



Hình 3.6: Mô hình use case chức năng người quản lý

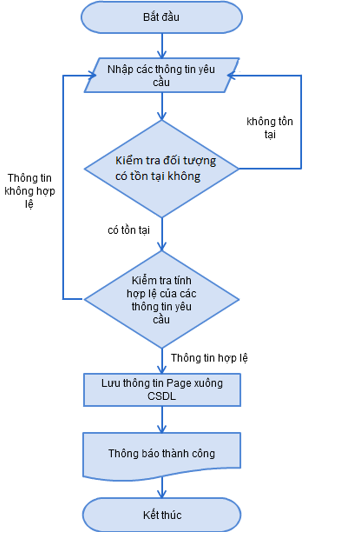
3.2.4. Flowchart

* Flowchart Thêm mới
* Input: Các thông tin thuộc tính của đối tượng
* Output: Kết quả thêm thành công



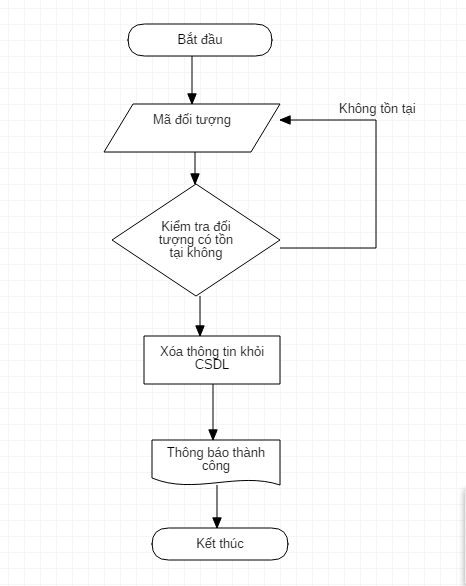
Hình 3.7: Sơ đồ flowchart chức năng thêm mới

* Flowchart Chỉnh sửa
* Input: Các thông tin thuộc tính của đối tượng
* Output: Kết quả chỉnh sửa thành công



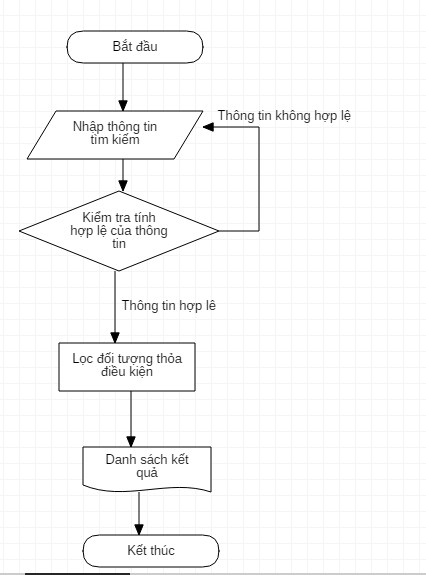
Hình 3.8: Sơ đồ flowchart chức năng chỉnh sửa

* Flowchart Xóa
* Input: Mã đối tượng
* Output: Thông báo xóa thành công



Hình 3.9: Sơ đồ flowchart chức năng xóa

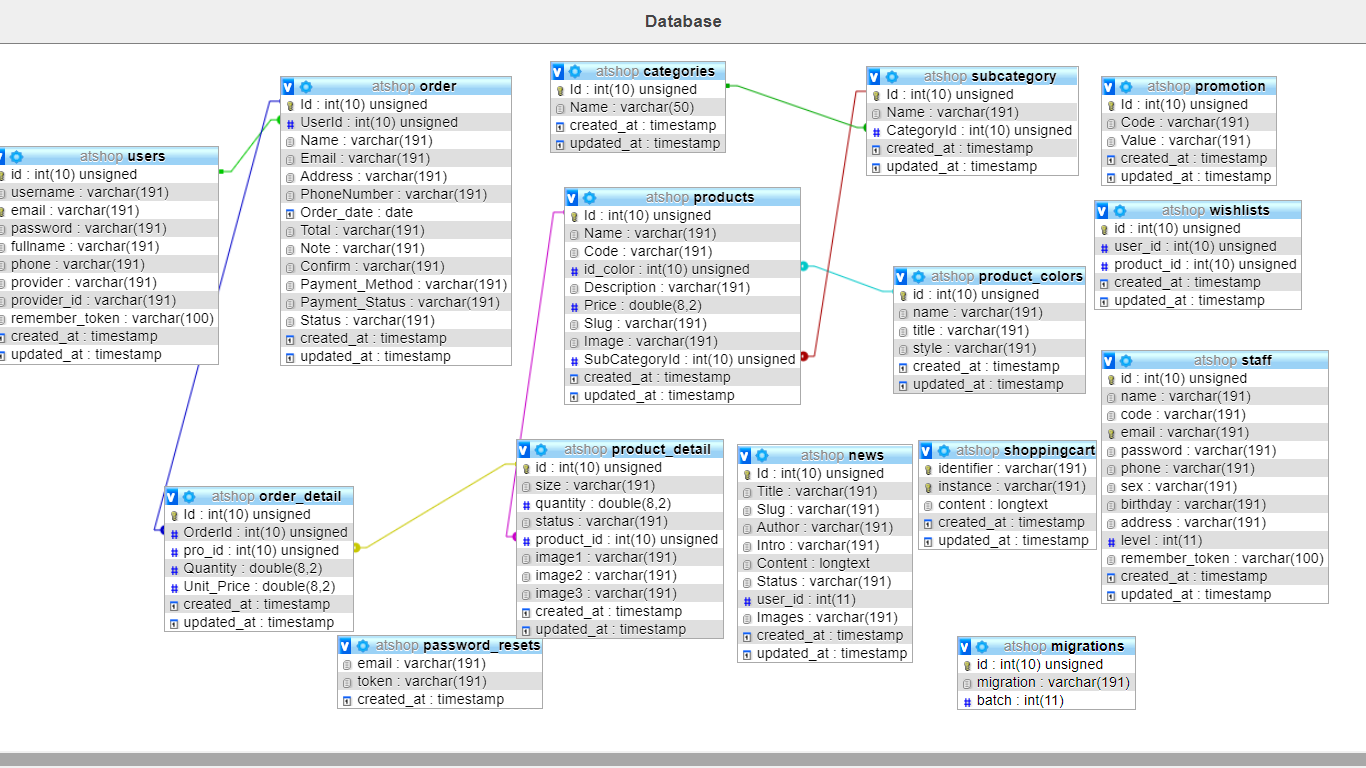
* Flowchart tìm kiếm
* Input: Thông tin tìm kiếm
* Output: Danh sách các đối tượng



Hình 3.10: Sơ đồ flowchart chức năng tìm kiếm

## 3.3. THIẾT KẾ CƠ SỞ DỮ LIỆU

3.3.1. Mô hình cơ sở dữ liệu



Hình 3.11: Mô hình cơ sở dữ liệu

3.3.2. Mô tả dữ liệu

-Bảng Categories: Lưu thông tin các danh mục chính của website

Bảng 3.1: Thực thể categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** |
| 1 | **id** | Interger | Mã danh mục | Khóa chính |
| 2 | **Name** | Varchar (255) | Tên thể loại | Bắt buộc |
| 3 | created\_at | Timestamp | Ngày tạo | Bắt buộc |
| 4 | updated\_at | Timestamp | Ngày sửa | Bắt buộc |

-Bảng SubCategory: Hiển thị loại sản phẩm của các danh mục

Bảng 3.2: Thực thể SubCategory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** |
| 1 | **id** | Interger | Mã loại sản phẩm | Khóa chính |
| 2 | **Name** | Varchar (255) | Tên loại sản phẩm | Bắt buộc |
| 3 | **CategoryId** | Interger | Mã danh mục | Khóa ngoại |
| 4 | created\_at | Timestamp | Ngày tạo | Bắt buộc |
| 5 | updated\_at | Timestamp | Ngày sửa | Bắt buộc |

-Bảng Users: Lưu thông tin của khách hàng như họ tên, địa chỉ, số điện thoại, email...trường ‘provide\_id’ để lưu mã mạng xã hội nếu khách hàng đăng nhập thông qua mạng xã hội

Bảng 3.3: Thực thể users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** |
| 1 | **id** | Interger | Mã khách hàng | Khóa chính |
| 2 | **username** | Varchar (255) | Tên đăng nhập | Bắt buộc |
| 3 | **fullname** | Varchar (255) | Tên người dùng | Bắt buộc |
| 4 | **email** | Varchar (255) | Địa chỉ email | Bắt buộc |
| 5 | **password** | Varchar (255) | Mật khẩu | Bắt buộc |
| 6 | **provider** | Varchar (255) | Tên của mạng xã hội | Tùy chọn |
| 7 | **provider\_id** | Interger | Mã mạng mạng xã hội | Tùy chọn |
| 8 | **remember\_token** | Varchar (255 | Ghi nhớ đăng nhập | Bắt buộc |
| 9 | created\_at | Timestamp | Ngày tạo | Bắt buộc |
| 10 | updated\_at | Timestamp | Ngày sửa | Bắt buộc |

-Bảng Products: Mỗi sản phẩm có một mã sản phẩm và thông tin cần thiết. Sản phẩm cần biết được thuộc thể loại nào. Bảng sản phẩm có trường giảm giá sản phẩm nếu trường giảm giá này lớn hơn 0 thì sản phẩm này được xem là đang khuyến mãi. Thuộc tính id\_color để tham chiếu đến bảng màu sắc. Một sản phẩm có thể có một hoặc nhiều chi tiết sản phẩm

