**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**~~~\*\*\*\*\*\*\*\*\*~~~**

****

**BÁO CÁO**

**ĐỒ ÁN TỐT NGHIỆP**

**Đề tài : Web bán hàng Home Shoppe**

*Sinh viên thực hiện: Đào Đức Thao (Mã SV: 171202723)*

*Lớp: CNNT1-K58*

*Giảng viên hướng dẫn:*

*Thầy Phạm Đình Phong*

*Hà Nội, ngày 20 tháng 6 năm 2021*

**LỜI CẢM ƠN**

**\*\*\***

Em chân thành cảm ơn Thầy Phạm Đình Phong trong thời gian qua đã giúp đỡ, hướng dẫn em để em có thể hoàn thành đề tài này một cách tốt nhất.

Qua đây, em cũng xin được gửi lời cảm ơn đến tất cả các thầy cô công tác tại khoa Công nghệ thông tin - Trường Đại học Giao thông vận tải đã dìu dắt, chỉ bảo cho em trong suốt quá trình học tập tại trường.

Cuối cùng, em xin gửi lời cảm ơn đến tất cả người thân, bạn bè, những người đã luôn động viên, ủng hộ và tạo điều kiện cho em trong thời gian hoàn thành đồ án.

Mặc dù em đã cố gắng rất nhiều nhưng cũng không tránh khỏi thiếu sót và hạn chế trong quá trình thực hiện. Em rất mong nhận được sự thông cảm và chỉ bảo cũng như những đóng góp chân thành của các thầy cô và bạn bè để bài đồ án được tốt hơn.

Em xin chân thành cảm ơn!

Hà Nội, Ngày 20 tháng 06 năm 2021

*Sinh viên thực hiện:*

Đào Đức Thao

**LỜI MỞ ĐẦU**

**\*\*\***

Với thời đại công nghệ 4.0 thì việc mua hàng online trở nên một cách phổ biến và phát triển đã nhanh chóng thu hút được nhiều khách hàng tin dùng và đặt hàng thường xuyên. Việc mua hàng online có nhiều điểm ưu việt so với cách mua trực tiếp như tiết kiệm được thời gian, chi phí phát sinh, xem được các đánh giá từ những người mua trước xem rằng nó có đẹp và hợp lý hay không.

Tài liệu này đưa ra các đặc tả chi tiết cho các yêu cầu phần mềm cho web bán hàng online. Website cho phép người dùng tạo tài khoản, đăng nhập vào và đặt mua các sản phẩm có trong web. Đối với người dùng có nhu cầu đặt hàng thì người dùng sẽ tìm kiếm đến loại hàng cần đặt rồi tiến hành đặt hàng. Còn đối với chủ cửa hàng thì họ sẽ thấy các đơn hàng và chấp nhận nó, tiếp sau đó là họ sẽ xử lý rồi giao mặt hàng cho khách hàng.

Website cung cấp giao diện sinh động và trực quan cho người dùng.

Nội dung

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT 6](#_Toc65422833)

[1.1 Tổng quan đề tài 6](#_Toc65422834)

[1.1.1 Lý do chọn đề tài 6](#_Toc65422835)

[1.1.2 Mục tiêu đề tài 6](#_Toc65422836)

[1.1.3 Phạm vi đề tài 6](#_Toc65422837)

[1.1.4 Đối tượng nghiên cứu và phạm vi hệ thống 6](#_Toc65422838)

[1.2 Cơ sở lý thuyết 7](#_Toc65422839)

[1.2.1 Tổng quan về C# 7](#_Toc65422840)

[1.2.2 Mô hình MVC 8](#_Toc65422841)

[1.2.3 LINQ trong C# 11](#_Toc65422842)

[1.2.4 Giới thiệu về trình quản lý tệp hình ảnh CKFinder 11](#_Toc65422843)

[1.2.5 Giới thiệu về trình soạn thảo mã nguồn mở CKEditor 14](#_Toc65422844)

[1.2.6 Giới thiệu về API 16](#_Toc65422845)

[1.2.7 Giới thiệu về Google Map API 17](#_Toc65422846)

[1.2.8 Giới thiệu về Facebook API 18](#_Toc65422847)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 20](#_Toc65422848)

[2.1 Tìm hiểu nghiệp vụ 20](#_Toc65422849)

[2.2 Xác định người dùng 20](#_Toc65422850)

[2.3 Chức năng 20](#_Toc65422851)

[2.4 Sơ đồ phân ra chức năng 21](#_Toc65422852)

[2.5 Thiết kế cơ sở dữ liệu 21](#_Toc65422853)

[2.5.1 Cấu trúc cơ sở dữ liệu 21](#_Toc65422854)

[2.5.2 Danh sách các bảng 22](#_Toc65422855)

[CHƯƠNG 3 XÂY DỰNG CHƯƠNG TRÌNH 29](#_Toc65422856)

[3.1 Xây dựng chức năng và giao diện cho người dùng 29](#_Toc65422857)

[3.1.1 Chức năng đăng kí 29](#_Toc65422858)

[3.1.2 Chức năng đăng nhập 30](#_Toc65422859)

[3.1.3 Chức năng tìm kiếm sản phẩm 30](#_Toc65422860)

[3.1.4 Chức năng gửi phản hồi 31](#_Toc65422861)

[3.1.5 Chức năng giỏ hàng 32](#_Toc65422862)

[3.1.6 Chức năng thanh toán 33](#_Toc65422863)

[3.1.7 Giao diện trang chủ 34](#_Toc65422864)

[3.1.8 Giao diện trang chi tiết sản phẩm 35](#_Toc65422865)

[3.1.9 Giao diện trang tin tức 35](#_Toc65422866)

[3.1.10 Giao diện trang liên hệ 36](#_Toc65422867)

[3.2 Xây dựng chức năng cho quản trị viên 36](#_Toc65422868)

[3.2.1 Chức năng đăng nhập của người quản trị 36](#_Toc65422869)

[3.2.2 Chức năng thêm, sửa, xóa, tìm kiếm sản phẩm 40](#_Toc65422870)

[3.2.3 Chức năng phân quyền tài khoản 40](#_Toc65422871)

[3.2.4 Chức năng tạo mới tin tức 41](#_Toc65422872)

[3.2.5 Chức năng quản lí hóa đơn 42](#_Toc65422873)

[KẾT LUẬN 43](#_Toc65422874)

[TÀI LIỆU THAM KHẢO 44](#_Toc65422875)

# **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT**

## **Tổng quan đề tài**

### **Lý do chọn đề tài**

Cuộc sống phát triển đi cùng với đó là nhu cầu kinh doanh, buôn bán ngày càng tăng cao với các hình thức kinh doanh khác nhau. Bên cạnh việc mở các cửa hàng kinh doanh theo kiểu truyền thống với lợi thế về mặt bằng, ngày càng có nhiều doanh nghiệp đi theo hướng phát triển hoạt động bán hàng trực tuyến, trong đó nổi bật nhất là bán hàng trực tuyến qua website bán hàng. Mặc dù vậy, không phải tất cả mọi người đều hiểu được **website bán hàng trực tuyến là gì** và tầm quan trọng của việc thiết kế website bán hàng trong việc thúc đẩy doanh thu và quảng bá thương hiệu hiệu quả đối với cửa hàng, doanh nghiệp trong thời đại Internet bùng nổ như hiện nay.

### **Mục tiêu đề tài**

Website bán hàng giúp việc bán hàng trở nên nhanh chóng và thuận tiện hơn, tất cả các khâu bán hàng đều gói gọn bên trong một website từ khâu trưng bày sản phẩm (thông tin mô tả, hình ảnh sản phẩm, giá bán), giao kết hợp đồng đến thanh toán. Điểm nổi bật của hình thức bán hàng thông qua website so với Facebook, Zalo đó chính là quy trình bán hàng khép kín từ A đến Z, khách mua hàng có thể lựa chọn hàng hóa và thanh toán trực tuyến qua thẻ ngân hàng, thẻ Visa,…ngay trên website rất nhanh chóng và thuận tiện

### **1.1.3 Phạm vi đề tài**

Website được thiết kế nhằm đáp ứng nhu cầu mua sắm cho tất cả mọi người, mọi lứa tuổi

### **Đối tượng nghiên cứu và phạm vi hệ thống**

**Đối tượng:**

* **Công cụ sử dụng:** Visual studio, Sublime text 3, SQl Server, IIS Express (Google Chrome)
* **Ngôn ngữ lập trình:** C#
* **Framework và library:** Entity, LinQ, CKFinder, CKEditor, ASP.Net MVC, Api Google, Api Facebook…
* **Hệ quản trị cơ sở dữ liệu:** NoSQL

**Phạm vi:** Đề tài chỉ dừng lại ở phạm vi xây dựng website trên nền tảng .Net Framework

## **1.2 Cơ sở lý thuyết**

### **1.2.1 Tổng quan về C#**

**1.2.1.1 Lịch sử hình thành**

Ngôn ngữ lập trình C# được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.

**1.2.1.2 C# là gì?**

* C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.
* C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.
* C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), . . . trở nên rất dễ dàng.

**1.2.1.3 Đặc trưng của C#**

* C# là ngôn ngữ đơn giản:

Như ta đã biết thì ngôn ngữ C# dựng trên nền tảng C++ và Java nên ngôn ngữ C# khá đơn giản. Nếu chúng ta thân thiện với C và C++ hoậc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn. Một vài trong các sự cải tiến là loại bỏ các dư thừa, hay là thêm vào những cú pháp thay đổi.

* C# là ngôn ngữ hiện đại

Một vài khái niệm như xử lý ngoại lệ, những kiểu dữ liệu mở rộng, bảo mật mã nguồn..v..v... Đây là những đặc tính được cho là của một ngôn ngữ hiện đại cần có. Và C# chứa tất cả các đặt tính ta vừa nêu trên.

* C# là ngôn ngữ lập trình thuần hướng đối tượng

Lập trình hướng đối tượng(tiếng Anh: Object-oriented programming, viết tắt: OOP) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (*abstraction*), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên.

* C# là một ngôn ngữ ít từ khóa

C được sử dụng để mô tả thôn# là ngôn ngữ sử dụng giới hạn những từ khóa (gồm khoảng 80 từ khóa và mười mấy kiểu dữ liệu xây dựng sẵn). Nếu bạn nghĩ rằng ngôn ngữ có càng nhiều từ khóa thì sẽ càng mạnh mẽ hơn. Điều này không phải sự thật, lấy ví dụ ngôn ngữ C# làm điển hình nhé. Nếu bạn học sâu về C# bạn sẽ thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

Ngoài những đặc điểm trên thì còn một số ưu điểm nổi bật của C#:

* C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, nên cũng khá dể dàng tiếp cận và học nhanh với C#.
* C# có thể biên dịch trên nhiều nền tảng máy tính khác nhau.
* C# được xây dựng trên nền tảng của C++ và Java nên nó được thừa hưởng những ưu điểm của ngôn ngữ đó.
* C# là một phần của .NET Framework nên được sự chống lưng khá lớn đến từ bộ phận này.
* C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ.

### **1.2.2 Mô hình MVC**

**1.2.2.1 Định nghĩa**

**MVC** là từ viết tắt bởi 3 từ **Model – View – Controller**. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

**1.2.2.2 Thành phần**

* **Model (M):**

Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một [file XML](https://monamedia.co/file-xml-la-gi/) bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,…

* **View (V):**

Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website.

Thông thường, các ứng dụng web sử dụng MVC View như một phần của  hệ thống, nơi các thành phần [HTML](https://monamedia.co/html-va-css-la-gi/) được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của  người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi.

Ví dụ: Nút “delete” được tạo bởi View khi người dùng nhấn vào nút đó sẽ có một hành động trong Controller.

* **Controller (C):**

Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

**1.2.2.3 Luồng tương tác giữa các thành phần trong MVC**

Ví dụ: Chức năng thêm bài viết mới trong trang quản trị website. Nơi đây có 2 trường nhập về tiêu đề và nội dung bài viết, thì trong đó:

* View sẽ hiển thị ra phần nhập form tiêu đề và nội dung.
* Controller lấy dữ liệu từ 2 trường và gửi tới Model.
* Model lúc này sẽ nhận dữ liệu từ Controller để lưu trữ trong cơ sở dữ liệu.

Mỗi bộ phận thực hiện chức năng nhất định, nhưng chúng có sự thống nhất, liên kết với nhau tạo nên **mô hình MVC**. Mô hình này tương đối nhẹ. Nó có thể tích hợp được nhiều tính năng có trong ASP.NET hiện giờ. Ví dụ như authentication (quá trình xác thực).

