**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Thiết kế và xây dựng website bán voucher**

**Hà Huy Chung**

[chung.hh210749P@sis.hust.edu.vn](mailto:chung.hh210749P@sis.hust.edu.vn)

**Ngành công nghệ thông tin**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | Dr. Nguyễn An Hưng |
|  |  |
| **Khoa:** | Công nghệ thông tin |

|  |
| --- |
| **Hà Nội, 03/2024** |

**LỜI CAM KẾT**

Họ và tên sinh viên: Hà Huy Chung . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  
Điện thoại liên lạc: 0383605884 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  
Email: chung.HH210749P . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   
Lớp: Liên thông k-67 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   
Hệ đào tạo: Liên Thông . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Tôi – Hà Huy Chung – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của học hàm học vị+điền tên giáo viên hướng dẫn. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường

Hà Nội, ngày tháng năm  
Tác giả ĐATN

Hà Huy Chung

**LỜI CÁM ƠN**

Lời đầu tiên, tôi xin gửi lời cảm ơn chân thành và sâu sắc nhất tới các thầy cô của khoa Công nghệ thông tin, trường Đại Học Bách Khoa Hà Nội, những người đã cung cấp những kiến thức quý báu, tạo điều kiện và môi trường học tập tốt nhất cho tôi trong suốt quãng thời gian học tập vừa qua.

Đặc biệt, tôi xin bày tỏ lòng biết ơn sâu sắc tới giáo viên hướng dẫn Nguyễn An Hưng, đã tận tình hướng dẫn, chia sẻ kiến thức chuyên môn, kinh nghiệm thực tiễn, những trải nghiệm bản thân và cung cấp những lời khuyên bổ ích, từ đó giúp tôi có thể hoàn thành đồ án tốt nghiệp này một cách tốt nhất.

Cuối cùng, tôi muốn gửi lời tri ân đến tất cả các giảng viên, bạn bè và mọi người đã đóng góp ý kiến và hỗ trợ tôi trong quá trình thực hiện đồ án này.

Lời cảm ơn này không đủ diễn đạt tất cả, nhưng nó là biểu hiện của lòng biết ơn chân thành và tôn trọng của tôi đối với những người đã giúp đỡ tôi có thể hoàn thành đồ án tốt nghiệp một cách tốt nhất.

Trân trọng,

**TÓM TẮT NỘI DUNG ĐỒ ÁN**

**Tổng quan**

Đồ án này tập trung vào việc thiết kế và phát triển một website bán voucher, nhằm cung cấp một giải pháp tiện lợi cho việc mua sắm và quản lý voucher điện tử. Website này cho phép người dùng dễ dàng tìm kiếm, so sánh và mua các loại voucher từ nhiều nhà cung cấp khác nhau trong một hệ thống thống nhất. Dựa trên nền tảng công nghệ web hiện đại, đồ án không chỉ tập trung vào việc cung cấp một giao diện người dùng thân thiện mà còn đảm bảo tính năng bảo mật cao và quy trình thanh toán linh hoạt.

**Chức năng và tính năng**

Đồ án này phát triển một website bán voucher với mục tiêu cung cấp một trải nghiệm mua sắm trực tuyến mượt mà và tiện lợi cho người dùng. Website bao gồm các chức năng chính như tìm kiếm và lọc voucher theo các tiêu chí khác nhau (ví dụ: loại, giá trị, nhà cung cấp), tính năng đăng nhập/đăng ký cho người dùng, hệ thống giỏ hàng, và quy trình thanh toán an toàn. Ngoài ra, website cũng tích hợp hệ thống quản lý voucher cho phép nhà cung cấp dễ dàng đăng ký và quản lý các sản phẩm voucher của mình.

**Công nghệ**

Trong đồ án này, chúng tôi sử dụng ngôn ngữ lập trình web phổ biến như HTML, CSS, JavaScript cho phía front-end và Java cho phía back-end, kết hợp với cơ sở dữ liệu Mysql để quản lý dữ liệu người dùng và voucher.

**Đối tượng, mục tiêu và lợi ích**

Đồ án này nhắm đến việc phục vụ nhu cầu mua sắm voucher điện tử của người tiêu dùng, từ các voucher giải trí, ẩm thực, du lịch đến giáo dục và dịch vụ. Website không chỉ giúp người dùng tiết kiệm thời gian và công sức trong việc tìm kiếm và so sánh các ưu đãi tốt nhất mà còn mang lại cơ hội cho các doanh nghiệp và nhà cung cấp để tiếp cận một lượng lớn khách hàng tiềm năng thông qua việc đăng ký và quảng bá voucher trên nền tảng này.

**Table of contents**

[CHAPTER 1: INTRODUCTION 1](#_2et92p0)

[1.1 Problem 1](#_3dy6vkm)

[1.2 Objectives and scope of the topic 1](#_1t3h5sf)

[1.3 Solution Orientation 1](#_4d34og8)

[1.4 Thesis organization 2](#_2s8eyo1)

[CHAPTER 2: SURVEY AND ANALYSIS SYSTEM 3](#_3rdcrjn)

[2.1 Current status survey 3](#_26in1rg)

[2.1.1 Survey the user 3](#_lnxbz9)

[2.1.2 Surveying system available 3](#_35nkun2)

[2.1.3 Conclude 5](#_1ksv4uv)

[2.2 Functional overview 6](#_44sinio)

[2.2.1 Use-case Specification 6](#_2jxsxqh)

[2.2.2 Decomposition of Use-case Category Management 7](#_3j2qqm3)

[2.2.3 Decomposition of Use-case Account Management 8](#_4i7ojhp)

[2.2.4 Decomposition of Use-case Product Management 8](#_1ci93xb)

[2.2.5 Decomposition of Use-case Order Management 9](#_2bn6wsx)

[2.2.6 Decomposition of Use-case Brand Management 9](#_3as4poj)

[2.2.7 Decomposition of Use-case Blog Management 10](#_49x2ik5)

[2.2.8 Decomposition of Use-case Promotion Management 10](#_147n2zr)

[2.3 Usecase detail specification 11](#_3o7alnk)

[2.3.1 Login 11](#_23ckvvd)

[2.3.2 Register account 12](#_32hioqz)

[2.3.3 Product search 12](#_41mghml)

[2.3.4 Buy the product 13](#_vx1227)

[2.3.5 View product details 14](#_1v1yuxt)

[2.3.6 Add a product 14](#_2u6wntf)

[2.3.7 Order management 15](#_3tbugp1)

[2.3.8 Category management 16](#_nmf14n)

[2.3.9 Brand management 16](#_1mrcu09)

[2.3.10 Post management 17](#_2lwamvv)

[2.3.11 Promotion management 18](#_111kx3o)

[2.4 Sequence diagram 19](#_3l18frh)

[2.4.1 Login function 19](#_206ipza)

[2.4.2 Register function 19](#_4k668n3)

[2.4.3 Search function 20](#_2zbgiuw)

[2.4.4 View product function 20](#_1egqt2p)

[2.4.5 Comment function 20](#_3ygebqi)

[2.4.6 Buy product function 21](#_2dlolyb)

[2.4.7 Category management function 21](#_sqyw64)

[2.4.8 Brand management function 22](#_3cqmetx)

[2.4.9 Blog management function 22](#_1rvwp1q)

[2.4.10 Promotion management function 23](#_4bvk7pj)

[2.4.11 Order management function 23](#_2r0uhxc)

[2.4.12 Product management function 24](#_1664s55)

[CHAPTER 3: TECHNOLOGY USED 25](#_3q5sasy)

[3.1 Introduction to Spring boot 25](#_25b2l0r)

[3.1.1 Spring boot overview 25](#_kgcv8k)

[3.1.2 Advantages of Spring boot 26](#_1jlao46)

[3.1.3 Disadvantages of Spring boot 26](#_43ky6rz)

[3.2 Introduction to HTML 27](#_xvir7l)

[3.3 CSS 28](#_1x0gk37)

[3.4 JavaScript 29](#_2w5ecyt)

[3.5 MySQL 30](#_3vac5uf)

[3.6 XAMPP 31](#_pkwqa1)

[CHAPTER 4: APPLICATION DEVELOPMENT AND IMPLEMENTATION 32](#_1opuj5n)

[4.1 Architectural design 32](#_48pi1tg)

[4.2](#_1302m92) Detailed design 33

[4.2.1 Database design 33](#_3mzq4wv)

[4.3](#_184mhaj) Application construction 39

[4.3.1 Libraries and tools used 39](#_3s49zyc)

[4.3.2 Illustrate the main functions 39](#_279ka65)

[4.4](#_1d96cc0) Testing 44

[4.4.1](#_j8sehv) About the interface 44

[4.4.2](#_338fx5o) About the function 44

[4.4.3](#_1idq7dh) About performance 44

[4.5](#_42ddq1a) Deployment 44

[4.5.1](#_2hio093) Install on website server 44

[CHAPTER 5: CONCLUSION AND DEVELOPMENT DIRECTION 45](#_2pta16n)

[4.6](#_14ykbeg) Result 45

[4.7](#_3oy7u29) Restrictions exist 45

[4.8](#_243i4a2) Development 46

[References 47](#_243i4a2)

**List of Figures**

[Figure 2.2.1: Use-case Specification 6](#_z337ya)

[Figure 2.2.2: Decomposition of Use-case Category Management 7](#_1y810tw)

[Figure 2.2.3: Decomposition of Use-case Account Management 8](#_2xcytpi)

[Figure 2.2.4: Decomposition of Use-case Product Management 8](#_3whwml4)

[Figure 2.2.5: Decomposition of Use-case Order Management 9](#_qsh70q)

[Figure 2.2.6: Decomposition of Use-case Brand Management 9](#_1pxezwc)

[Figure 2.2.7: Decomposition of Use-case Blog Management 10](#_2p2csry)

[Figure 2.2.8: Decomposition of Use-case Promotion Management 10](#_2p2csry)

[Figure 2.4.1: Sequence diagram login function 19](#_wnyagw)

[Figure 2.4.2: Sequence diagram register function 19](#_wnyagw)

[Figure 2.4.3: Sequence diagram search function 20](#_wnyagw)

[Figure 2.4.4: Sequence diagram view product function 20](#_wnyagw)

[Figure 2.4.5: Sequence diagram comment function 20](#_wnyagw)

[Figure 2.4.6: Sequence diagram buy product function 21](#_wnyagw)

[Figure 2.4.7: Sequence diagram category management function 21](#_wnyagw)

[Figure 2.4.8: Sequence diagram brand management function 22](#_wnyagw)

[Figure 2.4.9: Sequence diagram blog management function 22](#_wnyagw)

[Figure 2.4.10: Sequence diagram promotion management function 23](#_wnyagw)

[Figure 2.4.11: Sequence diagram order management function 23](#_wnyagw)

[Figure 2.4.12: Sequence diagram product management function 24](#_wnyagw)

[Figure 3.1: Logo of Spring boot 25](#_34g0dwd)

[Figure 3.2: HTML 27](#_3hv69ve)

[Figure 3.3: CSS 28](#_4h042r0)

[Figure 3.4: JavaScript 29](#_1baon6m)

[Figure 3.5: MySQL 30](#_2afmg28)

[Figure 3.6: XAMPP 31](#_39kk8xu)

[Figure 4.1: MVC pattern illustration 32](#_2nusc19)

[Figure 4.2: Database 33](#_3gnlt4p)

[Figure 4.3: Home page 39](#_meukdy)

[Figure 4.4: Account registration page 40](#_36ei31r)

[Figure 4.5: Account login page 40](#_1ljsd9k)

[Figure 4.6: The page displays user information 40](#_45jfvxd)

[Figure 4.7: The page displays order history 41](#_2koq656)

[Figure 4.8: The page displays all product 41](#_zu0gcz)

[Figure 4.9: Product search page 42](#_3jtnz0s)

[Figure 4.10: The page displays the product details 42](#_1yyy98l)

[Figure 4.11: Payment 43](#_4iylrwe)

[Figure 4.12: Admin Dashboard 43](#_2y3w247)

**List of tables**

[Table 2.1: Login 11](#_ihv636)

[Table 2.2: Register account 12](#_1hmsyys)

[Table 2.3: Product search 12](#_2grqrue)

[Table 2.4: Buy the product 13](#_3fwokq0)

[Table 2.5: View product details 14](#_4f1mdlm)

[Table 2.6: Add a product 14](#_19c6y18)

[Table 2.7: Order management 15](#_28h4qwu)

[Table 2.8: Category management 16](#_37m2jsg)

[Table 2.9: Brand management 16](#_46r0co2)

[Table 2.10: Post management 17](#_46r0co2)

[Table 2.11: Promotion management 18](#_46r0co2)

[Table 4.1: Table “User” 34](#_haapch)

[Table 4.2: Table “Blog” 34](#_319y80a)

[Table 4.3: Table “Images” 35](#_1gf8i83)

[Table 4.4: Table “Brand” 35](#_40ew0vw)

[Table 4.5: Table “Category” 35](#_2fk6b3p)

[Table 4.6: Table “ Product” 35](#_1vsw3ci)

[Table 4.7: Table “Product\_Category” 36](#_upglbi)

[Table 4.8: Table “Product\_Size” 36](#_4fsjm0b)

[Table 4.9: Table “Promotion” 36](#_3ep43zb)

[Table 4.10: Table “Order” 37](#_1tuee74)

[Table 4.11: Table “Comment” 38](#_4du1wux)

[Table 4.12: Table “Statistic” 38](#_2szc72q)

# Chương 1: Giới thiệu

Chương giới thiệu

Trong chương này, tôi sẽ thảo luận về việc giao dịch với sự phát triển của công nghệ và ứng dụng của công nghệ thông tin vào việc bán hàng với những ưu điểm nổi bật. Tiếp theo là một cái nhìn tổng quan về thông tin chung hơn về các sản phẩm hiện có và từ đó đưa ra nhận xét và định hướng trong quá trình thiết kế và phát triển phần mềm.