Bảng 3.4: Thực thể products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** |
| 1 | **id** | Interger | Khóa chính | Tự động tăng |
| 2 | Subcategoryid | Interger | Mã loại sản phẩm | Khóa ngoại |
| 4 | slug | Varchar (255) | Tên không dấu | Bắt buộc |
| 5 | code | Varchar (255) | Mã sản phẩm | Bắt buộc |
| 6 | Name | Varchar (255) | Tên sản phẩm | Bắt buộc |
| 7 | Image | Varchar (255) | Hình đại diện | Bắt buộc |
| 8 | Id\_color | Interger | Mã màu sắc | Bắt buộc |
| 9 | Description | LongText | Mô tả sản phẩm | Tùy chọn |
| 10 | Price | BigInt | Giá bán sản phẩm | Bắt buộc |
| 11 | Discount | BigInt | Giá giảm sản phẩm | Tùy chọn |
| 12 | created\_at | Timestamp | Ngày tạo | Bắt buộc |
| 13 | updated\_at | Timestamp | Ngày sửa | Bắt buộc |

-Bảng Product\_detail: Lưu thông tin chi tiết của sản phẩm như kích cỡ, số lượng, trạng thái, hình ảnh chi tiết của sản phẩm

Bảng 3.5: Thực thể product\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** |
| 1 | **id** | Interger | Mã chi tiết sản phẩm | Khóa chính |
| 2 | size | Varchar (255) | Kích cỡ sản phẩm | Bắt buộc |
| 4 | quanity | Varchar (255) | Số lượng sản phẩm | Bắt buộc |
| 5 | Product\_id | Interger | Mã sản phẩm | Khóa ngoại |
| 6 | Name | Varchar (255) | Tên sản phẩm | Bắt buộc |
| 7 | Image1 | Varchar (255) | Hình chi tiết | Tùy chọn |
| 8 | Image2 | Varchar (255) | Hình chi tiết | Tùy chọn |
| 9 | Image3 | Varchar (255) | Hình chi tiết | Tùy chọn |
| 12 | created\_at | Timestamp | Ngày tạo | Tùy chọn |
| 13 | updated\_at | Timestamp | Ngày sửa | Tùy chọn |

-Bảng Product\_colors: Lưu thông tin màu sắc như mã màu sắc, tên màu, mã css của màu.

Bảng 3.6: Thực thể product\_colors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** |
| 1 | **id** | Interger | Mã màu sắc | Khóa chính |
| 2 | **name** | Interger | Tên màu | Bắt buộc |
| 3 | **title** | Interger | Tiêu đề của màu | Tùy chọn |
| 4 | **style** | Varchar (20) | Mã css của màu | Bắt buộc |
| 5 | created\_at | Timestamp | Ngày tạo | Bắt buộc |
| 6 | updated\_at | Timestamp | Ngày sửa | Bắt buộc |

-Bảng Order (Đơn hàng): Mỗi đơn hàng có mã đơn hàng để phân biệt, đơn hàng cần biết được của khách hàng nào có phương thức vận chuyển, thanh toán là gì và tình trạng của đơn hàng cũng như biết được ngày nào khách hàng đó đã đặt hàng.

Bảng 3.7: Thực thể order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | **id** | Mã đơn hàng | Interger | Khóa chính |
| 2 | Name | Tên khách hàng | Varchar (255) | Bắt buộc |
| 3 | Email | Email khách hàng | Varchar (255) | Bắt buộc |
| 4 | Address | Địa chỉ khách hàng | Varchar (255) | Bắt buộc |
| 5 | PhoneNumber | SDT khách hàng | Varchar (255) | Bắt buộc |
| 6 | Order\_Code | Mã đơn hàng | Interger | Bắt buộc |
| 7 | UserId | Mã khách hàng | interger | Bắt buộc |
| 9 | Payment\_Method | Phương thức thanh toán | Varchar (255) | Bắt buộc |
| 10 | Payment\_status | Trạng thái thanh toán | Varchar (255) | Bắt buộc |
| 11 | Order\_date | Ngày đặt | DateTime | Bắt buộc |
| 12 | Note | Ghi chú của khách hàng | Varchar (255) | Tùy chọn |
| 13 | Order\_status | Tình trạng đơn hàng | Interger | Bắt buộc |
| 14 | created\_at | Ngày tạo | TimeStamp | Bắt buộc |
| 15 | updated\_at | Ngày sửa | TimeStamp | Bắt buộc |

-Bảng order\_detail: Mỗi chi tiết đơn hàng sẽ thuộc một đơn hàng duy nhất. Chi tiết đơn hàng cho biết số lượng mua và giá mua sản phẩm tại thời điểm đó và số lượng đã giao cho khách hàng

Bảng 3.8: Thực thể order\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | Id | Mã chi tiết đơn hàng | Interger | Khóa chính |
| 2 | Productid | ID sản phẩm | Interger | Bắt buộc |
| 3 | Total | Số lượng mua | Interger | Bắt buộc |
| 4 | Unit\_price | Đơn giá | BigInt | Bắt buộc |
| 5 | Quantity | Số lượng | Interger | Bắt buộc |
| 6 | Created\_at | Ngày tạo | TimeStamp | Bắt buộc |
| 7 | Update\_at | Ngày sửa | TimeStamp | Bắt buộc |

Bảng **Staff** (Nhân viên): Mỗi nhân viên sẽ có một mã số để phân biệt và có đầy đủ thông tin để liên lạc

Bảng 3.9: Thực thể staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | **id** | Id nhân viên | Interger | Khóa chính |
| 2 | code | Mã số nhân viên | Interger | Bắt buộc |
| 3 | email | Email nhân viên | Varchar (255) | Bắt buộc |
| 4 | password | Mật khẩu nhân viên | Varchar (255) | Bắt buộc |
| 5 | phone | Số điện thoại | Varchar (20) | Bắt buộc |
| 6 | name | Họ và tên | Varchar (100) | Bắt buộc |
| 7 | sex | Giới tính | Char (10) | Bắt buộc |
| 8 | birthday | Ngày sinh | Date | Bắt buộc |
| 9 | address | Địa chỉ | Varchar (255) | Bắt buộc |
| 10 | level | Quyền truy cập | Varchar (255) | Bắt buộc |
| 11 | remember\_token | Ghi nhớ đăng nhập | Varchar (100) | Tùy chọn |
| 13 | created\_at | Ngày tạo | TimeStamp | Tùy chọn |
| 14 | update\_at | Ngày sửa | TimeStamp | Tùy chọn |

-Bảng Wishlist: Lưu các sản phẩm vào danh sách yêu thích

Bảng 3.20: Thực thể wishlist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** |
| 1 | **id** | Interger | Mã danh sách yêu thích | Khóa chính |
| 2 | **UserId** | Interger | Mã khách hàng | Khóa ngoại |
| 3 | **Product\_id** | Interger | Mã sản phẩm | Khóa ngoại |
| 4 | created\_at | Timestamp | Ngày tạo | Bắt buộc |
| 5 | updated\_at | Timestamp | Ngày sửa | Bắt buộc |

-Bảng Promotion: Lưu các mã giảm giá

Bảng 3.31: Thực thể promotion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** |
| 1 | **id** | Interger | Id mã giảm giá | Khóa chính |
| 2 | **code** | Varchar (255) | Mã mã giảm giá | Bắt buộc |
| 3 | **value** | Float | Giá trị mã giảm giá | Bắt buộc |
| 4 | created\_at | Timestamp | Ngày tạo | Bắt buộc |
| 5 | updated\_at | Timestamp | Ngày sửa | Bắt buộc |

-Bảng News: Lưu thông tin tin tức

Bảng 3.42: Thực thể news

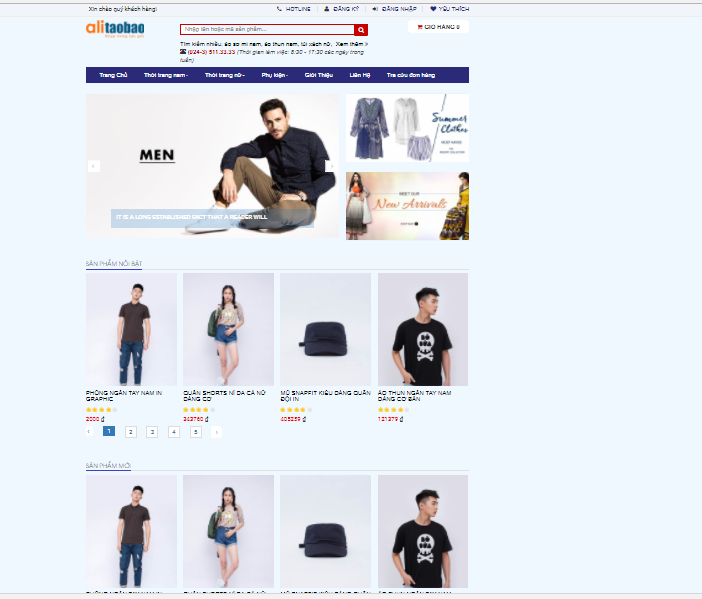
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | **id** | Mã tin tức | Interger | Khóa chính |
| 2 | Title | Tiêu đề tin tức | Interger | Bắt buộc |
| 3 | slug | Tên không dấu | Varchar (255) | Bắt buộc |
| 4 | Author | Tác giả | Varchar (255) | Bắt buộc |
| 5 | intro | Đoạn gới thiệu | Varchar (200) | Bắt buộc |
| 6 | content | Nội dung tin tức | Varchar (100) | Bắt buộc |
| 7 | images | Ảnh đại diện tin tức | Varchar (100) | Bắt buộc |
| 13 | created\_at | Ngày tạo | TimeStamp | Bắt buộc |
| 14 | update\_at | Ngày sửa | TimeStamp | Bắt buộc |

