TRƯỜNG ĐẠI HỌC NHA TRANG **KHOA CÔNG NGHỆ THÔNG TIN**



**XÂY DỰNG WEBSITE QUẢN LÝ CỬA HÀNG**

**GVHD : Ths. Phạm Thị Kim Ngoan**

**SVTH : Nguyễn Tuấn Tú**

**MSSV : 63135967**

**Lớp : 63.CNTT-5**

Khánh Hòa - 2023

**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 4](#_Toc152518599)

[1. Lý do chọn đề tài: 4](#_Toc152518600)

[2. Mục tiêu của đề tài: 4](#_Toc152518601)

[3. Giới hạn và phạm vi của đề tài: 4](#_Toc152518602)

[4. Tính khả thi: 4](#_Toc152518603)

[5. Khảo sát hiện trạng: 4](#_Toc152518604)

[CHƯƠNG 2. CỞ SỞ LÝ THUYẾT 6](#_Toc152518605)

[1. Tổng quan về ngôn ngữ Asp.NET MVC 6](#_Toc152518606)

[1.1Giới thiệu Asp.NET MVC Framework 6](#_Toc152518607)

[1.2 Lý do chọn ASP.NET MVC 9](#_Toc152518608)

[2. Tổng quan về SQL Server 10](#_Toc152518609)

[2.1 Khái niệm hệ quản trị CSDL SQL Server 10](#_Toc152518610)

[2.2 Đặc điểm của SQL Server và đối tượng làm việc 11](#_Toc152518611)

[CHƯƠNG 3: NỘI DUNG THỰC HIỆN 12](#_Toc152518612)

[1. Đối tượng sử dụng website 12](#_Toc152518613)

[2. Chức năng chính của website 12](#_Toc152518614)

[2.1. Chức năng của khách hàng: 12](#_Toc152518615)

[2.2. Chức năng của admin: 12](#_Toc152518616)

[3. Phân tích hệ thống 13](#_Toc152518617)

[3.1 Các yêu cầu chức năng chính của hệ thống 13](#_Toc152518618)

[3.2 Thiết kế và đặc tả giao diện 14](#_Toc152518619)

[CHƯƠNG 4: KẾT LUẬN 17](#_Toc152518620)

[1. Kết quả đạt được của đề tài 17](#_Toc152518621)

[2. Hạn chế của đề tài 17](#_Toc152518622)

[3. Kết luận 17](#_Toc152518623)

**DANH MỤC HÌNH**

[Hình 1 Mẫu Model-View-Controller 6](#_Toc152521213)

[Hình 2 Mô hình tuần tự của MVC 8](#_Toc152521214)

[Hình 3 Giao diện trang chủ 14](#_Toc152521216)

[Hình 4 Giao diện trang chi tiết sản phẩm 14](#_Toc152521217)

[Hình 5 Giao diện trang danh mục loại sản phẩm 15](#_Toc152521218)

[Hình 6 Giao diện danh mục nhà cung cấp 16](#_Toc152521219)

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1. Lý do chọn đề tài:

Trong thời đại công nghệ hiện đại, sự phát triển của công nghệ thông tin và sự gia tăng của người dùng laptop đã tạo ra nhu cầu ngày càng cao về phụ kiện laptop. Nhận thấy tiềm năng và cơ hội kinh doanh trong lĩnh vực này, chúng tôi đã quyết định chọn đề tài "Xây dựng trang web quản lý bán phụ kiện laptop trực tuyến cho cửa hàng Tuấn Tú Store".

## 2. Mục tiêu của đề tài:

Nâng cao kiến thức và kỹ năng trong việc xây dựng và phát triển trang web thương mại điện tử.

Tạo ra một giao diện thân thiện, dễ sử dụng cho khách hàng và nhân viên cửa hàng.

Cung cấp thông tin chi tiết và đầy đủ về các loại phụ kiện laptop.

Tạo ra một hệ thống đặt hàng và thanh toán an toàn, tiện lợi cho khách hàng.

Tăng khả năng cạnh tranh và mở rộng quy mô kinh doanh cho cửa hàng Tuấn Tú Store.

## 3. Giới hạn và phạm vi của đề tài:

Đề tài sẽ được triển khai ban đầu tại cửa hàng Tuấn Tú Store.

Đối tượng sử dụng bao gồm nhân viên admin và khách hàng.