## Đặt vấn đề

Trong thời đại số hóa ngày nay, việc tiếp cận thông tin và mua sắm trực tuyến đã trở nên không chỉ tiện lợi mà còn là một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta. Đặc biệt, thị trường voucher - phiếu giảm giá cho các dịch vụ và sản phẩm từ giải trí, ẩm thực đến du lịch và học tập - đang bùng nổ và cho thấy một tiềm năng tăng trưởng mạnh mẽ. Tuy nhiên, việc tìm kiếm và sử dụng voucher một cách hiệu quả lại không phải lúc nào cũng dễ dàng. Người tiêu dùng thường phải đối mặt với những thách thức như thông tin voucher không rõ ràng, quá trình thanh toán phức tạp, và thiếu tính minh bạch trong quy định sử dụng. Điều này đã tạo ra một nhu cầu cấp thiết cho việc phát triển một giải pháp trực tuyến, cụ thể là một website bán voucher, nhằm mục đích cung cấp một kênh mua sắm trực tuyến an toàn, minh bạch và tiện lợi cho cả người tiêu dùng và các nhà cung cấp. Đồ án này nhằm mục đích khám phá và giải quyết những vấn đề này bằng cách thiết kế và phát triển một website bán voucher, với hy vọng không chỉ cải thiện trải nghiệm mua sắm trực tuyến của người tiêu dùng mà còn hỗ trợ các doanh nghiệp trong việc quảng bá và quản lý hiệu quả sản phẩm voucher của mình.

## Mục tiêu và phạm vi của đề tài

Phát triển một nền tảng trực tuyến: Tạo ra một website bán voucher dễ sử dụng, thân thiện với người dùng, cho phép người tiêu dùng tìm kiếm, so sánh và mua voucher một cách tiện lợi.

Cải thiện trải nghiệm người dùng: Tối ưu hóa quy trình tìm kiếm và thanh toán trên website để đảm bảo trải nghiệm mua sắm suôn sẻ và nhanh chóng cho người tiêu dùng.

Tăng cường tính minh bạch và tin cậy: Đảm bảo tất cả thông tin về voucher, bao gồm giá cả, điều kiện sử dụng, và hạn sử dụng, được trình bày một cách rõ ràng và dễ hiểu.

Hỗ trợ cho các nhà cung cấp: Cung cấp một công cụ hiệu quả để các doanh nghiệp và nhà cung cấp có thể quảng bá và quản lý voucher của mình, từ đó mở rộng thị trường và tăng doanh thu.

Hỗ trợ Người Dùng: Thiết lập hệ thống hỗ trợ trực tuyến để giải quyết thắc mắc và vấn đề của người tiêu dùng và nhà cung cấp một cách kịp thời.

## Định hướng và giải pháp

Đề tài tập trung vào việc phát triển một nền tảng trực tuyến dành cho việc bán và quản lý voucher, giúp kết nối người tiêu dùng và các nhà cung cấp dịch vụ, sản phẩm một cách hiệu quả. Mục tiêu chính là tạo ra một trải nghiệm mua sắm thuận tiện, an toàn và minh bạch cho người dùng, đồng thời cung cấp một giải pháp quản lý và phân phối voucher linh hoạt cho doanh nghiệp.

Đầu tiên, người quản trị sẽ có nhiều quyền kiểm soát hơn đối với dữ liệu của trang web, có thể cập nhật lịch sử đơn hàng, thống kê sản phẩm, khởi động các chương trình khuyến mãi trong khoảng thời gian cụ thể cho từng sản phẩm, quản lý doanh thu và xử lý thanh toán của người dùng, quản lý doanh thu, và phân quyền người dùng.

Nhân viên có thể mở tài khoản cá nhân và quản lý thông tin đơn hàng, quản lý và phân chia đơn hàng cho người giao hàng, kiểm tra lịch sử giao hàng, chuyển đổi trạng thái đơn hàng, quản lý sản phẩm, quản lý kho.

Người giao hàng có thể đăng nhập và kiểm tra thông tin đơn hàng được phân công.

Người dùng có thể mở tài khoản cá nhân để có thể thực hiện mua sắm, thay đổi thông tin cá nhân, xem thông tin lịch sử đơn hàng mua sắm.

Bằng cách tập trung vào việc phát triển các giải pháp công nghệ tiên tiến và tối ưu hóa trải nghiệm người dùng, dự án này hướng đến việc tạo ra một nền tảng bán voucher đáng tin cậy, thuận tiện và bảo mật cho cả người tiêu dùng và nhà cung cấp.

## Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp sẽ được tổ chức như sau:

Trong chương 2, tôi sẽ mô tả tổng quan về các khảo sát hiện tại về người dùng và hệ thống. Sau đó, là phân tích các chức năng và quy trình hoạt động dựa trên biểu đồ UML, thiết kế biểu đồ lớp, mô tả ngắn gọn các lớp, thiết kế cơ sở dữ liệu và mô tả ngắn gọn các bảng trong cơ sở dữ liệu.

Trong chương 3, tôi sẽ liệt kê và giới thiệu các công nghệ và ngôn ngữ lập trình được sử dụng trong đồ án tốt nghiệp.

Trong chương 4, tôi sẽ bao gồm các kết luận, kết quả đạt được sau quá trình phát triển đồ án tốt nghiệp, các hạn chế tồn tại, hướng phát triển trong tương lai, và cuối cùng là danh sách các tài liệu tham khảo đã được sử dụng trong quá trình xây dựng đồ án tốt nghiệp.

Trong chương 5 là một tóm tắt về những gì bạn đã thực hiện trong hệ thống của mình, cùng với các hạn chế và phát triển trong tương lai để cải thiện hệ thống.

# Chương 2: Khảo sát và phân tích yêu cầu

Để giải quyết những vấn đề được mô tả trong chương 1 và phát triển hệ thống bán hàng trực tuyến, chương 2 sẽ tiến hành khảo sát và phân tích các yêu cầu trong một hệ thống bán hàng trực tuyến. Ở phần 1, đối tượng mà tôi nhắm đến là những người đã sử dụng các trang web bán hàng trực tuyến hiện nay. Ở phần 2, tôi chọn một hệ thống hiện tại và phân tích các chức năng cần thiết của hệ thống đó để đặt ra những mục tiêu rõ ràng cho hệ thống của mình.

## 2.1 Khảo sát hiện trạng

### 2.1.1 Khảo sát ý kiến người dùng

Khảo sát một số người dùng đang sử dụng các ưu đãi bán hàng trực tuyến đưa ra một số yêu cầu sau đây:

* Đầu tiên là đăng ký/đăng nhập để quản lý tài khoản đó.
* Các chức năng cơ bản của người dùng như thay đổi thông tin cá nhân, mật khẩu, hình đại diện.
* Kiểm tra trạng thái đơn hàng.
* Lịch sử giao dịch trên hệ thống.
* Sự đa dạng về sản phẩm và đầy đủ thông tin như tên sản phẩm, giá, danh mục, mô tả,...
* Nhiều loại sản phẩm khác nhau.
* Áp dụng chức năng giảm giá để hỗ trợ thu hút người dùng mới và người dùng quay lại.
* Có đầy đủ thông tin như địa chỉ liên hệ, số điện thoại, đơn vị quản lý,...
* Có chức năng chăm sóc khách hàng, trò chuyện trực tiếp và gián tiếp.
* Có chức năng tìm kiếm, sắp xếp và lọc.

### 2.1.2 Khảo sát các hệ thống có sẵn

#### 2.1.2.1 Khảo sát hệ thống quà tặng điện tử gotit

Gotit là một hệ thống ra đời vào năm 2015 và tính đến thời điểm hiện tại được liên kết với nhiều thương hiệu lớn như shopee, lazada, cgv, pnj,…bằng các khảo sát website này tôi đã rút ra được một số điểm trong hệ thống của họ

Hình thức mua: khá phức tạp để mua

Hình thức đăng nhập, đăng ký: không có đăng nhập đăng ký, gửi thông tin mua hàng trực tiếp qua email, SMS, zalo,… hoặc có thể nhận trực tiếp

Cách sử dụng: có thể dùng trên website, ứng dụng, cửa hàng, …

Trải nghiệm mua sắm: Bạn có thể mua ở nhà, tại văn phòng, quán cà phê hoặc bất kỳ đâu, bất cứ lúc nào miễn là thiết bị của bạn (máy tính để bàn, laptop, điện thoại thông minh...) được kết nối với internet.

Các chức năng có sẵn trên hệ thống Gotit:

* Cung cấp các giải pháp quà tặng: tặng một hoặc nhiều người,…
* Mua thẻ quà, voucher
* Tặng voucher qua zalo, email, sms,…
* Hướng dẫn sử dụng thẻ quà trên website, tại quầy và các hình thức khác, …
* Chuyển đồi thẻ quà giấy sang thẻ quà điện tử
* Chọn giờ gửi voucher

Có 3 phương thức thanh toán được Gotit sử dụng:

* Mua online bằng thẻ quốc tế
* Mua online bằng thẻ ATM nội địa
* Mua online qua Zalo Pay

Dựa trên các điểm phân tích trên, tôi rút ra một điểm cần lưu ý với hệ thống Gotit: Sản phẩm rất đa dạng, được phân phối từ các nhà cung cấp lớn, ai cũng có thể sử dụng được, tập trung nhiểu vào các ngày lễ. có thể thấy website này tập trung vào mọi đối tượng khách hàng nhưng điểm nhấn ở đây chính là các đối tượng được đề cập nhiều trong các ngày lễ lơn trong năm như thiếu nhi, phụ nữ.

#### 2.1.2.2 Khảo sát trang web dealtoday:

Các chức năng có sẵn trên website:

* Đăng nhập, đăng ký, quên mật khẩu,
* Quản lý tài khoản cá nhân, chỉnh sửa thông tin
* Có chức năng giỏ hàng để thuận tiện mua hàng
* Đa dạng sản phẩm voucher
* Sự đa dạng về sản phẩm, có nhiều sự lựa chọn cho khách hàng
* Có thể tìm kiếm theo tiêu chí người dùng
* Chính sách giao hàng phong phú

Khác với Gotit, dealtoday có nhiều chức năng và chính sách phong phú, giao diện người dùng bắt mắt khiến người dùng rất có thiện cảm từ lần đầu tiên truy cập. Có nhiều chức năng chăm sóc khách hàng như hotline, chatbot thuận tiện cho việc giải đáp thắc mắc cũng như tiếp nhận ý kến từ khách hàng.