**1.2.2.4 Ưu và nhược điểm của MVC**

Ưu điểm:

* + Nhẹ, tiết kiệm băng thông: MVC không sử dụng viewstate nên khá tiết kiệm diện tích băng thông. Khi sử dụng, người dùng có thể sử dụng ứng dụng trên web cần tương tác gửi và nhận dữ liệu một cách liên tục. Do đó, việc giảm băng thông giúp cho website hoạt động tốt và ổn định hơn
* Kiểm tra dễ dàng: Với **MVC**, bạn có thể dễ dàng kiểm tra, rà soát lỗi phần mềm trước khi tới tay người tiêu dùng, đảm bảo chất lượng và độ uy tín cao hơn.
* Chức năng control: Trên các nền website thì ngôn ngữ lập trình như CSS, HTML, [Javascript](https://monamedia.co/javascript-la-gi-gioi-thieu-ve-ngon-ngu-js-cho-nguoi-moi-hoc/) có một vai trò vô cùng quan trọng. Việc sử dụng mô hình MVC sẽ giúp bạn có một bộ control ưu việt trên nền tảng các ngôn ngữ hiện đại với nhiều hình thức khác nhau.
* View và size: View sẽ là nơi lưu trữ các dữ liệu. Càng nhiều yêu cầu được thực hiện thì kích thước càng tệp càng lớn. Khi  đó, đường truyền mạng cũng giảm tốc độ load. Việc sử dụng mô hình MVC sẽ giúp bạn tiết kiệm được diện tích băng thông một cách tối ưu.
* Chức năng Soc (Separation of Concern): Chức năng này cho phép bạn phân tách rõ ràng các phần như Model, giao diện, data, nghiệp vụ.
* Tính kết hợp: Việc tích hợp ở **mô hình MVC** cho phép bạn thoải mái viết code trên nền tảng website. Khi đó, server của bạn sẽ được giảm tải khá nhiều.
* Đơn giản: Đây là một mô hình với kết cấu tương đối đơn giản. Dù bạn không có quá nhiều chuyên môn cũng có thể sử dụng được.

Nhược điểm:

MVC thường được sử dụng vào những  dự án lớn. Do đó, với các dự án nhỏ, mô hình MVC có thể gây cồng kềnh, tốn thời gian trong quá trình phát triển cũng như thời gian trung chuyển dữ liệu.

**1.2.1.5 Ứng dụng mô hình MVC vào lập trình**

* Mô hình MVC được ứng dụng trong nhiều [ngôn ngữ lập trình](https://monamedia.co/top-10-ngon-ngu-lap-trinh-phan-mem-tot-nhat-hien-nay/) khác nhau, nhưng phổ biến nhất là ứng dụng ASP.NET MVC hay PHP MVC.
* MVC đang là mô hình được ứng dụng rất nhiều trong lập trình.
* Hệ thống MVC phát triển tốt sẽ cho phép phát triển front – end, back – end cùng trên hệ thống mà không có sự can thiệp, chia sẻ, chỉnh sửa các tập tin trong khi một hoặc hai bên vẫn đang làm việc.
* Việc sử dụng mô hình tương đối đơn giản. Chỉ cần hiểu rõ quy trình vận hành, nắm được các chức năng của từng bộ phận thì việc triển khai mô hình MVC tương đối dễ dàng.

### **1.2.3 LINQ trong C#**

**1.2.3.1 Giới thiệu**

**LINQ (Language Integrated Quey) -** ngôn ngữ truy vấn tích hợp – nó tích hợp cú pháp truy vấn (gần giống các câu lệnh SQL) vào bên trong ngôn ngữ lập trình C#, cho nó khả năng truy cập các nguồn dữ liệu khác nhau (SQL DB, XML, List…) với cùng cú pháp.

LINQ hoạt động trên những kiểu tập hợp có khả năng duyệt qua nó (xem thêm Collection, List trong C#). Để sử dụng LINQ thì nạp hai thư viện *Generic* và *Linq:*

using System.Collections.Generic;

using System.Linq;

Nguồn dữ liệu phục vụ cho LINQ, phải là các từ các lớp triển khai giao diện

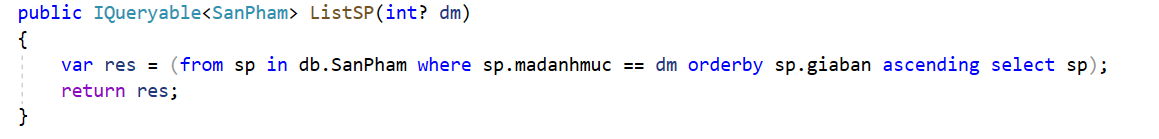
*IEnumerable, IEnumerable<T>* tức là các mảng, danh sách thuộc Collection đã biết ở phần trước

**Ví dụ:**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

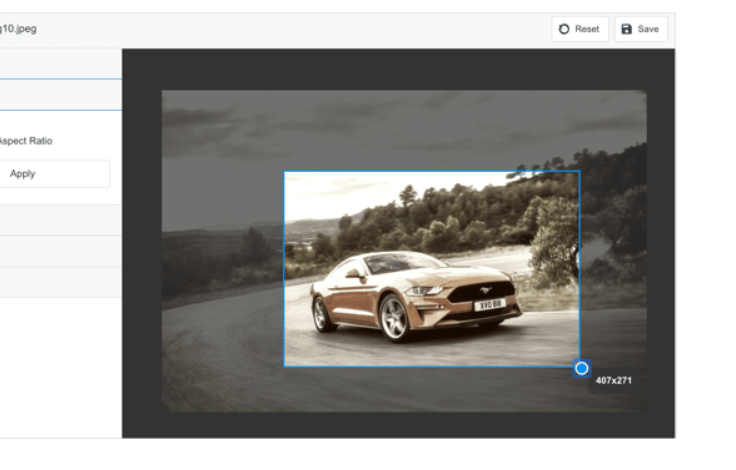
*Hình 1.1 Lấy ra danh sách sản phẩm trong bảng sản phẩm*

**

*Hình 1.2 Lấy ra danh sách sản phẩm trong bảng sản phẩm theo mã danh mục và sắp xếp chúng theo thứ tự tăng dần (ascending) của giá bán*

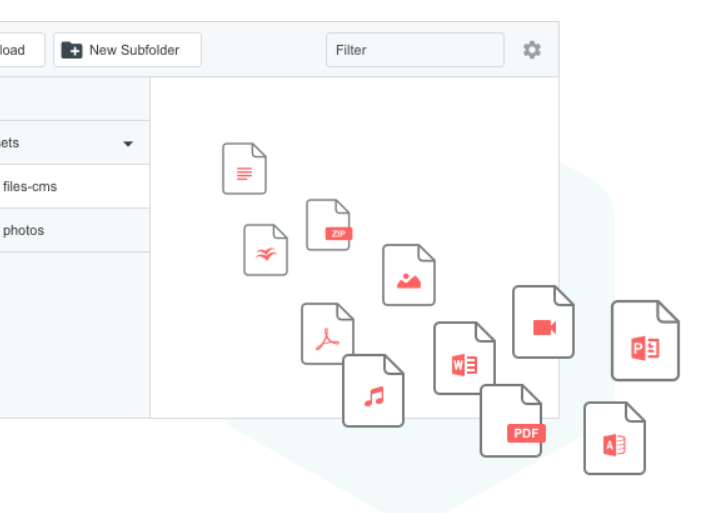
### **1.2.4 Giới thiệu về trình quản lý tệp hình ảnh CKFinder**

Truy cập [*https://ckeditor.com/ckfinder/download/#net*](https://ckeditor.com/ckfinder/download/#net) để download thư viện



*Hình 1.3 Cắt, xoay, chỉnh sửa hình ảnh*

* **Cắt, xoay, chỉnh sửa, thay đổi kích thước hình ảnh**
* CKFinder bao gồm một trình chỉnh sửa hình ảnh trực quan. Sử dụng nó để: cắt, thay đổi kích thước và xoay hình ảnh, đặt độ sáng, độ tương phản, độ bão hòa, độ phơi sáng và hơn thế nữa.
* Chọn một trong các bộ lọc được xác định trước. Đưa Instagram đó vào ngay trong trình chỉnh sửa WYSIWYG của bạn chỉ với một cú nhấp chuột!
* CKFinder cũng có thể tiết kiệm thời gian của bạn và tự động chia tỷ lệ hình ảnh lớn hơn đến kích thước và chất lượng tối đa được xác định trước.
* **Tải lên tệp ở bất kỳ đâu bạn thích**
* Các tệp của bạn hoàn toàn nằm trong tầm kiểm soát của bạn. Lưu trữ chúng trên máy chủ nơi bạn đã cài đặt CKFinder hoặc định cấu hình nó để gửi tệp bằng FTP.
* CKFinder đi kèm với một số bộ điều hợp sẵn sàng sử dụng để lưu tệp của bạn vào các kho lưu trữ đám mây phổ biến: Dropbox, Amazon S3 hoặc Microsoft Azure.
* Kiến trúc CKFinder cho phép viết bộ điều hợp để lưu trữ tệp hầu như ở mọi nơi, kể cả trong cơ sở dữ liệu.
* **Bất kỳ loại tệp nào bạn muốn**



*Hình 1.4 Các loại tệp tải lên*

* Hình ảnh PNG, JPG, GIF, TIF
* Video
* Tệp PDF hoặc DOC
* Lưu trữ ZIP… hoặc bất kỳ thứ gì khác.
* **Tích hợp**



*Hình 1.5 Tích hợp*

* **Tệp của bạn được bảo mật**
* Danh sách kiểm soát truy cập cho phép bạn xác định các quyền chi tiết cho mỗi thư mục và thư mục con. Bạn có thể đặt một số thư mục ở chế độ chỉ đọc và không cho phép những thư mục khác truy cập đối với những người dùng chưa được xác thực.
* Bạn có thể định cấu hình CKFinder để cung cấp cho người dùng các thư mục riêng tư của riêng họ và không gian chung được chia sẻ.
* Bạn có thể kiểm soát các loại phần mở rộng tệp được phép và không được phép.
* **Điều chỉnh nó**
* **Chế độ chỉ đọc**: thiết lập CKFinder không có tính năng tải lên, cho phép người dùng của bạn chọn từ danh sách tệp hoặc hình ảnh được xác định trước (ví dụ: ảnh lưu trữ).
* **Chế độ nhỏ gọn**: hiển thị các tệp không có cây thư mục theo cách nhỏ gọn đẹp mắt. Nhiều tùy chọn cấu hình, API thân thiện, tài liệu chi tiết và khả năng được mở rộng với các plugin tùy chỉnh cho phép bạn hoàn toàn điều chỉnh CKFinder theo nhu cầu của mình.