## 4. Tính khả thi:

Chúng tôi có thể xây dựng trang web bán phụ kiện laptop theo yêu cầu của đề tài.

Chúng tôi có thể tích hợp các tính năng quản lý và giao dịch cần thiết để đáp ứng nhu cầu của cửa hàng Tuấn Tú Store.

## 5. Khảo sát hiện trạng:

Cửa hàng Tuấn Tú Store là một cửa hàng phụ kiện laptop uy tín và được biết đến tại địa phương. Tuy nhiên, cửa hàng gặp khó khăn trong việc quảng bá sản phẩm và tìm kiếm khách hàng mới. Hiện tại, cửa hàng Tuấn Tú Store chỉ hoạt động trong phạm vi cục bộ và không có một hệ thống quản lý và giao dịch trực tuyến chuyên nghiệp.

Việc xây dựng một trang web quản lý bán phụ kiện laptop trực tuyến sẽ giúp cửa hàng Tuấn Tú Store mở rộng quy mô kinh doanh, tiếp cận được đông đảo khách hàng tiềm năng và thuận tiện cho khách hàng trong việc mua sắm và đặt hàng. Trang web sẽ cung cấp thông tin chi tiết về các loại phụ kiện laptop, cho phép khách hàng tìm kiếm và chọn lựa sản phẩm theo nhu cầu của mình. Hơn nữa, tính năng đặt hàng và thanh toán trực tuyến sẽ tạo thuận lợi cho khách hàng và giúp cửa hàng quản lý đơn hàng một cách hiệu quả.

Xây dựng trang web quản lý bán phụ kiện laptop trực tuyến cho cửa hàng Tuấn Tú Store sẽ đáp ứng nhu cầu ngày càng tăng về mua sắm trực tuyến và giúp cửa hàng nâng cao khả năng cạnh tranh, tăng doanh thu và mở rộng thị trường kinh doanh.

# CHƯƠNG 2. CỞ SỞ LÝ THUYẾT

## 1. Tổng quan về ngôn ngữ ASP.NET MVC

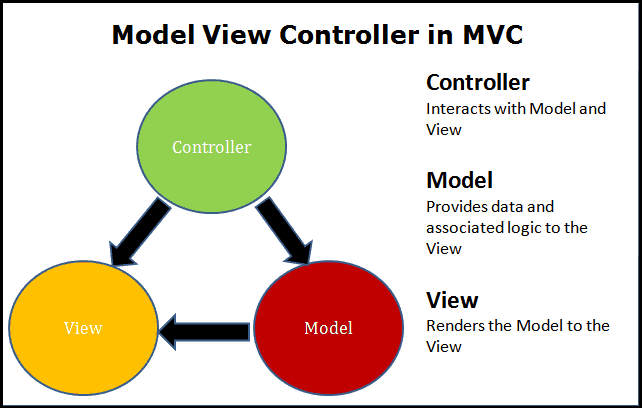
Mô hình kiến trúc Model-View-Controller (MVC) được sử dụng để phân chia ứng dụng thành ba thành phần chính: model, view và controller. Nền tảng ASP.NET MVC cung cấp cho chúng ta khả năng phát triển các ứng dụng web theo mô hình MVC, thay vì sử dụng mô hình ASP.NET Web Forms truyền thống. ASP.NET MVC có những ưu điểm nổi bật như nhẹ nhàng (lighweight), dễ kiểm thử giao diện người dùng (so với ứng dụng Web Forms), và tích hợp các tính năng sẵn có của ASP.NET. Nền tảng ASP.NET MVC được xác định trong namespace System.Web.Mvc và là một phần của namespace System.Web.

MVC là một mẫu thiết kế chuẩn mà nhiều lập trình viên đã quen thuộc. Có một số loại ứng dụng web phù hợp với kiến trúc MVC, trong khi một số khác vẫn sử dụng ASP.NET Web Forms và cơ chế postbacks. Đôi khi, có những ứng dụng kết hợp cả hai kiến trúc này.

### 1.1 Giới thiệu Asp.NET MVC Framework

ASP.NET là một nền tảng ứng dụng web được phát triển và cung cấp bởi Microsoft, cho phép nhà phát triển tạo ra các trang web động, ứng dụng web và dịch vụ web. ASP.NET MVC là một phần của ASP.NET, cho phép nhà phát triển xây dựng ứng dụng web dựa trên mô hình thiết kế MVC.