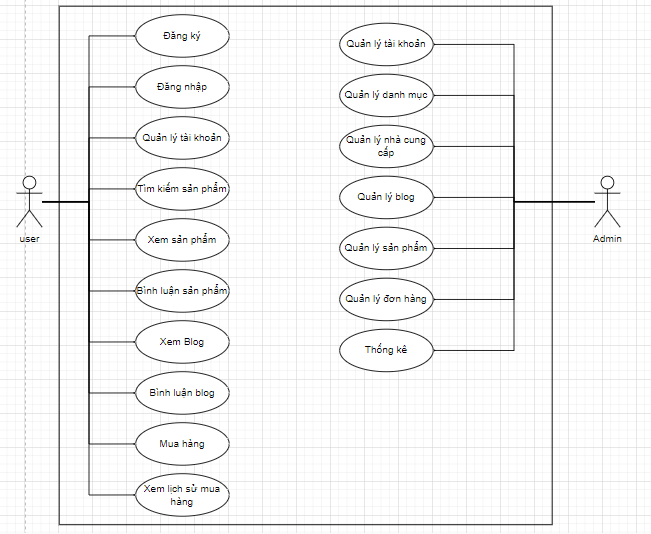
### 2.1.3 Kết luận

Dựa trên phân tích từ 2 hệ thống trên, tôi rút ra một số chức năng cần thiết trong trong hệ thống:

* Đăng nhập, đăng ký
* Quản lý tài khoản cá nhân, cho phép chỉnh sửa thông tin
* Kiểm tra trạng thái đơn hàng, lịch sử giao dịch trên hệ thống
* Đa dạng sản phẩm cũng như đầy đủ thông tin như tên sản phẩm, giá sản phẩm, danh mục, mô tả, bình luận về sản phẩm
* Các chính sách thanh toán và giao hàng
* Nhà cung cấp các loại voucher
* Hình ảnh sản phẩm bắt mắt
* Tìm kiếm sản phẩm theo yêu cầu, nhu cầu của khách hàng
* Cần quan tấm đến nhiều loại đối tượng khách hàng như phụ nữ, trẻ em,.. cũng như các ngày lễ lớn trong năm

## 2.2 Tổng quan chức năng

### 2.2.1 Biểu đồ use case tổng quát



*Hình 2.2.1: biểu đồ use case tổng quát*

Hình 2.1 hiển thị sơ đồ Use Case chung của hệ thống. Trong hình chúng ta có thể thấy trong hệ thống có 2 tác nhân và chức năng chính tương ứng với các tác nhân như sau:

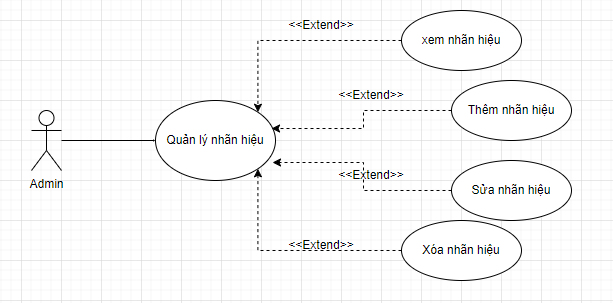
Tác nhân User : Là loại tác nhân cơ bản nhất, có thể thực hiện các chức năng cơ bản như:

* Đăng ký
* Đăng nhập
* Quản lý tài khoản
* Tìm kiếm sản phẩm
* Xem thông tin sản phẩm
* Bình luận sản phẩm
* Xem blog
* Bình luận blog
* Mua hàng
* Xem lịch sử mua hàng

Tác nhân quản trị viên (admin): Là loại tác nhân có quyền hạn cao nhất trong hệ thống, ngoài việc kế thừa các chức năng của người dùng, quản trị viên có thể:

* Quản lý thành viên trên hệ thống
* Quản lý danh mục
* Quản lý nhà cung cấp
* Quản lý blog
* Quản lý sản phẩm
* Thống kê

### 2.2.2 Biểu đồ phân rã chức năng quản lý nhãn hiệu



*Hình 2.2.2: Biểu đồ phân rã chức năng quản lý nhãn hiệu*

**Xem nhãn hiện**: Xem thông tin nhãn hiệu : tên, ảnh,...

**Sửa nhãn hiệu**: Sửa các thông tin về nhãn hiệu: Tên, ảnh,...

**Xóa nhãn hiệu**: Xóa nhãn hiệu

**Thêm nhãn hiệu**: Thêm nhãn hiệu mới

### 2.2.3 Biểu đồ phân rã chức năng Quản lý tài khoản



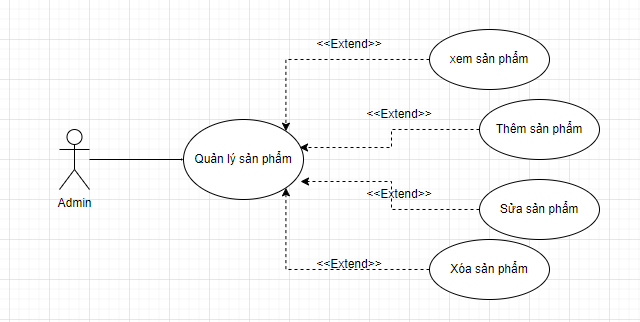
*Hình 2.2.3: Biểu đồ phân rã chức năng Quản lý tài khoản*

**Sửa thông tin cá nhân**: Sửa thông tin cá nhân: têm, email, điện thoại, ngày sinh , địa chỉ.

**Bảo mật**: Thay đổi mật khẩu

**Xem thông tin cá nhân**: Xem thông tin cá nhân của tài khoản hiện tại

### 2.2.4 Biểu đồ phân rã chức năng quản lý sản phẩm



*Figure 2.2.4: Biểu đồ phân rã chức năng quản lý sản phẩm*

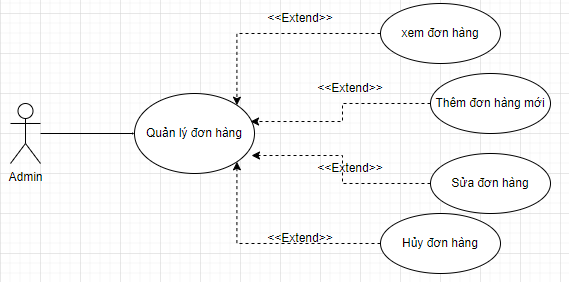
**Xem sản phẩm**: Xem thông tin của sản phẩm: Tên, nhãn hiệu, danh mục, giá nhập, giá bán

**Xóa sản phẩm**: Xóa sản phẩm

**Cập nhật sản phẩm**: Xem thông tin của sản phẩm: Tên, nhãn hiệu, danh mục, giá nhập, giá bán

**Thêm mới sản phẩm**: thêm mới sản phẩm với các thông tin: Tên, nhãn hiệu, danh mục, giá nhập, giá bán

### 2.2.5 Biểu đồ phân rã chức năng quản lý đơn hàng



*Figure 2.2.5: Biểu đồ phân rã chức năng quản lý đơn hàng*

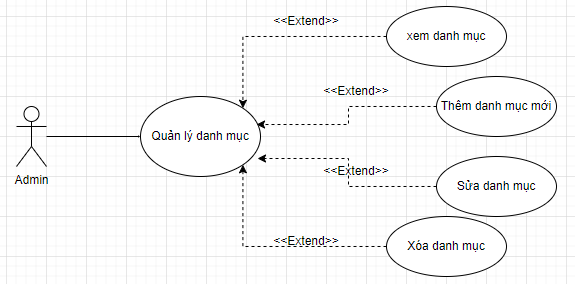
**Xem đơn hàng:** Xem thông tin của đơn hàng: Mã đơn hàng, Giá, sản phẩm,..

**Sửa đơn hàng:** Sửa thông tin người nhận, số điện thoại, ghi chú, trạng thái,..

**Hủy đơn hàng:** Hủy đơn hàng.

**Thêm đơn hàng mới:** Tạo mới đơn hàng với các thông tin: sản phẩm, họ tên, Số điện thoại, địa chỉ, ghi chú

### 2.2.6 Biểu đồ phân rã chức năng quản lý danh mục



*Figure 2.2.6: Biểu đồ phân rã chức năng quản lý danh mục*

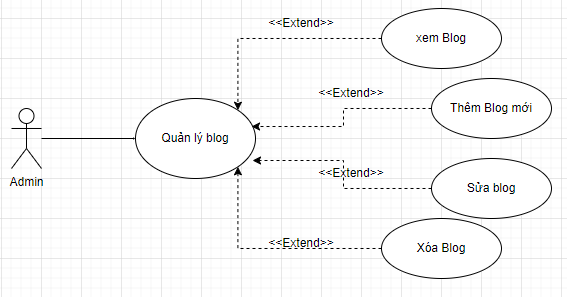
**Xem danh mục:** xem các thông tin của danh mục

**Chỉnh sửa danh mục:** Chỉnh sửa các thông tin của danh mục

**Xóa danh mục:** Xóa danh mục.

**Thêm mới danh mục:** Thêm một danh mục mới.

### 2.2.7 Biểu đồ phân rã chức năng quản lý blog



*Figure 2.2.7: Biểu đồ phân rã chức năng quản lý blog*

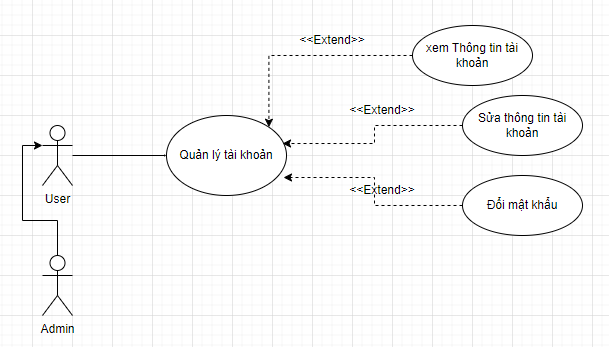
**Xem blog:** Xem các thông tin của blog: tiêu đề, nội dung, hình ảnh, trạng thái.

**Chỉnh sửa blog:** Chỉnh sửa các thông tin của blog như: tiêu đề, nội dung, hình ảnh

**Xóa blog:** Xóa Blog

**Thêm mới blog:** Thêm mới blog với các thông tin như: tiêu đề, nội dung, hình ảnh

### 2.2.8 Biểu đồ phân rã chức năng Quản lý tài khoản



*Figure 2.2.8: Biểu đồ phân rã chức năng Quản lý tài khoản*

**Xem thông tin tài khoản:** Xem các thông tin tài khoản: Họ tên, email, Số điện thoại, Địa chỉ.

**Cập nhật thông tin tài khoản:** Cập nhật các thông tin tài khoản: Họ tên, email, Số điện thoại, Địa chỉ.

**Đổi mật khẩu:** Thay đổi mật khẩu của tài khoản.

## 2.3 Đặc tả chi tiết use case

### 2.3.1 Đăng nhập

*Table 2.1: Đăng nhập*

|  |  |
| --- | --- |
| **Use case name** | Đăng nhập |
| **Usecase ID** | UC-01 |
| **Brief Description** | Hệ thống cho phép người dùng truy cập và đăng nhập tài khoản |
| **Trigger** | Khi người dùng muốn đăng nhập vào hệ thống |
| **Actors** | Tất cả User |
| **Preconditions** | URL trang web được truy cập và giao diện hệ thống được tải |
| **Main Flows** | 1. Người dùng nhấn nút “Đăng nhập”  2. Hệ thống hiển thị giao diện đã đăng ký.  3. Người dùng phải nhập email và mật khẩu.  4. Hệ thống lấy lại email và mật khẩu trên cơ sở dữ liệu.  5. Hệ thống đối chiếu email, mật khẩu đã nhập với email, mật khẩu từ cơ sở dữ liệu.  6. Hệ thống cho phép người dùng truy cập |
| **Alternative Flow** | 1. Mật khẩu và Email không hợp lệ  Hệ thống thông báo tới người dùng Email hoặc mật khẩu không hợp lệ  2. Người dùng nhập email hoặc mật khẩu không hợp lệ  Hệ thống thông báo email hoặc mật khẩu người dùng nhập không hợp lệ  3. Người dùng quên mật khẩu  Hệ thống hiển thị thông báo kết nối người quản trị hệ thống |
| **Exceptions** | Người dùng không có quyền truy cập hệ thống |
| **Post-conditions** | Trang web được mở và hiển thị các chức năng của người dùng |