### **1.2.5 Giới thiệu về trình soạn thảo mã nguồn mở CKEditor**

Truy cập *https://ckeditor.com/ckeditor-4/download/*  để download thư viện

* **Đầy đủ các tính năng**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

*Hình 1.6 Các tính năng của CKEditor*

* **Dán từ Word, Excel và Google Tài liệu** . Hỗ trợ bảng tuyệt vời với thay đổi kích thước cột, chọn hàng và cột.
* **Phương tiện nhúng** (chèn video, tweet, bài đăng trên Instagram và hơn thế nữa), widget, đoạn mã, công thức toán học.
* **Bảng tính** để tạo lưới dữ liệu bên trong trình chỉnh sửa.
* **Tự động điền, @ chú thích,** plugin **biểu tượng cảm xúc** .
* **Giao diện người dùng nội tuyến và iframe** , tự động duyệt, chế độ tối đa hóa để viết không bị phân tâm. Chế độ chỉ đọc.
* Nhiều plugin tạo kiểu và định dạng, tính năng sao chép định dạng.
* **Khả năng tiếp cận tuyệt vời**
* Hỗ trợ bàn phím tuyệt vời.
* Hỗ trợ độ tương phản cao và hỗ trợ các trình đọc màn hình như JAWS.
* Trình kiểm tra trợ năng tùy chọn để phân tích xem nội dung do người dùng tạo có thể truy cập được hay không.
* **Dễ dàng tích hợp**
* CKEditor 4 tích hợp dễ dàng với các công cụ, khuôn khổ và công nghệ JavaScript hiện đại, bao gồm các trình quản lý gói như **npm, NuGet, Composer, fiber và pnpm** hoặc các gói như **webpack** hoặc **Rollup.js** .
* Tích hợp gốc với **Angular** , **React** và **Vue.js** có sẵn để thuận tiện cho bạn.
* Xây dựng ứng dụng máy tính để bàn di động hoặc đa nền tảng với JavaScript? CKEditor 4 tương thích với **Electron** và các thiết bị di động ( **Android, iOS** ).
* **Thực sự toàn cầu**
* 70 phiên bản ngôn ngữ.
* Giao diện người dùng với hỗ trợ LTR và RTL (ví dụ: cho tiếng Ả Rập hoặc tiếng Do Thái).
* Các plugin chuyên dụng để thiết lập hướng văn bản (trộn nội dung RTL và LTR) và cho ngôn ngữ phần văn bản (ví dụ: trích dẫn một cụm từ tiếng Đức trong một câu tiếng Anh).
* Hỗ trợ thành phần (IME) tuyệt vời cho các ngôn ngữ như tiếng Trung, tiếng Nhật, tiếng Hàn.
* **Bảng tính**
* Tính năng bảng tính có sẵn trực tiếp trong tài liệu của bạn.
* Hoàn hảo cho các ngành tài chính, kỹ thuật và khoa học.
* Giúp người dùng của bạn tăng năng suất (không cần Excel)!
* **Xuất nhanh sang PDF và Word**
* **Phương trình toán học và kiểm tra chính tả**
* Nhập hoặc viết tay - tùy theo cách nào phù hợp với bạn hơn.
* Tạo phím tắt của riêng bạn và tùy chỉnh các thanh công cụ của bạn.
* Chỉnh sửa công thức của bạn ở chế độ chỉ dùng bàn phím.
* Lỗi chính tả và ngữ pháp được gạch chân khi di chuyển.
* Huy hiệu không bị phân tâm với quyền truy cập vào các đề xuất hiệu đính.
* Đề xuất chính tả và ngữ pháp có sẵn khi di chuột, không cần nhấp chuột.

### **1.2.6 Giới thiệu về API**

**API là gì?**

* API (Application Programming Interface) đây là một giao diện lập trình dùng phương thức kết nối với các phần mềm và thư viện khác. API có ở trong hệ điều hành Windows và thậm chí Twitter cũng có web này. Tuy nhiên, ở mỗi nền tảng thì API lại được sử dụng với nhiều ứng dụng khác nhau. Có thể hiểu API giống như bàn phím, chính là đồ để kết nối người dùng với máy tính. Đây là phần mềm giao tiếp giữa hệ điều hành và chương trình.
* Mỗi API dành cho các hệ điều hành Linux và Windows khác nhau hoàn toàn. API là nơi cung cấp khả năng truy xuât đến tập các hàm thường dùng. Hiểu đơn giản thì API giống như công cụ tạo ra các phần mềm. API là công nghệ mới nhất của Microsoft để xây dựng thành phần phân tán. API bao gồm: controller, routing, action resulf, loc container, filter, unit test, model binder, injection,… Ngoài ra nó còn hỗ trợ các phương thức như: Post/Get/Delete/Put dữ liệu.

**Ưu và nhược điểm của API**

**Ưu điểm**

* Giao tiếp hai chiều phải được xác nhận trong các giao dịch sử dụng API. Cũng chính vì vậy mà các thông tin rất đáng tin cậy.
* API là công cụ mã nguồn mở, có thể kết nối mọi lúc nhờ vào Internet.
* Hỗ trợ chức năng RESTful một cách đầy đủ.
* Cấu hình đơn giản khi được so sánh với WCF (Window Communication Foundation). Cung cấp cấp trải nghiệm thân thiện với người dùng.

**Nhược điểm**

* Tốn nhiều chi phí phát triển, vận hành, chỉnh sửa.
* Đòi hỏi kiến thức chuyên sâu.
* Có thể gặp vấn đề bảo mật khi bị tấn công hệ thống.

**Ứng dụng của AP**

* **Web API:** Hay còn gọi là hệ thống API dùng  trong website. Các website sử dụng API cho phép bạn cập nhật, kết nối hoặc lấy dữ liệu dễ dàng hơn. Vó dụ khi bạn tạo chức năng đăng nhập thông của Google, Twitter, Facebook thì bạn đang gọi đến API của Google. Bên cạnh đó, các ứng dụng trên điện thoại đều lấy dữ liệu thông qua API.
* **API trên hệ điều hành:** Windows và Linux sử dụng API để cung cấp dữ liệu, đặc tả các hàm và sử dụng chúng như phương thức để kết nối. API giúp tạo ra các phần mềm liên kết trực tiếp với hệ điều hành.
* **API của thư viện phần mềm hay framework:** API sẽ quy định, mô tả những hành động mà các thư viện dữ liệu cung cấp. Mỗi API có cách điều khiển khác nhau và chúng giúp cho chương trình ngôn ngữ này có thể sử dụng được thư viện được xây dựng bằng ngôn ngữ khác. Ví dụ, bạn có thể yêu cầu thư viện tạo ra file PDF nhưng viết bằng C++.

### **1.2.7 Giới thiệu về Google Map API**

**Google Map API là gì?**

* Google Map là bản đồ trực tuyến được ứng dụng trên app miễn phí cho di động và website do Google phát hành, quản lý. Ứng dụng này cho phép người dùng sử dụng các tính năng như: tìm vị trí, tìm đường, hiển thị tuyến đường tối ưu, hướng dẫn bắt xe bus, các địa điểm chỉ định như bệnh viện, cây xăng, ATM,…
* Google Map API chính là phương pháp sử dụng cho phép web B có thể dùng dịch vụ hiển thị nội dung ở web A – Google Map thông qua Google Map API. Bản đồ tại web A sẽ được nhúng vào web B (cá nhân). Sau đó, web B có thể sử dụng mọi dịch vụ, tính năng mà Google Map đã cung cấp.

**Một số ứng dụng của Google Map API**

* Khi dùng Google Map API bạn có thể đánh dấu trên bản đồ kèm theo mọi thông tin về địa điểm đó như: trường học, bệnh viện, ATM,…
* Tính năng chỉ đường tối ưu, tìm địa điểm hoặc cung cấp những cách tìm thông qua các dịch vụ có sẵn tại Google Map.
* Khoanh vùng rõ ràng các khu đặc biệt như: khu sản xuất, khu đô thị, khu ô nhiễm,…
* Theo dõi lưu lượng phương tiện giao thông tại các khu vực được chỉ định.

**Google Maps API thay đổi cách tính phí**

* Google Map API đã thay đổi cách tính chi phí cho các dịch vụ mà nó cung cấp. Khi tạo tài khoản mới thì người dùng sẽ được sử dùng miễn phí dịch vụ trong hạn mức 200$/tháng. Điều này đồng nghĩa với việc bạn có thể phát triển những ứng dụng nhỏ (request thấp) trong tháng đó để được miễn phí toàn bộ chi phí.
* Ngoài ra, một số dịch vụ như Dynamic Maps hay Static Maps dành cho di động đang được Google miễn phí để kích thích sự phát triển ứng dụng trên nền tảng có sẵn này.

**Google Map API bị chặn ở Việt Nam**

* Việt Nam đang nằm trong danh sách những vùng bị cấm sử dụng Google Map API. Vì thế, để có thể sử dụng MAP APO thì bạn cần có Billing Account mới và phải nằm ngoài danh sách các vùng bị chặn thì mới có thể sử dụng được.

**Cách sử dụng Google Map API**

Truy cập [*https://maps.googleapis.com/*](https://maps.googleapis.com/) để lấy key

Truy cập [*https://www.google.es/maps/*](https://www.google.es/maps/) để xác định tọa độ địa điểm

Sau đó sử dụng key và tọa độ địa điểm để tạo bản đồ

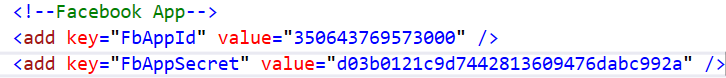
### **1.2.8 Giới thiệu về Facebook API**

**Facebook API là gì?**

* Facebook API là nền tảng do Facebook cung cấp cho người viết ứng dụng để dễ dàng trong việc tạo ứng dụng và đảm bảo người viết ứng dụng không can thiệp quá sâu vào hệ thống của Facebook.
* Thông qua Facebook API, ta có thể lấy được thông tin về người dùng như là thông tin cá nhân, ảnh profile.... nếu như được cấp quyền cho ta truy cập trang cá nhân.
* Facebook sẽ gửi một phương thức POST đến máy chủ Facebook API. Nó bao gồm một số các thông số yêu cầu như api\_key của ứng dụng, session\_key của người dùng đưa ra yêu cầu. Bên cạnh đó Facebook còn thêm vào tham số fb\_sig để thông báo ứng dụng đưa ra yêu cầu. Bằng cách này tất cả các lời gọi API sẽ được đảm bảo, Facebook có thể xác minh các yêu cầu được gửi từ một ứng dụng đã được chấp thuận.

**Cách sử dụng Facebook API**

Truy cập *https://developers.facebook.com/* để lấy id và key



Download Facebook trong Manage Nuget Packages



*Hình 1.7 SDK Facebook*

Sau đó sử dụng id và key vừa lấy được để đăng nhập bằng facebook

# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## **2.1 Tìm hiểu nghiệp vụ**

Hoạt động của website bán hàng như sau:

* Khi người dùng muốn mua hàng thì phải đăng nhập hoặc đăng kí nếu chưa có tài khoản
* Sau khi đăng nhập người dùng có thể thêm sản phẩm muốn mua vào giỏ hàng và chọn phương thức thanh toán
* Người dùng có thể tìm kiếm sản phẩm theo loại hoặc theo tên, chia sẻ các sản phẩm mình thích lên các nền tảng mạng xã hội (Facebook, Twitter…)
* Hóa đơn mua hàng sẽ được gửi đến cho khách hàng qua gmail
* Người dùng có thể xem địa chỉ của shop và đến mua trực tiếp
* Trang tin tức giúp người dùng biết được những sản phẩm mới ra mắt
* Người quản lí website có thể xem và cập nhật những sản phẩm cho trang web và phân quyền tài khoản

## **2.2 Xác định người dùng**

Xác định website bán nhiều mặt hàng phù hợp cho tất cả mọi người: dân văn phòng, học sinh, sinh viên...

## **2.3 Chức năng**

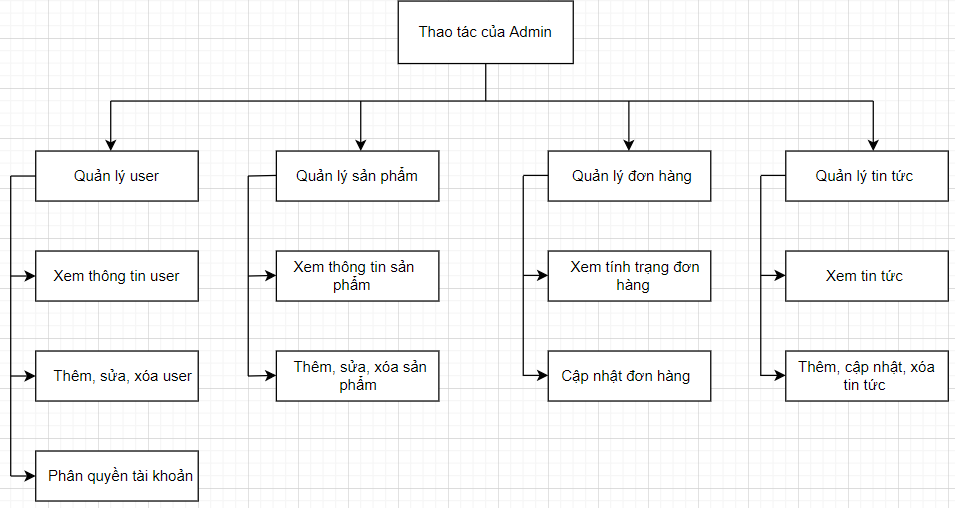
* Đăng kí tài khoản có sử dụng mã xác thực Captcha
* Đăng nhập bằng tài khoản vừa đăng kí hoặc đăng nhập bằng tài khoản facebook
* Xem sản phẩm theo loại, thêm vào giỏ hàng, thanh toán
* Hóa đơn thanh toán được gửi trực tiếp đến gmail của khách hàng
* Tìm kiếm nhanh sản phẩm theo tên
* Xem thông tin về những sản phẩm mới nhất
* Quản lý user: thêm, xóa, thay đổi quyền user
* Quản lý tin tức: thêm, sửa, xóa tin tức
* Quản lý sản phẩm: thêm mới, cập nhật, xóa sản phẩm
* Quản lý ngôn ngữ: tiếng Anh, tiếng Việt
* Giao diện thân thiện dễ sử dụng