## 3.4. KẾT QUẢ XÂY DỰNG HỆ THỐNG

3.4.1. Một số giao diện chính từ trang người dùng

* Giao diện từ trang chủ website

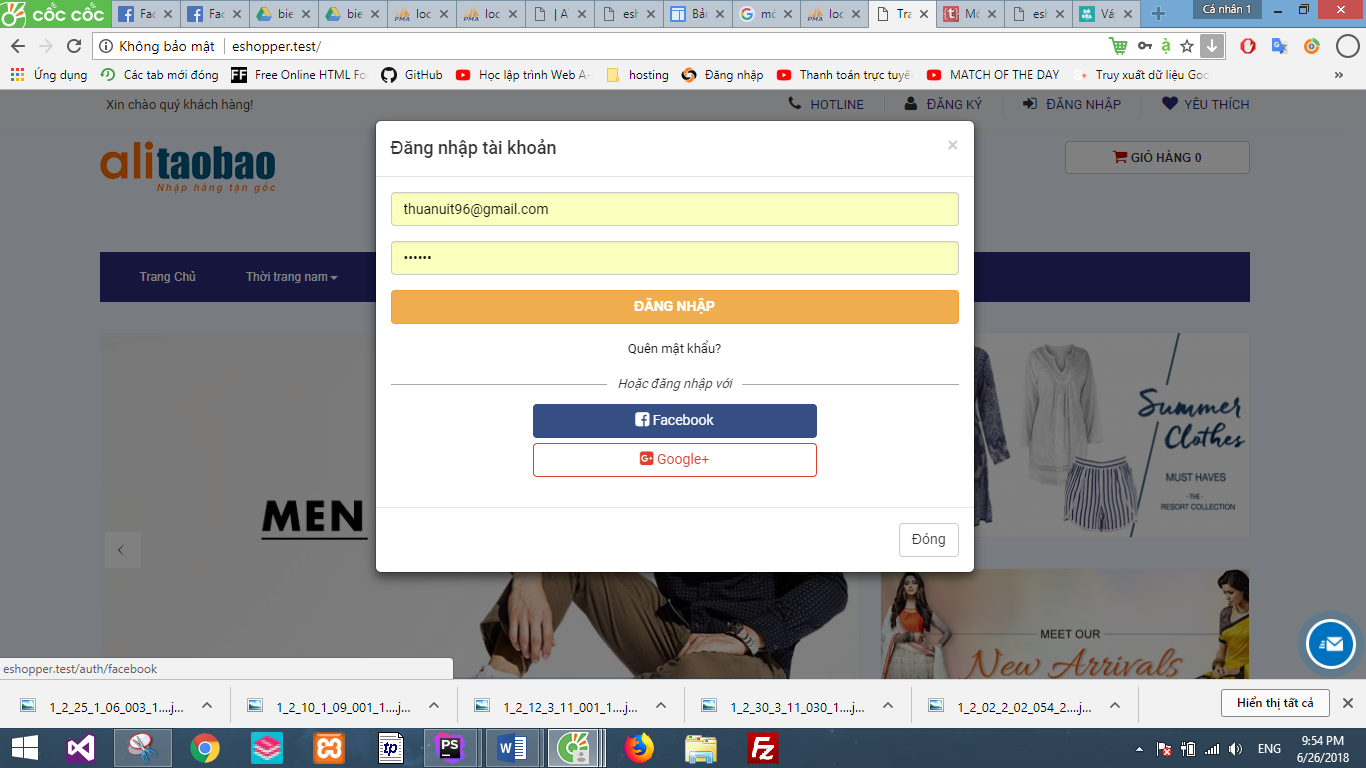
Khi khách hàng truy cập vào trang chủ sẽ thấy những mục như sau: sản phẩm mới, sản phẩm nổi bật, sản phẩm theo danh mục, tin tức mới.



Hình 3.12: Giao diện trang chủ của website

* Giao diện trang đăng nhập

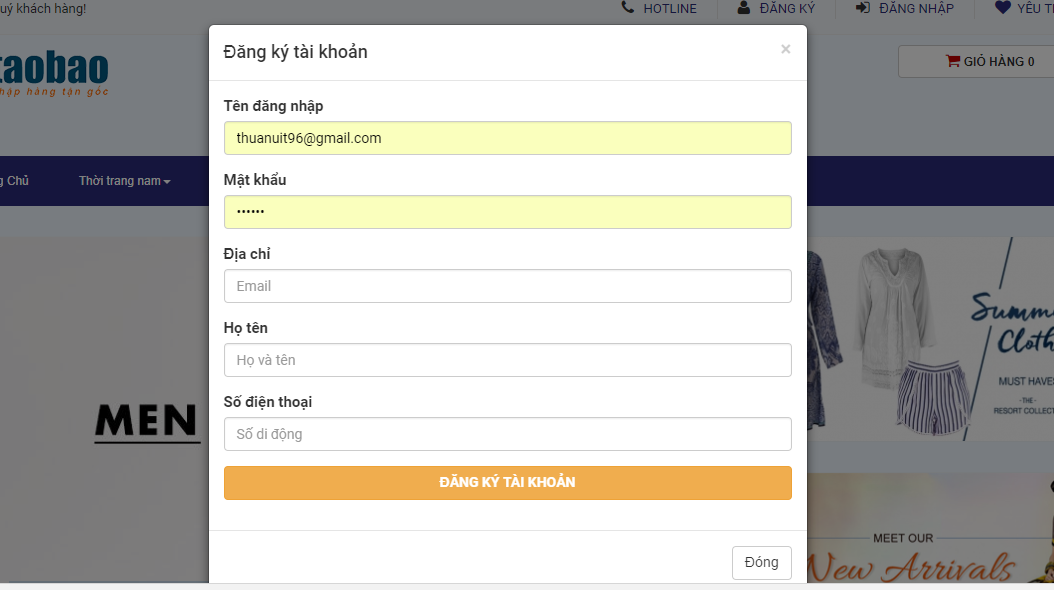
Khi khách hàng nhấn vào liên kết đăng nhập trên thanh menu thì sẽ xuất hiện một cửa sổ đăng nhập như sau:



Hình 3.13: Cửa sổ đăng nhập website

* Giao diện trang đăng ký

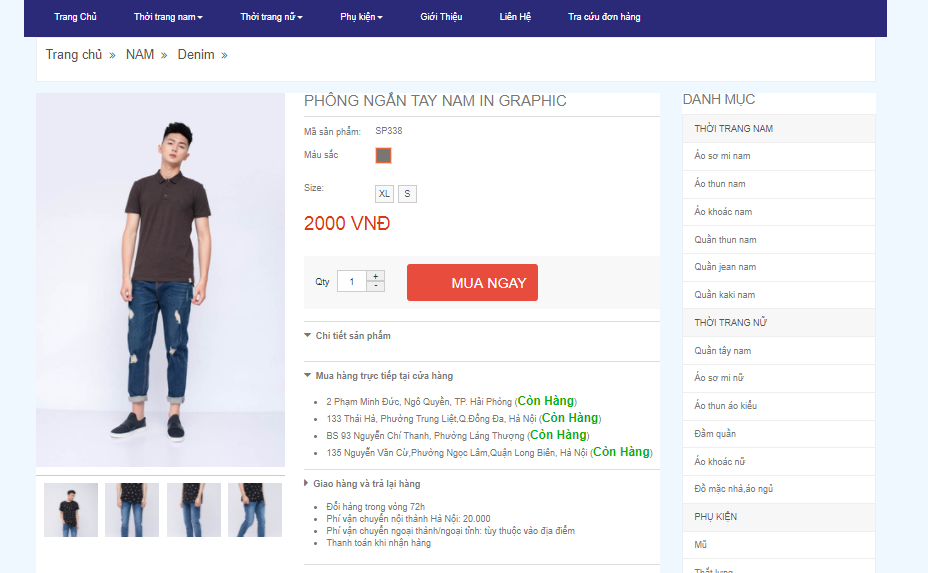
Khi khách hàng muốn đăng ký một tài khoản mới từ website thì sẽ nhấn vào liên kết đăng ký trên thanh menu khách hàng sẽ được dẫn đến trang giao diện đăng ký với giao diện như bên dưới.