### 2.3.2 Đăng ký tài khoản

*Table 2.2: Thêm mới tài khoản*

|  |  |
| --- | --- |
| **Use case name** | Đăng ký tài khoản |
| **Usecase ID** | UC-02 |
| **Brief Description** | Đăng ký tài khoản mới trên hệ thống |
| **Trigger** | Khi người dùng muốn đăng ký một tài khoản mới |
| **Actors** | Khách hàng |
| **Preconditions** | URL trang web được truy cập và giao diện hệ thống được tải |
| **Main Flows** | 1. Người dùng chọn đăng ký trên menu Hệ thống chính.  2. Hệ thống cung cấp cho người dùng một biểu mẫu bao gồm tất cả các trường bắt buộc.  3. Người dùng nhập tất cả các trường bắt buộc và gửi thông tin.  4. Hệ thống hiển thị thông báo đăng ký thành công |
| **Alternative Flow** | 1. Người dùng nhập tên sử dụng đã tồn tại  Yêu cầu nhập lại tên người dùng khác |
| **Exceptions** | Người dùng không có quyền truy cập hệ thống |
| **Post-conditions** | Thêm một tài khoản mới |

### 2.3.3 Tìm kiếm sản phẩm

*Table 2.3: Tìm kiếm sản phẩm*

|  |  |
| --- | --- |
| **Use case name** | Tìm kiếm sản phẩm |
| **Usecase ID** | UC-03 |
| **Brief Description** | Hệ thống sẽ tìm kiếm theo điều kiện của người dùng và hiển thị dữ liệu tìm kiếm lên màn hình |
| **Trigger** | Khi khách hàng muốn tìm kiếm một sản phẩm |
| **Actors** | Tất cả người dùng |
| **Preconditions** | Không |
| **Main Flows** | 1. Người dùng chọn ‘tìm sản phẩm’  2. Hệ thống hiển thị trang tìm kiếm  3. Người dùng nhập tiêu chí được yêu cầu  4. Hệ thống tìm kiếm sản phẩm phù hợp với tiêu chí của Người dùng  5. Nếu hệ thống tìm thấy một số sản phẩm phù hợp thì:  Hệ thống hiển thị sản phẩm.  6. Khác  Hệ thống thông báo cho Người dùng rằng không tìm thấy sản phẩm phù hợp |
| **Alternative Flow** | Không |
| **Exceptions** | Không |
| **Post-conditions** | Hiển thị danh sách sản phẩm khớp với điều kiện tìm kiếm |

### 2.3.4 Mua sản phẩm

*Table 2.4: Mua sản phẩm*

|  |  |
| --- | --- |
| **Use case name** | Mua sản phẩm |
| **Usecase ID** | UC-04 |
| **Brief Description** | Hệ thống cho phép người dùng thanh toán những sản phẩm mà họ đã chọn |
| **Trigger** | Khi người dùng bấm nút mua hàng trên trang sản phẩm được chọn |
| **Actors** | Khách hàng |
| **Preconditions** | Người dùng đăng nhập thành công và bấm mua hàng ở sản phẩm được chọn. |
| **Main Flows** | 1. Sản phẩm có thể lựa chọn của người dùng.  2. Hệ thống hiển thị thông tin về sản phẩm.  3. Người dùng chọn chức năng “mua ngay”  4. Hệ thống hiển thị hình thức thanh toán và thông tin người dùng  5. Người dùng lựa chọn hình thức thanh toán và thanh toán.  Hệ thống hiển thị thông báo mua sản phẩm thành công. |
| **Alternative Flow** | không |
| **Exceptions** | Người dùng chưa chọn sản phẩm cần mua.  Hệ thống thông báo chưa thêm sản phẩm vào giỏ hàng |
| **Post-conditions** | Sản phẩm được mua thành công tương ứng với người dùng đã mua nó. |

### 2.3.5 Xem chi tiết sản phẩm

*Table 2.5: xem chi tiết sản phẩm*

|  |  |
| --- | --- |
| **Use case name** | Xem chi tiết sản phẩm |
| **Usecase ID** | UC-05 |
| **Brief Description** | Hệ thống cho phép người dùng xem chi tiết của sản phẩm |
| **Trigger** | Khi người dùng bấm vào sản phẩm họ muốn mua |
| **Actors** | Khách hàng |
| **Preconditions** | None |
| **Main Flows** | 1. Người dùng chọn sản phẩm.  2. Hệ thống hiển thị chi tiết sản phẩm.  3. Người dùng chọn sản phẩm muốn xem để mua.  Hệ thống hiển thị sản phẩm tương ứng. |
| **Alternative Flow** | Không |
| **Exceptions Flow** | Không |
| **Post-conditions** | Không |

### 2.3.6 Thêm mới sản phẩm

*Table 2.6: Thêm mới sản phẩm*

|  |  |
| --- | --- |
| **Use case name** | Thêm mới sản phẩm |
| **Usecase ID** | UC-08 |
| **Brief Description** | Hệ thống cho phép nhân viên thêm mới sản phẩm |
| **Trigger** | Khi nhân viên muốn thêm một sản phẩm mới |
| **Actors** | Administrator |
| **Preconditions** | Tải khoản đăng nhập được phân quyền là admin |
| **Main Flows** | 1. Admin chọn sản phẩm bổ sung  2. Hệ thống hiển thị giao diện form điền thông tin cơ bản về sản phẩm  3. Admin điền đầy đủ thông tin sản phẩm.  4. Hệ thống thông báo sản phẩm thành công. |
| **Alternative Flow** | Không |
| **Exceptions Flow** | Admin điền thông tin sản phẩm còn thiếu.  Hệ thống thông báo cho bạn rằng sản phẩm đã bị lỗi. |
| **Post-conditions** | Thêm một sản phẩm mới |

### 2.3.7 Quản lý đơn hàng

*Table 2.7: Quản lý đơn hàng*

|  |  |
| --- | --- |
| **Use case name** | Quản lý đơn hàng |
| **Usecase ID** | UC-9 |
| **Brief Description** | Hệ thống cho phép admin quản lý các đơn hàng |
| **Trigger** | Không |
| **Actors** | Administrator |
| **Preconditions** | Người dùng đăng nhập với tài khoản admin |
| **Main Flows** | 1. Quản trị viên đăng nhập vào Hệ thống.  2. Đăng nhập thành công.  3. Admin sử dụng chức năng quản lý đơn hàng.  Hệ thống hiển thị danh sách đơn hàng. |
| **Alternative Flow** | Không |
| **Exceptions Flow** | Quản trị viên đăng nhập sai tên người dùng và mật khẩu.  Thông báo cho người dùng rằng thông tin này không chính xác. |
| **Post-conditions** | Xem thông tin của tất cả các đơn hàng |

### 2.3.8 Quản lý danh mục

*Table 2.8: Quản lý danh mục*

|  |  |
| --- | --- |
| **Use case name** | Category management |
| **Usecase ID** | UC-10 |
| **Brief Description** | The system allows admin to manage their category. |
| **Trigger** | When the admin wants to manage category |
| **Actors** | Administrator |
| **Preconditions** | The admin has logged in as a admin. |
| **Main Flows** | 1. Admin logins to System. 2. Login success. 3. Admin uses function manage categories. 4. System shows list of categories. |
| **Alternative Flow** | None |
| **Exceptions Flow** | 1. The admin logins wrong username and password.    1. Notify to user that this information is not correct. |
| **Post-conditions** | View information of all category. |

### 2.3.9 Brand management

*Table 2.9: Brand management*

|  |  |
| --- | --- |
| **Use case name** | Brand management |
| **Usecase ID** | UC-11 |
| **Brief Description** | The system allows administrators to brand manage. |
| **Trigger** | When the admin wants to brand management. |
| **Actors** | Admin |
| **Preconditions** | The admin has logged in as a admin. |
| **Main Flows** | 1. Admin logins to System. 2. Login success. 3. Admin uses function brand management. 4. System shows list of brand management. |
| **Alternative Flow** | None |
| **Exceptions Flow** | 1. The admin logins wrong username and password.    1. Notify to user that this information is not correct. |
| **Post-conditions** | View information of all brands. |

### 2.3.10 Post management

*Table 2.10: Post management*

|  |  |
| --- | --- |
| **Use case name** | Post management |
| **Usecase ID** | UC-12 |
| **Brief Description** | The system allows administrators to blog manage. |
| **Trigger** | When the admin wants to blog manage. |
| **Actors** | Admin |
| **Preconditions** | The admin has logged in as a admin. |
| **Main Flows** | 1. Admin logins to System. 2. Login success. 3. Admin uses function blog management. 4. System shows list of blog management. |
| **Alternative Flow** | None |
| **Exceptions Flow** | 1. The admin logins wrong username and password.    1. Notify to user that this information is not correct. |
| **Post-conditions** | View information of all blogs. |

### 2.3.11 Promotion management

*Table 2.11: Promotion management*

|  |  |
| --- | --- |
| **Use case name** | Promotion management |
| **Usecase ID** | UC-13 |
| **Brief Description** | The system allows admin to manage their promotion. |
| **Trigger** | When the admin wants to manage promotion |
| **Actors** | Administrator |
| **Preconditions** | The admin has logged in as a admin. |
| **Main Flows** | 1. Admin logins to System. 2. Login success. 3. Admin uses function manage promotions. 4. System shows list of promotions. |
| **Alternative Flow** | None |
| **Exceptions Flow** | 1. The admin logins wrong username and password.    1. Notify to user that this information is not correct. |
| **Post-conditions** | View information of all promotion. |

## 2.4 Sequence diagram

### 2.4.1 Login function

Diagram

Description automatically generated

*Figure 2.4.1: Sequence diagram login function*

### 2.4.2 Register function

Diagram

Description automatically generated

Figure 2.4.2: Sequence diagram register function

### 2.4.3 Search function

Diagram

Description automatically generated

Figure 2.4.3: Sequence diagram search function

### 2.4.4 View product function

Diagram

Description automatically generated

Figure 2.4.4: Sequence diagram view product function

### 2.4.5 Comment function

Diagram

Description automatically generated

Figure 2.4.5: Sequence diagram comment function

### 2.4.6 Buy product function

Diagram

Description automatically generated

Figure 2.4.6: Sequence diagram register function

### 2.4.7 Category management function

A picture containing calendar

Description automatically generated

Figure 2.4.7: Sequence diagram category management function

### 2.4.8 Brand management function

A picture containing diagram

Description automatically generated

Figure 2.4.8: Sequence diagram brand management function

### 2.4.9 Blog management function

Diagram

Description automatically generated

Figure 2.4.9: Sequence diagram register function

### 2.4.10 Promotion management function

Diagram

Description automatically generated

Figure 2.4.10: Sequence diagram promotion management function

### 2.4.11 Order management function

Diagram

Description automatically generated

Figure 2.4.11: Sequence diagram order management function

### 2.4.12 Product management function

Diagram

Description automatically generated

Figure 2.4.12: Sequence diagram product management function

# CHAPTER 3: TECHNOLOGY USED

By examining and analyzing in detail the functions needed in your system as described in chapter 2. Chapter 3 is the selection of technologies and platforms to use for your system to solve problems on.

## 3.1 Introduction to Spring boot

### 3.1.1 Spring boot overview

Spring Boot helps you to create stand-alone, production-grade Spring-based Applications that you can run. We take an opinionated view of the Spring platform and third-party libraries, so that you can get started with minimum fuss. Most Spring Boot applications need very little Spring configuration.

You can use Spring Boot to create Java applications that can be started by using java -jar or more traditional war deployments. We also provide a command line tool that runs “spring scripts”.

Our primary goals are:

* Provide a radically faster and widely accessible getting-started experience for all Spring development.
* Be opinionated out of the box but get out of the way quickly as requirements start to diverge from the defaults.
* Provide a range of non-functional features that are common to large classes of projects (such as embedded servers, security, metrics, health checks, and externalized configuration).
* Absolutely no code generation and no requirement for XML configuration.

Logo, company name

Description automatically generated

*Figure 3.1: Logo of Spring boot*

### 3.1.2 Advantages of Spring boot

   Simple configuration

    - The spring boot embeds the servlet container, which reduces the environmental requirements. The machine has a java operating environment, and the project can be packaged into a jar package, and the java command is used to java -jar \*\*\*\*.jar to execute.

    - Quickly integrate third-party frameworks without configuration files

    - Solve the shortcomings of Spring

    - Less code, fewer configuration files, no need to worry about third-party frameworks, simplified projects, and greater cost savings for the development and maintenance of the entire team.

    - It is very easy to develop Spring-based applications using Java or Groovy.

    - It reduces a lot of development time and increases productivity.

    - It avoids writing a lot of boilerplate code, comments and XML Configuration.

    - Spring Boot applications and their Spring ecosystem (such as Spring JDBC, Spring ORM, Spring Data, Spring Security, etc.) The integration is very easy.