## **2.4 Sơ đồ phân ra chức năng**

Ảnh có chứa văn bản, trong nhà, ảnh chụp màn hình

Mô tả được tạo tự động

*Hình 2.1 Sơ đồ phân rã chức năng dành cho user*

**

*Hình 2.2 Sơ đồ phân rã chức năng dành cho admin*

## **2.5 Thiết kế cơ sở dữ liệu**

### **2.5.1 Cấu trúc cơ sở dữ liệu**

Do website sử dụng cơ sở dữ liệu NOSQL nên cấu trúc cơ sở dữ liệu không phải là sự liên kết giữa các bảng mà cấu trúc được thể hiện dưới dạng JSON với mỗi bảng trong CSDL SQL tương đương với 1 collection

### **2.5.2 Danh sách các bảng**

**2.5.2.1 Bảng User**

**Chức năng:** Lưu những thông tin cơ bản của user

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | bigint | Mã user |
| 2 | UserName | varchar(50) | Tên user |
| 3 | Password | varchar(32) | Mật khẩu của user |
| 4 | GroupID | varchar(20) | Mã nhóm của user (phân quyền) |
| 5 | Name | nvarchar(50) | Tên người sở hữu user |
| 6 | Address | nvarchar(50) | Địa chỉ của user |
| 7 | Email | nvarchar(50) | Email của user |
| 8 | Phone | nvarchar(50) | Số điện thoại của user |
| 9 | ProvinceID | int | Tỉnh thành |
| 10 | DistrictID | int | Quận/Huyện |
| 11 | CreateDate | datetime | Ngày tạo |
| 12 | CreateBy | varchar(50) | Người tạo |
| 13 | ModifiedDate | datetime | Ngày sửa đổi |
| 14 | ModifiedBy | varchar(50) | Người sửa đổi |
| 15 | Status | bit | Trạng thái tài khoản (bị khóa hay không) |

Bảng 2.3: Bảng User

**2.5.2.2 Bảng Category**

**Chức năng:** Lưu loại tin tức

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | bigint | Mã loại tin tức |
| 2 | Name | nvarchar(250) | Tên loại tin tức |
| 3 | MetaTitle | varchar(250) | Tiêu đề |
| 4 | CreateDate | datetime | Ngày tạo tin |
| 5 | CreateBy | varchar(50) | Người tạo tin |
| 6 | ModifiedDate | datetime | Ngày chỉnh sửa |
| 7 | ModifiedBy | varchar(50) | Người chỉnh sửa |
| 8 | MetaKeywords | nvarchar(250) | Từ khóa |
| 9 | MetaDescriptions | nchar(250) | Mô tả |
| 10 | Status | bit | Trạng thái |
| 11 | Language | varchar(2) | Ngôn ngữ (tiếng Việt hoặc tiếng Anh) |

Bảng 2.4: Bảng Category

**2.5.2.3 Bảng Content**

**Chức năng:** Lưu nội dung tin tức

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | bigint | Mã tin tức |
| 2 | Name | nvarchar(250) | Tên tin tức |
| 3 | MetaTitle | varchar(250) | Tiêu đề |
| 4 | Description | nvarchar(500) | Mô tả |
| 5 | Images | nvarchar(250) | Ảnh |
| 6 | CategoryID | bigint | Mã thể loại |
| 7 | Detail | ntext | Chi tiết |
| 8 | CreateDate | datetime | Ngày tạo tin |
| 9 | CreateBy | varchar(50) | Người tạo tin |
| 10 | ModifiedDate | datetime | Ngày sửa đổi |
| 11 | ModifiedBy | varchar(50) | Người sửa đổi |
| 12 | Status | bit | Trạng thái |
| 13 | ViewCount | int | Số lượt xem |
| 14 | Language | varchar(2) | Ngôn ngữ |

Bảng 2.5: Bảng Content

**2.5.2.4 Bảng Contact**

**Chức năng:** Lưu địa chỉ liên hệ

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | int | Mã liên hệ |
| 2 | [Content] | ntext | Nội dung (địa chỉ liên hệ) |
| 3 | Status | bit | Trạng thái |

Bảng 2.6: Bảng Contact

**2.5.2.5 Bảng Feedback**

**Chức năng:** Lưu nội dung phản hồi

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | int | Mã phản hồi |
| 2 | Name | nvarchar(50) | Tên người phản hồi |
| 3 | Phone | nvarchar(50) | Số điện thoại |
| 4 | Email | nvarchar(50) | Email người phản hồi |
| 5 | Address | nvarchar(50) | Địa chỉ |
| 6 | [Content] | nvarchar(250) | Nội dung phản hồi |
| 7 | CreateDate | datetime | Ngày tạo phản hồi |
| 8 | Status | bit | Trạng thái |

Bảng 2.7: Bảng Feedback

**2.5.2.6 Bảng Language**

**Chức năng:** Lưu các loại ngôn ngữ

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | varchar(2) | Mã ngôn ngữ |
| 2 | Name | nvarchar(50) | Tên ngôn ngữ |
| 3 | IsDefault | bit | Ngôn ngữ mặc định |

Bảng 2.8: Bảng Language

**2.5.2.7 Bảng Menu**

**Chức năng:** Lưu menu của trang

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | int | Mã nội dung |
| 2 | Text | nvarchar(50) | Nội dung |
| 3 | Link | nvarchar(250) | Đường dẫn |
| 4 | DisplayOrder | Int | Thứ tự hiển thị |
| 5 | Target | nvarchar(50) | Mục tiêu |
| 6 | Status | bit | Trạng thái |
| 7 | TypeID | int | Loại nội dung |

Bảng 2.9: Bảng Menu

**2.5.2.8 Bảng Order**

**Chức năng:** Lưu thông tin cơ bản của khách hàng đã đặt hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | bigint | Mã đơn hàng |
| 2 | CreatedDate | datetime | Ngày đặt hàng |
| 3 | CustomerID | bigint | Mã khách hàng |
| 4 | ShipName | nvarchar(50) | Tên khách hàng |
| 5 | ShipMobile | varchar(50) | Số điện thoại của khách hàng |
| 6 | ShipAddress | nvarchar(50) | Địa chỉ của khách hàng |
| 7 | ShipEmail | nvarchar(50) | Email của khách hàng |
| 8 | Status | int | Trạng thái |

Bảng 2.10: Bảng Order

**2.5.2.8 Bảng OrderDetail**

**Chức năng:** Lưu thông tin chi tiết của đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ProductID | bigint | Mã sản phẩm |
| 2 | OrderID | bigint | Mã đơn hàng |
| 3 | Quantity | int | Số lượng |
| 4 | Price | decimal(18,0) | Tổng giá |

Bảng 2.10: Bảng OrderDetail

**2.5.2.9 Bảng Product**

**Chức năng:** Lưu thông tin chi tiết của sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | bigint | Mã sản phẩm |
| 2 | Name | nvarchar(250) | Tên sản phẩm |
| 3 | MetaTitle | varchar(250) | Tiêu đề sản phẩm |
| 4 | Description | nvarchar(500) | Mô tả sản phẩm |
| 5 | Images | nvarchar(250) | Ảnh sản phẩm |
| 6 | MoreImages | xml | Thông tin ảnh |
| 7 | Price | decimal(18,0) | Giá sản phẩm |
| 8 | PromotionPrice | decimal(18,0) | Giá ưu đãi |
| 9 | IncludeVAT | bit | Bao gồm thuế VAT |
| 10 | Quantity | int | Số lượng sản phẩm |
| 11 | CategoryID | bigint | Mã loại sản phẩm |
| 12 | Detail | ntext | Chi tiết sản phẩm |
| 13 | CreateDate | datetime | Ngày nhập |
| 14 | CreateBy | varchar(50) | Nhập bởi |
| 15 | ModifiedDate | datetime | Ngày sửa đổi |
| 16 | ModifiedBy | varchar(50) | Người sửa đổi |
| 17 | Status | bit | Trạng thái |
| 18 | TopHot | datetime | Ngày ra sản phẩm mới nhất |
| 19 | ViewCount | int | Số lượt xem |

Bảng 2.11: Bảng Product

**2.5.2.10 Bảng ProductCategory**

**Chức năng:** Lưu loại sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | bigint | Mã loại sản phẩm |
| 2 | Name | nvarchar(250) | Tên loại sản phẩm |
| 3 | MetaTitle | varchar(250) | Tiêu đề loại |
| 4 | ParentID | bigint |  |
| 5 | DisplayOrder | int | Thứ tự hiển thị |
| 6 | CreateDate | datetime | Ngày tạo loại sản phẩm |
| 7 | CreateBy | varchar(50) | Người tạo |
| 8 | ModifiedDate | datetime | Ngày sửa đổi |
| 9 | ModifiedBy | varchar(50) | Người sửa đổi |
| 10 | Status | bit | Trạng thái |

Bảng 2.12: Bảng ProductCategory

**2.5.2.11 Bảng Slide**

**Chức năng:** Lưu các slide chạy trên website

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | int | Mã ảnh |
| 2 | Image | nvarchar(250) | Tên ảnh |
| 3 | DisplayOrder | int | Thứ tự hiển thị |
| 4 | Link | nvarchar(250) | Đường dẫn ảnh |
| 5 | Description | nvarchar(250) | Mô tả |
| 6 | CreateDate | datetime | Ngày tạo |
| 7 | CreateBy | varchar(50) | Người tạo |
| 8 | ModifiedDate | datetime | Ngày sửa đổi |
| 9 | ModifiedBy | varchar(50) | Người sửa đổi |
| 10 | Status | bit | Trạng thái |

Bảng 2.13: Bảng Slide

**2.5.2.12 Bảng UserGroup**

**Chức năng:** Phân quyền người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Key** | **Value type** | **Mô tả** |
| 1 | ID | varchar(20) | Mã quyền (ADMIN hoặc MEMBER) |
| 2 | Name | nvarchar(50) | Tên quyền |

Bảng 2.14: Bảng UserGroup

# **CHƯƠNG 3 XÂY DỰNG CHƯƠNG TRÌNH**

## **3.1 Xây dựng chức năng và giao diện cho người dùng**

### **3.1.1 Chức năng đăng kí**

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 3.1 Giao diện trang đăng kí

**Quy trình thực hiện:**

* Người dùng nhập các thông tin như tên đăng nhập, mật khẩu, họ tên,email, mã xác nhận, nếu người dùng chưa nhập thì bắt buộc phải nhập, mã xác nhận phải nhập đúng như hình
* Kiểm tra xem dữ liệu đã đúng yêu cầu chưa: Email phải có dạng “[a@gmail.com](mailto:a@gmail.com)”, mật khẩu phải dài ít nhất 6 kí tự, mật khẩu và xác nhận mật khẩu phải giống nhau
* Kiểm tra trên cơ sở dữ liệu xem đã tồn tại tên đăng nhập đã đăng kí hay chưa?
* Nếu có rồi thì thông báo lỗi và yêu cầu nhập lại. Nếu chưa có thì tạo tài và lưu thông tin tài khoản vào cơ sở dữ liệu

### **3.1.2 Chức năng đăng nhập**

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 3.2 Giao diện trang đăng nhập

