BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ**

**ĐỀ TÀI:**

**WEBSITE QUẢN LÝ BÁN HÀNG THỜI TRANG**

**GVHD: ThS.GV Bùi Chí Thành**

**Sinh viên thực hiện: Hồ Thị Thanh Trang**

**MSSV: 63133125**

**Lớp: 63.TTQL-1**

Khánh Hòa - 2023

MỤC LỤC

[CHƯƠNG I: TỔNG QUAN VẤN ĐỀ. 5](#_Toc168294387)

[1.1 Giới thiệu đề tài: 5](#_Toc168294388)

[1.1.1 Khảo sát hiện trạng: 5](#_Toc168294389)

[1.1.2 Tính cấp thiết của đề tài 5](#_Toc168294390)

[1.1.3 Kết Luận: 5](#_Toc168294391)

[1.2 Mục đích nghiên cứu: 6](#_Toc168294392)

[1.3 Mục tiêu đề tài: 6](#_Toc168294393)

[2 CHƯƠNG II: TÌM HIỂU VỀ CƠ SỞ LÝ THUYẾT 7](#_Toc168294394)

[2.1 Tổng quan về ASP.NET: 7](#_Toc168294395)

[2.1.1 Giới thiệu về ASP.NET: 7](#_Toc168294396)

[2.1.2 Tìm hiểu về mô hình lập trình web MVC của ASP.NET: 7](#_Toc168294397)

[2.2 Giới thiệu mô hình lập trình Web bằng ASP.NET MVC5 9](#_Toc168294398)

[2.3 Hoạt động của các thành phần chính trong ASP.NET MVC: 11](#_Toc168294399)

[2.4 Tại sao phải sử dụng ASP.NET? 12](#_Toc168294400)

[2.5 Ưu điểm và khuyết điểm của ASP.NET: 12](#_Toc168294401)

[2.6 Sự khác biệt mô hình lập trình Web ASP.NET MVC và ASP.NET Webform: 14](#_Toc168294402)

[2.7 Ajax: 15](#_Toc168294403)

[2.7.1 Ajax là gì nhỉ? 15](#_Toc168294404)

[2.7.2 Ajax hoạt động thế nào? 16](#_Toc168294405)

[2.7.3 Nhược điểm của AJAX? 17](#_Toc168294406)

[2.8 HTML: 18](#_Toc168294407)

[2.8.1 HTML là gì? 18](#_Toc168294408)

[2.8.2 HTML được xử lý ra sao? 19](#_Toc168294409)

[2.8.3 Dùng chương trình gì để tạo tập tin HTML? 19](#_Toc168294410)

[2.8.4 HTML đóng vai trò gì trong website? 20](#_Toc168294411)

[2.9 CSS: 20](#_Toc168294412)

[2.10 Bootstrap: 20](#_Toc168294413)

[2.10.1 Bootstrap là gì? 20](#_Toc168294414)

[2.10.2 Vì sao phải dùng nó? 20](#_Toc168294415)

[2.11 Jquery: 21](#_Toc168294416)

[2.11.1 Jquery là gì? 21](#_Toc168294417)

[2.11.2 Jquery giúp ích gì cho người lập trình web? 22](#_Toc168294418)

[2.11.3 Tại sao phải dùng Jquery? 22](#_Toc168294419)

[2.12 HTML5: 23](#_Toc168294420)

[2.12.1 HTML5 là gì? 23](#_Toc168294421)

[2.12.2 HTML5 là vấn đề quan trọng của chúng ta? 23](#_Toc168294422)

[2.13 CSS3: 24](#_Toc168294423)

[3 CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 26](#_Toc168294424)

[3.1 Phân tích thiết kế các chức năng ứng dụng: 26](#_Toc168294425)

[3.1.1 Đặc tả chức năng ứng dụng: 26](#_Toc168294426)

[3.1.2 Sơ đồ luồng dữ liệu: 33](#_Toc168294427)

[3.2 Thiết kế dữ liệu: 38](#_Toc168294428)

[3.2.1 Mô hình vật lý: 38](#_Toc168294429)

[3.2.2 Mô tả chi tiết các bảng: 39](#_Toc168294430)

[CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM 41](#_Toc168294431)

[3.3 Giao diện User: 41](#_Toc168294432)

[3.3.1 Giao diện trang chủ: 41](#_Toc168294433)

[3.3.2 Giao diện thanh Menu: 41](#_Toc168294434)

[3.3.3 Giao diện trang sản phẩm: 41](#_Toc168294435)

[3.3.4 Giao diện trang chi tiết sản phẩm: 42](#_Toc168294436)

[3.3.5 Giao diện trang đăng ký tài khoản: 42](#_Toc168294437)

[3.3.6 Giao diện trang đăng nhập: 43](#_Toc168294438)

[3.3.7 Giao diện trang giỏ hàng: 43](#_Toc168294439)

[3.3.8 Giao diện trang xác nhận đặt hàng: 44](#_Toc168294440)

[3.3.9 Giao diện trang xác nhận đã đặt hàng: 44](#_Toc168294441)

[3.3.10 Giao diện trang liên hệ: 45](#_Toc168294442)

[3.3.11 Giao diện trang giới thiệu: 45](#_Toc168294443)

[3.4 Giao diện Admin: 46](#_Toc168294444)

[3.4.1 Giao diện trang chủ Admin: 46](#_Toc168294445)

[3.4.2 Giao diện trang quản lý sản phẩm: 46](#_Toc168294446)

[3.4.3 Giao diện trang quản lý nhà cung cấp: 47](#_Toc168294447)

[3.4.4 Giao diện trang quản lý loại hàng: 47](#_Toc168294448)

[3.4.5 Giao diện trang quản lý tài khoản: 48](#_Toc168294449)

[3.4.6 Giao diện trang quản lý phân quyền: 48](#_Toc168294450)

[3.4.7 Giao diện trang quản lý đơn hàng: 49](#_Toc168294451)

[3.4.8 Giao diện trang thống kê: 49](#_Toc168294452)

[4 CHƯƠNG 5: TỔNG KẾT. 50](#_Toc168294453)

[4.1 Kết quả đạt được: 50](#_Toc168294454)

[4.2 Hoàn thành: 50](#_Toc168294455)

[4.3 Hướng phát triển: 51](#_Toc168294456)

[4.4 Tài liệu tham khảo: 52](#_Toc168294457)

# CHƯƠNG I: TỔNG QUAN VẤN ĐỀ.

## Giới thiệu đề tài:

### Khảo sát hiện trạng:

Ngành thời trang hiện nay đang phát triển mạnh mẽ với nhiều thương hiệu và cửa hàng mới xuất hiện. Điều này dẫn đến sự cạnh tranh gay gắt và nhu cầu về một hệ thống quản lý bán hàng hiệu quả trở nên cấp thiết.

Bên cạnh đó, sự bùng nổ của công nghệ và Internet đã thay đổi thói quen mua sắm của người tiêu dùng. Mua sắm trực tuyến đang trở thành xu hướng phổ biến, đòi hỏi các cửa hàng thời trang phải có một hệ thống bán hàng trực tuyến tốt để đáp ứng nhu cầu của khách hàng.

### Tính cấp thiết của đề tài

**Tăng Khả Năng Tiếp Cận Khách Hàng:** Website bán hàng giúp cửa hàng tiếp cận được lượng khách hàng lớn hơn, không bị giới hạn bởi địa lý. Khách hàng có thể dễ dàng xem và mua sắm các sản phẩm thời trang mọi lúc, mọi nơi.

**Tối Ưu Hóa Quy Trình Quản Lý:** Website quản lý bán hàng tích hợp các tính năng từ quản lý kho, theo dõi đơn hàng, quản lý khách hàng đến phân tích doanh thu, giúp tối ưu hóa các quy trình làm việc và giảm thiểu sai sót.

**Nâng Cao Trải Nghiệm Khách Hàng:** website bán hàng chuyên nghiệp với giao diện thân thiện, dễ sử dụng, cùng với các chức năng hỗ trợ khách hàng như chat trực tuyến, thanh toán linh hoạt, giúp nâng cao trải nghiệm mua sắm của khách hàng.

**Cạnh Tranh Với Các Đối Thủ:** Trong môi trường cạnh tranh gay gắt, việc sở hữu một website quản lý bán hàng chuyên nghiệp là yếu tố quan trọng giúp cửa hàng thời trang tạo lợi thế cạnh tranh và thu hút khách hàng.

### Kết Luận:

Việc xây dựng một website quản lý bán hàng thời trang không chỉ là một xu hướng tất yếu mà còn là một nhu cầu cấp thiết để đáp ứng sự phát triển của thị trường và nâng cao hiệu quả kinh doanh. Nghiên cứu và phát triển hệ thống này sẽ mang lại lợi ích lớn cho các cửa hàng thời trang, giúp họ quản lý tốt hơn, tiếp cận khách hàng hiệu quả hơn và nâng cao trải nghiệm mua sắm của khách hàng.

## Mục đích nghiên cứu:

* Phát triển lỹ năng quản lý bán hàng và phân tích dữ liệu.
* Tìm hiểu kĩ thuật lập trình trên ASP.NET MVC5.
* Tìm hiểu các công nghệ hỗ trợ như: HTML5, CSS3, Ajax, Jquery, Web Sevice…
* Tìm hiểu về ngôn ngữ và kiến thức tương tác CSDL (LINQ to SQL và Entity FrameWork).
* Cách điều khiển dữ liệu (Controller).
* Tìm hiểu về chức năng thanh toán trực tuyến qua website.

## Mục tiêu đề tài:

Là một website bán hàng, điều quan trọng cần chú trọng là tính thân thiện với người dùng và giao diện đẹp mắt. Cách trình bày sản phẩm, thông tin chi tiết, và giá cả hợp lý đều đóng vai trò quan trọng trong việc tạo ấn tượng tốt với khách hàng. Một trang web hấp dẫn và dễ sử dụng sẽ mang lại sự thoải mái cho cả khách hàng và người quản lý khi truy cập và thao tác.

* **Người dùng (User):**
* Tham quan trang web.
* Tìm kiếm và lựa chọn sản phẩm cần mua.
* Xem chi tiết 1 sản phẩm.
* Đăng ký, đăng nhập phục vụ cho việc mua bán online.
* Thực hiện việc mua hàng và thanh toán trực tuyến.
* **Người quản trị (Admin):**
* Quản lý trang web của mình thông qua tài khoản Admin với giao diện rõ ràng, dễ sử dụng, bảo mật cao.
* Quản lý tất cả các sản phẩm một cách dễ dàng.
* Xem sản phẩm danh sách sản phẩm mới và danh sách sản phẩm bán chạy.
* Kiểm tra và xử lý đơn hàng.

# CHƯƠNG II: TÌM HIỂU VỀ CƠ SỞ LÝ THUYẾT

## Tổng quan về ASP.NET:

### Giới thiệu về ASP.NET:

Đầu năm 2002, Microsoft giới thiệu một kỹ thuật lập trình Web khá mới mẻ với tên gọi ban đầu là ASP+, tên chính thức sau này là ASP.Net. Với ASP.Net, không những không cần đòi hỏi bạn phải biết các tag HTML, thiết kế web, mà nó còn hỗ trợ mạnh lập trình hướng đối tượng trong quá trình xây dựng và phát triển ứng dụng Web. ASP.Net là kỹ thuật lập trình và phát triển ứng dụng web ở phía Server (Server-side) dựa trên nền tảng của Microsoft .Net Framework. Hầu hết, những người mới đến với lập trình web đều bắt đầu tìm hiểu những kỹ thuật ở phía Client (Client-side) như: HTML, Java Script, CSS (Cascading Style Sheets). Khi Web browser yêu cầu một trang web (trang web sử dụng kỹ thuật client-side), Web server tìm trang web mà Client yêu cầu, sau đó gởi về cho Client. Client nhận kết quả trả về từ Server và hiển thị lên màn hình. ASP.Net sử dụng kỹ thuật lập trình ở phía server thì hoàn toàn khác, mã lệnh ở phía server (ví dụ: mã lệnh trong trang ASP) sẽ được biên dịch và thi hành tại Web Server. Sau khi được Server đọc, biên dịch và thi hành, kết quả tự động được chuyển sang HTML/JavaScript/CSS và trả về cho Client. Tất cả các xử lý lệnh ASP.Net đều được thực hiện tại Server và do đó, gọi là kỹ thuật lập trình ở phía server. ASP.NET được Microsoft phát triển qua nhiều phiên bản từ ASP.NET 1.0, 1.1, 2.0 và gần đây nhất là phiên bản ASP.NET 5.

### Tìm hiểu về mô hình lập trình web MVC của ASP.NET:

Mô hình MVC (viết tắt chữ cái đầu của 3 từ Model - View - Controller) là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm (đặc biệt đối với phát triển ứng dụng web). Nó giúp cho tổ chức ứng dụng (phân bố source code ứng dụng) thành 3 phần khác nhau *Model*, *View* và *Controller*. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

**Model**: là các thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất CSDL, đối tượng mô tả dữ liệu như các Class, hàm xử lý... Model được giao nhiệm vụ cung cấp dữ liệu cho CSDL và lưu dữ liệu vào các kho chứa dữ liệu. Tất cả các nghiệp vụ logic được thực thi ở Model. Dữ liệu vào từ người dùng sẽ thông qua View để kiểm tra ở Model trước khi lưu vào cơ sở dữ liệu. Việc truy xuất, xác nhận và lưu dữ liệu là một phần của Model.

**View**: View hiển thị các thông tin cho người dùng của ứng dụng và được giao nhiệm vụ cho việc nhận các dữ liệu vào từ người dùng, gởi đi các yêu cầu người dùng đến bộ điều khiển (Controller), sau đó là nhận lại các phản hồi từ bộ điều khiển và hiển thị kết quả cho người dùng. Các trang HTML, JSP, các thư viện thể và các file nguồn là một phần của View.

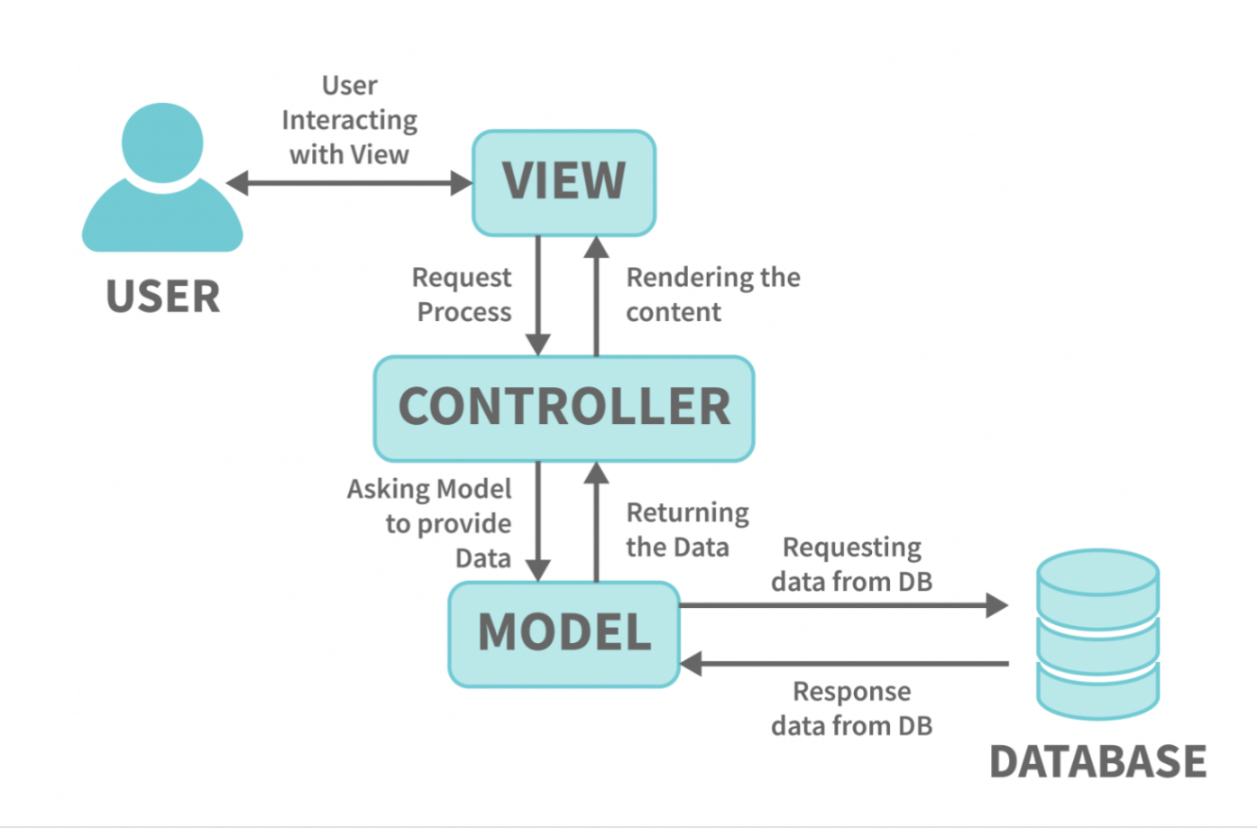
**Controller**: Controller là tầng trung gian giữa Model và View. Controller được giao nhiệm vụ nhận các yêu cầu từ người dùng (phía máy khách). Một yêu cầu được nhận từ máy khách được thực hiện bởi một chức năng logic thích hợp từ thành phần Model và sau đó sinh ra các kết quả cho người dùng và được thành phần View hiển thị. ActionServlet, Action, ActionForm, struts-config.xml là các thành phần của Controller.