Mô hình MVC (Model-View-Controller) là một mô hình thiết kế chuẩn được sử dụng để phân chia một ứng dụng thành ba thành phần chính. Các thành phần này bao gồm:



Hình 1 Mẫu Model-View-Controller

**Model:** Models đóng vai trò quan trọng trong việc thiết lập logic cho phần dữ liệu. Thông thường, các đối tượng model được sử dụng để truy xuất và lưu trạng thái của dữ liệu trong Cơ sở dữ liệu. Ví dụ, khi làm việc với một đối tượng Product (sản phẩm), tôi sẽ lấy dữ liệu từ CSDL, thực hiện các thao tác và sau đó cập nhật lại dữ liệu vào bảng Products trong SQL Server.

Tuy nhiên, trong các ứng dụng nhỏ, việc sử dụng model thường chỉ là một khái niệm để phân biệt phần dữ liệu, mà không cần thiết phải triển khai một tầng model và các lớp liên quan. Ví dụ, nếu ứng dụng chỉ cần đọc dữ liệu từ CSDL và truyền chúng cho view, tôi không cần phải có một tầng model và các lớp tương ứng. Trong trường hợp này, tôi có thể coi dữ liệu như một đối tượng model (thay vì tầng model).

**View:** Views đóng vai trò quan trọng trong việc hiển thị giao diện người dùng (UI). Views được sử dụng để xây dựng các thành phần giao diện như hộp văn bản, danh sách thả xuống (drop-down list) và các ô kiểm (checkbox) dựa trên thông tin dữ liệu từ model. Ví dụ, nếu tôi cần tạo một view để cập nhật bảng Products, thì view đó sẽ hiển thị các thành phần giao diện như hộp văn bản, danh sách thả xuống và các ô kiểm dựa trên trạng thái hiện tại của một đối tượng Product.

**Controller:** Trong một ứng dụng MVC, controllers được đảm nhận vai trò quản lý tương tác người dùng, làm việc với model và lựa chọn view để hiển thị giao diện người dùng.

Controllers xử lý đầu vào và tương tác người dùng, chẳng hạn như nhận dữ liệu từ người dùng (giá trị query-string) và truyển dữ liệu đó cho model. Model sau đó truy xuất dữ liệu từ cơ sở dữ liệu dựa trên các giá trị này. Controllers cũng quyết định view nào sẽ được render và hiển thị cho người dùng.

Mô hình MVC giúp phân chia các khía cạnh khác nhau của ứng dụng thành các thành phần riêng biệt (logic đầu vào, logic xử lý tác vụ và logic giao diện người dùng). Mỗi loại logic được giao cho một thành phần cụ thể: views xử lý logic giao diện người dùng, controllers xử lý logic đầu vào, và models xử lý logic kinh doanh (mục đích chính của ứng dụng). Phân chia này giảm độ phức tạp của ứng dụng và cho phép nhà phát triển tập trung vào từng khía cạnh cụ thể. Ví dụ, bạn có thể tập trung vào giao diện người dùng (views) mà không cần quan tâm đến logic xử lý dữ liệu của ứng dụng.

Mô hình MVC cũng tạo điều kiện cho việc kiểm thử dễ dàng hơn so với các mô hình khác như Web Forms. Trong ứng dụng Web Forms, một lớp thông thường được sử dụng để hiển thị thông tin cho người dùng và xử lý dữ liệu người dùng nhập vào. Việc xây dựng các bộ kiểm thử tự động cho ứng dụng Web Forms rất phức tạp, vì bạn cần khởi tạo các đối tượng trang, các điều khiển được sử dụng trên trang và các lớp phụ thuộc khác trong ứng dụng. Với MVC, các thành phần được phân tách và sử dụng giao diện (interface), cho phép kiểm thử các thành phần riêng biệt trong một môi trường cô lập với các yếu tố còn lại của ứng dụng.

Sự phân chia ba thành phần của MVC cũng cho phép phát triển song song. Ví dụ, một nhà phát triển có thể làm việc với view, nhà phát triển thứ hai có thể triển khai logic của controller, và nhà phát triển thứ ba có thể tập trung vào logic tác vụ của model cùng một lúc.

A diagram of a software system

Description automatically generated

Hình Mô hình tuần tự của MVC

Để minh họa điều này, đoạn văn bản cung cấp ví dụ về một thành phần GUI đơn giản là checkbox. Checkbox có thành phần model để quản lý trạng thái (được chọn hoặc không được chọn), thành phần view để hiển thị nó trên màn hình với trạng thái tương ứng, và thành phần controller để xử lý các sự kiện khi có tương tác từ người dùng hoặc các đối tượng khác đối với checkbox.