**Quy trình thực hiện:**

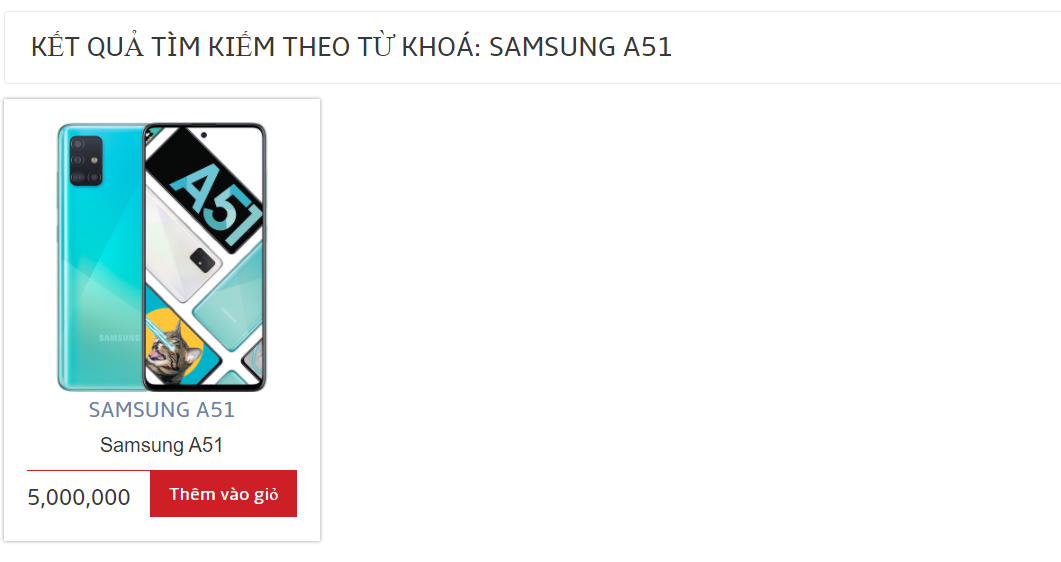
* Người dùng đăng nhập bằng tài khoản đã đăng kí hoặc đăng nhập bằng tài khoản facebook
* Kiểm tra xem tài khoản có tồn tại hay không, có đúng mật khẩu hay không?
* Nếu đăng nhập thành công bằng tài khoản đã đăng kí sẽ trả về tên tài khoản, nếu đăng nhập thành công bằng tài khoản facebook sẽ trả về gmail của tài khoản facebook

### **3.1.3 Chức năng tìm kiếm sản phẩm**

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 3.3 Chức năng tìm kiếm



Hình 3.4 Sản phẩm được tìm kiếm

**Quy trình thực hiện:**

* Gõ tên sản phẩm cần tìm kiếm, sẽ hiển thị luôn gợi ý sản phẩm
* Chọn vào sản phẩm muốn tìm kiếm sẽ hiển thị ra sản phẩm đó, kích vào sản phẩm để xem chi tiết

### **3.1.4 Chức năng gửi phản hồi**

Ảnh có chứa bản đồ

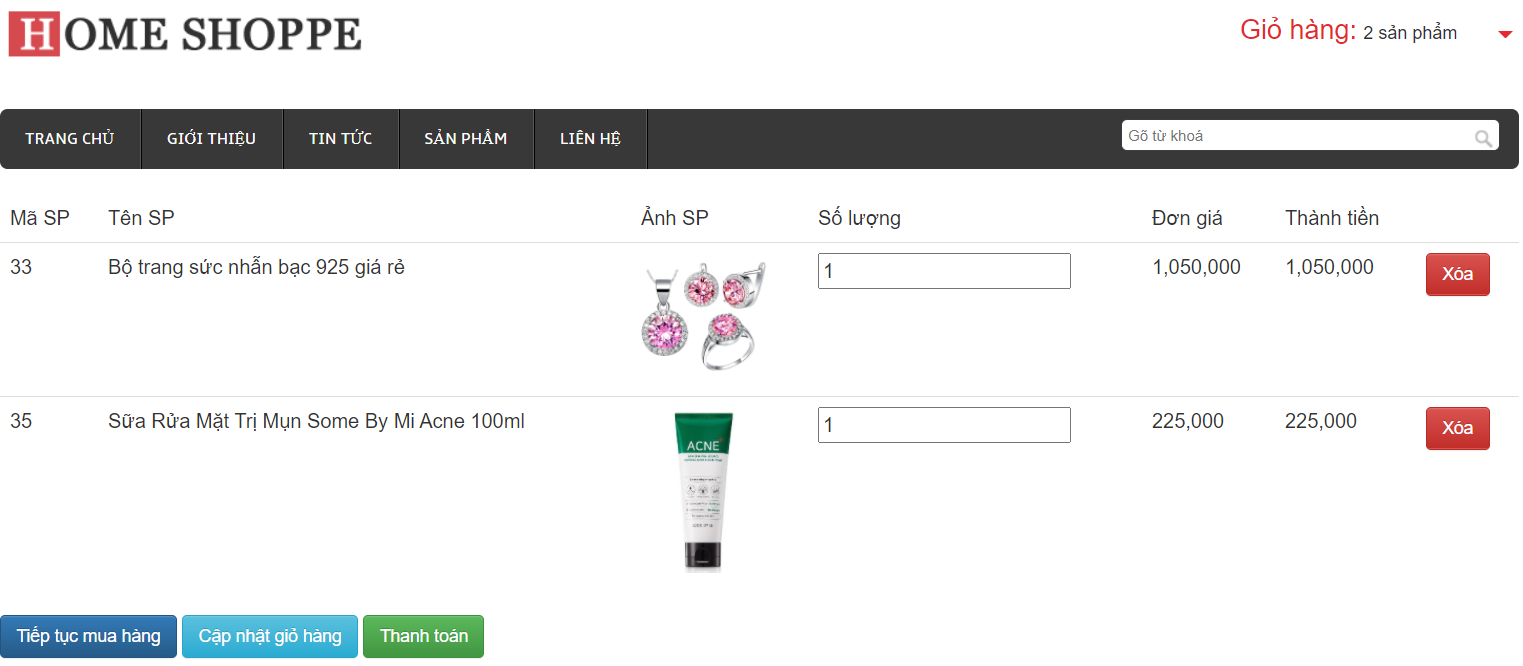
Mô tả được tạo tự động

Hình 3.5 Giao diện trang liên hệ

**Quy trình thực hiện:**

* Người dùng nhập những thông tin cơ bản như họ tên, số điện thoại, địa chỉ, email và yêu cầu, phản hồi của người dùng
* Người quản trị sẽ tiếp nhận những phản hồi, góp của về sản phẩm của người dùng

### **3.1.5 Chức năng giỏ hàng**

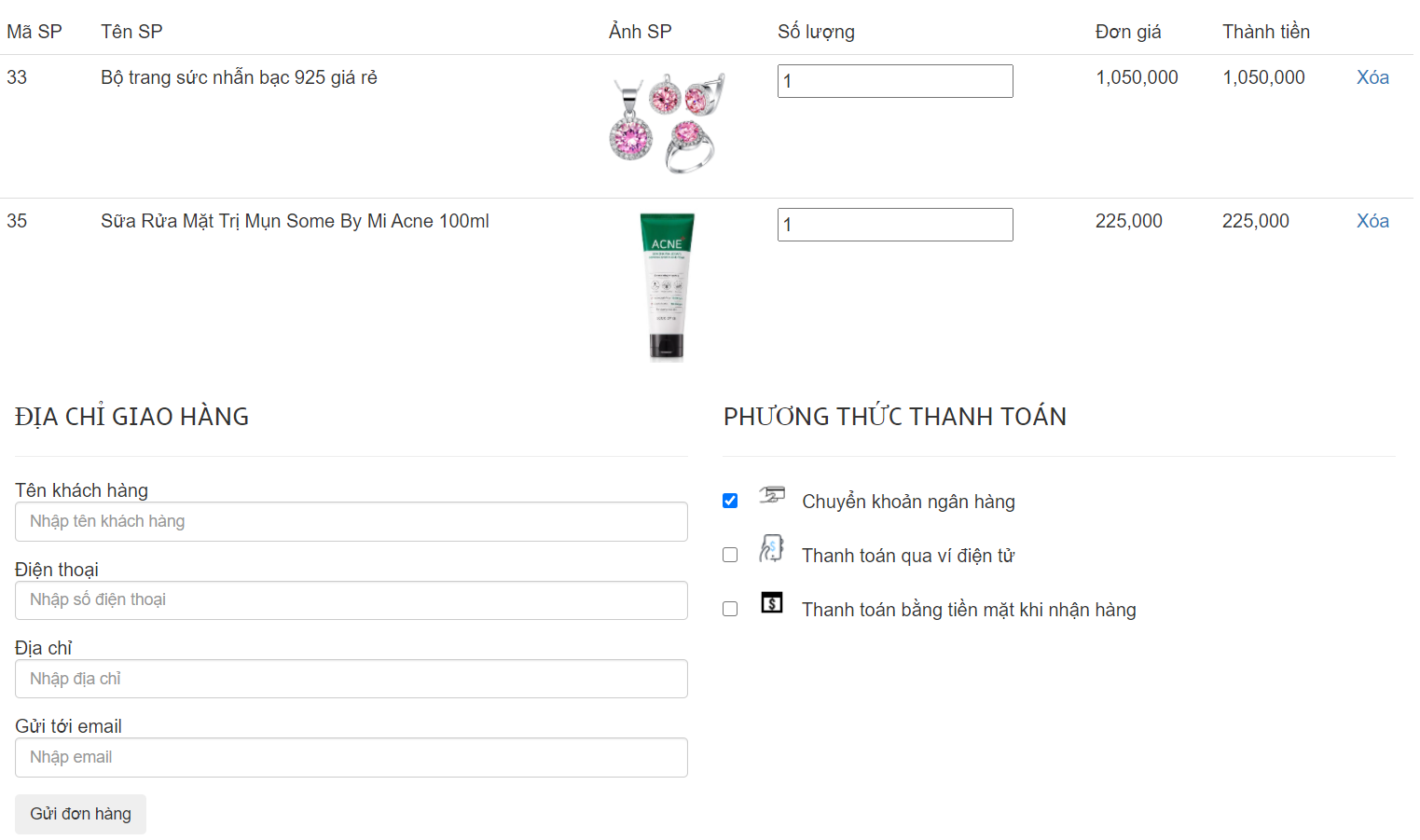


Hình 3.6 Giao diện trang giỏ hàng

**Quy trình thực hiện:**

* Sau khi đã thêm các sản phẩm vào giỏ hàng, khách hàng có thể tiếp tục mua hàng, cập nhật lại số lượng hoặc xóa các sản phẩm vừa thêm

### **3.1.6 Chức năng thanh toán**



Hình 3.7 Giao diện trang thanh toán

**Quy trình thực hiện:**

* Khi đã chọn được sản phẩm phù hợp, khách hàng nhập các thông tin để thanh toán
* Thanh toán thành công sẽ có thông báo



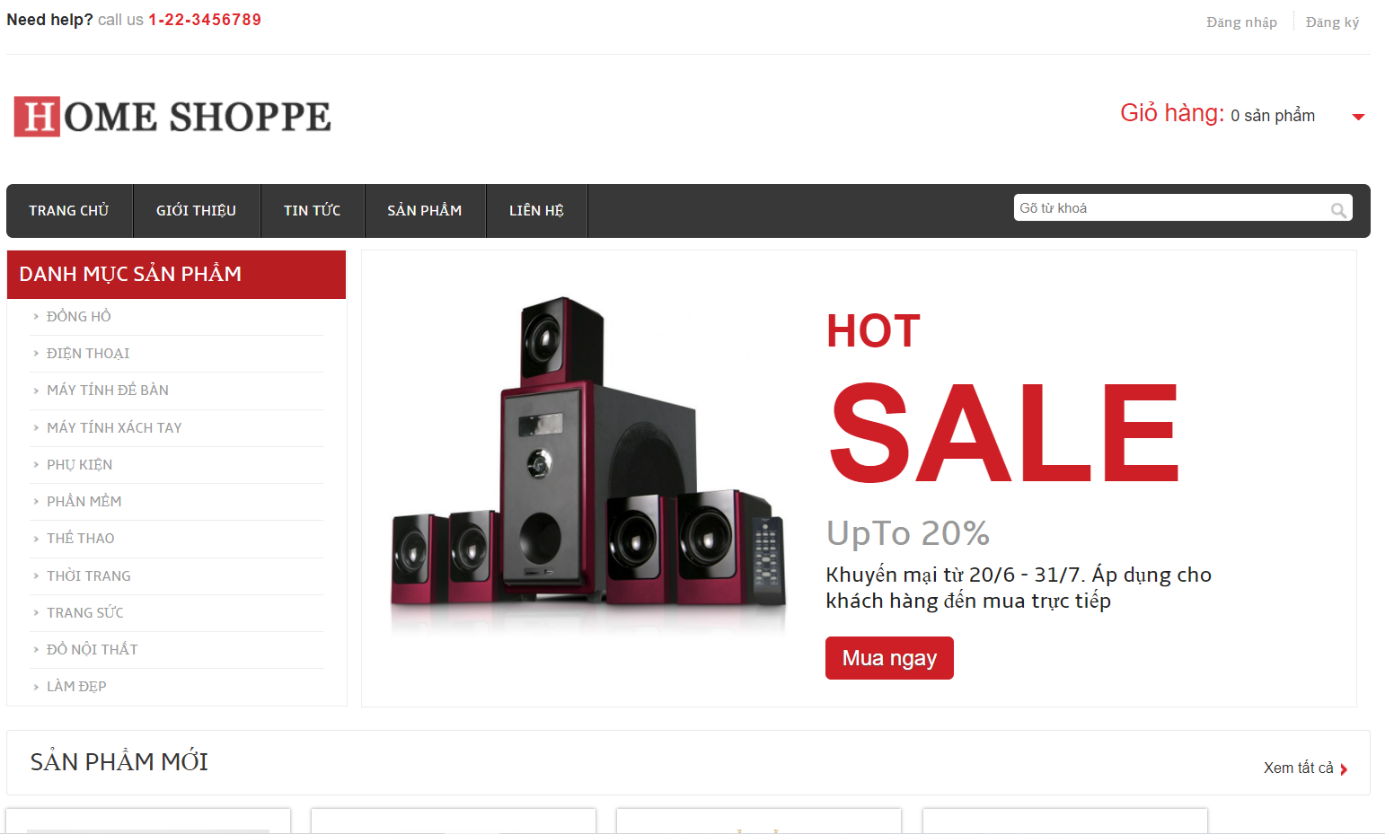
Hình 3.8 Thông báo khi thanh toán thành công