         It follows the "self-use default configuration" method to reduce development workload.

         It provides embedded HTTP server, such as Tomcat, Jetty, etc., to develop and test web applications very easily.

         It provides CLI (Command Line Interface) tools from the command prompt to develop and test Spring Boot (Java or Groovy) applications very easily and quickly.

         It provides many plugins to develop and test Spring Boot applications very easy to use build tools such as Maven and Gradle.

         It provides many plug-ins to make working with embedded and in-memory databases very easy.

.

### 3.1.3 Disadvantages of Spring boot

The biggest challenge many developers face when using Spring Boot is the lack of control. The opinionated style installs many additional dependencies (that often go unused) which increases the deployment file size.

The Spring Boot artifact may be run directly in Docker containers. This is useful to get when you need to quickly create microservices. Yet, some developers argue that since Spring Boot was designed to be lightweight and agile, it should therefore not be used for monolithic applications.

Though Spring Boot comes with some basic tools for logs and your app health monitoring, these aren’t sufficient. Tools like Retrace help teams to monitor Java apps with ease. This tool helps to detect slow SQL queries, provides performance and CPU usage reports and shows the most common errors by interpreting the logs.

On top of that, it can be quite challenging to update your legacy Spring code. You can overcome this problem by using tools such as the Spring Boot CLI (Command Line Interface) that will help you convert your legacy code.

Some other disadvantages are:

* If you have never worked with Spring before and want to learn about proxies, dependency injection, and AOP programming, it is not recommended to start with Spring Boot because it doesn’t cover most of these details.
* You really have to understand a lot of the underlying Spring systems (and a bit of Spring history too), along with some advanced topics in order to modify and troubleshoot it.
* Spring Boot works well with microservices. The Spring Boot artifacts can be deployed directly into Docker containers. However, some developers don’t recommend the framework for building large and monolithic apps.
* If you are not familiar with other projects of the Spring ecosystem like Spring Security, Spring AMQP, Spring Integration, etc), using them with Spring Boot will make you miss many concepts that you would grasp if you had started using them independently.

## 3.2 Introduction to HTML



*Figure 3.2: HTML*

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the » W3C, the organisation charged with designing and maintaining the language.

The definition of HTML is HyperText Markup Language.

HyperText is the method by which you move around on the web - by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear - i.e. you can go to any place on the Internet whenever you want by clicking on links - there is no set order to do things in.

Markup is what HTML tags do to the text inside them. They mark it as a certain type of text (italicised text, for example).

HTML is a Language, as it has code-words and syntax like any other language.

HTML is the language for describing the structure of Web pages. HTML gives authors the means to:

* Publish online documents with headings, text, tables, lists, photos, etc.
* Retrieve online information via hypertext links, at the click of a button.
* Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
* Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

With HTML, authors describe the structure of pages using markup. The elements of the language label pieces of content such as “paragraph,” “list,” “table,” and so on.

## 3.3 CSS



*Figure 3.3: CSS*

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation.

## 3.4 JavaScript



*Figure 3.4: JavaScript*

JavaScript was initially created to “make web pages alive”.

The programs in this language are called scripts. They can be written right in a web page’s HTML and run automatically as the page loads. Scripts are provided and executed as plain text. They don’t need special preparation or compilation to run. In this aspect, JavaScript is very different from another language called Java.

Today, JavaScript can execute not only in the browser, but also on the server, or actually on any device that has a special program called the JavaScript engine.

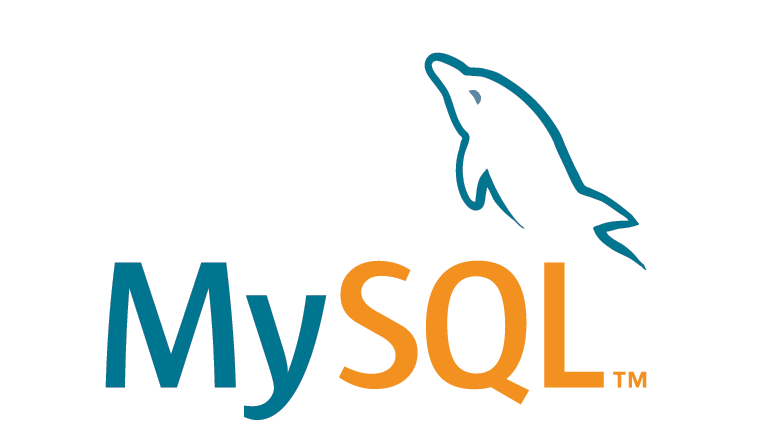
The browser has an embedded engine sometimes called a “JavaScript virtual machine”.

Different engines have different “codenames”. For example:

* V8 – in Chrome and Opera.
* SpiderMonkey – in Firefox.
* …There are other codenames like “Trident” and “Chakra” for different versions of IE, “ChakraCore” for Microsoft Edge, “Nitro” and “SquirrelFish” for Safari, etc.

The terms above are good to remember because they are used in developer articles on the internet. We’ll use them too. For instance, if “a feature X is supported by V8”, then it probably works in Chrome and Opera.

## 3.5 MySQL



*Figure 3.5: MySQL*

MySQL is a leading open source database management system. It is a multi-user, multithreaded database management system. MySQL is especially popular on the web. It is one of the parts of the very popular LAMP platform. Linux, Apache, MySQL and PHP. MySQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows or Mac. Wikipedia, YouTube, Facebook use MySQL. These sites manage millions of queries each day. MySQL comes in two versions: MySQL server system and MySQL embedded system. The MySQL server software and the client libraries are dual-licensed: GPL version 2 and proprietary license.

The development of MySQL began in 1994 by a Swedish company MySQL AB. Sun Microsystems acquired MySQL AB in 2008. Sun was bought by Oracle in 2010.

MySQL, PostgreSQL, Firebird, SQLite, Derby, and HSQLDB are the most well known open source database systems.

MySQL is developed in C/C++. Except for C/C++, APIs exist for PHP, Python, Java, C#, Eiffel, Ruby, Tcl or Perl.

## 3.6 XAMPP



*Figure 3.6: XAMPP*

XAMPP is the title used for a compilation of free software. The name is an acronym, with each letter representing one of the five key components. The software packet contains the web server Apache, the relational database management system MySQL (or MariaDB), and the scripting languages Perl and PHP. The initial X stands for the operating systems that it works with: Linux, Windows, and Mac OS X.

Apache: ihe open source web server Apache is the most widely used server worldwide for delivery of web content. The server application is made available as a free software by the Apache Software Foundation.

MySQL/MariaDB: in MySQL, XAMPP contains one of the most popular relational database management systems in the world. In combination with the web server Apache and the scripting language PHP, MySQL offers data storage for web services. Current XAMPP versions have replaced MySQL with MariaDB (a community-developed fork of the MySQL project, made by the original developers).

PHP: the server-side programming language PHP enables users to create dynamic websites or applications. PHP can be installed on all platforms and supports a number of diverse database systems.

Perl: the scripting language Perl is used in system administration, web development, and network programming. Like PHP, Perl also enables users to program dynamic web applications.

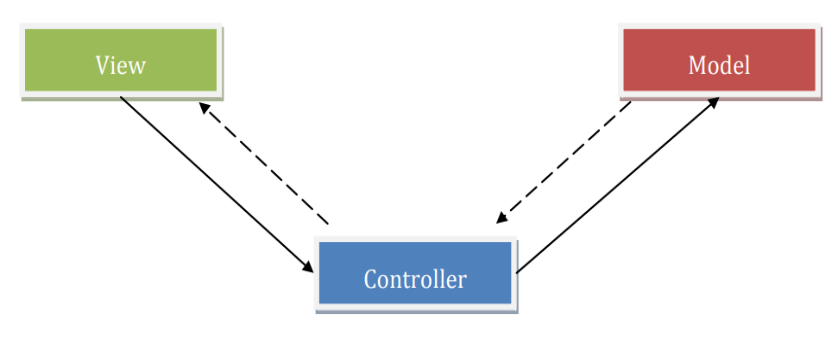
Alongside these core components, this free-to-use Apache distribution contains some other useful tools, which vary depending on your operating system. These tools include the mail server Mercury, the database administration tool phpMyAdmin, the web analytics software solutions Webalizer, OpenSSL, and Apache Tomcat, and the FTP servers FileZilla or ProFTPd.

# CHAPTER 4: APPLICATION DEVELOPMENT AND IMPLEMENTATION

Combined with the technology selection in chapter 3, the choice of software architecture for my system will be outlined in chapter 4. Chapter 4 will describe the architecture specific to my system.

## 4.1 Architectural design

Model-View-Control (MVC) is a software architecture, currently considered a design pattern in software engineering. The MVC model separates the data processing part from the interface, allowing the development, testing and maintenance of components independently.



*Figure 4.1: MVC pattern illustration*

Whereby:

* Model showing data structures Classes of the Model component often perform tasks such as querying, adding, deleting, and updating data. When the data in the Model changes, the View component will be updated.
* View is a component representing data in the Model into interfaces that interact with users. A model can have multiple views depending on different purposes.
* Controller acts as an intermediary between Model and View. The user information from the View is sent to the Controller for processing, then the Controller interacts with the Model to retrieve the requested data, and the Controller finally returns this data to the View.

The MVC pattern is often used in web applications, because the View component (HTML / XHTML code) is generated from web programming languages. Controller component will receive GET / POST data, process these data, then switch to Processing Model. The model will return data to the Controller, then the Controller generates HTML / XHTML code to be shown in the View.

## Detailed design

### 4.2.1 Database design

Diagram, schematic

Description automatically generated

*Figure 4.2: Database*

#### 4.2.1.1 Table “User”

*Table 4.1: Table “User”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | id | BIGINT | Used to contain a unique identifier for each data and as a primary key |
| 2 | email | VARCHAR(255) | Contains user email |
| 3 | password | VARCHAR(255) | Contains passwords when logged in by users |
| 4 | full\_name | VARCHAR(255) | Contains the first name and last name of the user |
| 5 | phone | VARCHAR(255) | Contains telephone user information |
| 6 | created\_at | TIMESTAMP | Account creation time |
| 7 | role | JSON | Permission array  Default: ["USER"] |
| 8 | address | VARCHAR(255) | Contains address of user |
| 9 | status | BOOLEAN | Account status: true - activate, false - disable |

#### 4.2.1.2 Table “Blog”

*Table 4.2: Table “Blog”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | id | BIGINT | Used to contain a unique identifier for each data and as a primary key |
| 2 | content | TEXT | Contains content |
| 3 | description | TEXT | Contains description |
| 4 | slug | VARCHAR(600) | Contains path |
| 5 | title | VARCHAR(300) | Contains title |
| 6 | thumbnail | VARCHAR(255) | Contains thumbnail |
| 7 | created\_at | TIMESTAMP | Contains post creation date |
| 8 | created\_by | BIGINT | Contains gender information |
| 9 | modified\_at | TIMESTAMP | Last modified time |
| 10 | modified\_by | BIGINT | Last modified person ID |
| 11 | published\_at | TIMESTAMP | Public time |
| 12 | status | INT | Post status: 0 - draft, 1 - public |

#### 4.2.1.3 Table “Images”

*Table 4.3: Table “Images”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | Id | VARCHAR(255) | Used to contain a unique identifier for each data and as a primary key |
| 2 | link | VARCHAR(255) | Image link |
| 3 | name | VARCHAR(255) | Contains file name of image |
| 4 | size | BIGINT | Image Size (Kb) |
| 5 | type | VARCHAR(255) | Picture type |
| 6 | uploaded\_at | TIMESTAMP | Upload time |
| 7 | uploaded\_by | BIGINT | Upload user ID |

#### 4.2.1.4 Table “Brand”

*Table 4.4: Table “Brand”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | id | INT | Used to contain a unique identifier for each data and as a primary key |
| 2 | name | VARCHAR(255) | Contains name of brand |
| 3 | thumbnail | VARCHAR(255) | Logo image link |

#### 4.2.1.5 Table “Category”

*Table 4.5: Table “Category”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | id | INT | Used to contain a unique identifier for each data and as a primary key |
| 2 | name | VARCHAR(255) | Contains name of category |