Khi người dùng nhấp vào checkbox, thành phần controller của checkbox xử lý sự kiện này và yêu cầu thành phần model thay đổi dữ liệu trạng tháĐoạn văn bản trên giải thích về khái niệm controllers trong mô hình kiến trúc MVC (Model-View-Controller). Trong một ứng dụng MVC, controllers được sử dụng để quản lý tương tác của người dùng, làm việc với model và quyết định view để hiển thị giao diện người dùng.

Controllers xử lý đầu vào và tương tác của người dùng, ví dụ như nhận dữ liệu từ người dùng (giá trị truy vấn) và truyền dữ liệu này cho model. Model sau đó truy xuất dữ liệu từ cơ sở dữ liệu dựa trên các giá trị này. Controllers cũng quyết định view nào sẽ được hiển thị cho người dùng.

Mô hình MVC giúp phân chia các khía cạnh khác nhau của ứng dụng thành các thành phần riêng biệt (logic đầu vào, logic xử lý, logic giao diện người dùng). Mỗi thành phần có nhiệm vụ cụ thể: views xử lý logic giao diện người dùng, controllers xử lý logic đầu vào, và models xử lý logic kinh doanh (mục đích chính của ứng dụng). Phân chia này giúp giảm độ phức tạp của ứng dụng và cho phép nhà phát triển tập trung vào từng khía cạnh cụ thể. Ví dụ, bạn có thể tập trung vào giao diện người dùng (views) mà không phải lo lắng về logic xử lý dữ liệu của ứng dụng.

Mô hình MVC cũng giúp dễ dàng kiểm thử hơn so với các mô hình khác như Web Forms. Trong ứng dụng Web Forms, thường sử dụng một lớp duy nhất để hiển thị thông tin cho người dùng và xử lý đầu vào của người dùng. Viết kiểm thử tự động cho ứng dụng Web Forms có thể phức tạp vì bạn cần khởi tạo các đối tượng trang, các điều khiển được sử dụng trên trang và các lớp phụ thuộc. Với MVC, các thành phần được phân chia và sử dụng giao diện, cho phép kiểm thử các thành phần riêng biệt trong một môi trường cô lập.

Sự phân chia ba thành phần của MVC cũng giúp phát triển song song. Ví dụ, một nhà phát triển có thể làm việc với view, một nhà phát triển khác có thể triển khai logic của controller và một nhà phát triển thứ ba có thể tập trung vào logic kinh doanh của model cùng một lúc.

Để minh họa điều này, đoạn văn bản sử dụng ví dụ về một thành phần GUI đơn giản là checkbox. Checkbox có thành phần model để quản lý trạng thái (được chọn hoặc không được chọn), thành phần view để hiển thị checkbox trên màn hình với trạng thái tương ứng và thành phần controller để xử lý các sự kiện khi checkbox được tương tác bởi người dùng hoặc các đối tượng khác.

### 1.2 Lý do chọn ASP.NET MVC

Tách bạch các tác vụ của ứng dụng: ASP.NET MVC cho phép tách bạch các tác vụ của ứng dụng như logic nhập liệu, business logic, và logic giao diện. Điều này giúp dễ dàng kiểm thử và áp dụng phát triển theo hướng TDD (Test-driven development). Bạn có thể kiểm thử unit-test cho ứng dụng mà không cần chạy controller trong tiến trình ASP.NET, giúp tiết kiệm thời gian và công sức.

Khả năng mở rộng và nhúng thêm: ASP.NET MVC là một nền tảng khả mở rộng và khả nhúng. Các thành phần của nó có thể dễ dàng thay thế hoặc tùy chỉnh. Bạn có thể nhúng thêm view engine, cơ chế định tuyến 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), giúp việc kiểm thử dễ dàng hơn.

Ánh xạ URL mạnh mẽ: ASP.NET MVC cung cấp thành phần ánh xạ URL mạnh mẽ, cho phép xây dựng các địa chỉ URL ngắn gọn và dễ tìm kiếm. Các địa chỉ URL không cần phải có phần mở rộng của tên tập tin và được thiết kế để tối ưu hóa tìm kiếm và phù hợp với lập địa chỉ theo kiểu REST.