* Chi tiết hóa đơn sẽ được gửi đến cho khách hàng qua gmail

*Ảnh có chứa văn bản

Mô tả được tạo tự độngHình 3.9 Chi tiết hóa đơn được gửi tới gmail khách hàng*

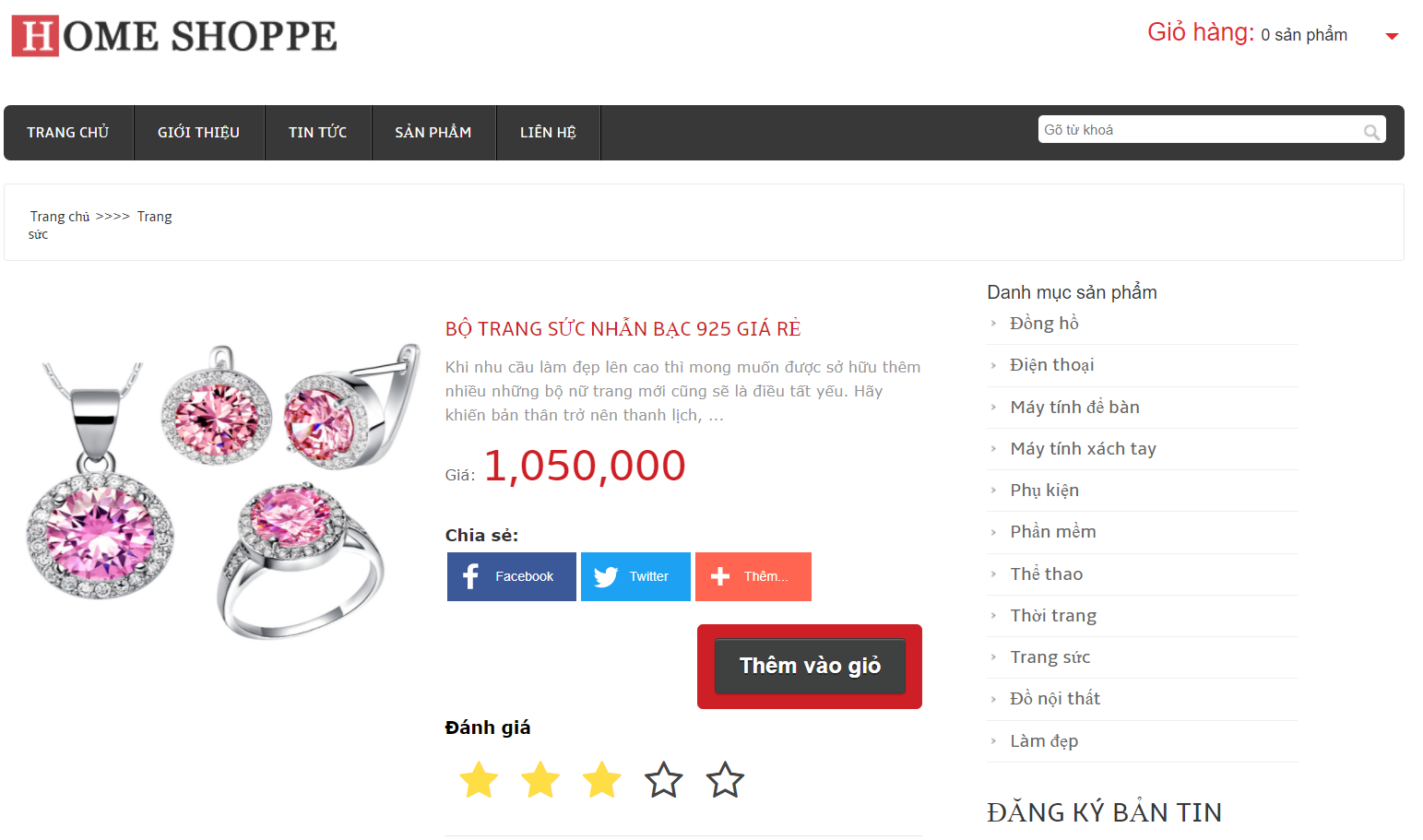
### **3.1.7 Giao diện trang chủ**



*Hình 3.10 Giao diện trang chủ*

Người dùng có thể xem các sản phẩm theo thể loại, các sản phẩm mới ra và các sản phẩm hot nhất hiện nay

### **3.1.8 Giao diện trang chi tiết sản phẩm**



*Hình 3.11 Giao diện trang chi tiết sản phẩm*

Người dùng có thể xem hình ảnh và mô tả chi tiết của sản phẩm trước khi thêm vào giỏ hàng

### **3.1.9 Giao diện trang tin tức**

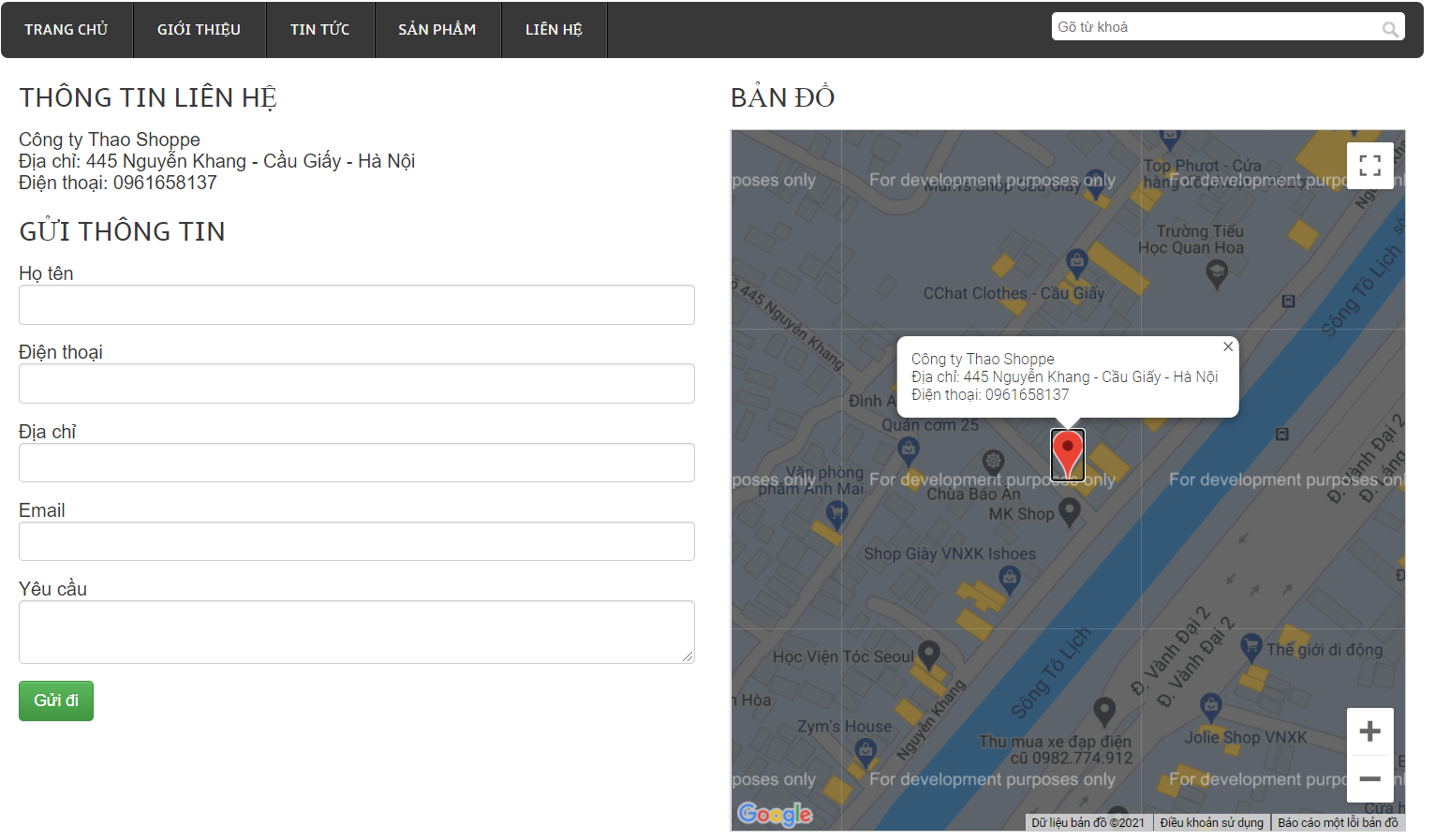
Ảnh có chứa văn bản

Mô tả được tạo tự động

*Hình 3.12 Giao diện trang tin tức*

Người dùng có thể xem tin tức về những sản phẩm mới nhất, tính năng mới của sản phẩm…