#### 4.2.1.6 Table “Product”

*Table 4.6: Table “Product”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | id | VARCHAR(255) | Used to contain a unique identifier for each data and as a primary key |
| 2 | brand\_id | BIGINT | Contains the binding key value of the “Brand” table |
| 3 | image | JSON | Contains a representative image of the product |
| 4 | Image\_feedback | JSON | Contains a feedback image of the product |
| 5 | name | VARCHAR(255) | Contains name of product |
| 6 | slug | VARCHAR(255) | Support seo |
| 7 | price | BIGINT | Contains price of product |
| 8 | Sale\_price | BIGINT | Contains discount of product |
| 9 | description | VARCHAR(255) | Contains description of product |
| 10 | status | INT | Open sale status |
| 11 | view | INT | Contains the number of views of the product |
| 12 | created\_at | TIMESTAMP | Contains product creation date |
| 13 | modified\_at | TIMESTAMP | Last modified time |

#### 4.2.1.7 Table “Product\_Category”

*Table 4.7: Table “Product\_Category”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | product\_id | VARCHAR(255) | Product ID |
| 2 | category\_id | BIGINT | Category ID |

#### 4.2.1.8 Table “Product\_size”

*Table 4.8: Table “Product\_size”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | product\_id | VARCHAR(255) | Product ID |
| 2 | size | INT | Contains size of product |
| 3 | quantity | INT | Contains quantity of product |

#### 4.2.1.9 Table “Promotion”

*Table 4.9: Table “Promotion”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | id | BIGINT | Used to contain a unique identifier for each data and as a primary key |
| 2 | coupon\_code | VARCHAR(255) | Coupon code |
| 3 | created\_at | TIMESTAMP | Creation time |
|  | discount\_type | INT | Discount type:  1 - percent (%)  2 - at a fixed rate |
|  | discount\_value | BIGINT | Discount value |
|  | expired\_at | TIMESTAMP | Expiration time |
|  | is\_active | boolean | Activation status |
|  | is\_public | boolean | Public status |
|  | maximum\_discount\_value | BIGINT | Maximum discount value |
|  | name | VARCHAR(300) | Name of promotion |

#### 4.2.1.10 Table “Order”

*Table 4.10: Table “Order”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | id | BIGINT | Used to contain a unique identifier for each data and as a primary key |
| 2 | created\_at | TIMESTAMP | Creation time |
| 3 | modified\_at | TIMESTAMP | Last modified time |
| 4 | note | VARCHAR(255) | Notes |
| 5 | product\_price | BIGINT | Product price |
| 6 | promotion | JSON | Promotions used:  {  "coupon\_code": varchar "discount\_type": int  "discount\_value": int "maximum\_discount\_value": int  } |
| 7 | receiver\_address | VARCHAR(255) | Recipient address |
| 8 | receiver\_name | VARCHAR(255) | Recipient name |
| 9 | receiver\_phone | VARCHAR(255) | Recipient phone number |
| 10 | size | INT | Product size |
| 11 | status | INT | Order status:  1 - Waiting for delivery  2 - on delivery  3 - delivered  4 - returns  5 - cancellation of application |
| 12 | total\_price | BIGINT | Total order value |
| 13 | buyer | BIGINT | Buyer ID |
| 14 | created\_by | BIGINT | Order creator |
| 15 | modified\_by | BIGINT | Last modified person ID |
| 16 | product\_id | VARCHAR(255) | Product ID |

#### 4.2.1.11 Table “Comment”

*Table 4.11: Table “Promotion”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | id | BIGINT | Used to contain a unique identifier for each data and as a primary key |
| 2 | content | TEXT | Contains content of comment |
| 3 | user\_id | INT | Contains the binding key value of the “User” table |
| 4 | product\_id | VARCHAR(255) | Contains the binding key value of the “Product” table |
| 5 | post\_id | BIGINT | Contains the binding key value of the “Post” table |
| 6 | created\_at | TIMESTAMP | Creation time |

#### 4.2.1.12 Table “Statistic”

*Table 4.12: Table “Statistic”*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Datatypes | Describe |
| 1 | id | BIGINT | Used to contain a unique identifier for each data and as a primary key |
| 2 | sales | BIGINT | Contains sales |
| 3 | profit | BIGINT | Contains profit |
| 4 | quantity | INT | Contains quantity |
| 5 | created\_at | TIMESTAMP | Creation time |
| 6 | order\_id | BIGINT | Contains the binding key value of the “Order” table |

## Application construction

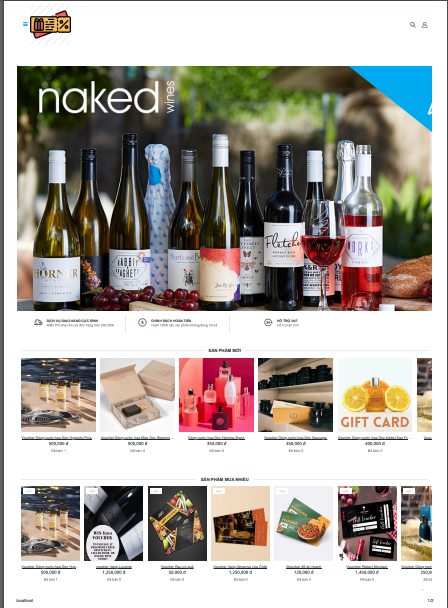
### 4.3.1 Libraries and tools used

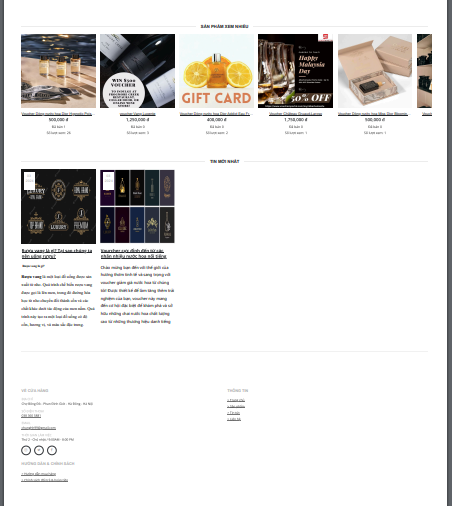
|  |  |  |
| --- | --- | --- |
| Purpose | Tool | URL address |
| IDE programming | Intellij idea | https://www.jetbrains.com/idea/ |
| Database design | MySQL Workbench | https://www.mysql.com/products/workbench/ |
| JAVA development environment | JDK | https://www.oracle.com/java/technologies/javase-downloads.html |

### Illustrate the main functions

#### 4.3.2.1 Trang chủ

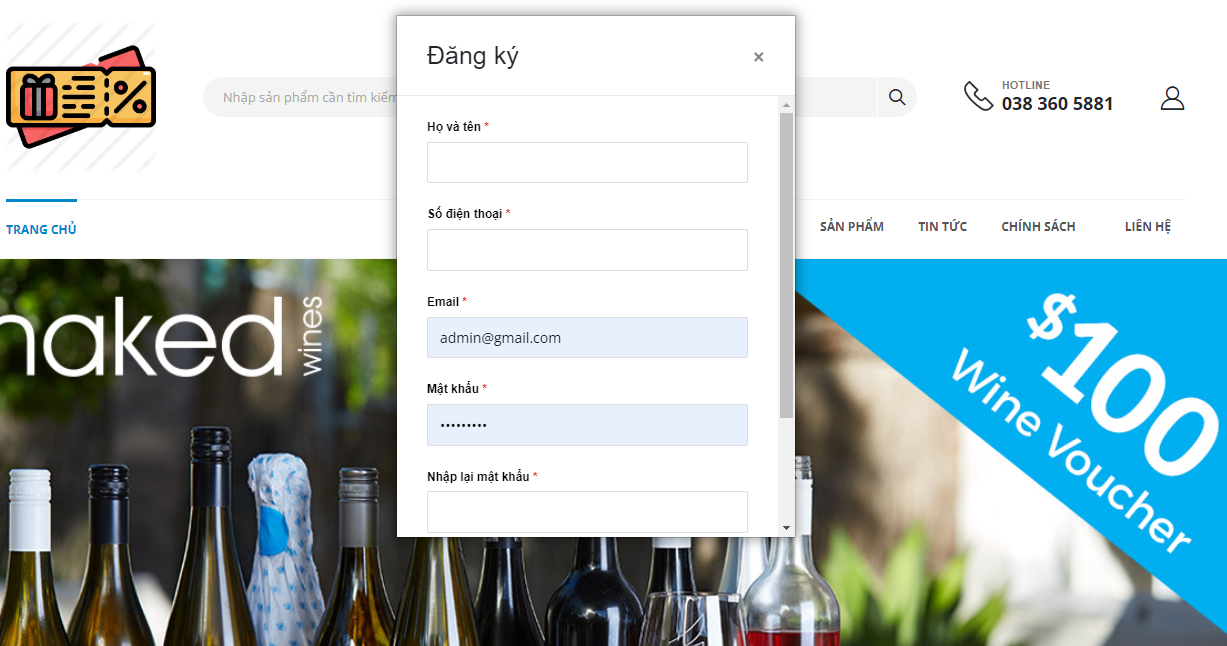
Trang chủ là nơi đầu tiên người dùng tiếp cận trang web. Vì vậy trang chủ được làm để người dùng qua hệ thống của website có thể hiểu được những gì nổi bật để người dùng cảm thấy thích thú hơn trong quá trình sử dụng.





*Figure 4.3: Trang chủ*

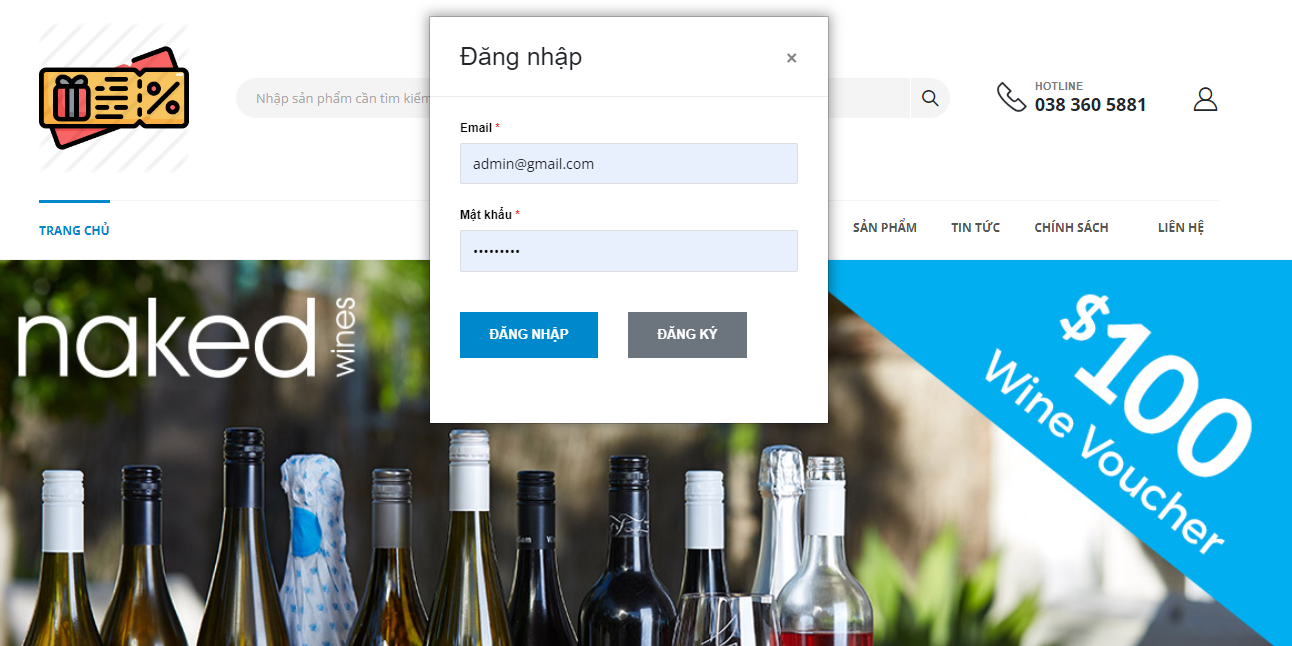
#### 4.3.2.2 Đăng ký tài khoản

Tại trang đăng ký thành viên, khách hàng có thể đăng ký tài khoản thành viên của website bằng cách nhập chính xác và đầy đủ tất cả các yêu cầu nêu trong mẫu như: Email, mật khẩu, tên, địa chỉ,…