Hỗ trợ sử dụng các đặc tả của ASP.NET: ASP.NET MVC hỗ trợ sử dụng các đặc tả của các trang ASP.NET như trang .aspx, điều khiển người dùng .ascx và trang master .master. Bạn có thể sử dụng các tính năng có sẵn của ASP.NET như lồng các trang master, sử dụng server controls, data-binding, địa phương hóa và nhiều hơn nữa.

Hỗ trợ các tính năng có sẵn của ASP.NET: ASP.NET MVC hỗ trợ sử dụng 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 và nhiều tính năng khác.

Razor View Engine: ASP.NET MVC5 bổ sung một view engine mới là Razor View Engine, giúp thiết lập các view nhanh chóng, dễ dàng và ít tốn công sức hơn so với việc sử dụng Web Forms view engine.

Tóm lại, ASP.NET MVC có nhiều tính năng và lợi ích như tách bạch tác vụ, khả năng mở rộng, ánh xạ URL mạnh mẽ, sử dụng các đặc tả của ASP.NET, hỗ trợ các tính năng có sẵn của ASP.NET và Razor View Engine. Đây là những lý do tại sao nên chọn ASP.NET MVC để phát triển ứng dụng web.

## 2. Tổng quan về SQL Server

### 2.1 Khái niệm hệ quản trị CSDL SQL Server

SQL Server là một hệ thống quản lý cơ sở dữ liệu RDBMS được sử dụng để lưu trữ, truy xuất và quản lý dữ liệu. Nó sử dụng ngôn ngữ Transact-SQL (T-SQL) để trao đổi dữ liệu giữa máy tính Client và máy tính SQL Server. Một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) bao gồm các thành phần như Databases, Database Engine và các ứng dụng quản lý dữ liệu khác.

SQL Server được tối ưu để chạy trên môi trường cơ sở dữ liệu rất lớn, có thể lên đến Tera-Byte và phục vụ hàng ngàn người dùng cùng lúc. Ngoài ra, SQL Server có khả năng kết hợp với các máy chủ khác như Microsoft Internet Information Server (IIS), ECommerce Server, Proxy Server để cung cấp các giải pháp toàn diện cho các ứng dụng web và doanh nghiệp.

### 2.2 Đặc điểm của SQL Server và đối tượng làm việc

SQL Server có các đặc điểm và đối tượng làm việc quan trọng. SQL là một ngôn ngữ tựa tiếng Anh, dễ hiểu và ít gây lỗi. Nó cung cấp các lệnh phong phú để thực hiện các công việc như chèn, cập nhật, xoá dữ liệu trong một bảng, tạo, sửa đổi và xoá các đối tượng trong cơ sở dữ liệu, điều khiển truy cập và đảm bảo tính bảo mật, nhất quán và ràng buộc của cơ sở dữ liệu.

Trong SQL Server, người dùng có thể làm việc với các đối tượng như bảng, thủ tục lưu trữ, chức năng, chỉ mục và ràng buộc. Các đối tượng này giúp tổ chức và quản lý dữ liệu trong cơ sở dữ liệu.

Để sử dụng SQL Server hiệu quả, người dùng cần hiểu cấu trúc cơ sở dữ liệu của mình và nắm vững các câu lệnh SQL để thực hiện các tác vụ truy vấn, cập nhật và quản lý dữ liệu.

# CHƯƠNG 3: NỘI DUNG THỰC HIỆN

## 1. Đối tượng sử dụng website

**Đối với khách hàng:** Khách hàng có thể truy cập vào website để xem và mua hàng. Họ không cần đăng ký và đăng nhập để thực hiện các hoạt động này.

**Đối với admin:** Admin có quyền xem, thêm, sửa, tìm kiếm sản phẩm, nhà cung cấp và danh mục sản phẩm.

## 2. Chức năng chính của website

### 2.1. Chức năng của khách hàng:

**a. Xem hàng:** Khách hàng có thể lựa chọn sản phẩm phụ kiện laptop bằng cách click vào tên sản phẩm đã được hiển thị trên thanh trượt. Sau đó, hệ thống sẽ xử lý và tìm kiếm thông tin sản phẩm trong cơ sở dữ liệu. Sau khi xử lý hoàn tất, khách hàng sẽ có thể xem chi tiết thông tin về sản phẩm đó, bao gồm tên sản phẩm, mã sản phẩm, số lượng còn lại, mô tả, giá bán, sản phẩm tương tự và bình luận của những khách hàng khác.