Hình 3.14: Giao diện trang đăng ký tài khoản

* Giao diện trang chi tiết sản phẩm

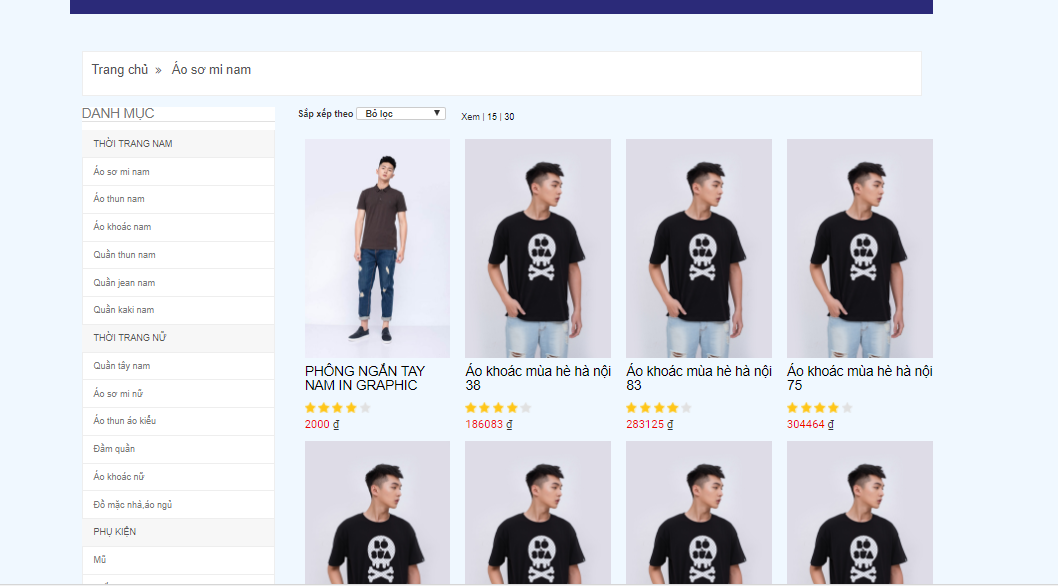
Khi chọn xem chi tiết một sản phẩm nào đó sẽ được dẫn sang trang chi tiết của sản phẩm, nội dung trang chi tiết sản phẩm bao gồm các mục cơ bản như: giới thiệu sản phẩm, thuộc tính sản phẩm, sản phẩm liên quan, danh sách bình luận từ khách hàng.



Hình 3.15: Giao diện trang chi tiết sản phẩm

* Giao diện trang thể loại hay nhóm sản phẩm

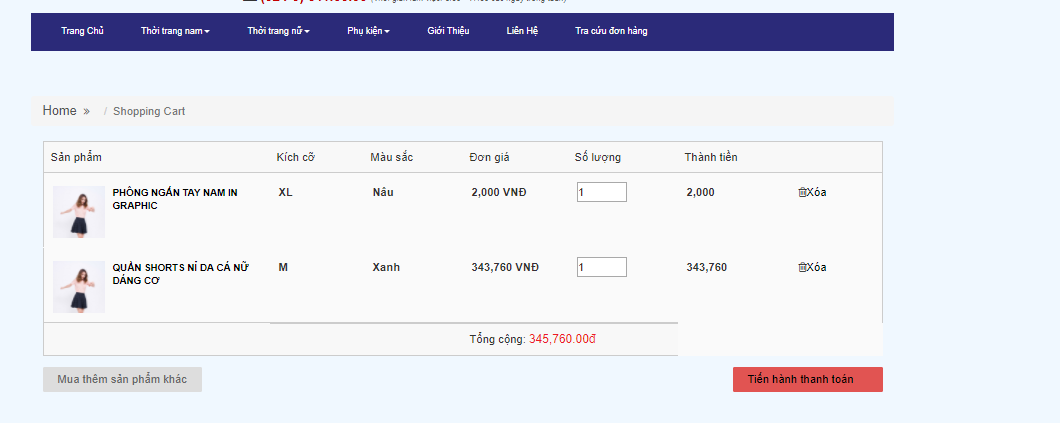
Khi chọn vào một thể loại hay một nhóm sản phẩm nào đó thì sẽ được chuyển sang trang hiện thị danh sách các sản phẩm từ thể loại hay nhóm sản phẩm đó. Giao diện trang thể loại hay nhóm sản phẩm được hiện thị như sau.



Hình 3.16: Giao diện trang thể loại hay nhóm sản phẩm

* Giao diện trang giỏ hàng

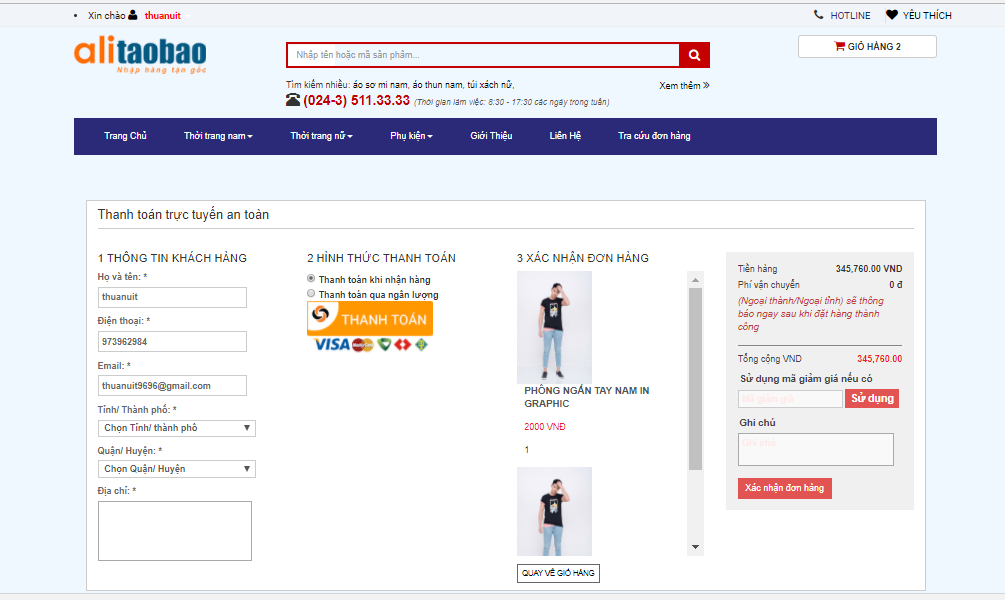
Khi ta nhấn vào liên kết đến trang giỏ hàng thì sẽ có giao diện như bên dưới, với trang giỏ hàng này thì ta có thể: cập nhật lại số lượng cần mua, xóa bỏ sản phẩm đã thêm.



Hình 3.17: Giao diện trang giỏ hàng

* Giao diện trang thanh toán đơn hàng (đã đăng nhập)

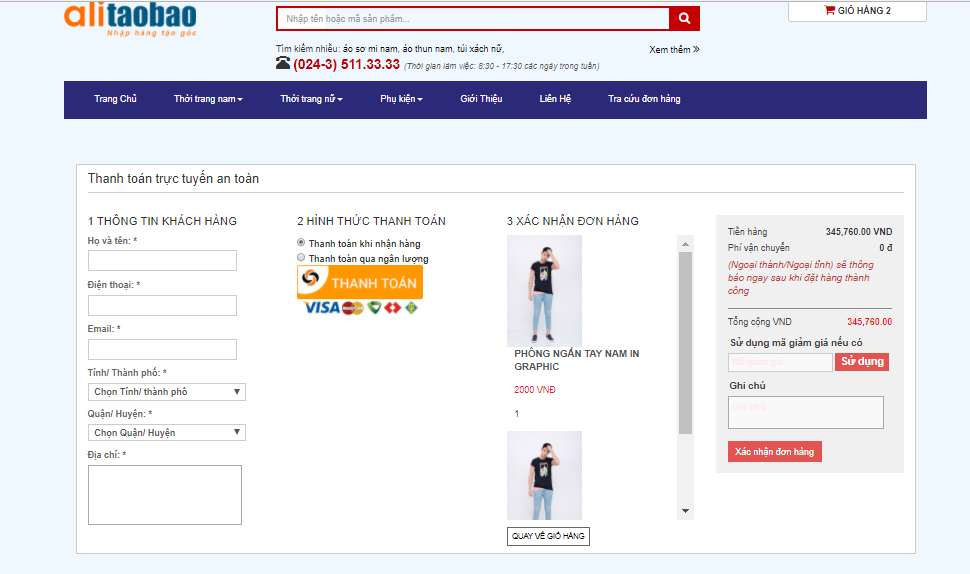
Sau khi vào trang giỏ hàng ta nhấn thanh toán đơn hàng sẽ xảy ra hai trường hợp là nếu khách hàng đã đăng nhập website sẽ hiển thị thông tin của khách hàng đã đăng ký. Khách hàng có thể lựa chọn phương thức thanh toán ngoài ra nếu khách hàng có mã giảm giá có thể áp dụng để được giảm chi phí khi đặt hàng.