*Figure 4.4: Account registration page*

#### 4.3.2.3 Đăng nhập

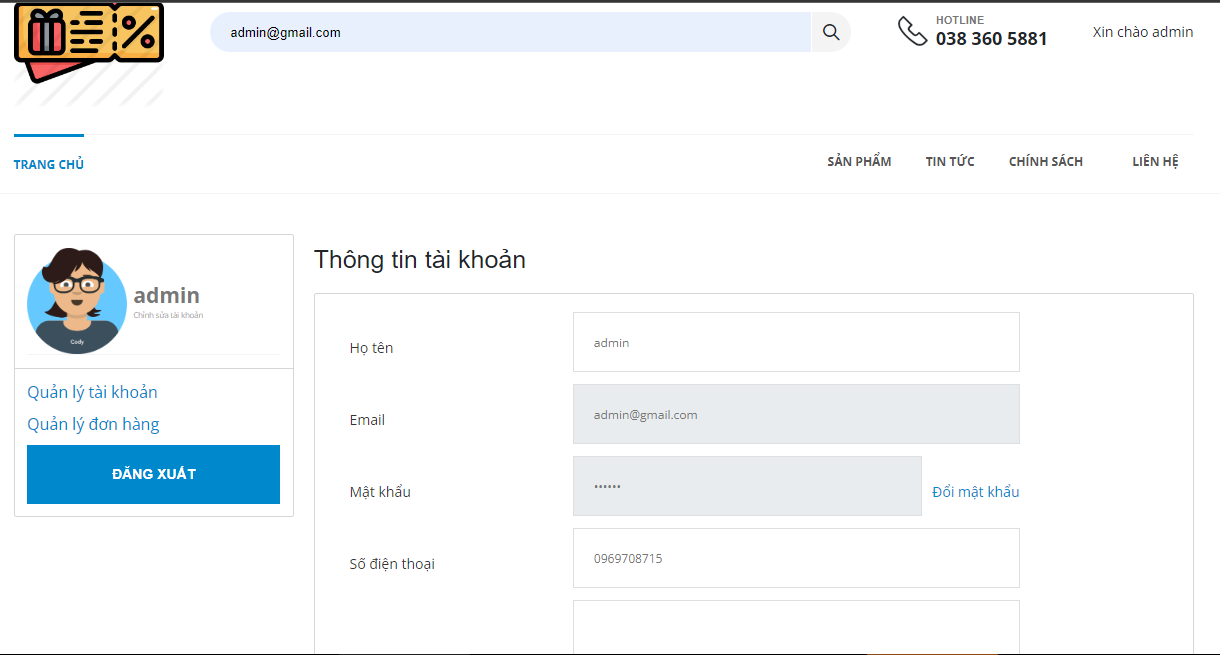
Sau khi nhấp vào liên kết Đăng nhập trên thanh menu, khách hàng sẽ được chuyển đến trang đăng nhập dành cho khách hàng đã đăng ký tài khoản trước đó.



*Figure 4.5: Account login page*

#### 4.3.2.4 Thông tin người dùng

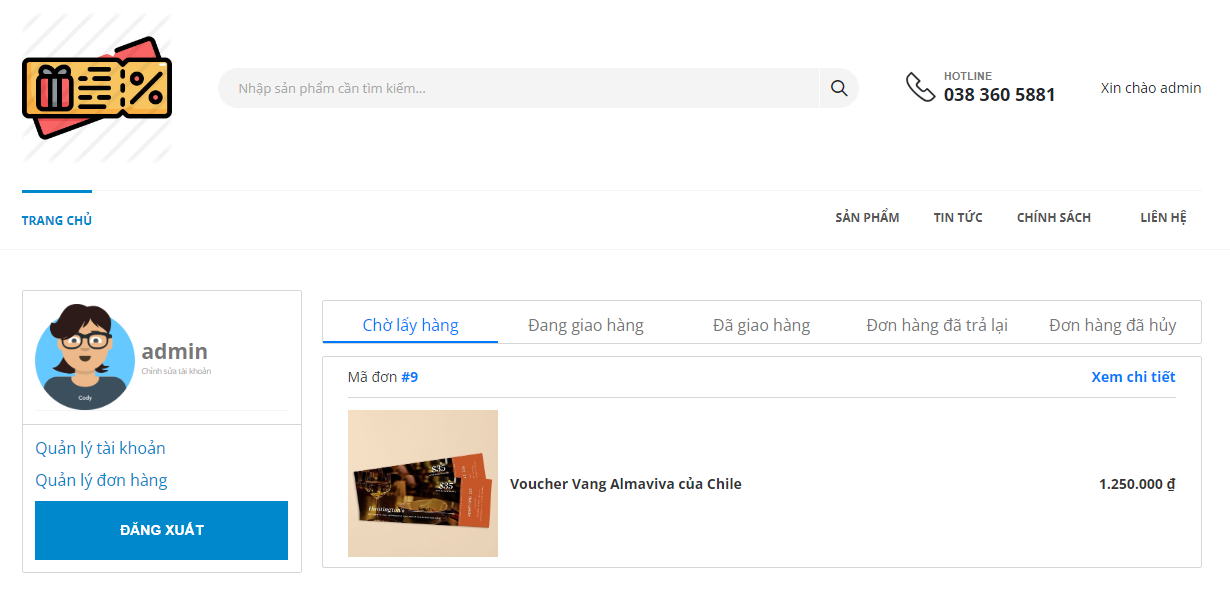
Trên trang này người dùng có thể cập nhật thông tin của mình cho chính xác, thay đổi mật khẩu để bảo mật tài khoản.



*Figure 4.6: The page displays user information*

#### 4.3.2.5 Lịch sử mua hàng

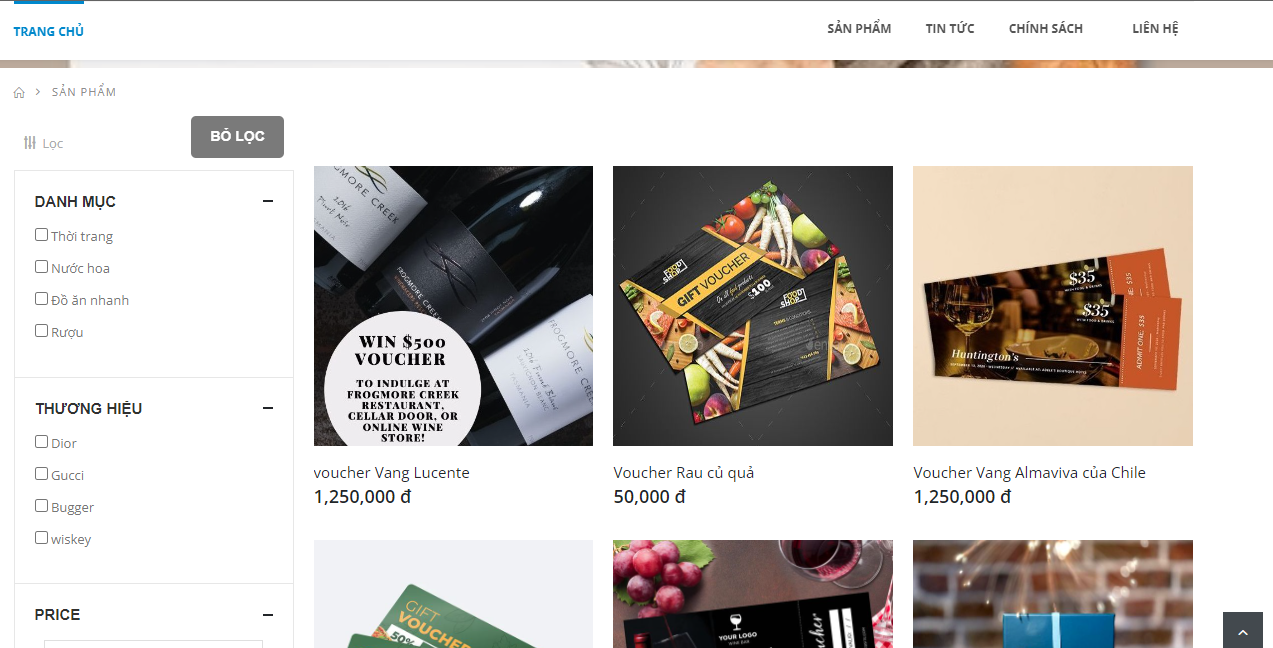
Tại trang này người dùng có thể kiểm tra trạng thái đơn hàng, lịch sử giao dịch trên hệ thống

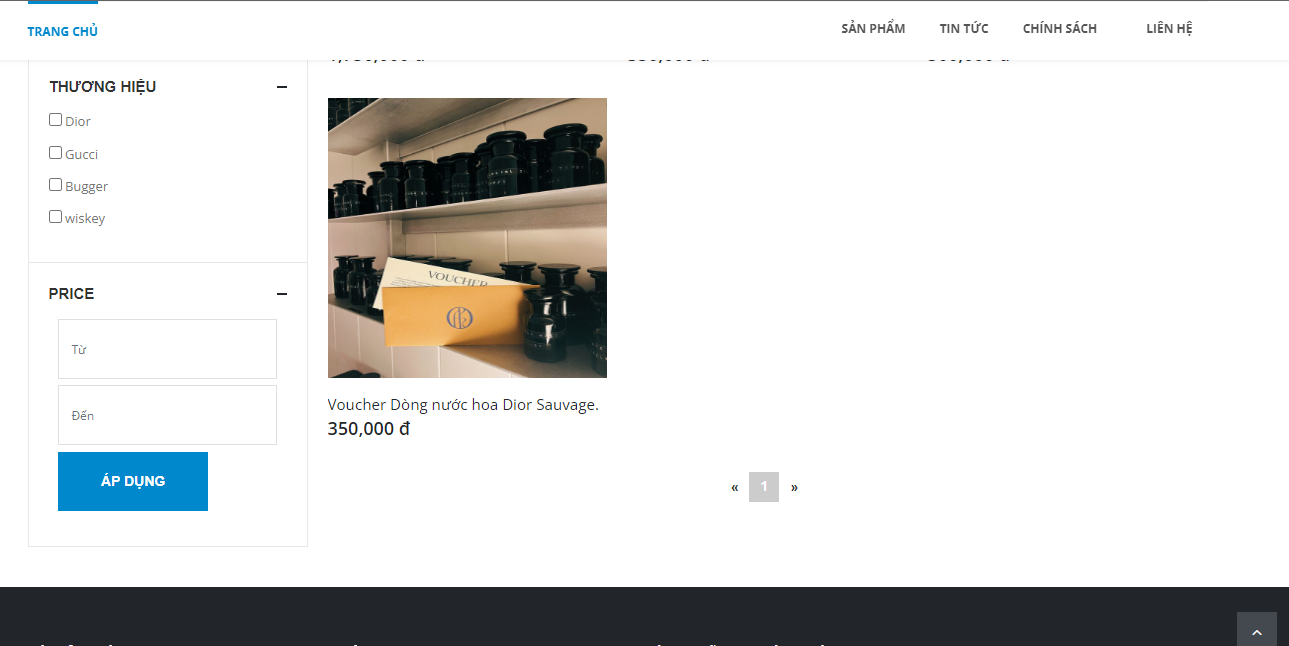


*Figure 4.7: The page displays order history*

#### 4.3.2.6 Danh sách voucher

Trang này sẽ hiển thị toàn bộ sản phẩm có sẵn trên hệ thống, khách hàng có thể lựa chọn sản phẩm mà mình quan tâm.