**b. Mua hàng:** Sau khi chọn sản phẩm, khách hàng có thể mua hàng trực tiếp, khách hàng có thể thay đổi địa điểm nhận hàng hoặc để địa chỉ mặc định mà lúc đầu khách hàng đã điền. Hệ thống sẽ xử lý thông tin sản phẩm để ghi nhận hóa đơn vào cơ sở dữ liệu. Hiển thị thông tin về sản phẩm mà khách hàng đã đặt hàng.

### 2.2. Chức năng của admin:

**a. Quản lý danh mục sản phẩm:** Admin có quyền xem, thêm, sửa và xóa danh mục sản phẩm

Xem: Admin có thể xem danh sách tất cả các danh mục sản phẩm từ cơ sở dữ liệu. Mỗi danh mục hiển thị tên, mô tả và số lượng sản phẩm thuộc danh mục đó.

Thêm: Admin có thể thêm danh mục sản phẩm mới vào cơ sở dữ liệu. Khi thêm, admin cung cấp tên và mô tả cho danh mục mới.

Sửa: Admin có thể sửa thông tin của danh mục sản phẩm. Khi sửa, admin có thể thay đổi tên và mô tả của danh mục.

Xóa: Admin có khả năng xóa danh mục sản phẩm khỏi cơ sở dữ liệu. Trước khi xóa, hệ thống sẽ yêu cầu xác nhận từ admin để đảm bảo rằng việc xóa là chính xác.

**b. Quản lý nhà cung cấp:** Admin có quyền xem, thêm, sửa và xóa thông tin nhà cung cấp

Xem: Admin có thể xem danh sách tất cả các nhà cung cấp từ cơ sở dữ liệu. Mỗi nhà cung cấp hiển thị tên, địa chỉ, số điện thoại và số lượng sản phẩm mà nhà cung cấp đó cung cấp.

Thêm: Admin có thể thêm thông tin của nhà cung cấp mới vào cơ sở dữ liệu. Khi thêm, admin cung cấp tên, địa chỉ, số điện thoại và các chi tiết khác cho nhà cung cấp mới.

Sửa: Admin có thể sửa thông tin của nhà cung cấp. Khi sửa, admin có thể thay đổi tên, địa chỉ, số điện thoại và các chi tiết khác của nhà cung cấp.

Xóa: Admin có khả năng xóa thông tin của nhà cung cấp khỏi cơ sở dữ liệu. Trước khi xóa, hệ thống sẽ yêu cầu xác nhận từ admin để đảm bảo rằng việc xóa là chính xác.

**c. Quản lý sản phẩm:** Admin có quyền xem, thêm, sửa và xóa thông tin sản phẩm

Xem: Admin có thể xem danh sách tất cả các sản phẩm từ cơ sở dữ liệu. Mỗi sản phẩm hiển thị tên sản phẩm, giá, mô tả, danh mục sản phẩm và nhà cung cấp.

Thêm: Admin có thể thêm sản phẩm mới vào cơ sở dữ liệu. Khi thêm, admin cung cấp thông tin như tên sản phẩm, giá, mô tả, danh mục sản phẩm và nhà cung cấp cho sản phẩm mới.

Sửa: Admin có thể sửa thông tin của sản phẩm. Khi sửa, admin có thể thay đổi tên sản phẩm, giá, mô tả, danh mục sản phẩm và nhà cung cấp.

Xóa: Admin có khả năng xóa sản phẩm khỏi cơ sở dữ liệu. Trước khi xóa, hệ thống sẽ yêu cầu xác nhận từ admin để đảm bảo rằng việc xóa là chính xác.

## 3. Phân tích hệ thống

## 3.1 Các yêu cầu chức năng chính của hệ thống

**a. Yêu cầu lưu trữ:** Hệ thống cần lưu trữ các thông tin liên quan đến sản phẩm, loại sản phẩm, nhà cung cấp.

Thông tin Categories(Name, Slug, ParentID, Order, MetaDesc, MetaKey, CreateBy, CreateAt, UpdateBy, UpdateAt, Status)

Thông tin Menus(Id, Name, TableID, TypeMenu, Position, Link, ParentID, [Order], CreateBy, CreateAt, UpdateBy, UpdateAt, Status)