### **3.1.10 Giao diện trang liên hệ**



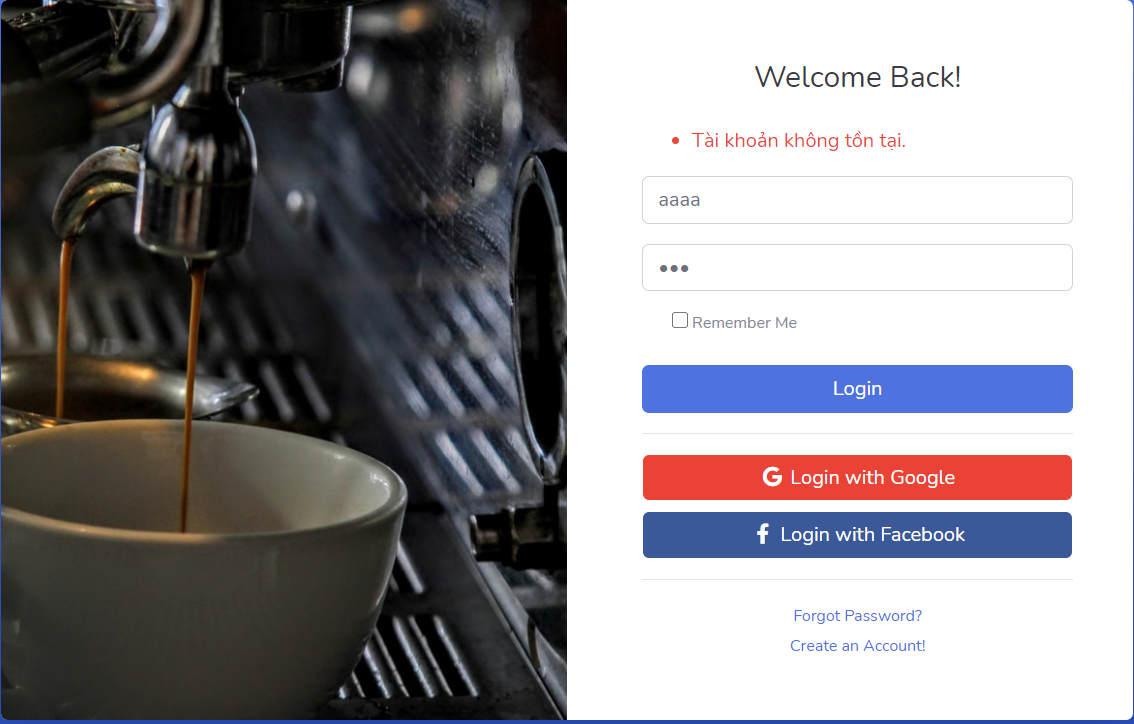
*Hình 3.13 Giao diện trang liên hệ*

Người dùng có thể xem địa chỉ của cửa hàng để đến trực tiếp trải nghiệm mua sắm

## **3.2 Xây dựng chức năng cho quản trị viên**

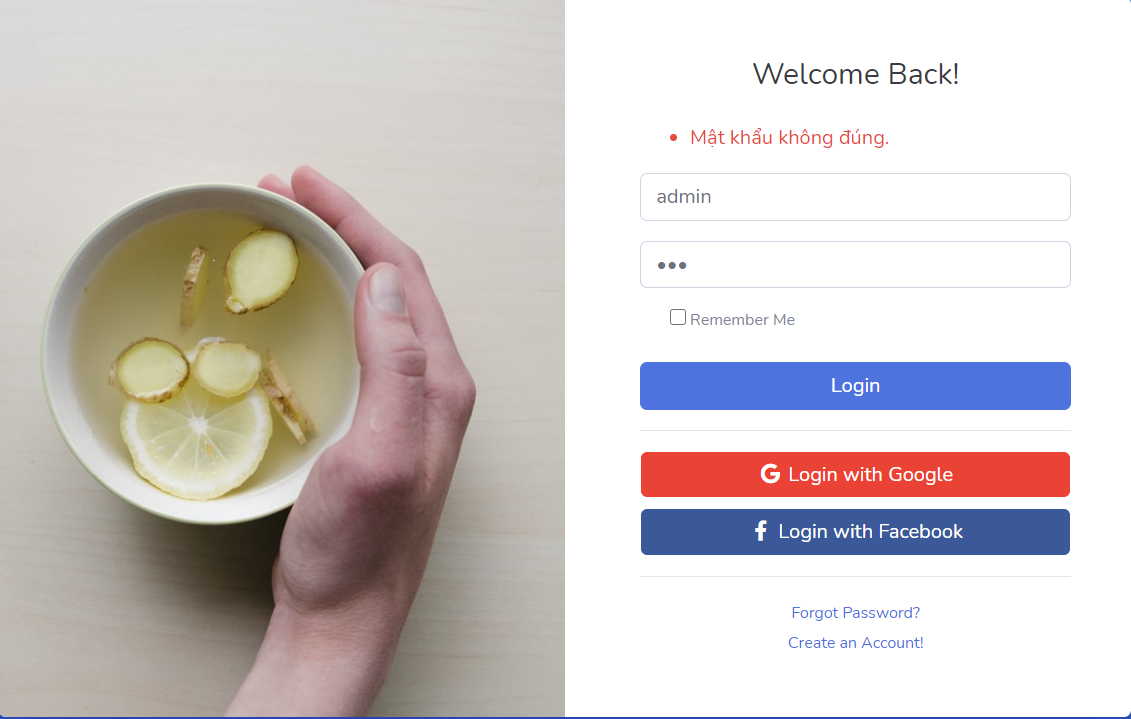
### **3.2.1 Chức năng đăng nhập của người quản trị**

* Trường hợp 1: Tài khoản không tồn tại



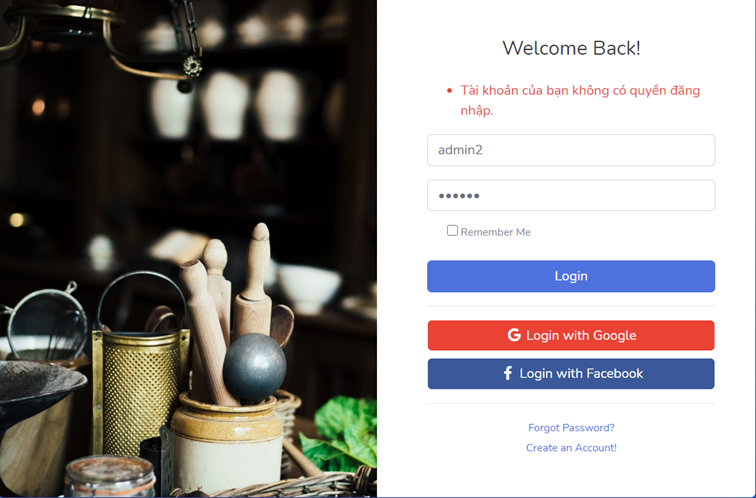
Hình 3.14 Giao diện trang quản trị khi tài khoản không tồn tại

* Người quản trị nhập một tài khoản bất kì chưa được đăng kí thì sẽ hiển thị ngay thông báo tài khoản không tồn tại
* Trường hợp 2: Đúng tên đăng nhập nhưng sai mật khẩu



Hình 3.15 Giao diện trang quản trị khi mật khẩu không đúng

* Người quản trị nhập đúng tên đăng nhập nhưng sai mật khẩu sẽ hiển thị thông báo mật khẩu không đúng
* Trường hợp 3: Tài khoản không có quyền đăng nhập (không phải tài khoản của người quản trị)



Hình 3.16 Giao diện trang quản trị khi tài khoản không có quyền đăng nhập

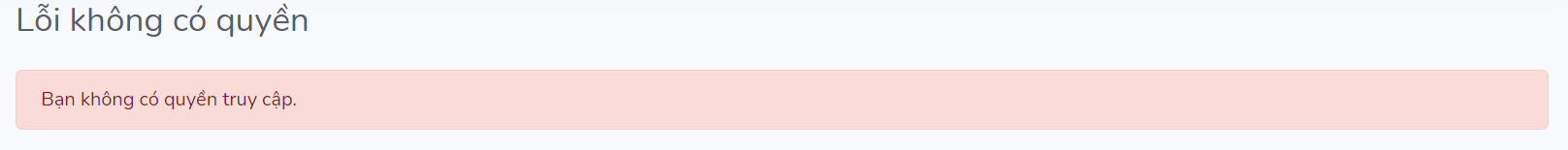
* Người quản trị nhập đúng tài khoản nhưng tài khoản lại là tài khoản người dùng
* Trường hợp 4: Đúng tài khoản người quản trị nhưng tài khoản đang bị khóa

Ảnh có chứa trong nhà, ảnh chụp màn hình

Mô tả được tạo tự động

Hình 3.17 Giao diện trang quản trị khi tài khoản đang bị khóa

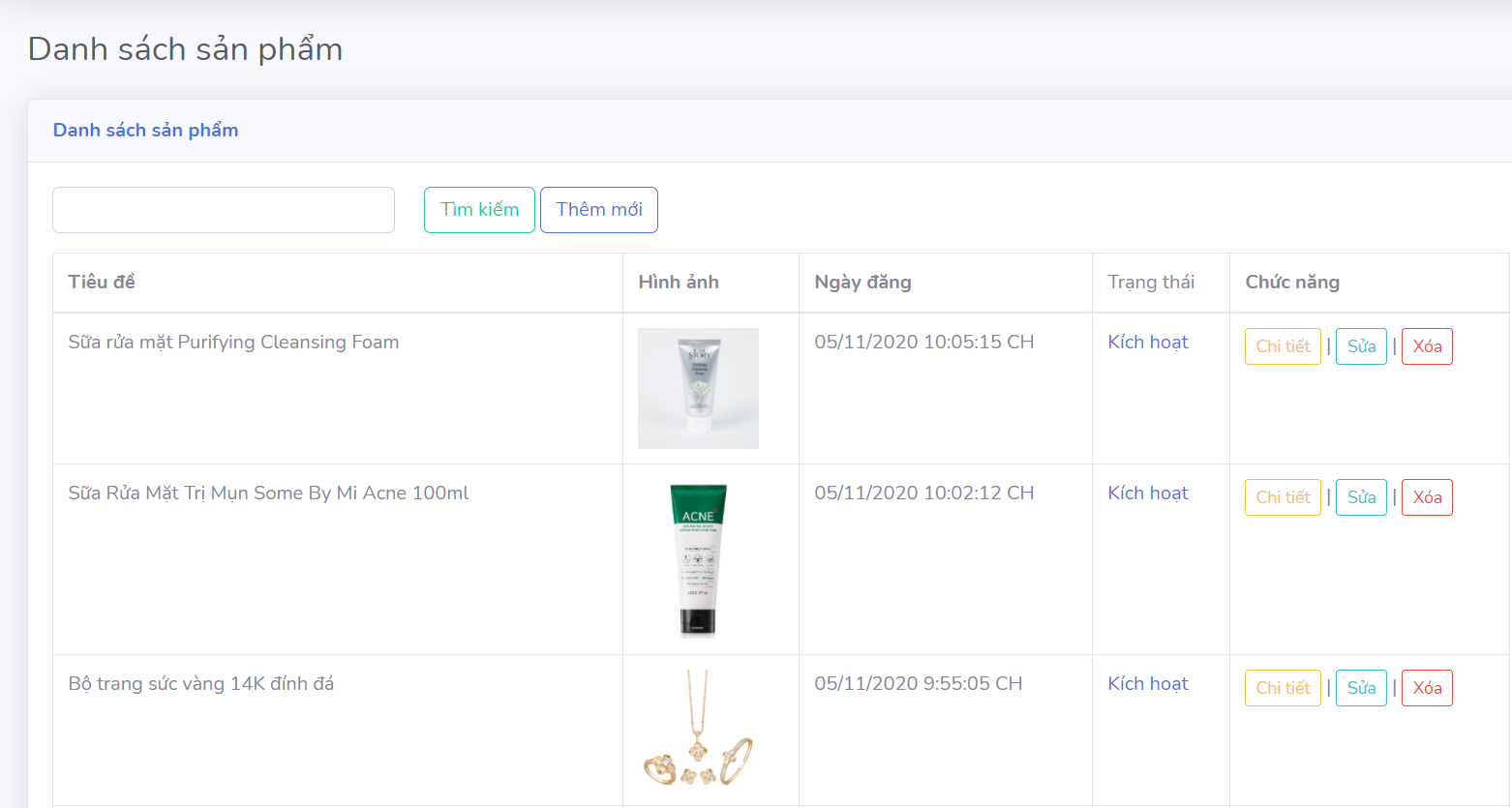
* Người quản trị nhập đúng tài khoản nhưng tài khoản đang bị khóa
* Trường hợp 5: Đúng tài khoản nhưng là tài khoản người điều hành



Hình 3.18 Người điều hành không có quyền vào trang quản lý tài khoản

* Người điều hành có thể vào được tất cả mọi trang quản lý trừ trang quản lý tài khoản
* Trường hợp 6: Đúng tài khoản người quản trị và tài khoản không bị khóa, chuyển hướng tới trang quản trị