Một ứng dụng web được phát triển theo mô hình MVC, có nguyên lý hoạt động xử lý yêu cầu của người dùng như sau:

- Khi có yêu cầu phát sinh từ người dùng (phía Client), yêu cầu này sẽ được gởi đến phía Server, tại đây Contronler sẽ tiếp nhận yêu cầu để xử lý.

- Bên trong Controller chứa nhiều phương thức (action), mỗi action tương ứng với 1 yêu cầu cụ thể từ người dùng. Controller sẽ lựa chọn action phù hợp với yêu cầu để xử lý. Trong quá trình xử lý, Controller tương tác với Model để có được dữ liệu mà người dùng mong muốn.

- Sau khi Controller làm việc với model để có được dữ liệu theo yêu cầu, Controller sẽ gởi về cho View thành phần Data Model, và nhiệm vụ của View sẽ chuyển Data Model nhận được thành dữ liệu ứng dụng và gởi trả về phía Client để hiển thị kết quả yêu cầu.



Hình 2.1: Mô hình MVC

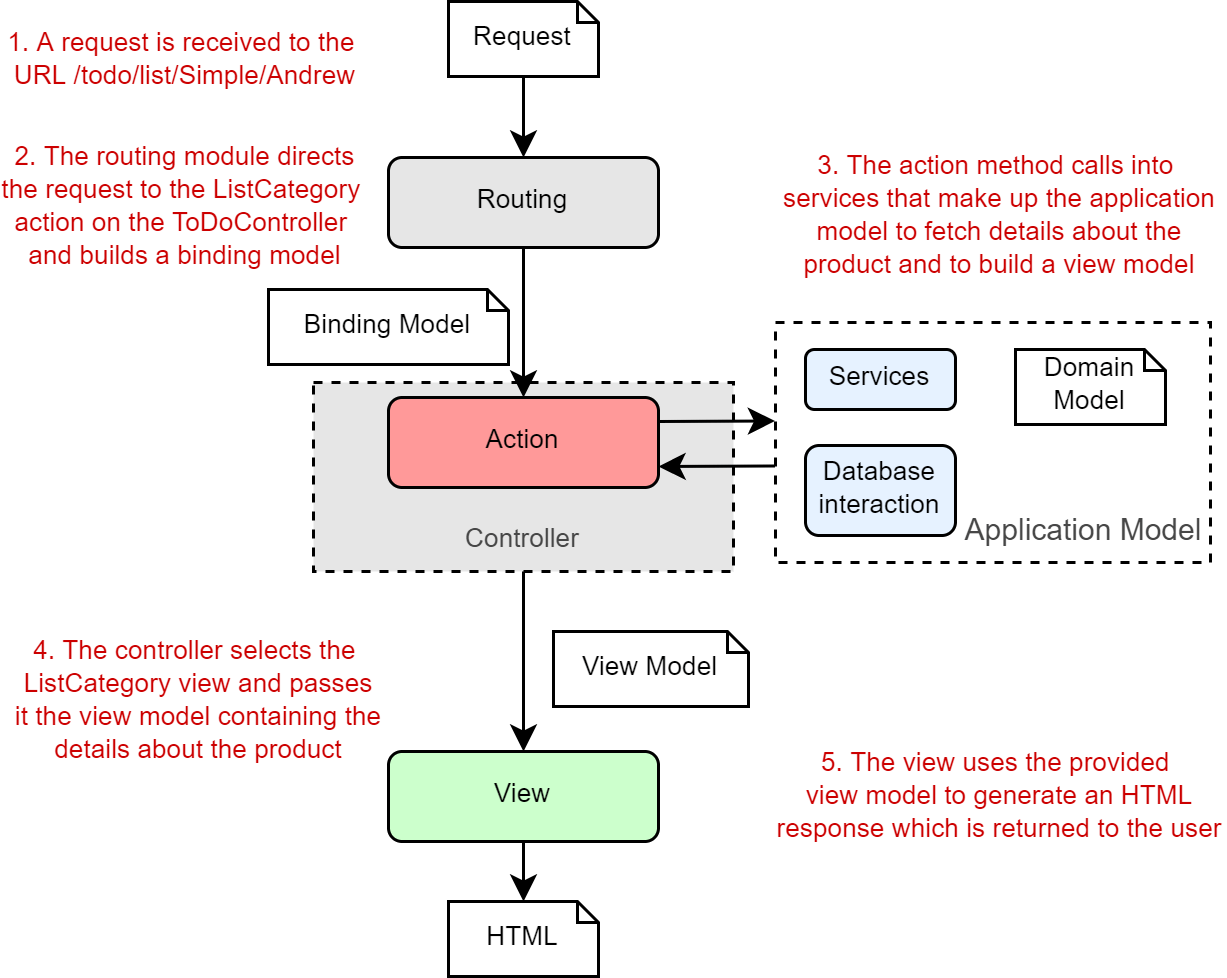
## Giới thiệu mô hình lập trình Web bằng ASP.NET MVC5

ASP.NET MVC5 giúp cho chúng ta có thể tạo được các ứng dụng web áp dụng mô hình MVC thay vì tạo ứng dụng theo mẫu ASP.NET Web Form. Nền tảng ASP.NET MVC có đặc điểm nổi bật là nhẹ (lightweight), dễ kiểm thử phần giao diện (so với ứng dụng Web Forms), tích hợp các tính năng có sẵn của ASP.NET. Nền tảng ASP.NET MVC được định nghĩa trong namespace System.Web.MVC và là một phần của name space System.Web. MVC là một mẫu thiết kế (design pattern) chuẩn mà nhiều lập trình viên đã quen thuộc. Một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC. Một số khác vẫn thích hợp với ASP.NET Web Forms và cơ chế postbacks. Đôi khi có những ứng dụng kết hợp cả hai kiến trúc trên.

* + Tách bạch các tác vụ của ứng dụng (logic nhập liệu, business logic, và logic giao diện), dễ dàng kiểm thử và mặc định áp dụng hướng phát triển TDD. Tất cả các tính năng chính của mô hình MVC được cài đặt dựa trên Interface và được kiểm thử bằng cách sử dụng các đối tượng mocks, mock object là các đối tượng mô phỏng các tính năng của những đối tượng thực sự trong ứng dụng. Bạn có thể kiểm thử unit-test cho ứng dụng mà không cần chạy Controller trong tiến trình ASP.NET, và điều đó giúp unit test được áp dụng nhanh chóng và tiện dụng. Bạn có thể sử dụng bất kỳ nền tảng unit-testing nào tương thích với nền tảng.
  + NET. MVC là một nền tảng khả mở rộng (extensible) & khả nhúng (pluggable). Các thành phần của ASP.NET MVC được thiết kể để chúng có thể được thay thế một cách dễ dàng hoặc dễ dàng tùy chỉnh. Bạn có thể nhúng thêm view engine, cơ chế định tuyến cho URL, cách kết xuất tham số của action-method và các thành phần khác. ASP.NET MVC cũng hỗ trợ việc sử dụng Dependency Injection (DI) và Inversion of Control (IoC). DI cho phép bạn gắn các đối tượng vào một lớp cho lớp đó sử dụng thay vì buộc lớp đó phải tự mình khởi tạo các đối tượng. IoC quy định rằng, nếu một đối tượng yêu cầu một đối tượng khác, đối tượng đầu sẽ lấy đối tượng thứ hai từ một nguồn bên ngoài, ví dụ như từ tập tin cấu hình. Và nhờ vậy, việc sử dụng DI và IoC sẽ giúp kiểm thử dễ dàng hơn.
  + ASP.NET MVC có thành phần ánh xạ URL mạnh mẽ cho phép bạn xây dựng những ứng dụng có các địa chỉ URL xúc tích và dễ tìm kiếm. Các địa chỉ URL không cần phải có phần mở rộng của tên tập tin và được thiết kế để hỗ trợ các mẫu định dạng tên phù hợp với việc tối ưu hóa tìm kiếm (URL) và phù hợp với lập địa chỉ theo kiểu REST.
  + Hỗ trợ sử dụng đặc tả (các thẻ) của các trang ASP.NET (.aspx), điều khiển người dùng (.ascx) và trang master page. Bạn có thể sử dụng các tính năng có sẵn của ASP.NET như là sử dụng lồng các trang master page, sử dụng in-line expression (<%= %>), sử dụng server controls, mẫu, data-binding, địa phương hóa (localization) và hơn thế nữa.
  + Hỗ trợ các tính năng có sẵn của ASP.NET như cơ chế xác thực người dùng, quản lý thành viên, quyền, output caching và data caching, session và profile, quản lý tình trạng ứng dụng, hệ thống cấu hình…
  + ASP.NET MVC bổ sung một view engine mới là Razor View Engine cho phép thiết lập các view nhanh chóng, dễ dàng và tốn ít công sức hơn so với việc sử dụng Web Forms view engine.

## Hoạt động của các thành phần chính trong ASP.NET MVC:

* Khi 1 request phát sinh từ web browser đi đến IIS Web Server thì request đó cuối cùng được đưa đến 1 MVC Handler.
* MVC Handler có nhiệm vụ chọn ra đúng 1 Controller để xử lý request đó. Controller được tạo ra bởi 1 thành phần được gọi là Controller Factory.
* Sau khi được tạo ra, Controller sẽ xác định request này được xử lý bởi Action Method cụ thể nào và sau đó sẽ thực thi Action Method đó. Action method có thể tương tác với những Model Class để có thể truy xuất dữ liệu hoặc thực thi một số business logic.
* Sau khi hoàn tất xử lý Action Method đó sẽ trả về 1 Action Result. ASP.NET MVC cung cấp nhiều Action Result, trong đó có 1 Action Result đặc biệt là View Result. Action Result này có nhiệm vụ làm việc với 1 View nhất định để tạo ra mã HTML để trả về cho web browser và người dùng có thể nhận được kết quả xử lý.
* View Engine là thành phần thực hiện hiển thị 1 View, đi kèm với ASP.NET MVC là Webform View Engine tức là chúng ta có thể viết View bằng ASPX.



Hình 2.2: Mô tả hoạt động của MVC trong ASP.NET

## Tại sao phải sử dụng ASP.NET?

Yêu cầu về xây dựng các ứng dụng thương mại điện tử ngày càng đuợc phát triển và nâng cao. Khi đó ASP không còn đáp ứng được yêu cầu đặt ra. ASP được thiết kế riêng biệt và nằm ở tầng phía trên hệ điều hành Windows và Internet Information Service, do đó các công dụng của nó hết sức rời rạc và giới hạn. ASP.Net đưa ra một phương pháp phát triển hoàn toàn mới khác hẳn so với ASP trước kia và đáp ứng được các yêu cầu đặt ra.

## Ưu điểm và khuyết điểm của ASP.NET:

**Ưu điểm:**

* ASP chỉ sử dụng VBScript và JavaScript mà không sử dụng được các ngôn ngữ mạnh khác: Visual Basic, C++… Trong khi đó ASP.NET cho phép viết nhiều ngôn ngữ: VBScript, JavaScript, C#, Visual Basic.Net, …
* ASP.Net sử dụng phong cách lập trình mới: Code behide. Tách code riêng, giao diện riêng 🡪dễ đọc, dễ quản lý và bảo trì.
* Trong các trang ASP chúng ta phải viết mã để kiểm tra dữ liệu nhập từ người dùng, ASP.NET hỗ trợ các validation controls để kiểm tra chúng ta không cần viết mã, ...
* Hỗ trợ phát triển Web được truy cập trên các thiết bị di động: PocketPC, Smartphone…
* Hỗ trợ nhiều web server control.
* Hỗ trợ thiết kế và xây dựng MasterPage lồng nhau.
* Hỗ trợ bẫy lỗi (debug) JavaScript.
* Cho phép người dùng thiết lập giao diện trang Web theo sở thích cá nhân sử dụng Theme, Profile, WebPart.
* Tăng cường các tính năng bảo mật (security).
* Hỗ trợ kỹ thuật truy cập dữ liệu mới LINQ.
* Hỗ trợ kỹ thuật xây dụng các ứng dụng đa phương tiện SilverLight.
* Hỗ trợ kỹ thuật bất đồng bộ ASP.Net Ajax.
* ASP.Net hỗ trợ mạnh mẽ bộ thư viện phong phú và đa dạng của .Net Framework, làm việc với XML, Web Service, truy cập cơ sở dữ liệu qua ADO.Net, …
* ASPX và ASP có thể cùng hoạt động trong 1 ứng dụng.
* Kiến trúc lập trình giống ứng dụng trên Windows.
* Hỗ trợ quản lý trạng thái của các control.
* Tự động phát sinh mã HTML cho các Server control tương ứng với từng loại Browser.
* Hỗ trợ nhiều cơ chế Cache.
* Triển khai cài đặt: Không cần lock, không cần đăng ký DLL, cho phép nhiều hình thức cấu hình ứng dụng.
* Hỗ trợ quản lý ứng dụng ở mức toàn cục: Global.aspx có nhiều sự kiện hơn, quản lý session trên nhiều Server, không cần Cookies.
* Trang ASP.Net được biên dịch trước. Thay vì phải đọc và thông dịch mỗi khi trang web được yêu cầu, ASP.Net biên dịch những trang web động thành những tập tin DLL mà Server có thể thi hành nhanh chóng và hiệu quả. Yếu tố này làm gia tăng tốc độ thực thi so với kỹ thuật thông dịch của ASP.

**Khuyết điểm:**

* Đổi với dự án nhỏ việc áp dụng mô hình MVC gây cồng kềnh, tốn thời gian trong quá trình phát triển.
* Tốn thời gian trung chuyển dữ liệu của các thành phần.

## Sự khác biệt mô hình lập trình Web ASP.NET MVC và ASP.NET Webform:

|  |  |  |
| --- | --- | --- |
| **Các tính năng** | **ASP.NET WebForm** | **ASP.NET MVC** |
| Kiến trúc chương trình | Kiến trúc mô hình WebForm –> Bussiness –> Database | Kiến trúc sử dụng việc phân chia chương trình thành: Models, Views, Controllers |
| Cú pháp chương trình | Sử dụng cú pháp của WebForm, tất cả các sự kiện và controls do server quản lý | Các sự kiện được điều khiển bởi controllers, các controls không do server quản lý |
| Truy cập dữ liệu | Sử dụng hầu hết các công nghệ truy cập dữ liệu trong ứng dụng | Phần lớn dùng LINQ và SQL class để tạo mô hình truy cập đối tượng |
| Debug | Debug phải thực hiện tất cả bao gồm các lớp truy cập dữ liệu, sự hiển thị, điều khiển các controls | Debug có thể sử dụng các unit test để kiểm tra các phương thức trong controllers |
| Tốc độ phân tải | Tốc độ phân tải chậm khi trong trang có quá nhiều các controls vì ViewState quá lớn | Phân tải nhanh hơn do không phải quản lý ViewState để quản lý các controls trong trang |
| Tương tác với JavaScript | Tương tác với JavaScript khó khăn vì các controls được điều khiển bởi server | Tương tác với JavaScript dễ dàng vì các đối tượng không do server quản lý điều khiển không khó |
| URL address | Cấu trúc địa chỉ URL có dạng:  <filename>.aspx?&<các tham số> | Cấu trúc địa chỉ rành mạch, dễ hiểu theo dạng Controllers/Action/ID |

## Ajax:

### Ajax là gì nhỉ?

Ajax là một khái niệm có thể mới lạ với những bạn mới học lập trình web đôi lúc các bạn nghĩ nó là một ngôn ngữ lập trình mới. Nhưng thực tế không như vậy, ajax là một kỹ thuật viết tắt của chữ AJAX = Asynchronous JavaScript and XML, đây là một công nghệ giúp chung ta tạo ra những Web động mà hoàn toàn không reload lại trang nên rất mượt và đẹp. Đối với công nghệ web hiện nay thì ajax không thể thiếu, nó là một phần làm nên sự sinh động cho website. Còn đối với ngành SEO thì khi sử dụng ajax lại không tốt, tại vì con bọ google sẽ không index được, nhưng thực tế ta có cách khắc phục lỗi này, vấn đề này ta sẽ tìm hiểu nó ở một bài khác.