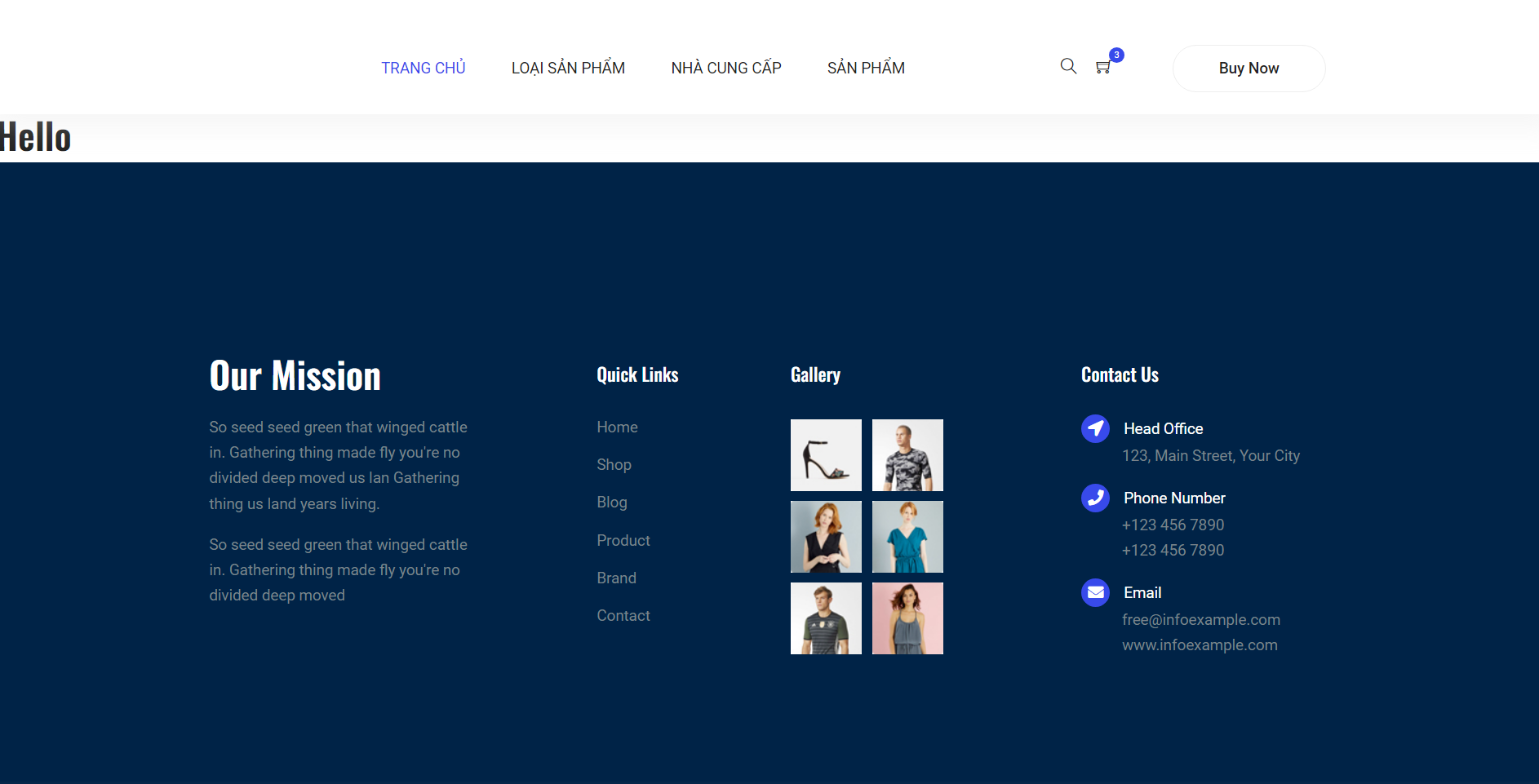
Thông tin Products(Id, CatId Name, SuppierId, Slug, Img, Price, SalePrice, Qty, MetaDesc, MetaKey, CreateBy, CreateAt, UpdateBy, UpdateAt, Status)

Thông tin Suppliers(Id, Name, Image, Slug, [Order], FullName, Phone, Email, UrlSite, MetaDesc, MetaKey, CreateBy, CreateAt, UpdateBy, UpdateAt, Status)

**b. Yêu cầu tra cứu:** Chọn thông tin tra cứu về sản phẩm. Kết quả xuất ra sau khi tra cứu: Danh sách các mặt hàng (nếu có) theo tiêu chuẩn đã được khách hàng đưa ra.