Hình 3.18: Giao diện thanh toán đơn hàng (đã đăng nhập)

* Giao diện trang thanh toán đơn hàng (chưa đăng nhập)

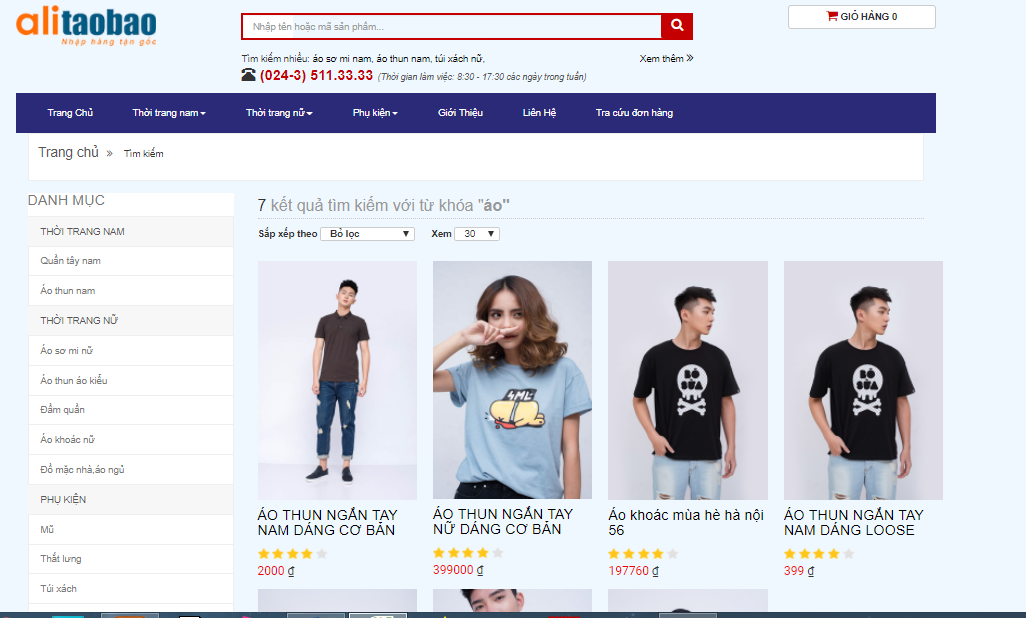
Đối với trường hợp khách hàng chưa đăng nhập thì website sẽ yêu cầu điền thông tin liên lạc.



Hình 3.19: Giao diện trang thanh toán đơn hàng (chưa đăng nhập)

* Giao diện trang tìm kiếm sản phẩm

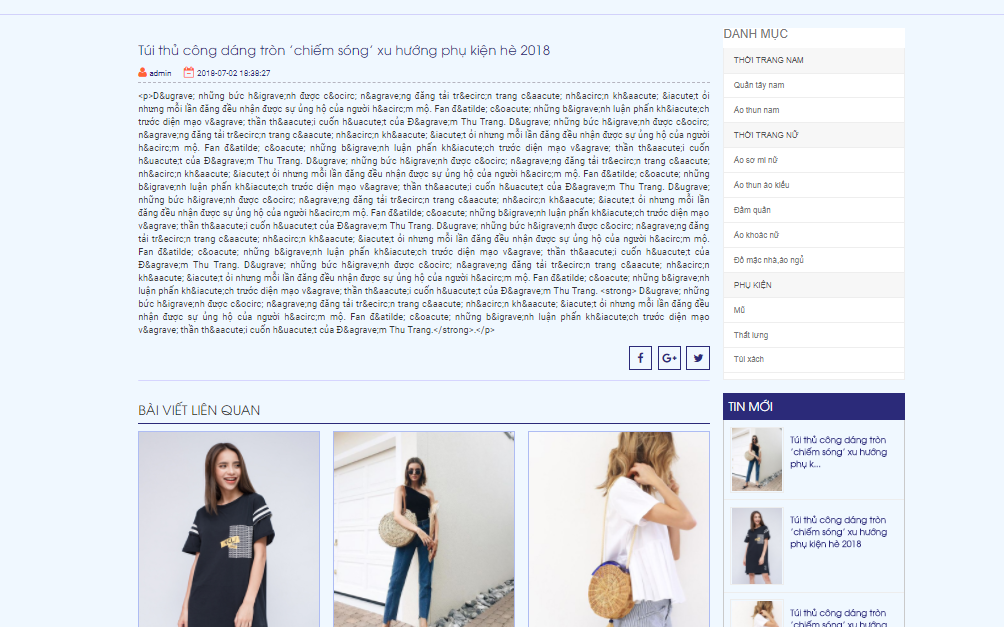
Khi ta nhập từ khóa tìm kiếm sản phẩm trên thanh tìm kiếm thì website sẽ liệt kê tất cả sản phẩm phù hợp với từ khóa đã nhập và trang này sẽ có giao diện như sau:



Hình 3.20: Giao diện trang tìm kiếm sản phẩm

* Giao diện trang chi tiết tin tức

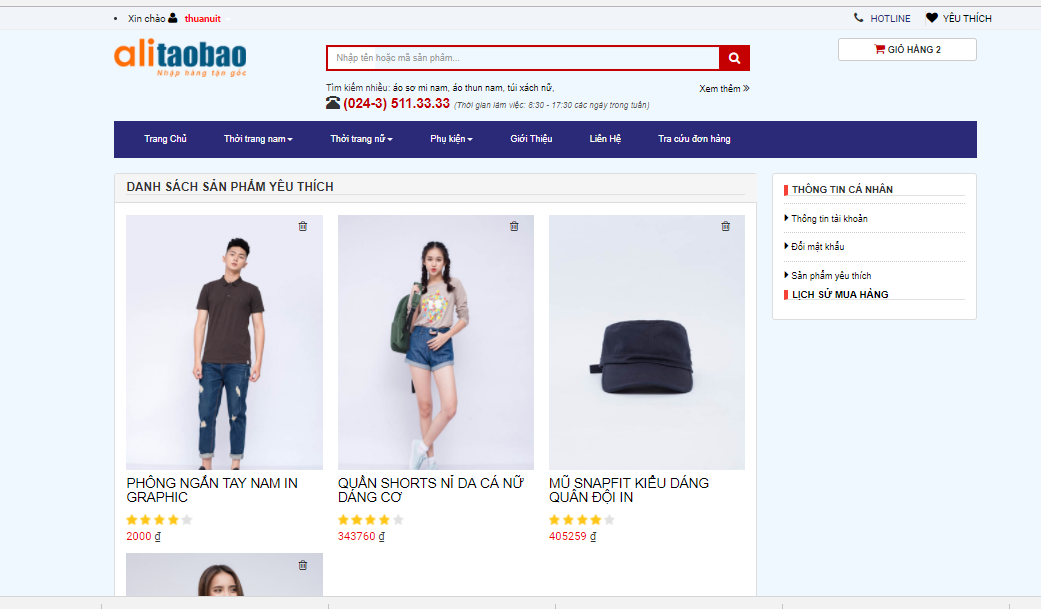
Giao diện trangchi tiết tin tức sẽ hiển thị nội dung của tin tức được ngoài ra còn có mục tin mới và tin liên quan.



Hình 3.21: Giao diện trang chi tiết tin tức

* Giao diện sản phẩm yêu thích

Khi khách hàng đã đăng nhập thì có thể xem danh sách các sản phẩm yêu thích mà mình đã thêm vào danh sách yêu thích