Ajax được viết bằng ngôn ngữ Javascript nên nó chạy trên client, tức là mỗi máy (user) sẽ chạy độc lập không hoàn toàn ảnh hưởng lẫn nhau. Hiện nay có nhiều thư viện javascript như **jQuery**hay **Angular**đều hỗ trợ kỹ thuật này nhằm giúp chúng ta thao tác dễ dàng hơn.

### Ajax hoạt động thế nào?

Từ lâu, mọi người đã tưởng tượng ứng dụng máy tính rồi sẽ được lưu và chạy hoàn toàn trên web thay vì nằm bó buộc trong ổ cứng. Dù vậy, viễn cảnh đó vẫn chưa thể xảy ra do ứng dụng web bị hạn chế bởi nguyên lý rằng tất cả các thao tác phải được thực hiện thông qua HTTP (HyperText Transfer Protocol - Giao thức truyền tải qua siêu liên kết). Những hoạt động của người sử dụng trên trang web sẽ tạo ra một yêu cầu HTTP tới server. Máy chủ thực hiện một số khâu xử lý như lấy lại dữ liệu, tính toán, kiểm tra sự hợp lệ của thông tin, sửa đổi bộ nhớ, sau đó gửi lại một trang HTML hoàn chỉnh tới máy khách. Về mặt kỹ thuật, phương pháp này nghe có vẻ hợp lý nhưng cũng khá bất tiện và mất thời gian, bởi khi server đang thực hiện vai trò của nó thì người dùng sẽ làm gì? Tất nhiên là chờ đợi. Để khắc phục hạn chế trên, các chuyên gia phát triển giới thiệu hình thức trung gian - cơ chế xử lý AJAX - giữa máy khách và máy chủ. Điều này giống như việc tăng thêm một lớp giữa cho ứng dụng để giảm quá trình “đi lại” của thông tin và giảm thời gian phản ứng. Thay vì tải lại (refresh) toàn bộ một trang, nó chỉ nạp những thông tin được thay đổi, còn giữ nguyên các phần khác. Vì thế, khi duyệt một trang hỗ trợ AJAX, người sử dụng không bao giờ nhìn thấy một cửa sổ trắng (blank) và biểu tượng đồng hồ cát - dấu hiệu cho thấy máy chủ đang thực hiện nhiệm vụ. Ví dụ, trong một website ảnh, với ứng dụng truyền thống, toàn bộ trang chứa các ảnh sẽ phải mở lại từ đầu nếu có một thay đổi nào đó trên trang. Còn khi áp dụng AJAX, DHTML chỉ thay thế đoạn tiêu đề và phần vừa chỉnh sửa, do vậy tạo nên các giao dịch trơn tru, nhanh chóng.

Nhiều công ty khác cũng đang nối gót xu thế này như trang chia sẻ ảnh Flickr (hiện thuộc Yahoo) hay công cụ tìm kiếm A9.com của Amazon. Yahoo dự định trong vài tuần tới sẽ ra mắt bản Yahoo Mail Beta 1 sử dụng AJAX trên toàn thế giới (đang được thử nghiệm giới hạn). Hòm thư mới sẽ được trang bị thêm nhiều tính năng thuộc Web 2.0 như RSS, chế độ xem thư trước (preview)… Microsoft cũng đang triển khai chương trình Windows Live Mail và Windows Live Messenger hỗ trợ AJAX.

Những dự án trên cho thấy AJAX không phải là một công nghệ quá xa xôi mà đang hiện diện ngay trong thế giới thực, từ mô hình rất đơn giản như Google Suggest đến tinh vi và phức tạp như Google Maps.

Thách thức lớn nhất khi tạo ứng dụng AJAX không nằm ở khâu kỹ thuật bởi những thành phần của nó đã xuất hiện từ lâu, hoạt động ổn định và được hiểu người biết đến. Vấn đề ở đây chỉ là “các chuyên gia thiết kế cần quên đi những giới hạn của web, bắt đầu nghĩ rộng hơn, sâu hơn về khả năng của công nghệ và sáng tạo nó theo cách riêng của mỗi người”, Alexei White, Giám đốc sản xuất của công ty eBusiness (Mỹ), nhận xét. “AJAX sẽ làm mất dần ảnh hưởng của Microsoft trên thị trường. Tuy chưa thể lấn át hoàn toàn, nó sẽ mang lại những tính năng thay thế cho hầu hết các sản phẩn như Office”.

### Nhược điểm của AJAX?

AJAX có thể góp phần tạo nên một thế hệ mới cho ứng dụng web (như colr.org hay backpackit.com). Tuy nhiên, nó cũng là một công nghệ “nguy hiểm” khi gây ra không ít rắc rối về giao diện người dùng. Chẳng hạn, phím “Back” được đánh giá cao trong giao diện website chuẩn. Đáng tiếc, chức năng này không hoạt động ăn khớp với Javascript và mọi người không thể tìm lại nội dung trước đó khi bấm phím Back. Bởi vậy, chỉ một sơ xuất nhỏ là dữ liệu trên trang đã bị thay đổi và khó có thể khôi phục lại được. Đây là một trong những nguyên nhân chính khiến nhiều người không ủng hộ ứng dụng Javascript.

Bên cạnh đó, mọi người không thể lưu lại địa chỉ web vào thư mục Favorite (Bookmark) để xem lại về sau. Do áp dụng lớp trung gian để giao dịch, các ứng dụng AJAX không có một địa chỉ cố định cho từng nội dung. Khiếm khuyết này làm cho AJAX dễ “mất điểm” trong mắt người dùng.

Những trình duyệt hỗ trợ AJAX là Microsoft Internet Explorer 5.0 trở lên; browser dựa trên Gecko như Mozilla, Firefox, SeaMonkey, Epiphany, Galeon và Netscape 7.1; trình duyệt chứa KHTML API 3.2 trở lên như Konqueror, Apple Safari…

CSS - tập tin định kiểu theo tầng (Cascading Style Sheets) - được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ HTML, XHTML, XML, SVG, XUL… Các thông số kỹ thuật của CSS do tổ chức World Wide Web Consortium (W3C) quản lý.

DOM - mô hình đối tượng tài liệu (Document Object Model) - là một giao diện lập trình ứng dụng (API). Thông thường DOM có dạng một cây cấu trúc dữ liệu và được dùng để truy xuất các tài liệu HTML và XML. Mô hình DOM hoạt động độc lập với hệ điều hành và dựa theo kỹ thuật lập trình hướng đối tượng để mô tả tài liệu.

DHTML, hay HTML động, tạo một trang web bằng cách kết hợp các thành phần: ngôn ngữ đánh dấu HTML tĩnh, ngôn ngữ lệnh máy khách (như Javascript) và ngôn ngữ định dạng CSS và DOM. Do có khả năng phong phú, DHTML còn được dùng như một công cụ xây dựng các trò chơi đơn giản trên trình duyệt.

## HTML:

### HTML là gì?

HTML là chữ viết tắt của cụm từ HyperText Markup Language (Ngôn ngữ đánh dấu siêu văn bản), được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu, tập tin HTML.

Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc ngọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví <strong> dụ </strong> và). Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ (ví dụ <strong>Đây là chữ in đậm</strong>). Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ<img>).

Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là .html hoặc .htm.

### HTML được xử lý ra sao?

Khi một tập tin HTML được hình thành, việc xử lý nó sẽ do trình duyệt web đảm nhận. Trình duyệt sẽ đóng vai trò đọc hiểu nội dung HTML từ các thẻ bên trong và sẽ chuyển sang dạng văn bản đã được đánh dấu để đọc, nghe hoặc hiểu (do các bot máy tính hiểu).

### Dùng chương trình gì để tạo tập tin HTML?

HTML là một tập tin siêu văn bản nên bạn có thể dùng các chương trình soạn thảo văn bản không có chức năng định dạng văn bản để tạo ra một tập tin HTML. Trong Windows, bạn có thể dùng Notepad để tạo ra một tập tin HTML, còn trên Mac thì có thể dùng TextEdit và Vim trên các hệ điều hành Linux khác. Miễn là sau đó bạn phải lưu tập tin thành đuôi .html và sử dụng trình duyệt website để đọc nó.

Tuy nhiên Notepad lại quá đơn giản để sử dụng cho mục đích soạn thảo, nên chúng ta sẽ dùng một chương trình khác tương tự để soạn thảo HTML. Và mình cũng không bao giờ khuyến khích bạn sửa một tập tin HTML bất kỳ bằng chương trinh Notepad để tránh gặp các lỗi hiển thị ký tự tiếng Việt.

### HTML đóng vai trò gì trong website?

Như mình đã nói, HTML là một ngôn ngữ đánh dấu siêu văn bản nên nó sẽ có vai trò xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình ảnh, video, nhạc.

## CSS:

*CSS là gì?*

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). Bạn 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,…rất nhiều.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

## Bootstrap:

### Bootstrap là gì?

Bootstrap là một framework CSS được Twitter phát triển. Nó là một tập hợp các bộ chọn, thuộc tính và giá trị có sẵn để giúp web designer tránh việc lặp đi lặp lại trong quá trình tạo ra các class CSS và những đoạn mã HTML giống nhau trong dự án web của mình. Ngoài CSS ra, thì bootstrap còn hỗ trợ các function tiện ích được viết dựa trên Jquery (Carousel, Tooltip, Popovers, …)

### Vì sao phải dùng nó?

Được viết bởi những người có óc thẩm mỹ và tài năng trên khắp thế giới. Sự tương thích của trình duyệt với thiết bị đã được kiểm tra nhiều lần nên có thể tin tưởng kết quả mình làm ra và nhiều khi không cần kiểm tra lại. Vì vậy, giúp cho dự án của bạn tiết kiệm được thời gian và tiền bạc.

Chỉ cần biết sơ qua HTML, CSS, Javascript, Jquery là bạn có thể sử dụng Bootstrap để tạo nên một trang web sang trọng và đầy đủ. Nhưng lại không cần code quá nhiều CSS.

Với giao diện mặc định là màu xám bạc sang trọng, hỗ trợ các component thông dụng mà các website hiện nay cần có. Vì nó là opensource nên bạn có thể vào mã nguồn của nó để thay đổi theo ý thích của bản thân.

Do có sử dụng Grid System nên Bootstrap mặc định hỗ trợ Responsive. Bootstrap được viết theo xu hướng Mobile First tức là ưu tiên giao diện trên Mobile trước. Nên việc sử dụng Bootstrap cho website của bạn sẽ phù hợp với tất cả kích thước màn hình. Nhờ đó mà chúng ta không cần xây dựng thêm một trang web riêng biệt cho mobile.

Đội ngũ phát triển Bootstrap đã bổ sung thêm tính năng Customizer. Giúp cho designer có thể lựa chọn những thuộc tính, component phù hợp với project của họ. Chức năng này giúp ta không cần phải tải toàn bộ mã nguồn về máy.

## Jquery:

### Jquery là gì?

Jquery 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 tương tác Ajax vào trong trang web của bạn.

JQuery được thiết kế để thay đổi cách viết Javascript của bạn. Chỉ với 10 dòng lệnh JQuery bạn có thể thay thế cả 20 chục dòng lệnh DOM JavaScript.

JQuery là 1 bộ thư viện khổng lồ hỗ trợ cho mọi ngôn ngữ lập trình mà người lập trình chỉ cần có kiến thức về HTML đều có thể học được. Một lập trình viên sử dụng jquery để lập trình sẽ tiết kiệm được nhiều thời gian của người lập trình web.

### Jquery giúp ích gì cho người lập trình web?

1. Truy cập các phần tử trong nội dung trang web:
2. Jquery cho phép bạn chọn bất cứ thành phần nào của tài liệu để “vọc” một cách dễ dàng như sử dụng CSS.
3. Thay đổi hình thức giao diện của trang web:
4. Jquery ra đời để lấp chỗ trống này, vì vậy bạn có thể sử dụng nó để giúp trang web có thể hiển thị tốt trên hầu hết các trình duyệt. Hơn nữa jQuery cũng có thể thay đổi class hoặc những định dạng CSS đã được áp dụng lên bất cứ thành phần nào của HTML ngay cả khi trang web đó đã được trình duyệt load thành công.
5. Thay đổi nội dung trang web.
6. jquery không chỉ có thể thay đổi bề ngoài của trang web, nó cũng có thể thay đổi nội dung của chính tài liệu đó chỉ với vài dòng code. Nó có thể thêm hoặc bớt nội dung trên trang, hình ảnh có thể được thêm vào hoặc đổi sang hình khác, danh sách có thể được sắp xếp lại hoặc thậm chí cả cấu trúc HTML của một trang web cũng có thể được viết lại và mở rộng.
7. Jquery cho phép các nhà phát triển web viết code javascript đơn giản hơn nhiều so với cách truyền thống như là các vòng lặp và điều khiển mảng.

### Tại sao phải dùng Jquery?

Có rất nhiều lí do để mình khuyên các bạn nên dùng jquery thay vì dùng javascript phải nói đến đầu tiên đó chính là số plugin của jquery khá là khổng lồ giúp bạn tiết kiệm được kha khá thời gian viết các ứng dụng liên quan đến giao diện, các hiệu ứng …

Hơn nữa, jQuery đơn giản hóa cách viết Javascript và tăng tốc độ xử lý các sự kiện trên trang web, giúp tiết kiệm thời gian và công sức rất nhiều so với việc ngồi viết javascript theo cách thông thường. Đúng như khẩu hiệu của nó Viết ít hơn, làm nhiều hơn, nếu không dùng jQuery thì một chuyên gia cũng phải viết mất vài trang giấy mới làm được một nút Back to Top với hiệu ứng trượt, trong khi đó dân nghiệp dư chỉ cần vài dòng code ngắn ngủi. jQuery là đã có thể “múa rìu qua mắt thợ”. Bên cạnh đó, việc sử dụng jQuery giúp giải quyết tốt vấn đề xung đột giữa các trình duyệt web.

## HTML5:

### HTML5 là gì?

HTML5 là một chuẩn mới và là thế hệ tiếp theo của ngôn ngữ đánh dấu siêu văn bản – HyperText Markup Language explained (gọi tắt là HTML). Các phiên bản trước của HTML, như HTML 4.01 đã ra đời từ năm 1999. Cho đến nay các trang web đã có những thay đổi rất nhiều kể từ đó. HTML5 vẫn còn trong giai đoạn phát triển và hoàn thiện. Tuy nhiên, nhiều phiên bản trình duyệt mới hiện nay đã có các hỗ trợ cho những phần tử, thẻ mới có trong HTML5 và các APIs.  
HTML5 là kết quả của sự hợp tác giữa tổ chức W3C và nhóm WHATWG. WHATWG làm việc với các web form và các ứng dụng, còn W3C thì được làm việc với XHTML 2.0. Trong năm 2006, họ quyết định hợp tác và tạo ra một phiên bản mới của HTML, đó chính là HTML5.

### HTML5 là vấn đề quan trọng của chúng ta?

Như mọi người sử dụng web, bạn sẽ được hưởng lợi từ HTML5 bởi vì nó sửa chữa những vấn đề sai sót rõ ràng nhất trong HTML4. Các trang wen sẽ có tiêu chuẩn web tốt hơn, kết quả của nội dung hiệu quả hơn và cải thiện hiệu suất. Một khi HTML5 được áp dụng trên diện rộng, các trang web sẽ bắt đầu chạy nhanh hơn, băng thông sẽ tốn ít hơn khi được và thời gian “thọ” của pin trên các thiết bị di động sẽ được kéo dài hơn.  
 Một điều thêm nữa là bạn sẽ không cần phải giữ quá nhiều plugin như Flash và Java vì sự update. Hiện nay vấn đề này vẫn là một nguyên nhân gây phiền nhiễu cho người dùng khá nhiều. Người dùng phải liên tục cập nhật quá nhiều addon và plugin cho mỗi trình duyệt. Và điều gì sẽ xảy ra khi một trong số chúng là phiên bản bị sai? Tất nhiên là các trang web ngừng hoạt động cho đến khi bạn khắc phục bằng sự cập nhật đúng. Tất cả các điều đó sẽ được giải quyết ngay khi HTML5 trở thành tiêu chuẩn chính.  
 Nếu bạn chỉ là một người sử dụng web thông thường và bạn không có ý định mã hóa hoặc duy trì trang web riêng của bạn, bạn không cần phải làm bất cứ điều gì để thưởng thức các tính năng tuyệt vời của HTML5. Tất cả các trình duyệt hiện nay được HTML5 hỗ trợ ở một mức độ lớn và bạn đã có thể được tận dụng lợi thế của nó mà không hề hay biết. Chỉ cần giữ cho trình duyệt của bạn được cập nhật thường xuyên và bạn sẽ nhận được những gì tốt đẹp nhất.  
Và nếu bạn là một nhà phát triển web, HTML5 sẽ làm cho mọi thứ trở nên đơn giản và dễ dàng hơn cho bạn. Nếu mọi việc suôn sẻ, bạn sẽ không còn phải đối phó với trường hợp xấu, lỗi, trục trặc trong thiết kế web bởi vì tất cả các trình duyệt cần phải tuân thủ các tiêu chuẩn như nhau.