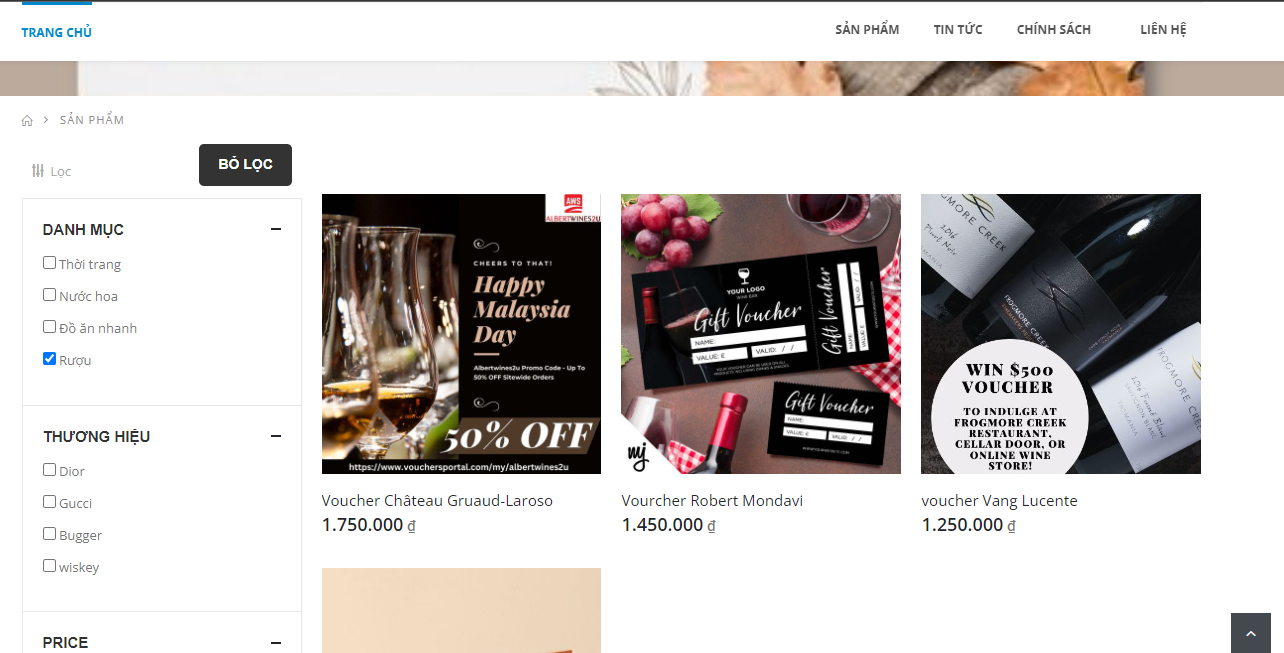




*Figure 4.8: The page displays all product*

#### 4.3.2.7 Tìm kiếm voucher

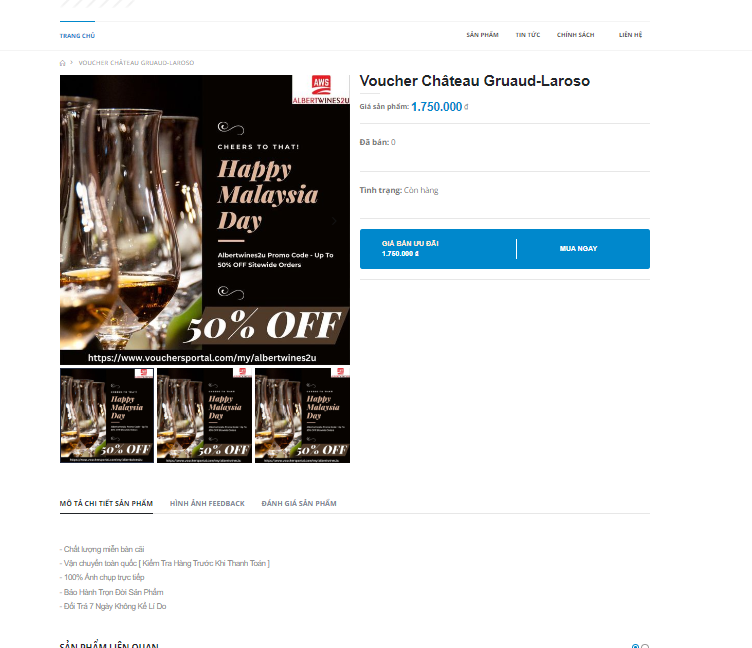
Khi người dùng bị làm phiền bởi quá nhiều sản phẩm hoặc bị làm phiền vì sản phẩm được sắp xếp không theo thứ tự thì người dùng khó tìm được sản phẩm mình yêu thích. Họ có thể tìm kiếm bằng cách gõ từ khóa vào ô “Tìm kiếm” và website sẽ tìm kiếm những sản phẩm liên quan đến từ khóa đó.



*Figure 4.9: Product search page*

#### 4.3.2.8 Chi tiết voucher

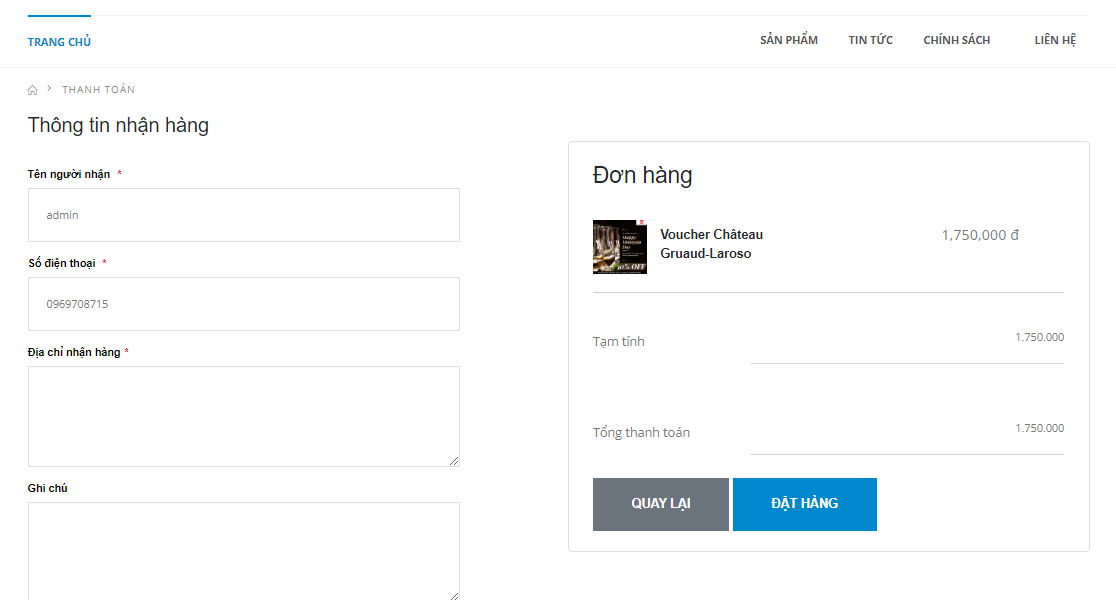
Trang này sẽ hiển thị thông tin chi tiết về sản phẩm để khách hàng biết sản phẩm này có đáp ứng được mục tiêu của mình hay không để có thể mua sản phẩm. Sau khi hài lòng với sản phẩm quý khách vui lòng nhấn nút “Thêm vào giỏ hàng” để tiến hành thanh toán



*Figure 4.10: The page displays the product details*

#### 4.3.2.9 Mua voucher

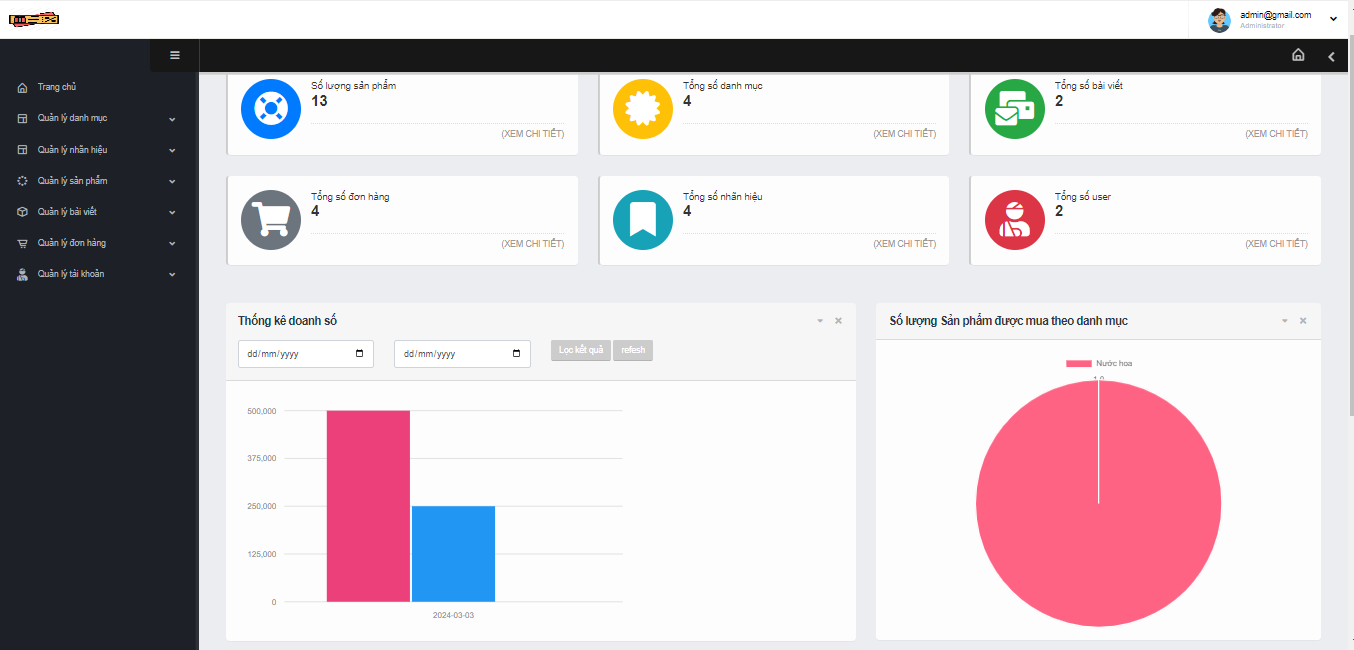
Once they have chosen the product that are right for them, they must confirm that they already own the product by clicking the "Buy product" button and choose payment methods.



*Figure 4.11: Payment*

#### 4.3.1.10 Admin Dashboard

Dashboard giúp quản trị viên biết được số liệu thống kê tổng hợp, những thay đổi mới nhất và đặc biệt là giúp thao tác dễ dàng hơn.



*Figure 4.12: Admin Dashboard*

## Testing

This test is performed on the basis of deploying the system to a server and then accessing the equipment table and testing each part of the system. Then, sum it up with the following result:

### 4.4.1 About the interface

The system interface is not changed from the design when used on computers with many different screen resolutions.

### 4.4.2 About the function

The functions operate as designed, do not generate errors during use. Requests from the user are handled correctly.

### 4.4.3 About performance

The system has fast response speed, in this section has not been tested with a large number of accesses.

## Deployment

### 4.5.1 Install on website server

* Requires server configuration

The minimum configuration for deploying a web server is as follows:

|  |  |  |
| --- | --- | --- |
| Ingredient | Minimum requirements | Note |
| CPU | 1.5 GHz |  |
| RAM | 1.0 GB |  |
| Hard drive is empty | 10 GB |  |

The recommended configuration for deploying a web server is as follows:

|  |  |  |
| --- | --- | --- |
| Ingredient | Minimum requirements | Note |
| CPU | 3 GHz |  |
| RAM | 2 GB |  |
| Hard drive is empty | 50 GB |  |

# CHAPTER 5: CONCLUSION AND DEVELOPMENT DIRECTION

## 5.1 Result

Within the framework of the time to implement the project with the effort of myself and the guidance of the teacher, I have achieved the following results:

* For customers:
  + - Customer can search for products, register to buy products, make suggestions and view news.
    - Customers review their registered information, review information on orders, and update the information that has been registered for purchase.
* For administrator:
  + - Admin can: add, edit, delete, update and statistics products.
    - Manage customers and update customer information register and manage customer lists.
    - View reports on sales situation.
    - In terms of processing speed (search, lookup, statistics...) fast, the results search results are clearly listed, convenient for customers to use.
    - The system uses a Web interface, a common interface that commonly used network applications today. The interface design is simple, easy to understand and clear, so it is suitable for users.

## 5.2 Restrictions exist

In the process of surveying and selecting solutions and limited time, the study also has the following limitations:

* Not mentioning the issue of security and data safety.
* Some accounting operations such as: sales invoice management ... not yet rigid.

## 5.3 Development

To overcome the above limitations, I will try to develop the topic well in the future and add the following functions:

* Manage configuration, landing-page, more interface to fit the promotions.
* Create customer care function, online support to add credibility to the website.
* Secure product as well as websites to prevent hackers from attacking.
* Apply many payment methods such as payment by bank card.
* The program will support more programming languages.
* Perform non-installed functions such as: send automatic emails to
* customers, completing and supporting the credit card payment process

**References**

|  |  |
| --- | --- |
| [1] | Aziz Ali " HTML, CSS, Bootstrap, Javascript and jQuery," [Online]. Available: https://www.quora.com/What-is-HTML-CSS-and-JavaScript |
| [2] | JAVA Programming Language (Spring Framework), [Online]. Available: https://spring.io/. |