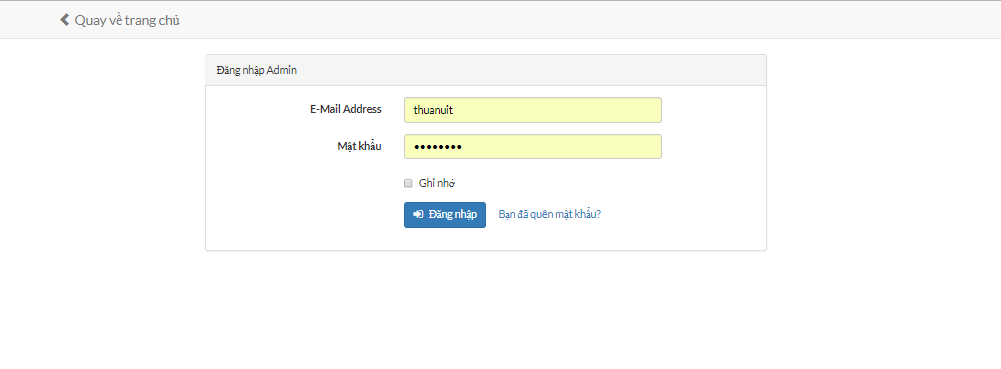


Hình 3.22: Giao diện trang sản phẩm yêu thích

3.4.2. Một số giao diện chính từ trang quản lý

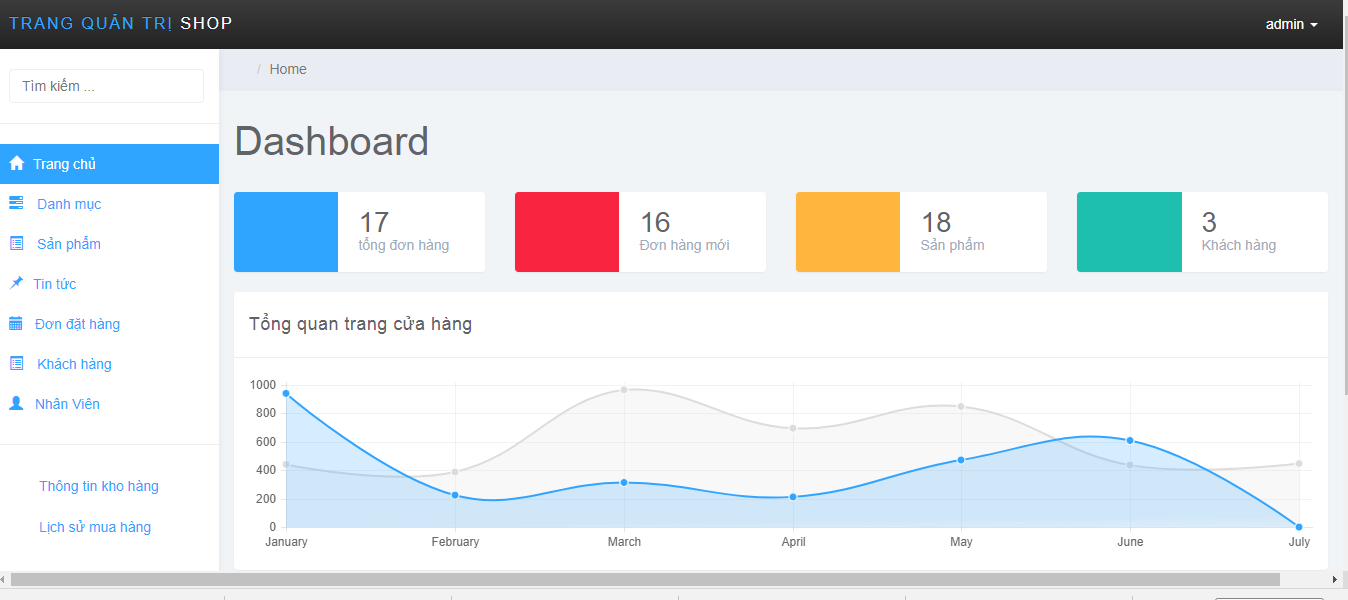
* Giao diện trang đăng nhập quản lý

Khi nhân viên truy cập đến trang quản lý của hệ thống thì giao diện đăng nhập vào hệ thống sẽ như sau:



Hình 3.23: Giao diện trang đăng nhập quản lý

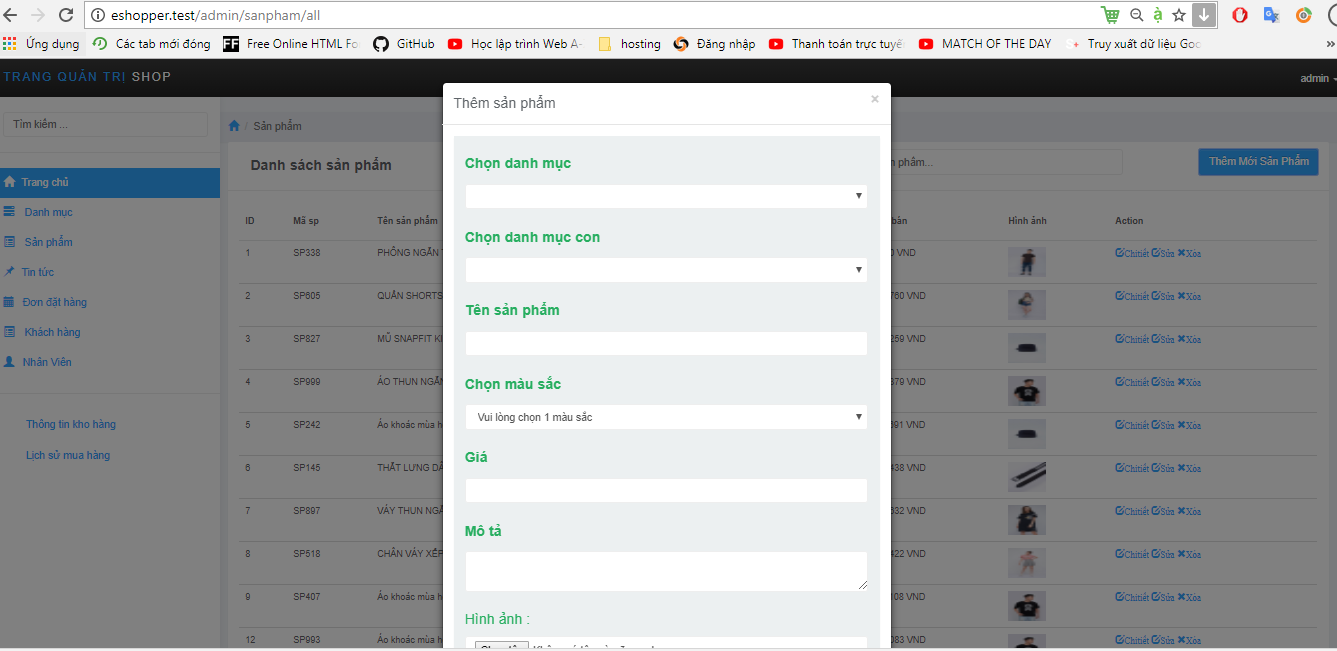
* Giao diện trang chủ: Thống kê tổng đơn hàng, đơn đặt hàng mới sản phẩm và khách hàng



Hình 3.24: Giao diện trang chủ trang admin

* Giao diện thêm sản phẩm

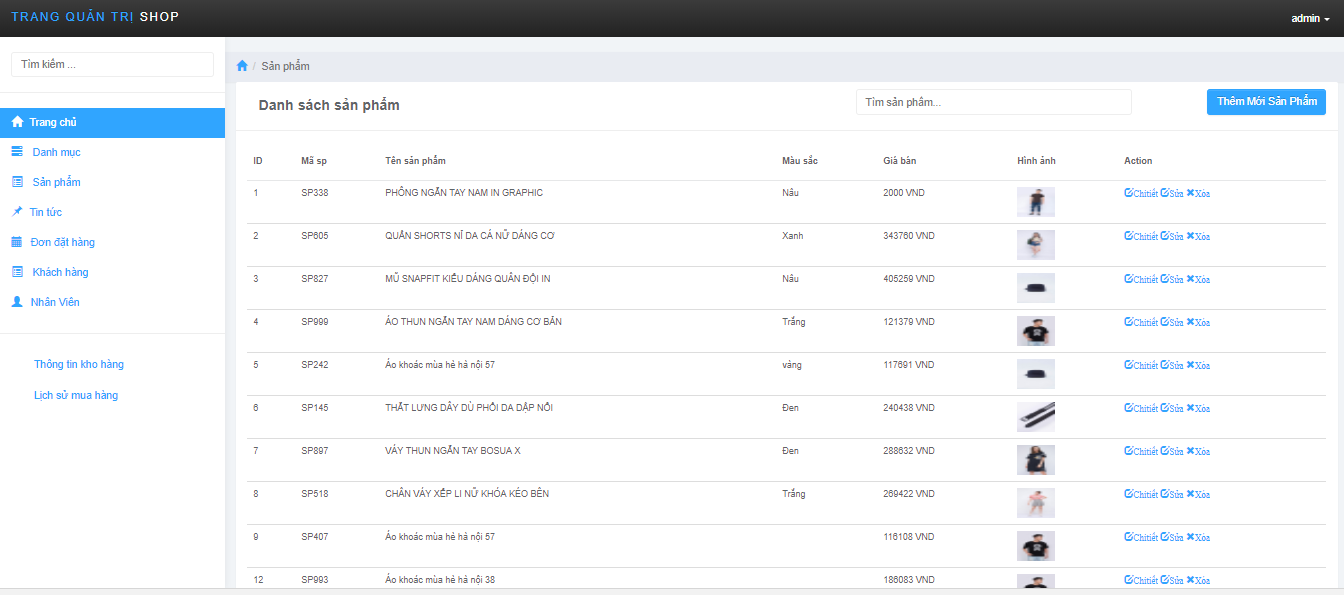
Để thêm sản phẩm người quản lý nhân vào danh mục sản phẩm và chọn thêm sản phẩm, thông tin sản phẩm thêm mới bao gồm: mã sản phẩm, tên sản phẩm, hãng sản xuất, giá sản phẩm …Sau khi nhập hết đầy đủ thông tin quản lý nhấn nút lưu lại để kết thúc quá trình thêm một sản phẩm mới (để có số lượng cho sản phẩm cần nhập hàng cho sản phẩm ở mục chi tiết sản phẩm).