### **3.2.2 Chức năng thêm, sửa, xóa, tìm kiếm sản phẩm**

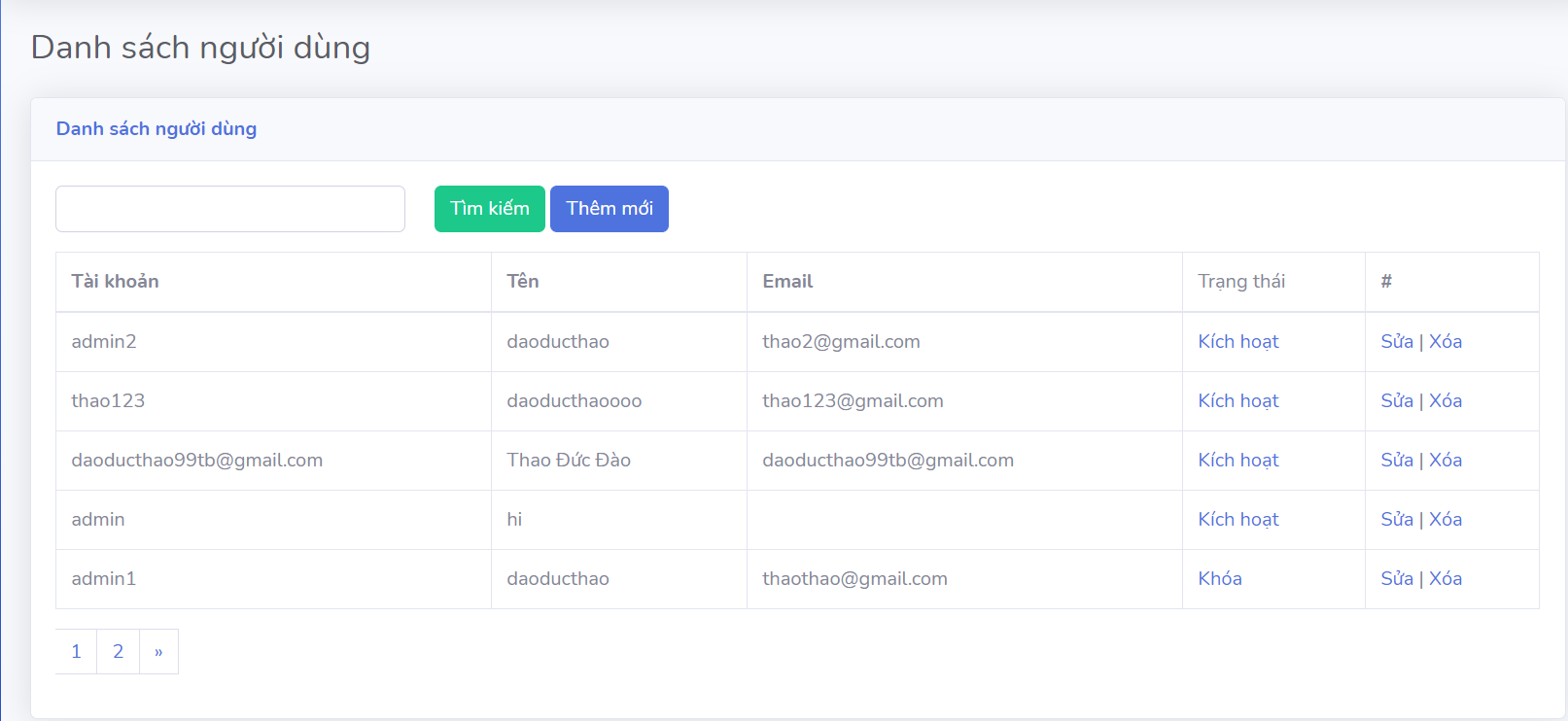


Hình 3.19 Giao diện trang quản lý sản phẩm

**Quy trình thực hiện:**

* Người quản trị có thể thêm mới, cập nhật, xóa sản phẩm trong website

### **3.2.3 Chức năng phân quyền tài khoản**

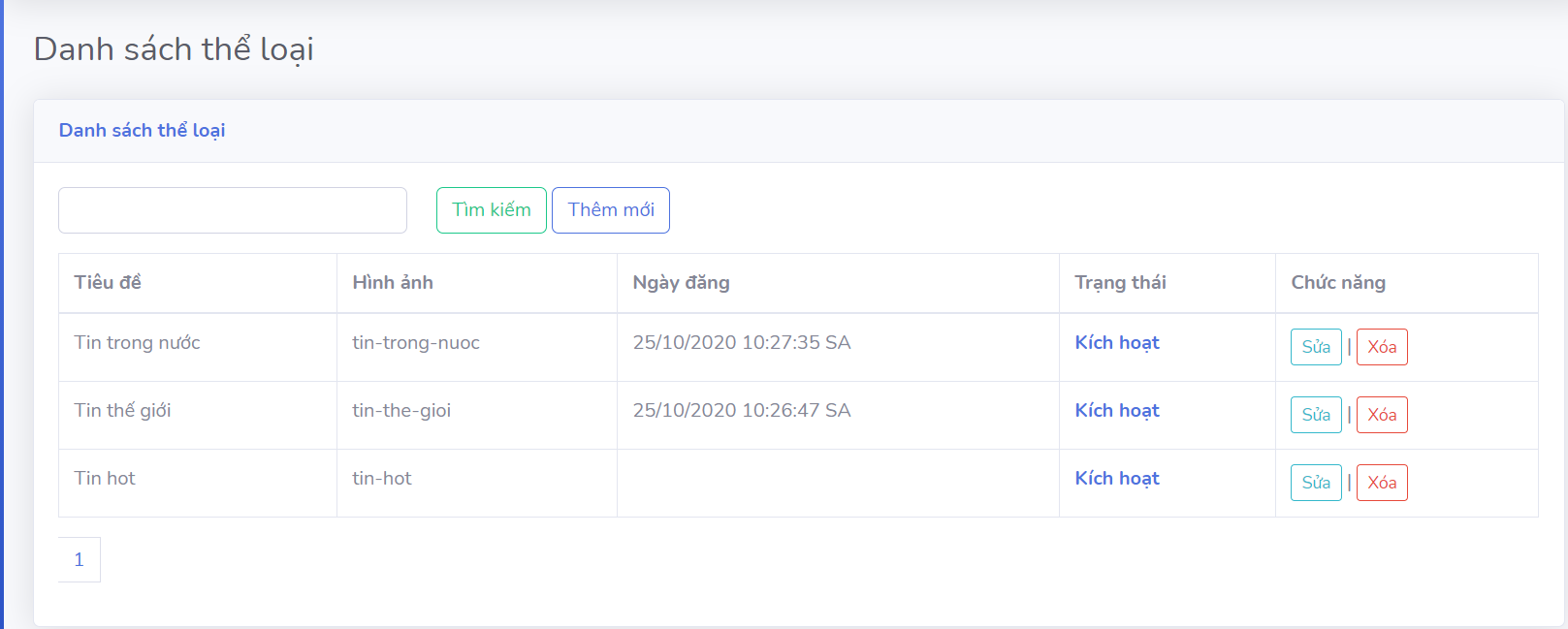


*Hình 3.20 Giao diện trang quản lý tài khoản*

**Quy trình thực hiện:**

* Người quản trị có thể thêm mới, cập nhật, tìm kiếm, xóa tài khoản
* Với những tài khoản bị khóa, người quản trị có thể cho kích hoạt trở lại, với những tài khoản người dùng thì phân quyền không cho phép đăng nhập vào trang quản trị

### **3.2.4 Chức năng tạo mới tin tức**



*Hình 3.21 Giao diện trang quản lý loại tin tức*

Ảnh có chứa văn bản

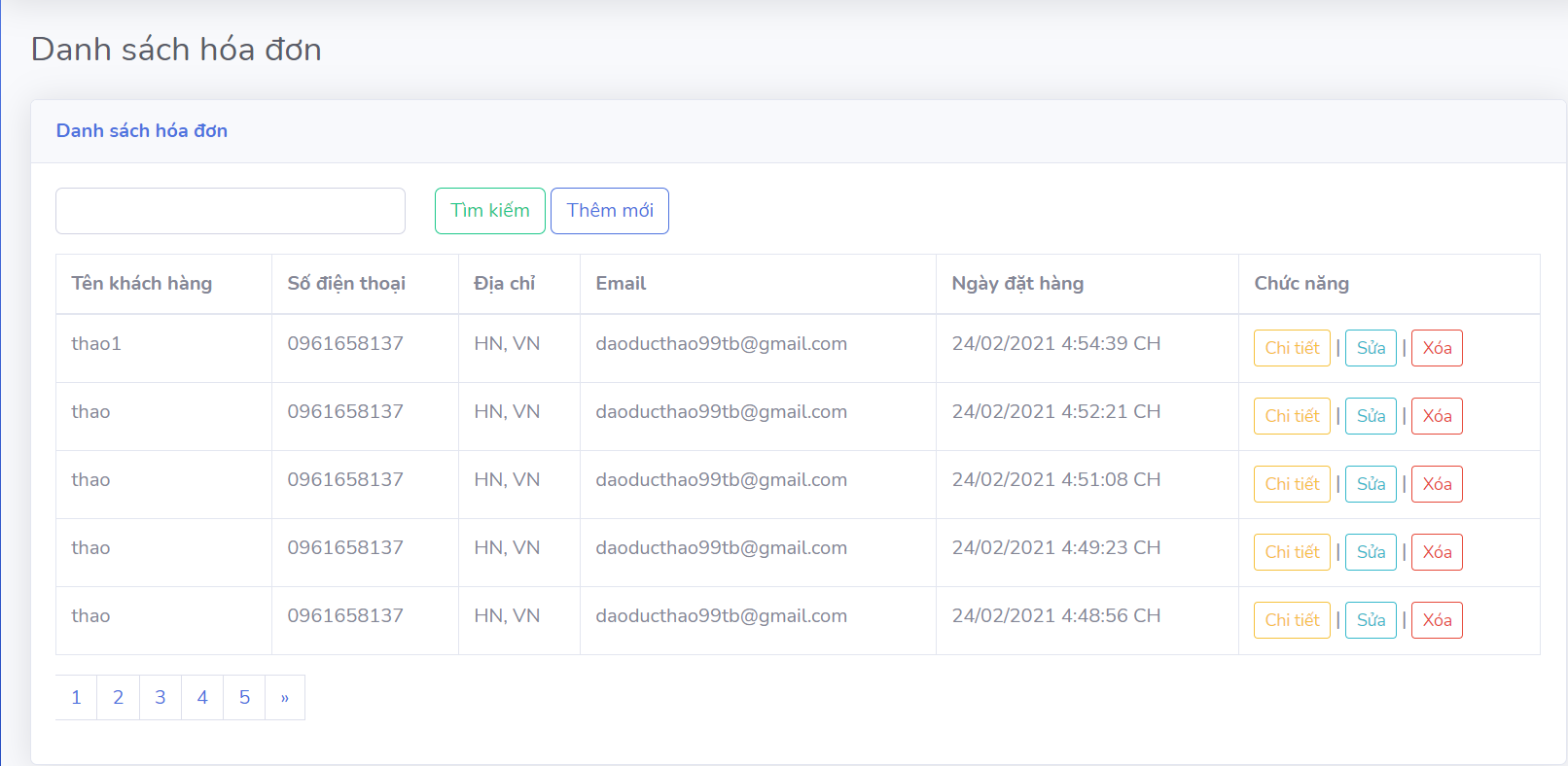
Mô tả được tạo tự động

*Hình 3.22 Giao diện trang quản lý nội dung tin tức*

**Quy trình thực hiện:**

* Người quản trị có thể thêm mới loại tin tức, trong từng loại tin tức có thể thêm nội dung chi tiết của tin tức

### **3.2.5 Chức năng quản lí hóa đơn**



*Hình 3.23 Giao diện trang quản lý hóa đơn*

**Quy trình thực hiện:**

* Người quản trị có thể xem chi tiết các hóa đơn mà khách hàng đã đặt mua, sửa thông tin hóa đơn nếu có sai sót và được khách hàng phản ánh lại.

# **KẾT LUẬN**

Trong quá trình thực hiện đề tài ***Xây dựng web bán hàng Home Shoppe***, do phải tiếp cận với nhiều công nghệ nền web, các thư viện mã nguồn mở liên quan nên em đã gặp phải không ít khó khăn. Tuy nhiên với sự giúp đỡ của thầy cô, anh chị, bạn bè cùng với sự nỗ lực của bản thân đồ án đã được hoàn thành đúng hạn và thu được kết quả khả quan giúp người dùng có được website tương đối tốt và hữu ích trong cuộc sống

Kết thúc đề tài em đã tích góp được cho mình nhiều kinh nghiệm và kiến thức quý báu như lập trình website với các thư viện mã nguồn mở, cách sử dụng api của Google, Facebook,cách phân quyền người dùng, tối ưu SEO cho website. Những kiến thức này sẽ là những hành trang quý báu trong công việc tương lai của bản thân

Bên cạnh đó do thời gian thực hiện có hạn nên một vài phần của đồ án chưa được khả quan như mong đợi, ví dụ như một số chức năng cần phải mở rộng phát triển thêm. Trong thời gian tới, em sẽ tiếp tục nghiên cứu và phát triển website này nhằm tăng các tính năng và tối ưu hóa tốc độ xử lý để đem lại hiểu quả cao hơn, tối ưu lại giao diện để mang lại trải nghiệm tốt nhất và phát triển để đưa website vào thực tế. Tiếp tục nghiên cứu cải thiện trải nghiệm người dùng, cập nhật phát triền thêm các tính năng như gửi bình luận sản phẩm, cập nhật thêm các thông tin của người dùng qua đó sử dụng AI để thực hiện tìm hiểu tâm lý mua hàng của khách hàng,…

# **TÀI LIỆU THAM KHẢO**

**Các tài liệu tiếng việt**

[1] Slide dạy học môn Phân tích thiết kế hệ thống thông tin - Ths. Nguyễn Đức Dư - ĐH Giao Thông Vận Tải

[2] Bài giảng môn đặc tả phần mềm TS.Trần Văn Dũng - ĐH Giao Thông Vận Tải

[3] Bài giảng môn Phân tích thiết kế hướng đối tượng TS.Nguyễn Hiếu Cường - ĐH Giao Thông Vận Tải.

[4] Bài giảng môn lập trình web – Ths. Nguyễn Kim Sao - ĐH Giao Thông Vận Tải

**Tài liệu tiếng anh**

[5] <https://developer.facebook.com/>

[6] https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/

[7] <https://stackoverflow.com/>