## CSS3:

Là phiên bản nâng cấp của CSS, CSS3 tuy chưa được chính thức sử dụng rộng rãi nhưng những gì mà nó có thể làm được quả là rất đáng mong đợi. Về mặt nguyên lý thì có vẻ sự ra đời của CSS3 đã phần nào lấn sân của Javascript. Bởi vì người ta nói rằng HTML dùng để thể hiện nội dung, CSS dùng để trình bày cấu trúc và Javascript thì tương tác với người dùng. Nhưng những tính năng mới của CSS3 như transition và animation phần nào đã làm thay đổi nguyên lý này.  
 Hiện tại khi sử dụng Javscript người lập trình bao giờ cũng phải tính đến điều kiện trình duyệt của người dùng không bật Javascript. Nhưng nếu trong tương lai (hy vọng không xa) chúng ta có thể tạo ra những hiệu ứng động đơn giản mà không cần sự hỗ trợ của Javascript thì cũng rất tiện.  
 CSS3 được phát triển nhằm tăng sự tương thích với HTML5 trong việc bố cục trình bày nội dung. Giờ đây, những tài liệu HTML đã có một bố cục hết sức rõ ràng và dễ hiểu, các thuộc tính CSS và các thẻ HTML giờ đã không còn bị chồng lấn mà được phân tách bằng các hàng “class” và “id”.  
 Ngoài việc hỗ trợ về giao diện thì CSS3 còn giúp bạn loại bỏ những đoạn code thừa trong HTML khi việc sử dụng những thẻ thuộc tính trong HTML không còn cần thiết nữa.  
 Media Queries mới được ra mắt trong CSS3 hỗ trợ tính tương thích với nhiều kích thước màn hình khác nhau mà không cần chỉnh sửa nội dung hiển thị, làm tăng tính uyển chuyển của ứng dụng.  
 CSS3 là ngôn ngữ được hỗ trợ bởi hầu hết các trình duyệt trên thế giới. Và nó thể hiện một website trên nhiều trình duyệt khác nhau cũng khá nhất quán. Tuy nhiên, người thiết kế web vẫn cần xác định rõ trình duyệt được người dùng sử dụng nhiều nhất để có sự điều chỉnh tốt nhất cho trình duyệt đó hoặc bạn có thể sử dụng một số hàm fix hỗ trợ CSS3.

# CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Phân tích thiết kế các chức năng ứng dụng:

### Đặc tả chức năng ứng dụng:

Mục đích của đề tài này là xây dựng một hệ thống phần mềm hoạt động trên môi trường web để quản lý việc bán hàng một cách thuận tiện và hiệu quả. Hệ thống này sẽ giúp người quản lý dễ dàng kiểm soát các sản phẩm, theo dõi khách hàng mua sắm, và thống kê doanh thu của cửa hàng. Khách hàng cũng sẽ hưởng lợi từ việc mua sắm trực tuyến mà không cần đến cửa hàng, tiết kiệm thời gian và công sức.

#### 3.1.1.1 Bộ phận quản lý:

Bao gồm các nghiệp vụ sau (người dùng phải đăng nhập bằng tài khoản và mật khẩu để sử dụng các chức năng, và hệ thống sẽ phân quyền cho nhân viên để thực hiện các chức năng khác nhau; chỉ người quản lý chính mới có đầy đủ quyền kiểm soát):

**1. Xem sản phẩm danh sách sản phẩm mới và danh sách sản phẩm bán chạy**

Mô tả: Người bán có thể quản lý và kiểm tra danh sách các sản phẩm mới và các sản phẩm bán chạy.

Chi tiết:

- Danh sách sản phẩm mới: Hiển thị theo thứ tự thời gian thêm vào hệ thống.

- Danh sách sản phẩm bán chạy: Hiển thị theo thứ tự doanh số bán hàng.

- Giao diện: Trang quản trị sản phẩm, có bộ lọc để chuyển đổi giữa "Mới nhất" và "Bán chạy".

**2. Xem tình trạng sản phẩm còn hàng hay hết hàng**

Mô tả: Người bán có thể kiểm tra và cập nhật tình trạng còn hàng của sản phẩm.

Chi tiết:

- Cập nhật tình trạng sản phẩm: Thay đổi số lượng tồn kho.

- Giao diện: Trang quản trị sản phẩm, cho phép cập nhật số lượng tồn kho.

**3. Cập nhật giá mới cho sản phẩm**

Mô tả: Người bán có thể cập nhật giá mới cho sản phẩm.

Chi tiết:

- Giao diện: Trang quản trị sản phẩm, cho phép thay đổi giá sản phẩm hiện tại và lưu lại.

**4. Nhập loại sản phẩm**

Mô tả: Người bán có thể thêm loại sản phẩm mới vào hệ thống.

Chi tiết:

- Giao diện: Trang quản trị loại sản phẩm, cho phép thêm, chỉnh sửa và xóa loại sản phẩm.

**5. Nhập nhà cung cấp**

Mô tả: Người bán có thể thêm nhà cung cấp mới vào hệ thống.

Chi tiết:

- Giao diện: Trang quản trị nhà cung cấp, cho phép thêm, chỉnh sửa và xóa thông tin nhà cung cấp.

**6. Nhập sản phẩm mới**

Mô tả: Người bán có thể thêm sản phẩm mới vào hệ thống.

Chi tiết:

- Giao diện: Trang quản trị sản phẩm, cho phép nhập thông tin chi tiết sản phẩm mới như tên, mô tả, giá, số lượng, loại sản phẩm và nhà sản xuất.

**7. Xem danh sách người liên hệ**

Mô tả: Người bán có thể xem danh sách các liên hệ từ khách hàng (ví dụ: hỗ trợ khách hàng, yêu cầu thông tin).

Chi tiết:

- Giao diện: Trang quản trị liên hệ, hiển thị danh sách các liên hệ cùng thông tin chi tiết như tên, email, nội dung liên hệ.

**8. Quản lý đơn hàng**

Mô tả: Người bán có thể xem và quản lý tất cả các đơn hàng của cửa hàng.

Chi tiết:

- Giao diện: Trang quản trị đơn hàng, hiển thị danh sách các đơn hàng, cho phép cập nhật tình trạng đơn hàng (đang xử lý, đã xác nhận, đã hủy) và xem chi tiết từng đơn hàng.

#### 3.1.1.2 Khách hàng

**1. Tìm kiếm sản phẩm**

Mô tả: Người mua có thể tìm kiếm sản phẩm dựa trên tên sản phẩm, tên loại, tên nhà cung cấp.

Chi tiết:

- Giao diện: Thanh tìm kiếm trên trang chủ hoặc trang danh mục sản phẩm.

- Chức năng:

+ Nhập tên hãng sản xuất vào thanh tìm kiếm.

+ Hiển thị danh sách sản phẩm thuộc hãng sản xuất đó.

**2. Xem sản phẩm**

Mô tả: Người mua có thể xem danh sách các sản phẩm có trên trang web.

Chi tiết:

- Giao diện: Trang danh mục sản phẩm.

- Chức năng: Hiển thị các sản phẩm theo danh mục, bộ lọc và sắp xếp (giá, tên, đánh giá, mới nhất, bán chạy).

**3. Chọn sản phẩm và xem chi tiết sản phẩm**

Mô tả: Người mua có thể chọn một sản phẩm cụ thể để xem chi tiết thông tin về sản phẩm đó.

Chi tiết:

- Giao diện: Trang chi tiết sản phẩm.

- Chức năng: Hiển thị thông tin chi tiết về sản phẩm như tên, giá, mô tả, thông số kỹ thuật, hình ảnh, đánh giá khách hàng, trạng thái còn hàng.

**4. Xem quảng cáo**

Mô tả: Người mua có thể xem các quảng cáo xuất hiện trên trang web.

Chi tiết:

- Giao diện: Banner quảng cáo trên trang chủ, trang danh mục sản phẩm và trang chi tiết sản phẩm.

- Chức năng: Hiển thị các quảng cáo khuyến mãi, sản phẩm nổi bật hoặc sự kiện đặc biệt.

**5. Đặt mua sản phẩm**

Mô tả: Người mua có thể đặt mua sản phẩm mà họ muốn.

Chi tiết:

- Giao diện: Trang chi tiết sản phẩm và trang giỏ hàng.

- Chức năng:

+ Nhấn nút "Thêm vào giỏ hàng" trên trang chi tiết sản phẩm.

+ Thực hiện các bước đặt hàng từ giỏ hàng.

**6. Thanh toán trực tuyến sản phẩm**

Mô tả: Người mua có thể thực hiện thanh toán trực tuyến cho sản phẩm đã đặt mua.

Chi tiết:

- Giao diện: Trang thanh toán.

- Chức năng:

+ Chọn phương thức thanh toán (thẻ tín dụng, thẻ ghi nợ, ví điện tử, ngân hàng trực tuyến).

+ Nhập thông tin thanh toán và xác nhận thanh toán.

**7. Xem giỏ hàng đã đặt mua những gì**

Mô tả: Người mua có thể xem danh sách các sản phẩm trong giỏ hàng của mình.

Chi tiết:

- Giao diện: Trang giỏ hàng.

- Chức năng: Hiển thị các sản phẩm đã thêm vào giỏ hàng, bao gồm tên sản phẩm, giá, số lượng và tổng tiền.

**8. Cập nhật giỏ hàng (Thêm, xóa, cập nhật, xóa tất cả trong giỏ hàng)**

Mô tả: Người mua có thể thêm, xóa, cập nhật số lượng hoặc xóa tất cả sản phẩm trong giỏ hàng.

Chi tiết:

- Giao diện: Trang giỏ hàng.

- Chức năng:

+ Thêm sản phẩm: Nhấn nút "Thêm vào giỏ hàng" từ trang chi tiết sản phẩm.

+ Xóa sản phẩm: Nhấn nút "Xóa" bên cạnh sản phẩm trong giỏ hàng.

+ Cập nhật số lượng: Thay đổi số lượng và nhấn nút "Cập nhật".

+ Xóa tất cả: Nhấn nút "Xóa tất cả" để làm trống giỏ hàng.

**9. Đặt hàng (chức năng này cần phải đăng nhập trước)**

Mô tả: Người mua có thể đặt hàng sau khi đăng nhập vào tài khoản.

Chi tiết:

- Giao diện: Trang giỏ hàng và trang thanh toán.

- Chức năng:

+ Yêu cầu người dùng đăng nhập nếu chưa đăng nhập.

+Sau khi đăng nhập, tiếp tục quá trình đặt hàng và thanh toán.

**10. Đăng ký tài khoản**

Mô tả: Người mua có thể tạo tài khoản mới trên trang web.

Chi tiết:

- Giao diện: Trang đăng ký.

- Chức năng:

+ Nhập thông tin cá nhân (tên, email, mật khẩu, số điện thoại, địa chỉ).

+ Xác nhận và tạo tài khoản.

**11. Đăng nhập tài khoản**

Mô tả: Người mua có thể đăng nhập vào tài khoản của mình.

Chi tiết:

- Giao diện: Trang đăng nhập.

- Chức năng:

+ Nhập email và mật khẩu.

+ Xác nhận và đăng nhập vào hệ thống.

**12. Liên hệ với công ty thông qua trang liên hệ**

Mô tả: Người mua có thể liên hệ với công ty để yêu cầu hỗ trợ hoặc thông tin.

Chi tiết:

- Giao diện: Trang liên hệ.

- Chức năng:

+ Nhập thông tin liên hệ (tên, email, nội dung liên hệ).

+Gửi yêu cầu liên hệ.

**Các chức năng chính của hệ thống đối với quản trị (Admin) như sau:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Loại** |
| 1 | Đăng nhập hệ thống để quản lý | Hiển thị |
| 2 | Quản lý sản phẩm | Hiển thị |
| 3 | Nhập sản phẩm mới | Hiển thị |
| 4 | Nhập loại sản phẩm | Hiển thị |
| 5 | Thống kê | Hiển thị |
| 6 | Nhập nhà cung cấp | Hiển thị |
| 7 | Nhập loại khách hàng | Hiển thị |
| 8 | Xem danh sách tài khoản khách hàng | Hiển thị |
| 9 | Quản lý đơn hàng | Hiển thị |

**Các chức năng chính của hệ thống đối với khách hàng như sau:**

|  |  |  |
| --- | --- | --- |
| **Quy tắc** | **Chức năng** | **Loại** |
| 1 | Xem sản phẩm | Hiển thị |
| 2 | Đặt hàng | Hiển thị |
| 3 | Xem giỏ hàng | Hiển thị |
| 4 | Gửi thông tin liên hệ | Hiển thị |
| 5 | Thanh toán trực tuyến | Hiển thị |
| 6 | Quản lý xem giỏ hàng (thêm, xóa, sửa….) | Hiển thị |
| 8 | Đăng ký | Hiển thị |
| 9 | Đăng nhập | Hiển thị |
| 10 | Xem thông tin cá nhân | Hiển thị |
| 12 | Đổi mật khẩu | Hiển thị |
| 13 | Đặt hàng | Hiển thị |

### Sơ đồ luồng dữ liệu:

#### Đăng ký tài khoản (User):

|  |  |
| --- | --- |
|  | D1: Thông tin khách hàng cần đăng ký: Họ tên, Email, Số điện thoại, Địa chỉ, Mật khẩu.  D3: Các danh mục tài khoản cần lưu trữ  D4: Thông tin tài khoản đã được đăng ký  D5: Thiết bị xuất (màn hình)  D6: Thông tin khách hàng đăng ký tài khoản thành công |

#### Đăng nhập tài khoản (User, Admin):

|  |  |
| --- | --- |
|  | D1: Email, mật khẩu cần đăng nhập  D3: Các danh mục tài khoản cần lưu trữ  D4: Thông tin tài khoản đã đăng nhập  D5: Thiết bị xuất (màn hình)  D6: Thông tin đăng nhập tài khoản thành công |

#### Tìm kiếm sản phẩm (User, Admin):

|  |  |
| --- | --- |
|  | D1: Thông tin sản phẩm cần tìm kiếm theo: tên sản phẩm, tên loại, tên nhà cung cấp  D3: Danh sách thông tin sản phẩm  D4: Kết quả thông tin sản phẩm  D5: Thiết bị xuất (màn hình)  D6: Trả về thông tin sản phẩm cần tìm |

#### Cập nhật giỏ hàng (User):

|  |  |
| --- | --- |
|  | D1: Thông tin sản phẩm và số lượng sản phẩm cần thêm, sửa, xóa  D3: Các danh mục cần lưu trữ trong giỏ hàng  D4: Thông tin giỏ hàng đã được cập nhật  D5: Thiết bị xuất (màn hình)  D6: Thông tin giỏ hàng cập nhật hiện thành công |

#### Đặt hàng (User):

|  |  |
| --- | --- |
|  | D1: Thông tin về giỏ hàng cần xử lý  D3: Dữ liệu cần thiết cho việc tính thành tiền  D4: Kết quả tính thành tiền  D5: Dữ liệu được kết xuất (bao gồm D1, D3, D4)  D6: Thông tin đặt hàng thành công và trả về D1, D3, D4 |

#### Cập nhật thông tin tài khoản cá nhân (User):

|  |  |
| --- | --- |
|  | D1: Thông tin tài khoản cá nhân cần thêm, sửa, xóa  D3: Các danh mục cần lưu trữ trong kho tài khoản  D4: Thông tin tài khoản cá nhân đã được cập nhật  D5: Thiết bị xuất (màn hình)  D6: Thông tin tài khoản cập nhật hiện thành công |

#### Cập nhật sản phẩm (Admin):

|  |  |
| --- | --- |
|  | D1: Thông tin sản phẩm cần thêm, sửa, xóa  D3: Các danh mục cần lưu trữ trong kho sản phẩm  D4: Thông tin sản phẩm đã được cập nhật  D5: Thiết bị xuất (màn hình)  D6: Thông tin sản phẩm cập nhật hiện thành công |

#### Cập nhật nhà cung cấp (Admin):

|  |  |
| --- | --- |
|  | D1: Thông tin nhà cung cấp cần thêm, sửa, xóa  D3: Các danh mục cần lưu trữ trong kho nhà cung cấp  D4: Thông tin nhà cung cấp đã được cập nhật  D5: Thiết bị xuất (màn hình)  D6: Thông tin nhà cung cấp cập nhật hiện thành công |

#### Cập nhật loại hàng (Admin):

|  |  |
| --- | --- |
|  | D1: Thông tin loại hàng cần thêm, sửa, xóa D3: Các danh mục cần lưu trữ trong kho loại hàng  D4: Thông tin loại hàng đã được cập nhật  D5: Thiết bị xuất (màn hình)  D6: Thông tin loại hàng cập nhật hiện thành công |

#### Cập nhật đơn hàng (Admin):

|  |  |
| --- | --- |
|  | D1: Thông tin đơn hàng cần thêm, sửa, xóa  D3: Các danh mục cần lưu trữ trong kho đơn hàng  D4: Thông tin đơn hàng đã được cập nhật  D5: Thiết bị xuất (màn hình)  D6: Thông tin đơn hàng cập nhật hiện thành công |