Hình 3.25: Giao diện trang thêm sản phẩm

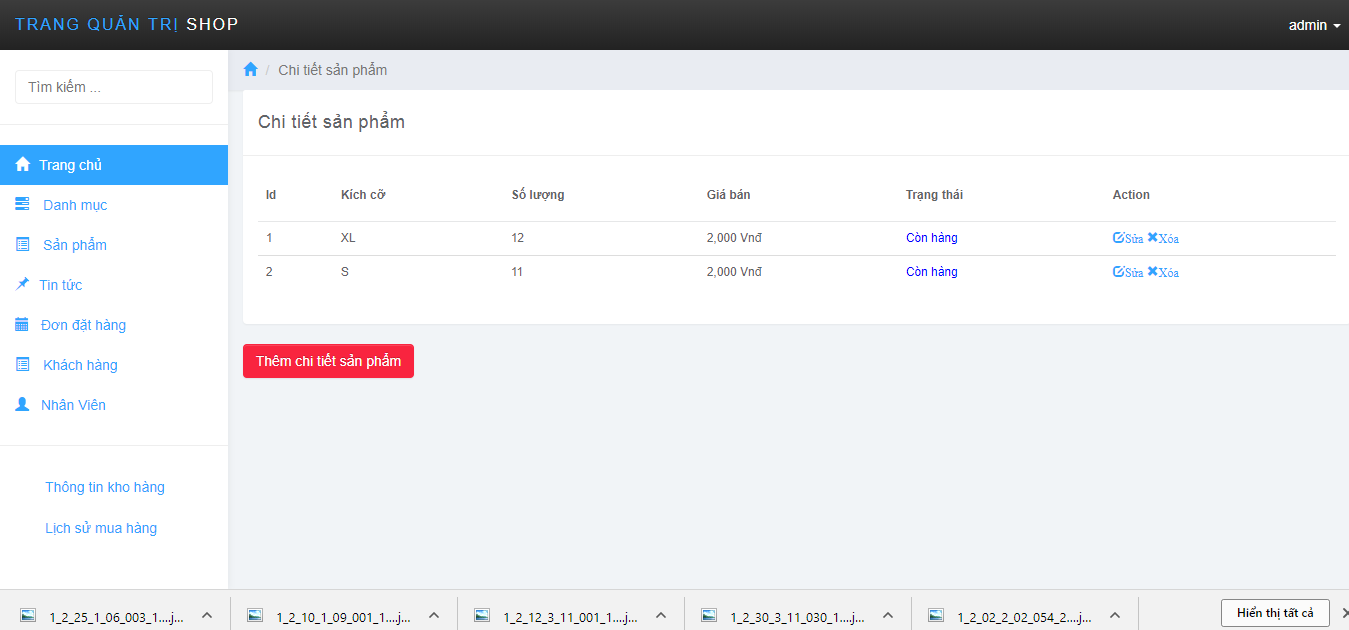
* Giao diện trang danh sách sản phẩm

Giao diện trang danh sách sản phẩm bao gồm: hiện thị tất cả sản phẩm, công cụ tìm kiếm nhanh một sản phẩm, nút thêm mới, xóa, sửa một sản phẩm bất kỳ…



Hình 3.26 Giao diện trang danh sách sản phẩm

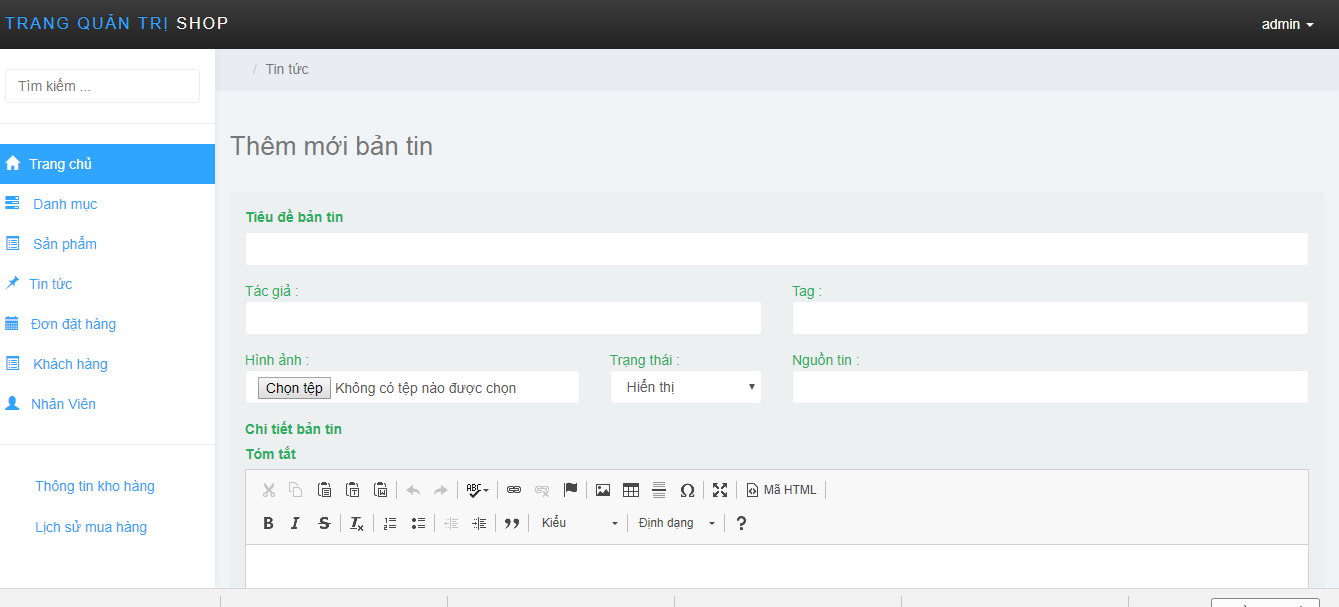
* Giao diện trang chi tiết sản phẩm



Hình 3.27: Giao diện trang chi tiết sản phẩm

* Giao diện thêm tin tức

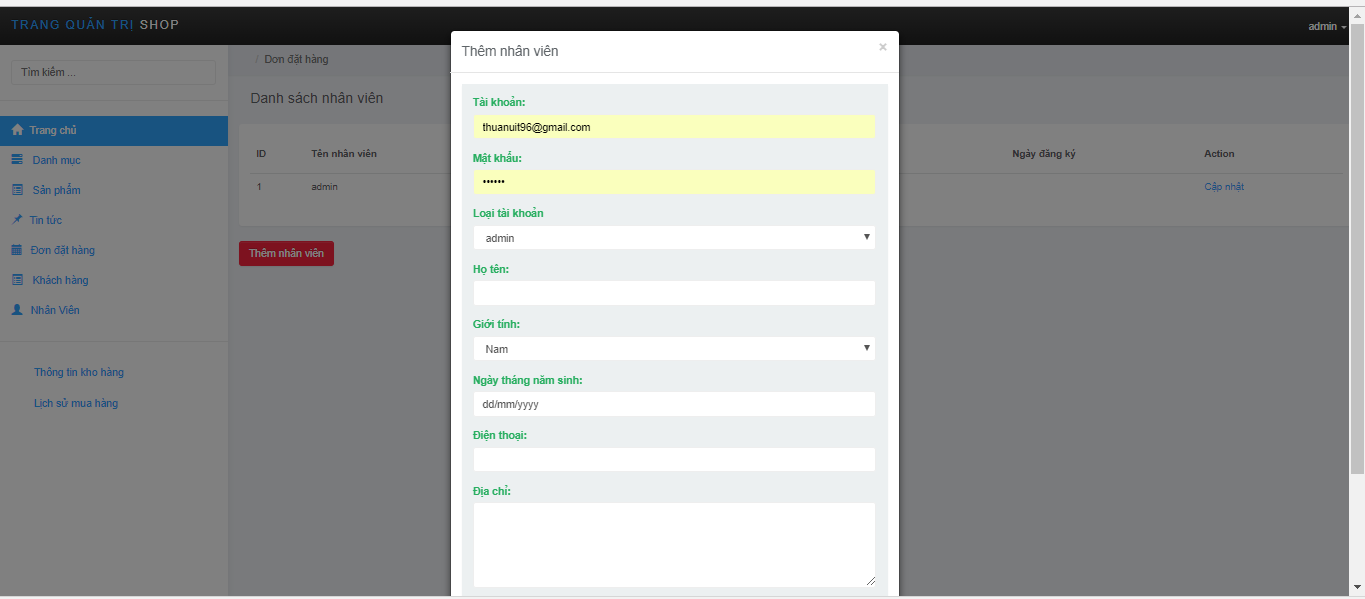
Giao diện thêm tin tức bao gồm các mục như: thông tin tin tức (tiêu đề, ảnh đại diện, trạng thái tin) và nội dung tin tức.



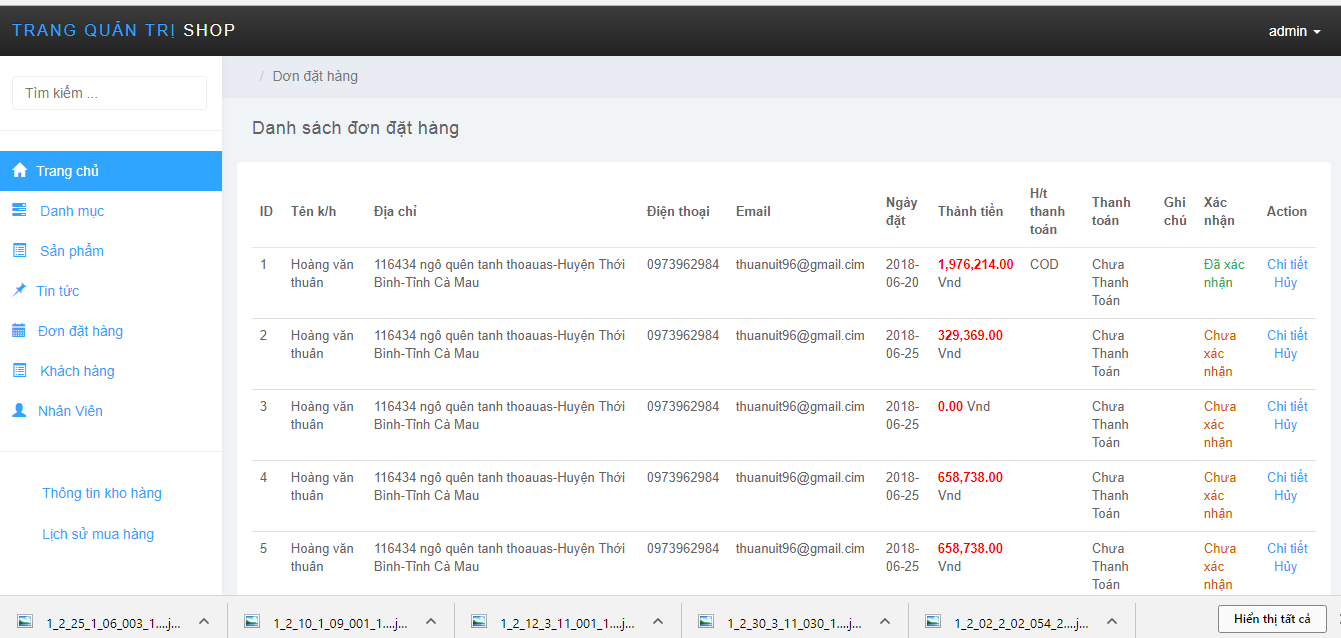
Hình 3.28: Giao diện trang thêm tin tức

* Giao diện trang nhân viên

Các chức năng của trang nay bao gồm cập nhật thông tin cá nhân, thêm nhân viên mới

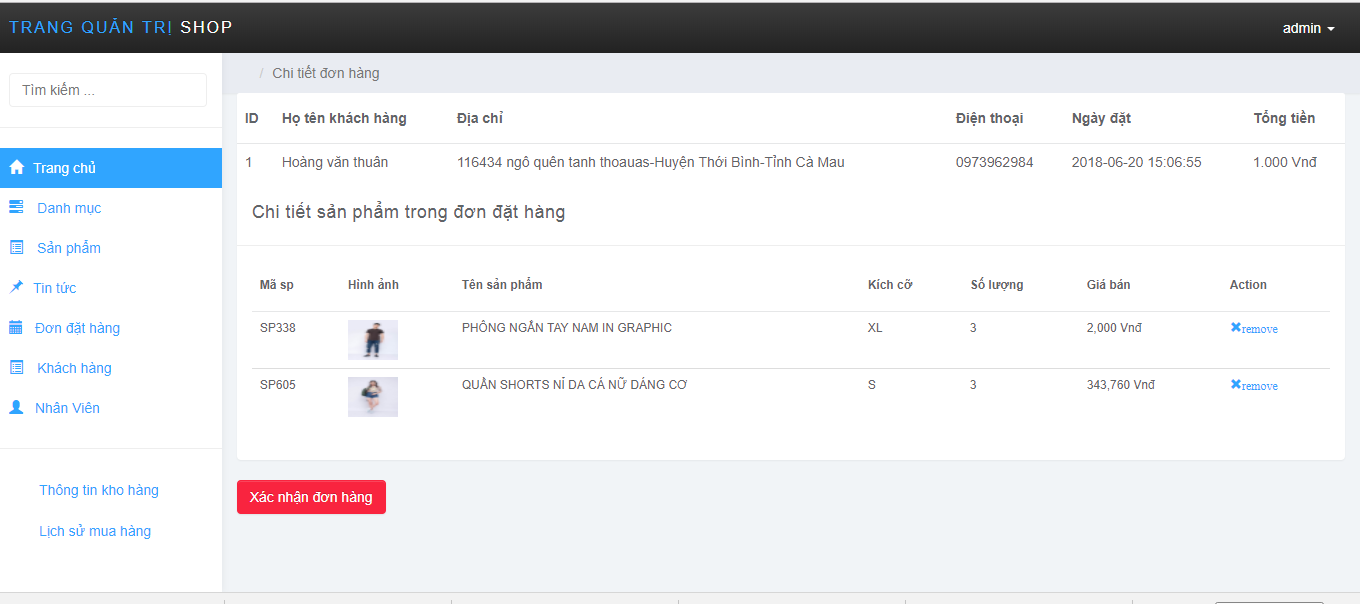


Hình 3.29: Giao diện trang nhân viên

* Giao diện trang đơn đặt hàng

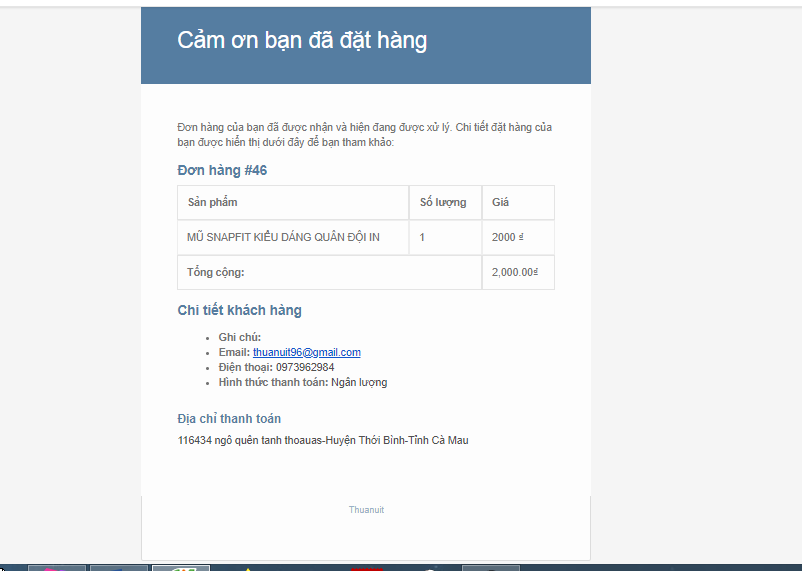
Hình 3.30: Giao diện đơn đặt hàng

* Giao diện trang chi tiết đơn hàng



Hình 3.31: Giao diện chi tiết đơn hàng

* Giao diện trang mail đơn hàng: Khi khách hàng đặt hàng thành công hệ thống sẽ gửi mail thống báo về mail của khách hàng.



Hình 3.32: Giao diện mail thông báo đơn hàng

# CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 4.1. KẾT QUẢ ĐẠT ĐƯỢC

**Về mặt kiến thức**

* Hiểu được lập trình website, mặc khác hiểu được kỹ thuật lập trình website với Laravel framework.
* Nâng cao khả năng phân tích thiết kế hệ thống thông tin, khắc phục điểm yếu, phát triển điểm mạnh cho bản thân.
* Bổ sung thêm nhiều kiến thức về cơ sở dữ liệu, nâng cao kỹ năng PHP, Javascript, jQuery, Ajax …
* Tiếp thu được khả năng lập trình website với mô hình MVC.

**Về kinh nghiệm thực tiễn**

* Phát huy tinh thần sáng tạo, học hỏi giải quyết các vấn đề thực tiễn
* Có kinh nghiệm trong việc phân chia công việc lớn thành việc nhỏ để dễ dàng hoàn thành nhanh chóng, thuận tiên cho việc quản lý và phù hợp với thời gian

## 4.2. ƯU ĐIỂM VÀ HẠN CHẾ CỦA ĐỀ TÀI

**Ưu điểm:**

* Giao diện thân thiện với người dùng, thao tác đơn giản và nhanh chóng.
* Có khả năng liên kết đăng nhập qua facebook, có thể dễ dàng tích hợp thêm vào hệ thống đăng nhập khác như google, github, twitter.
* Việc quảng bá sản phẩm đến email khách hàng nhanh chóng thông qua công cụ makerting.
* Mua sắm trực tuyến nhanh chóng, thực hiện ở mọi nơi có kết nối internet trên laptop hay điện thoại cá nhân của khách hàng.

-Người dùng có thể chia sẻ sản phẩm trên facebook và google plus.

**Hạn chế:**

* Giao diện khách hàng hiển thị trên điện thoại chưa được tối ưu.
* Chức năng còn nhiều hạn chế trong thực tế.
* Việc quản lý sản phẩm không được tối ưu so với thực tế.

## 4.3. HƯỚNG PHÁT TRIỂN

* Tối ưu hóa giao diện người dùng trên điện thoại.
* Phát triển chức năng so sánh sản phẩm hoàn thiện hơn, có chức năng gợi ý sản phẩm cần so sánh.
* Liên kết tài khoản khách hàng với nhiều mạng xã hội khác nhau.

**Tài Liệu Tham Khảo**

*Danh mục tài liệu tiếng việt*

[1] X. C. Vũ, "Tìm hiểu về JQuery và PHP trong việc xây dựng các ứng dụng WEB," Đại học Dân lập Hải Phòng, 2012.

*Danh mục tài liệu tiếng anh*

[2] J. Greenspan and B. Bulger, *MySQL/PHP database applications*. John Wiley & Sons, Inc., 2001.

[3] C. Scollo and S. Shumann, *Professional PHP programming*. Wrox Press Ltd., 1999.

[4] E. Muhardin, "PHP Programming Fundamental dan MySQL Fundamental," *ArtiVisi Intermedia,* 2003.

[5] P. Coad and J. Nicola, *Object-oriented programming*. Yourdon Press Englewood Cliffs, 1993.

[6] H. R. Yu, "Design and implementation of web based on Laravel framework," *Atl. Press, no. Iccset,* vol. 2014, pp. 301-304, 2015.

[7] R. Das and L. P. Saikia, "Comparison of Procedural PHP with Codeigniter and Laravel Framework," *International Journal of Current Trends in Engineering & Research,* vol. 2, no. 6, 2016.

[8] D. D. Dvorski, "Installing, configuring, and developing with Xampp," *Skills Canada,* 2007.

[9] N. Adermann and J. Boggiano, "Composer: Dependency Manager for PHP," *Dostupné z<* <https://getcomposer>*. org/>[cit. 5. 5. 2014],* 2014.