#### Cập nhật tài khoản (Admin):

|  |  |
| --- | --- |
|  | D1: Thông tin tài khoản cần thêm, sửa, xóa D3: Các danh mục cần lưu trữ trong kho tài khoản  D4: Thông tin tài khoản đã được cập nhật  D5: Thiết bị xuất (màn hình)  D6: Thông tin tài khoản cập nhật hiện thành công |

#### Cập nhật phân quyền (Admin):

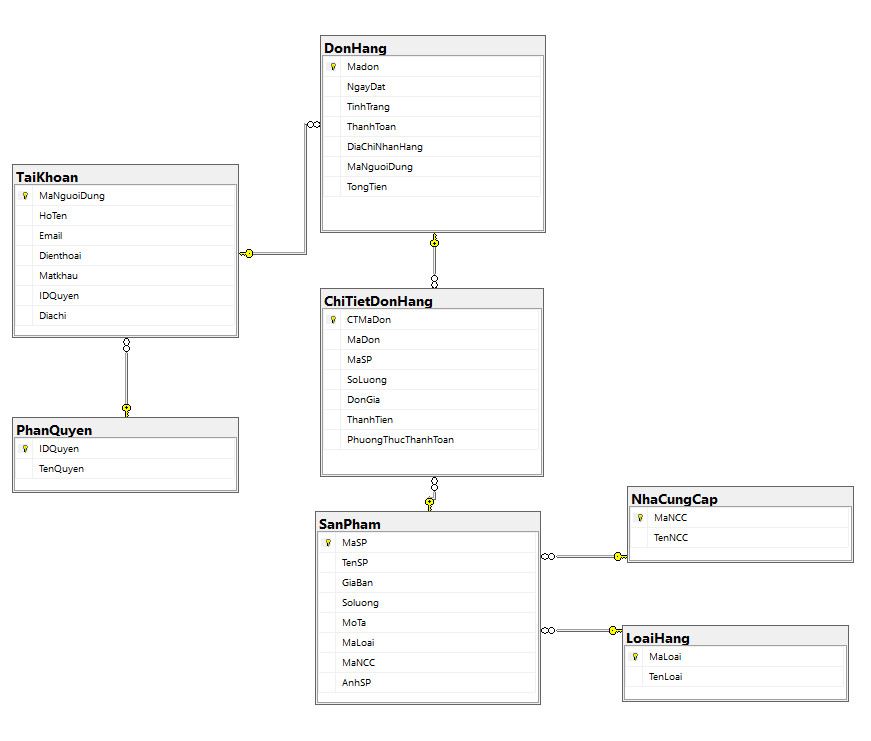
|  |  |
| --- | --- |
|  | D1: Thông tin phân quyền cần thêm, sửa, xóa D3: Các danh mục cần lưu trữ trong kho phân quyền  D4: Thông tin phân quyền đã được cập nhật  D5: Thiết bị xuất (màn hình)  D6: Thông tin phân quyền cập nhật hiện thành công |

#### Thống kê (Admin):

|  |  |
| --- | --- |
|  | D1: Thông tin cần thống kê: theo doanh thu, theo số lượng khách hàng, …  D3: Danh sách các đơn hàng  D4: D1  D5: D4  D6: Thông tin thông kê được |

## Thiết kế dữ liệu:

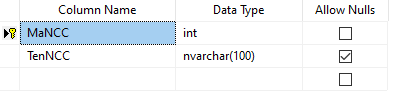
### Mô hình vật lý:



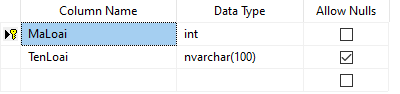
Hình 3.7: Mô hình CSDL

### Mô tả chi tiết các bảng:

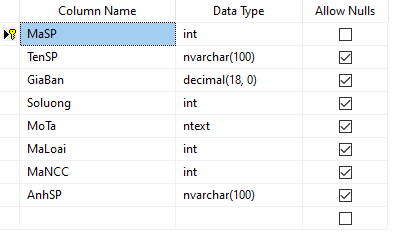
#### Bảng NhaCungCap:



#### Bảng LoaiHang:



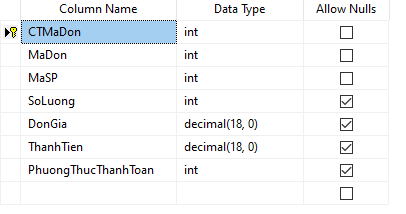
#### Bảng SanPham:



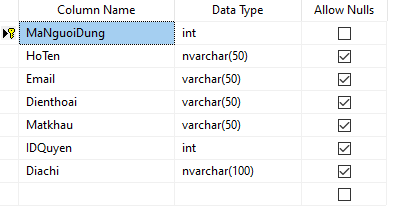
#### Bảng DonHang:



#### Bảng ChiTietDonHang:



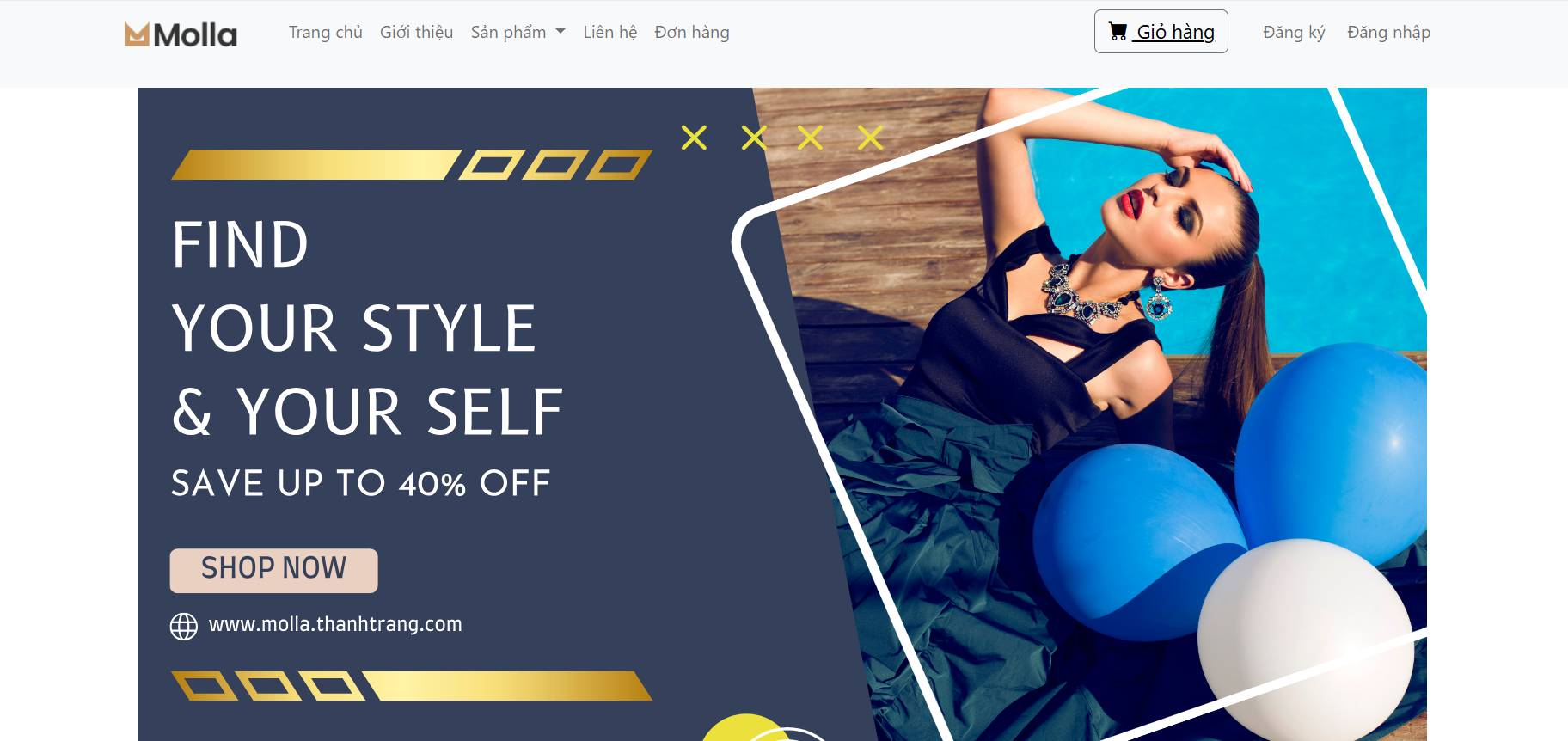
#### Bảng TaiKhoan:



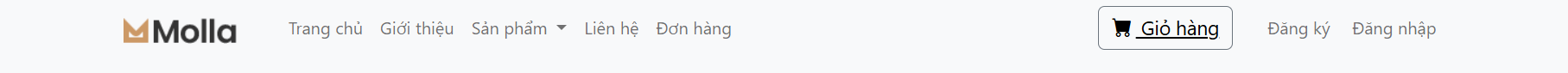
# CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

## Giao diện User:

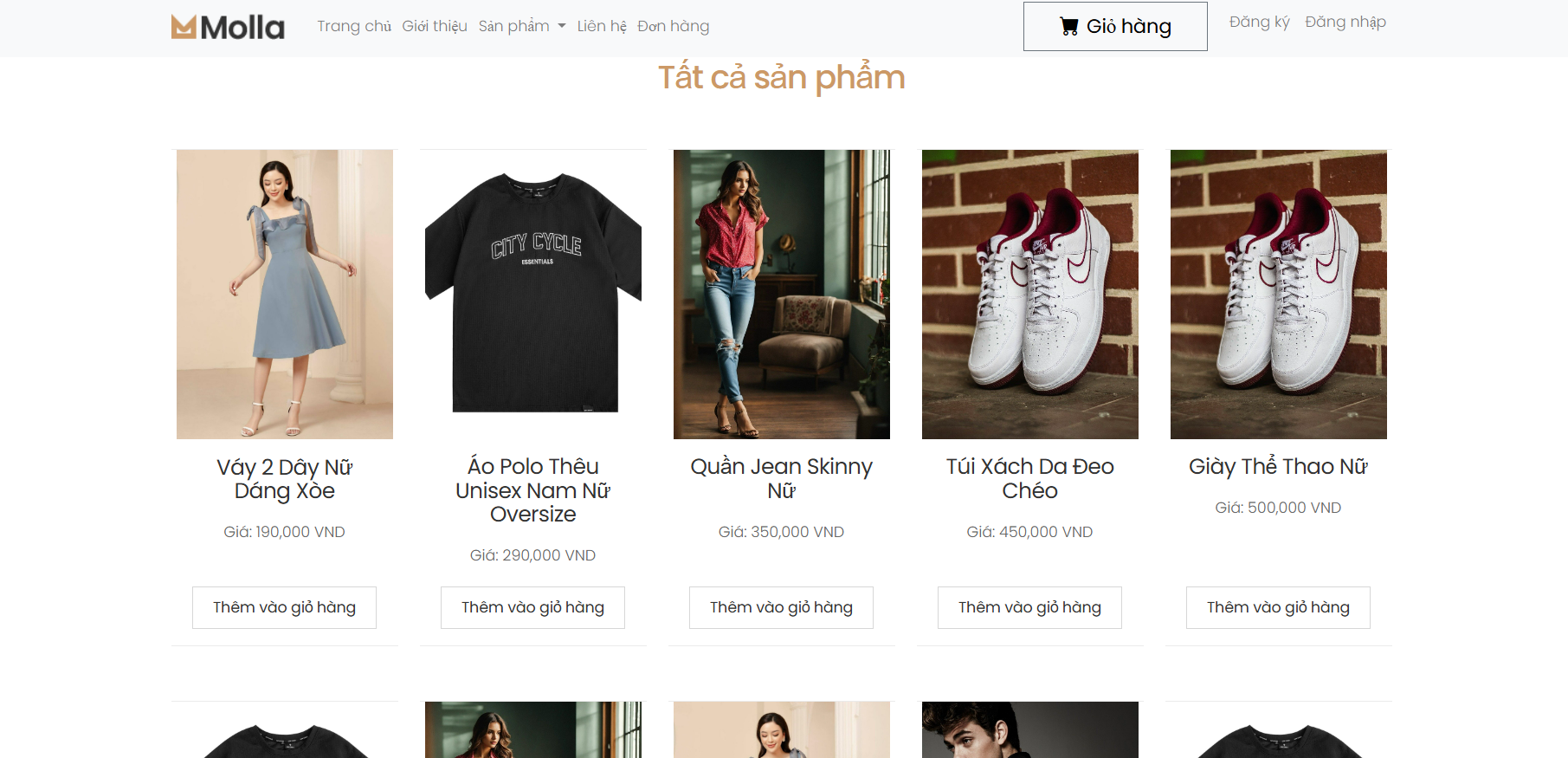
### Giao diện trang chủ:



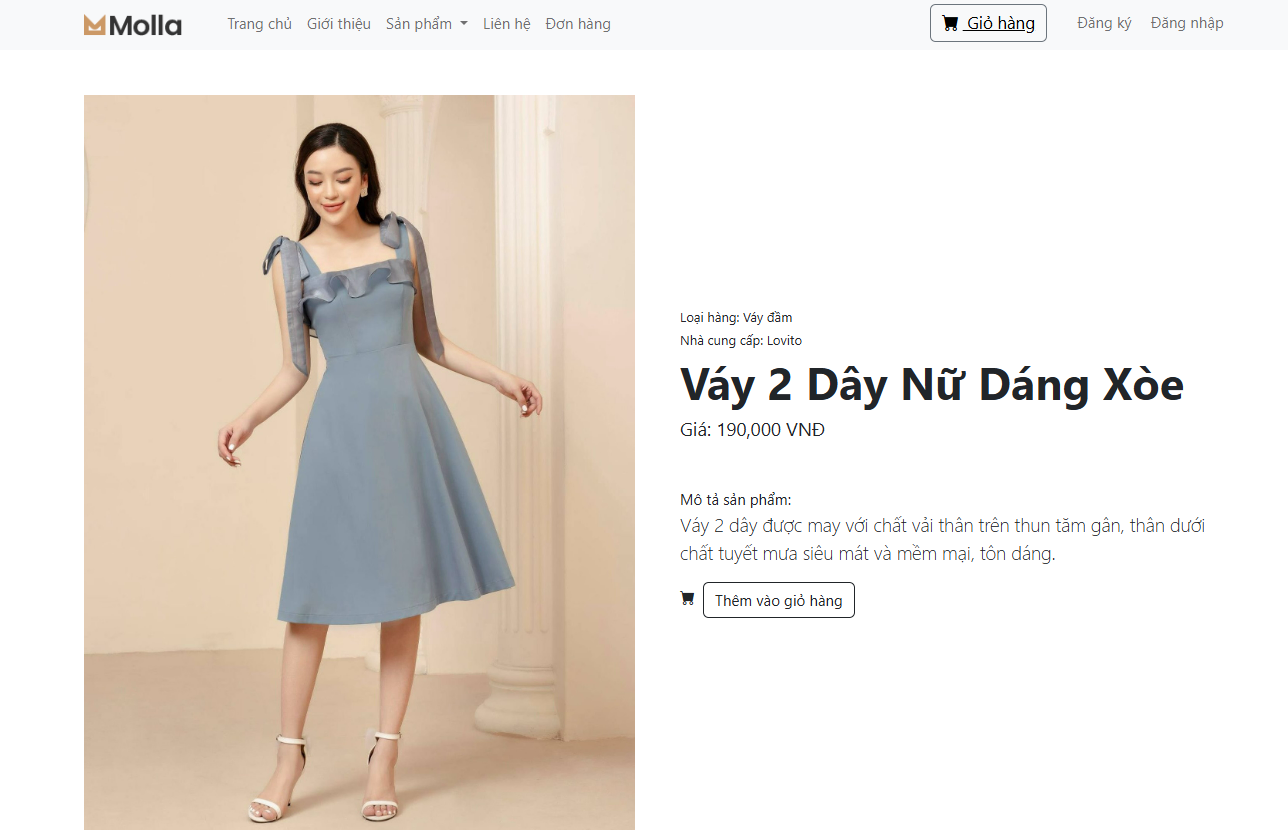
### Giao diện thanh Menu:



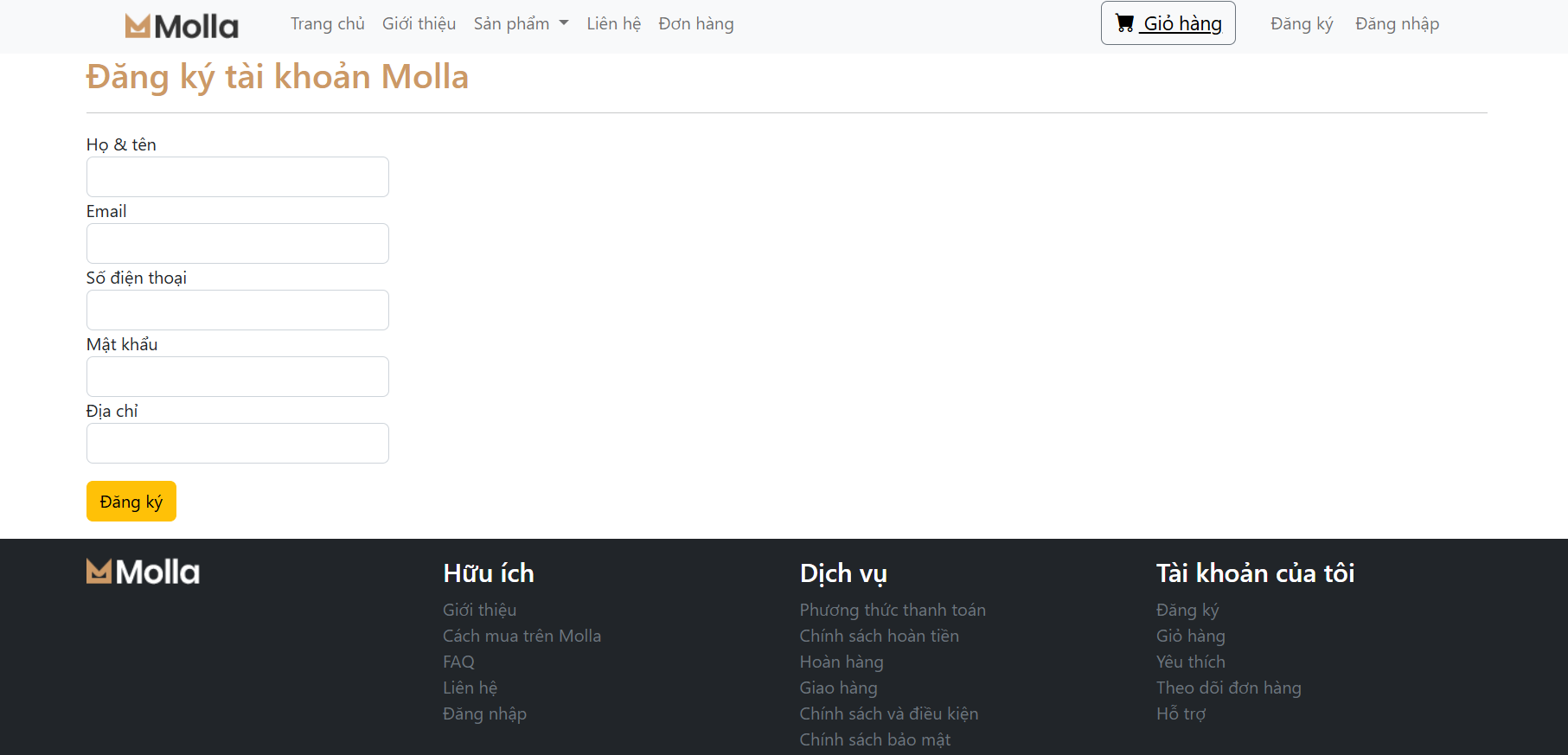
### Giao diện trang sản phẩm:



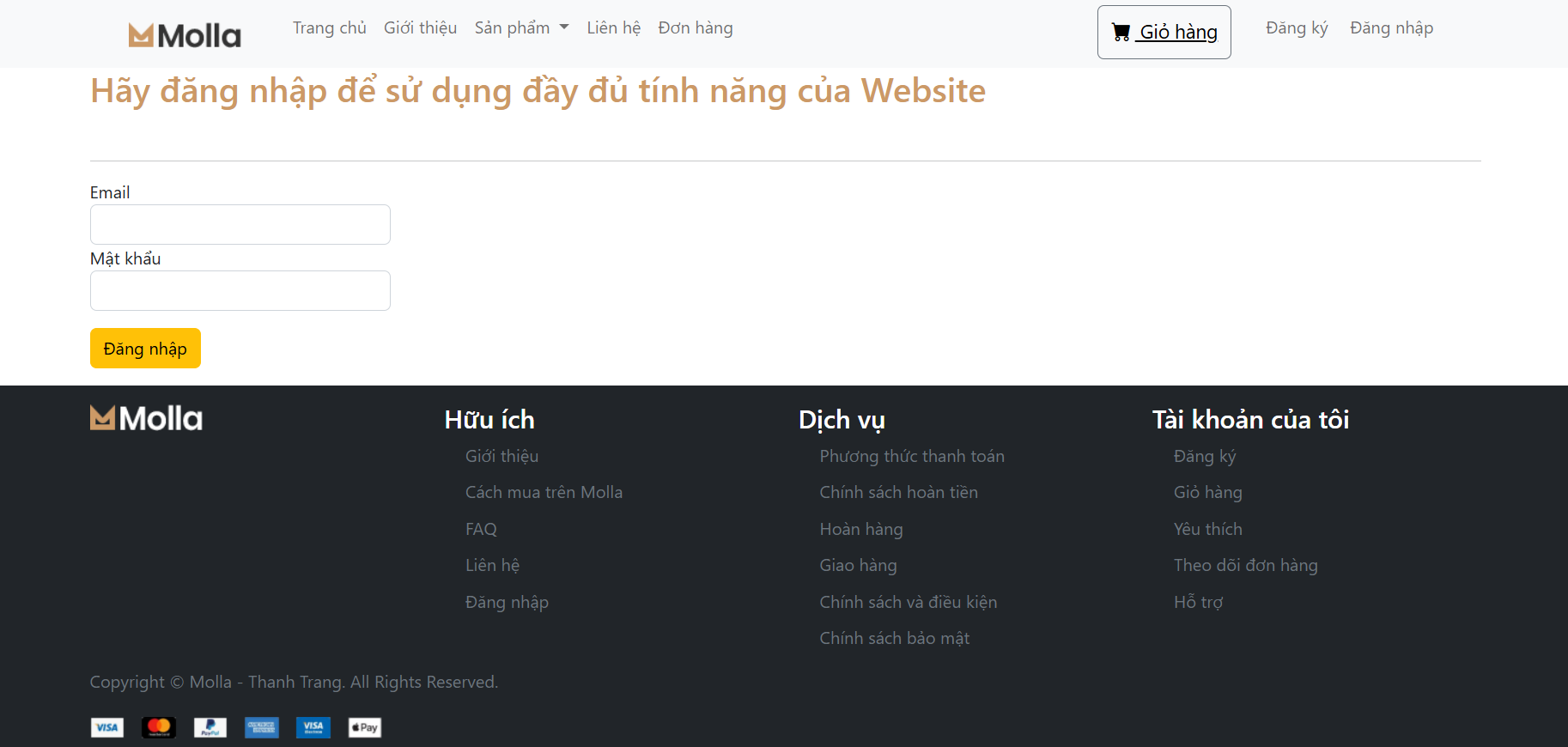
### Giao diện trang chi tiết sản phẩm:



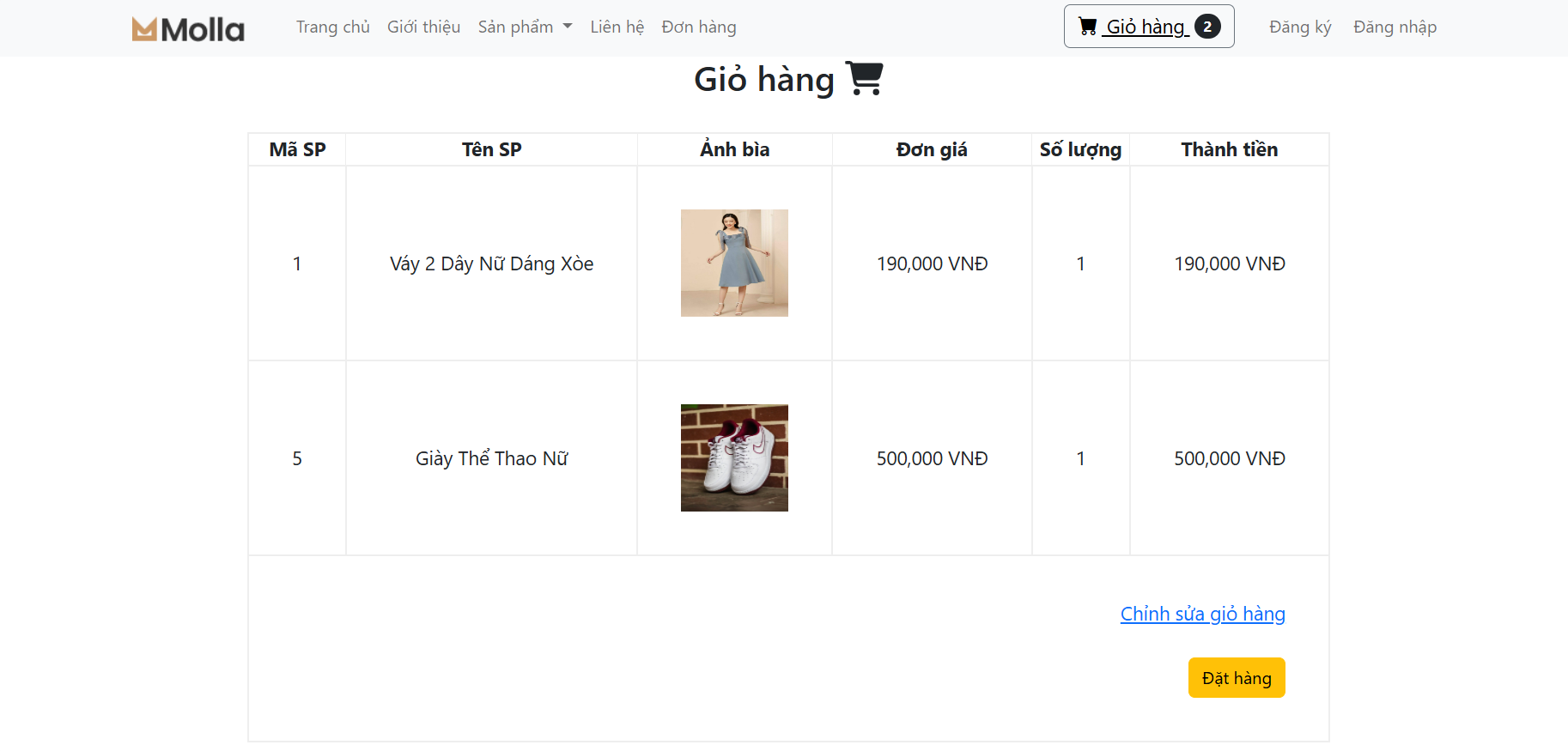
### Giao diện trang đăng ký tài khoản:



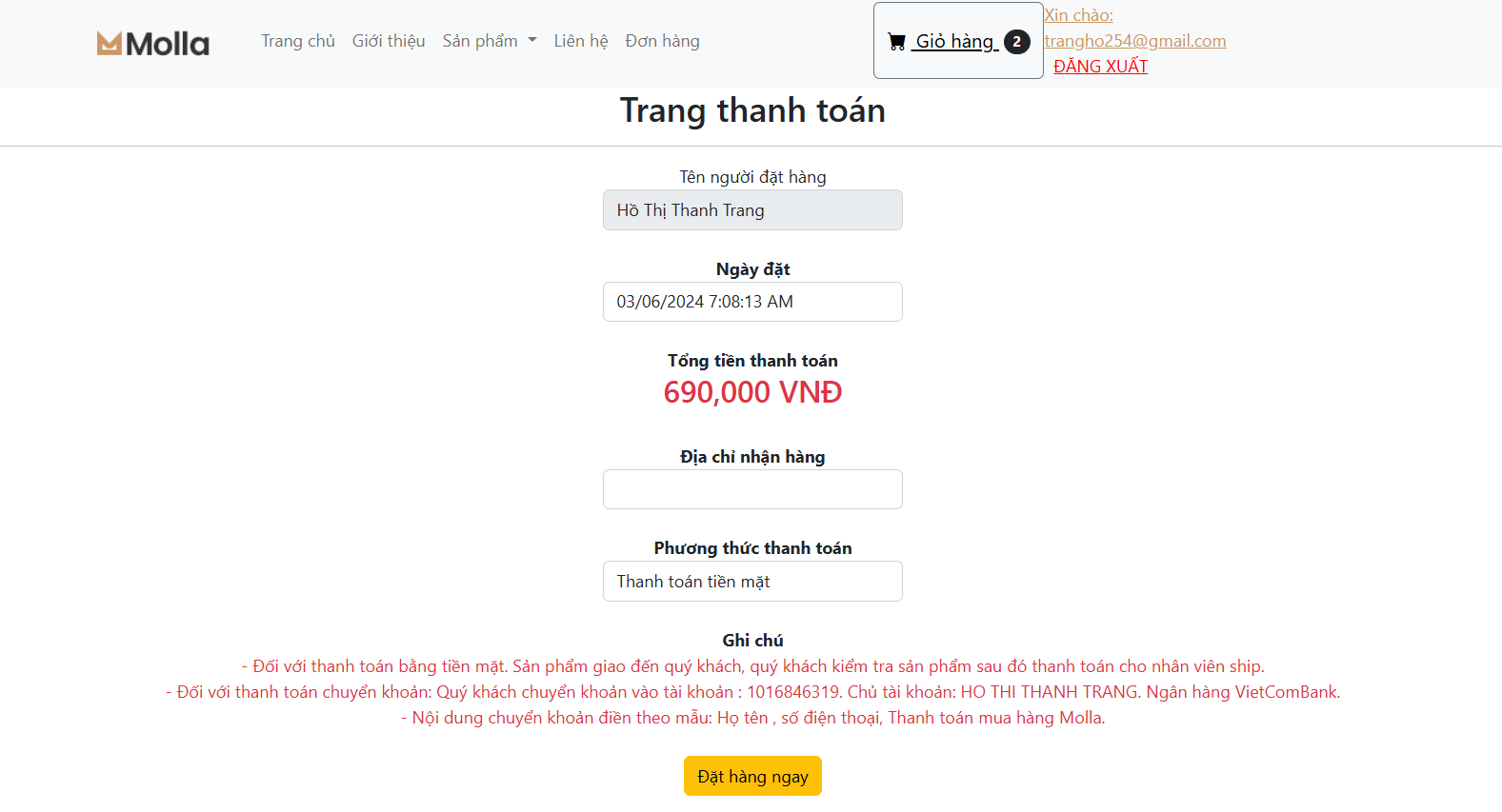
### Giao diện trang đăng nhập:



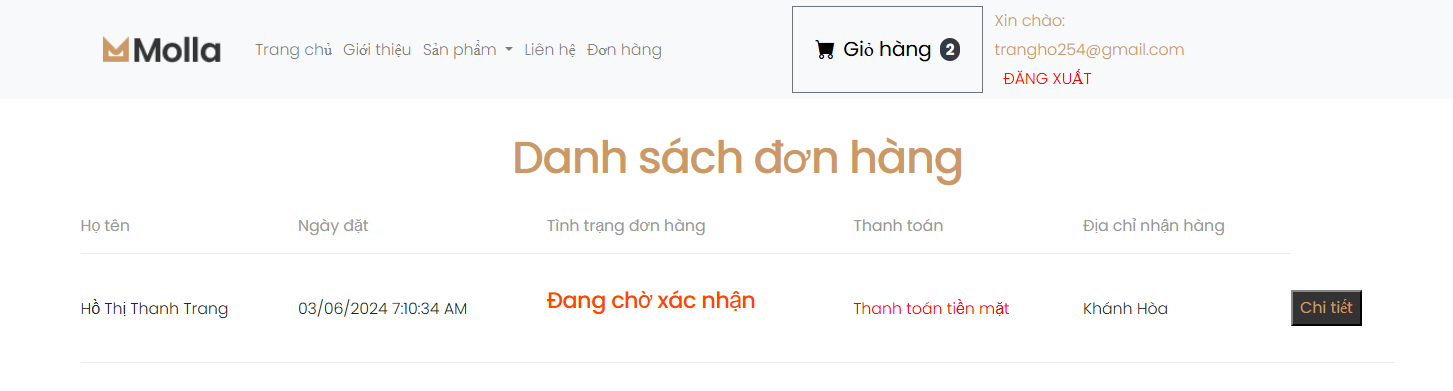
### Giao diện trang giỏ hàng:



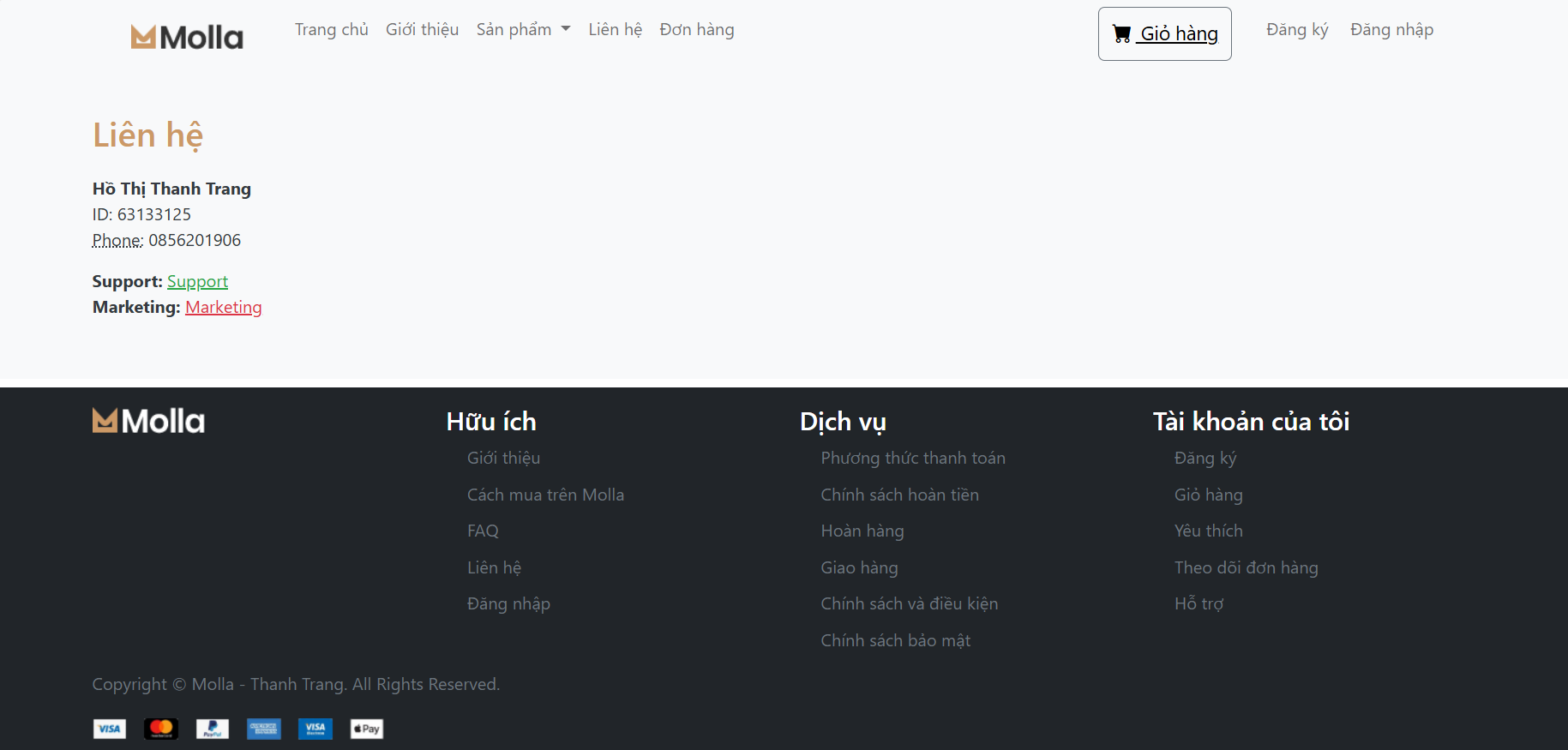
### Giao diện trang xác nhận đặt hàng:



### Giao diện trang xác nhận đã đặt hàng:



### Giao diện trang liên hệ:

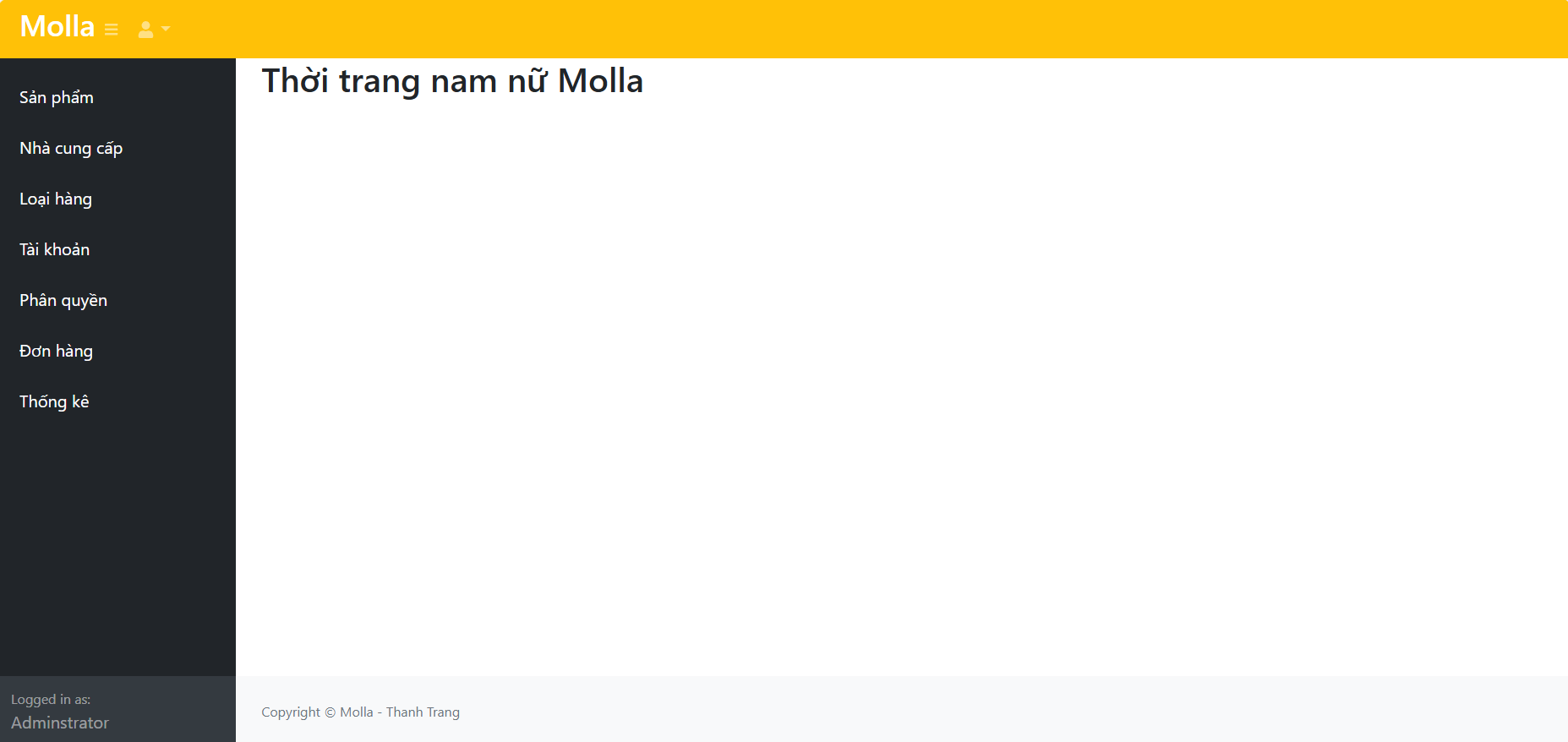


### Giao diện trang giới thiệu:

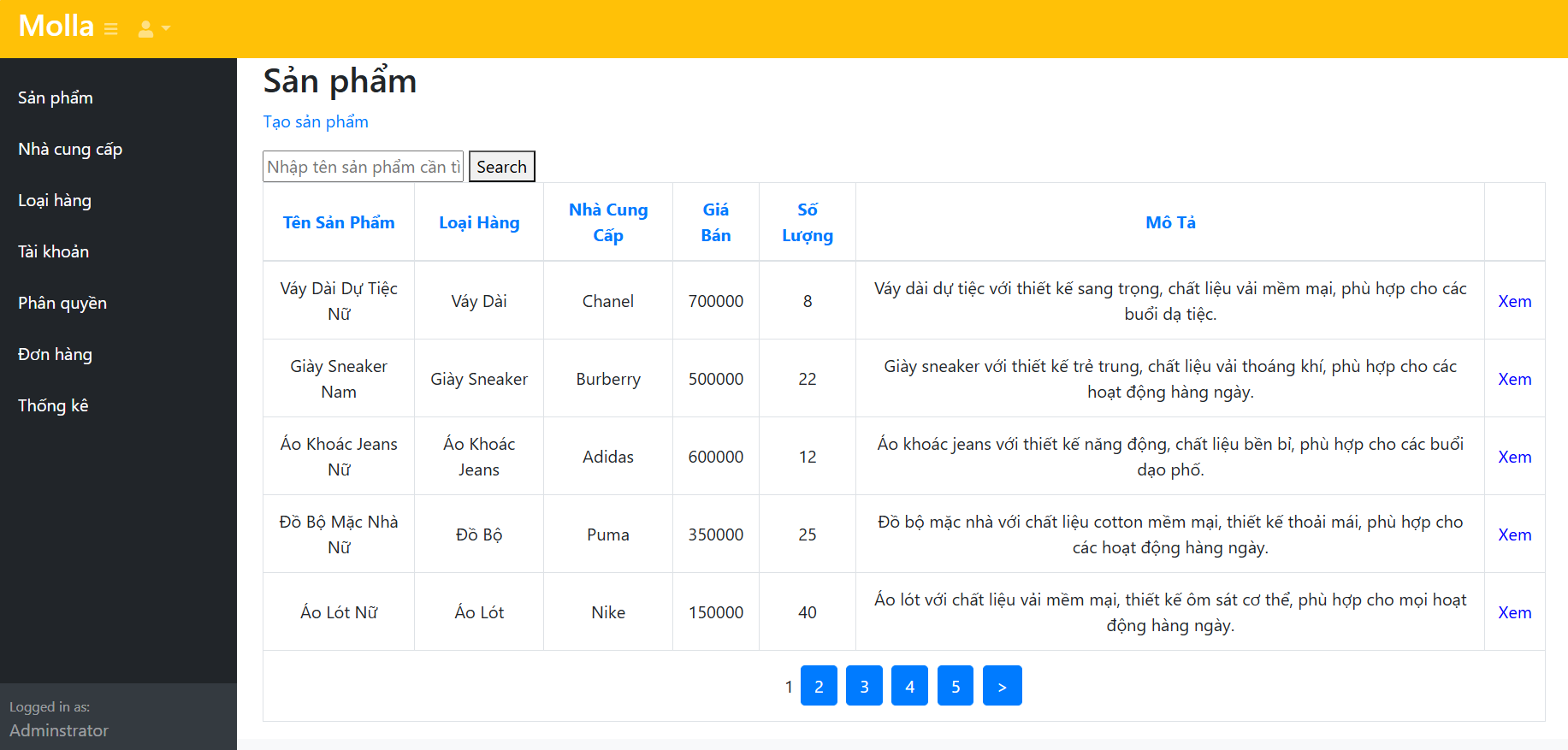


## Giao diện Admin:

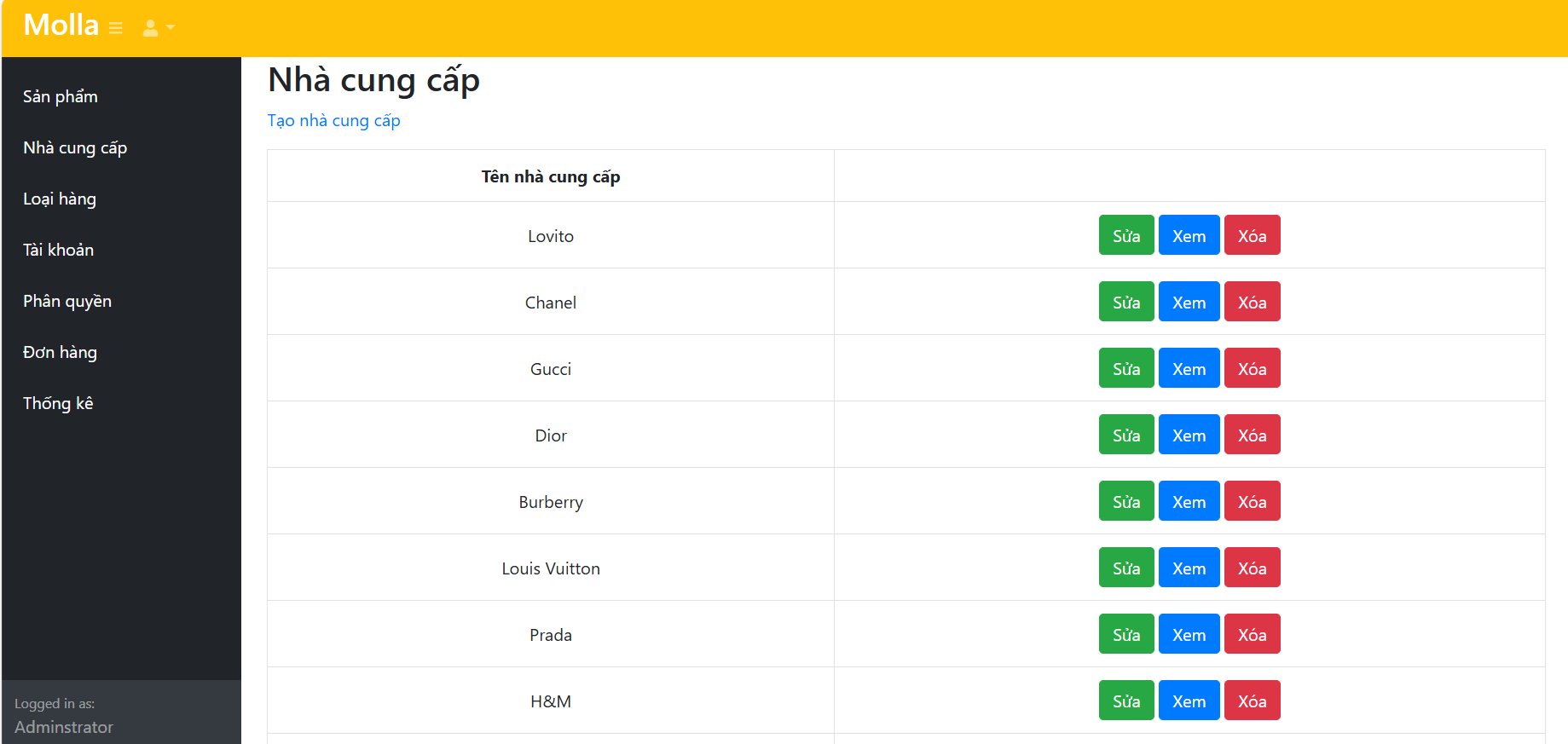
### Giao diện trang chủ Admin:



### Giao diện trang quản lý sản phẩm:



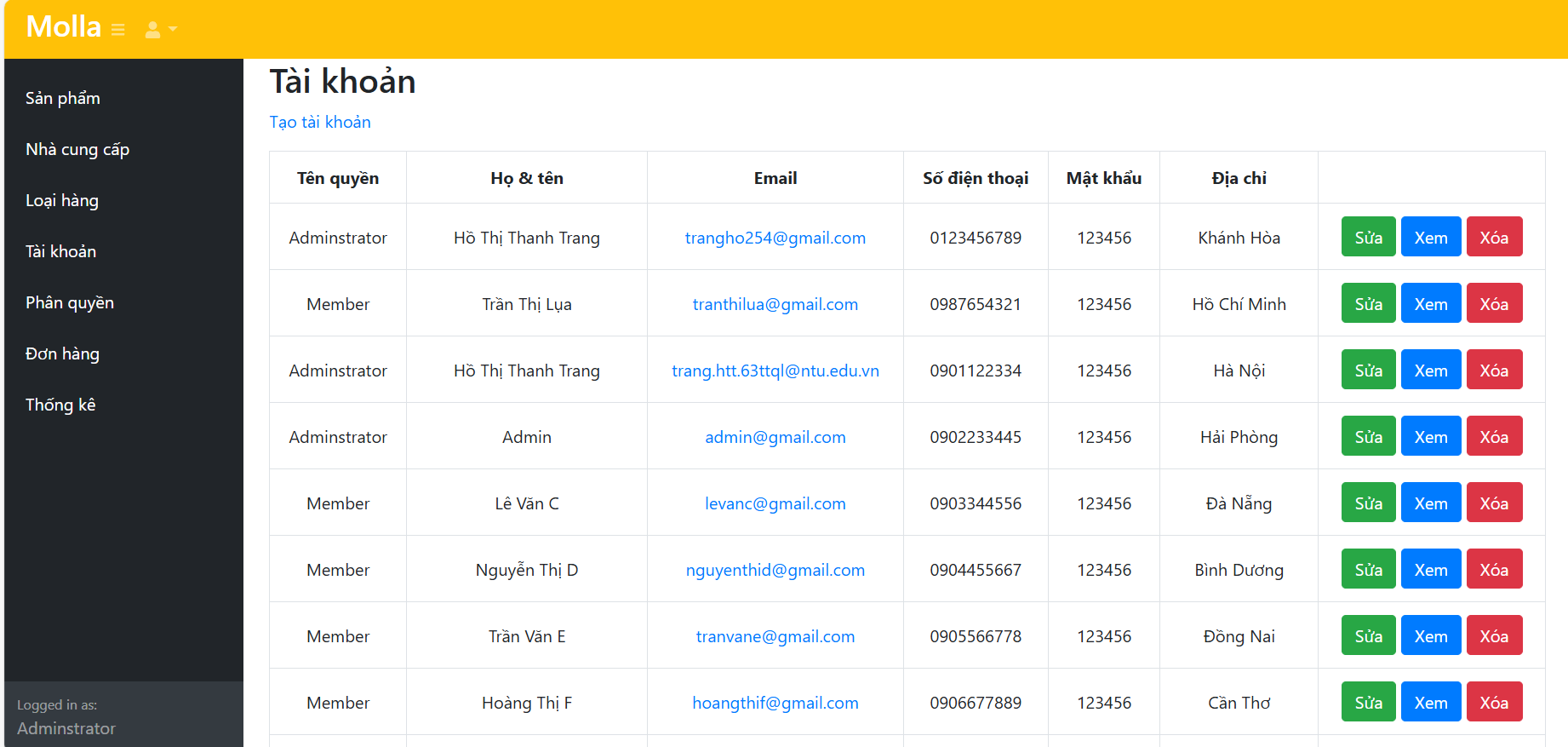
### Giao diện trang quản lý nhà cung cấp:



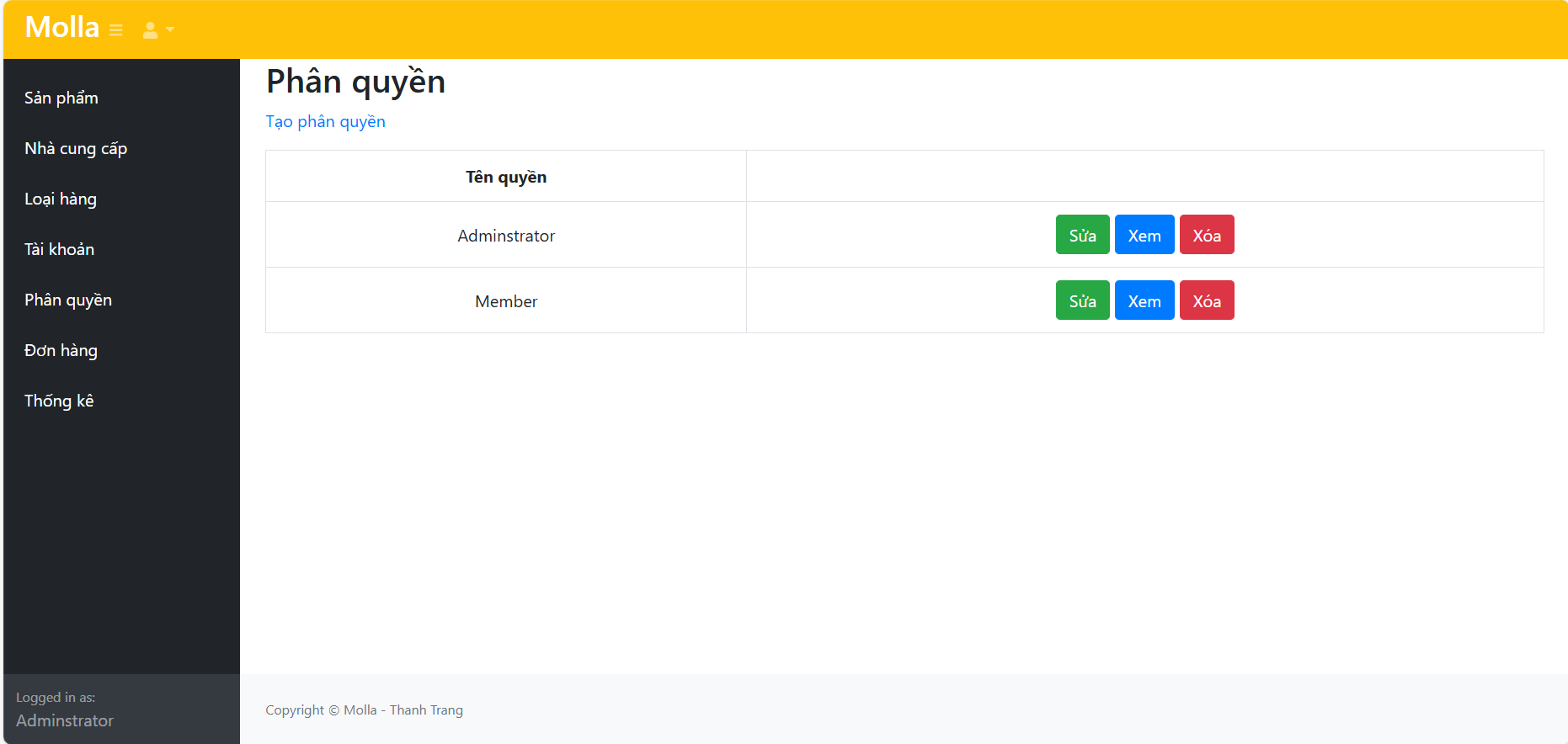
### Giao diện trang quản lý loại hàng:



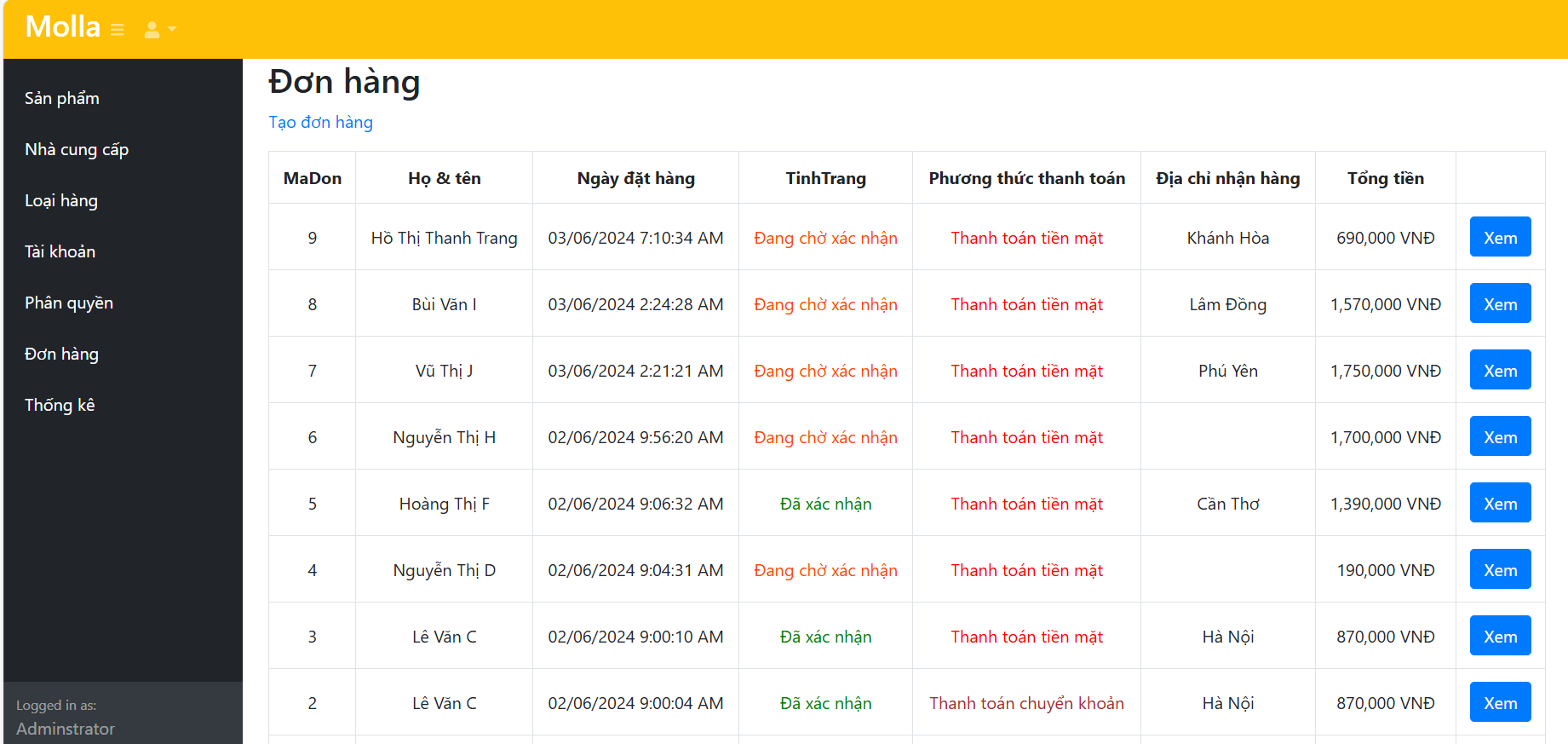
### Giao diện trang quản lý tài khoản:



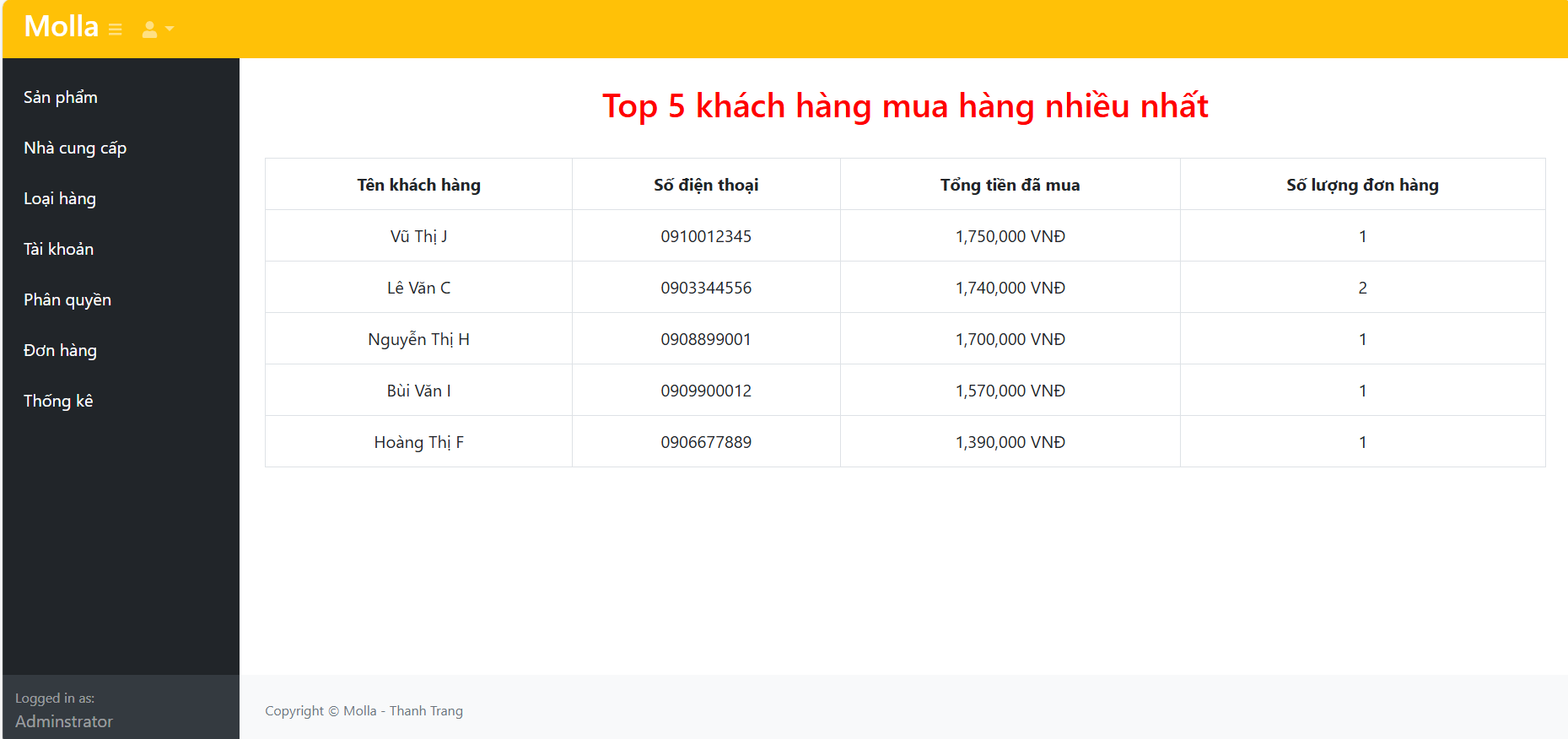
### Giao diện trang quản lý phân quyền:



### Giao diện trang quản lý đơn hàng:



### Giao diện trang thống kê:

****

# CHƯƠNG 5: TỔNG KẾT.

## Kết quả đạt được:

Mở rộng hiểu biết về ứng dụng web sử dụng công nghệ ASP.NET. Đồng thời nắm bắt được cách ứng dụng ngôn ngữ lập trình C# vào web thông qua Microsoft ASP.NET MVC5 và học thêm một số ngôn ngữ, công nghệ khác như HTML5, CSS3, jQuery, AJAX, …

Hiểu biết về ngôn ngữ truy vấn cơ sở dữ liệu LINQ, một ngôn ngữ truy vấn theo dạng hướng đối tượng, giúp lập trình viên dễ dàng tương tác với các cơ sở dữ liệu khác nhau như XML, cơ sở dữ liệu quan hệ, …

Dựa trên nền tảng kiến thức đã học, em đã phát thảo cơ bản một ứng dụng web cơ bản phục vụ cho nhu cầu học tập. Tuy rằng ứng dụng web này còn khá đơn giản, nội dung chưa đầy đủ các chức năng cần thiết, giao diện chưa tối ưu, giải thuật và xử lý lỗi còn hạn chế, nhưng đây sẽ là nền tảng để có thể phát triển và xây dựng các ứng dụng web thực tế hơn trong tương lai.

## Hoàn thành:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Chức năng** | **Mức độ hoàn thành** | **Ghi chú** |
| 1 | Đăng ký | Hoàn thành |  |
| 2 | Đăng nhập | Hoàn thành |  |
| 3 | Xem sản phẩm trừng bày, xem chi tiết sản phẩm | Hoàn thành |  |
| 4 | Đặt hàng, thêm vào giỏ hàng | Hoàn thành |  |
| 5 | Giới thiệu về chúng tôi | Hoàn thành |  |
| 6 | Phân loại sản phẩm | Hoàn thành |  |
| 7 | Trang chủ hiển thị các loại sản phẩm | Hoàn thành |  |
| 8 | Liên hệ với người quản lý | Hoàn thành |  |
| 9 | Quản lý sản phẩm | Hoàn thành |  |
| 10 | Quản lý nhà cung cấp | Hoàn thành |  |
| 11 | Quản lý loại hàng | Hoàn thành |  |
| 12 | Quản lý tài khoản người dùng | Hoàn thành |  |
| 13 | Phân quyền người dùng | Hoàn thành |  |
| 14 | Đơn hàng | Hoàn thành |  |
| 15 | Quản lý thống kê | Hoàn thành |  |

## Hướng phát triển:

Tại trang chủ dành cho người dùng, cần cải thiện bố cục và thiết kế để trang web trở nên thân thiện hơn với người dùng. Đặc biệt, cần nâng cao khả năng tương thích của trang web để nó hoạt động mượt mà trên cả máy tính để bàn và trình duyệt di động.

Ngoài ra, xu hướng hiện nay cho thấy hầu hết người dùng đều sử dụng tài khoản Facebook hoặc Google. Vì vậy, việc tích hợp chức năng đăng nhập bằng các tài khoản này sẽ giúp người dùng đăng nhập nhanh chóng mà không cần đăng ký tài khoản mới trên trang web.

Cần thêm một chức năng trực tiếp phản hồi trên website. Thay vì phải liên hệ qua ứng dụng thứ ba như Gmail, người dùng có thể trực tiếp nhắn tin trên website để có được phản hồi một cách nhanh chóng và tiện lợi hơn.

Thêm các công nghệ Web Service để nâng cao hiệu suất và tính năng của trang web.

## Tài liệu tham khảo:

**Tài liệu tiếng Việt:**

[1] ThS.GV Bùi Chí Thành, *Slides Bài giảng Thiết kế và lập trình Web 2*

[2] ThS.GV Lê Thị Bích Hằng, ThS.GV Bùi Chí Thành, *Slides Bài giảng Thiết kế và lập trình Web 2*

**Tài liệu Internet:**

[3]<https://learn.microsoft.com/vi-vn/aspnet/mvc/overview/getting-started/introduction/getting-started> (13/10/2023, Getting started with ASP.NET MVC 5| Microsoft Learn), truy cập ngày 02/06/2024

[4]<https://getbootstrap.com/docs/5.3/getting-started/introduction/> (Get started with Boostrap), truy cập ngày 02/06/2024