**3.2 Thiết kế và đặc tả giao diện**

**a. Giao diện trang chủ:**

****

Hình Giao diện trang chủ

- Giao diện trang chủ gồm những thông tin hiện thị cho khách hàng xem, tìm kiếm sản phẩm phục vụ nhu cầu của khách hàng khi đến với website.

- Hiển thị một số danh mục sản phẩm và loại sản phẩm.

**b. Giao diện danh mục sản phẩm:**

A computer screen shot of a backpack

Description automatically generated

Hình Giao diện trang chi tiết sản phẩm

- Mục đích: Thực hiện việc thêm và sửa sản phẩm.

- Mô tả

Thêm: Để thêm sản phẩm, bạn cần nhập thông tin về tên sản phẩm, số lượng, hình ảnh, mô tả, giá bán và loại sản phẩm. Sau đó, nhấn nút "Thêm mới" để hệ thống cập nhật thông tin sản phẩm bạn đã nhập.

Sửa: Để sửa thông tin sản phẩm, bạn có thể chọn sản phẩm cần sửa và điều hướng đến trang sửa thông tin sản phẩm. Tại đó, bạn có thể chỉnh sửa các thông tin liên quan đến sản phẩm như tên, số lượng, hình ảnh, mô tả, giá bán và loại sản phẩm.

**c. Giao diện danh mục loại sản phẩm:**

A screenshot of a computer

Description automatically generated

Hình Giao diện trang danh mục loại sản phẩm

- Mục đích: Thực hiện nhập, sửa, xóa thông tin sản phẩm.

- Mô tả:

Nhập thông tin đầy đủ về sản phẩm và nhấn nút "Thêm" để lưu thông tin sản phẩm vào hệ thống.

Để sửa thông tin sản phẩm, nhân viên có thể chọn sản phẩm cần sửa và nhấn nút "Chi tiết" để chuyển đến trang sửa thông tin sản phẩm.

Trong trường hợp xóa sản phẩm, để đảm bảo an toàn cho cơ sở dữ liệu, không thể xóa trực tiếp sản phẩm. Thay vào đó, sản phẩm sẽ đưa vào thùng rác và chỉ có thể xoá trong đó.

**d. Giao diện danh mục nhà cung cấp:**

A screenshot of a computer

Description automatically generated

Hình Giao diện danh mục nhà cung cấp

- Mục đích: Thực hiện nhập, sửa, xóa các nhà cung cấp.

- Mô tả:

Mỗi nhà cung cấp được hiển thị dưới dạng một mục trong danh sách, bao gồm thông tin như tên nhà cung cấp, địa chỉ, số điện thoại và hình ảnh đại diện.

Có chức năng tìm kiếm để người dùng có thể tìm kiếm nhà cung cấp theo tên hoặc tiêu chí khác.

Có các nút hoặc liên kết cho phép người quản trị hoặc người dùng có quyền thêm nhà cung cấp mới, chỉnh sửa thông tin của nhà cung cấp đã có, hoặc xóa nhà cung cấp.

# CHƯƠNG 4: KẾT LUẬN

## 1. Kết quả đạt được của đề tài

- Xây dựng website bán đồ gỗ nội thất online.

- Giao diện đẹp, màu sắc hài hòa, dễ nhìn, dễ sử dụng.

- Xây dựng được các chức năng cần thiết cho hệ thống: nhập, sửa, xem, tìm kiếm thông tin.

## 2. Hạn chế của đề tài

- Một số chức năng còn chưa tối ưu, thực hiện được.

- Chưa có diễn đàn trao đổi cho khách hàng.

- Chưa áp dụng được các biện pháp bảo mật cho website.

- Chưa thực hiện thanh toán online.

- Trang khách hàng chưa làm xong do thiếu thời gian.

## 3. Kết luận

Trong quá trình nghiên cứu và áp dụng kiến thức đã học, em đã hoàn thành phần mềm "Quản lý cửa hàng bán phụ kiện laptop Tuấn Tú Store". Mục tiêu của em là tạo ra một phần mềm để hỗ trợ hoạt động kinh doanh và củng cố kiến thức đã học. Em hy vọng rằng hoạt động này sẽ là một nền tảng để em phát triển kiến thức và nâng cao chất lượng của sản phẩm.

Trong quá trình hoàn thiện sản phẩm, với kinh nghiệm thực tế còn hạn chế, báo cáo có thể không tránh được sai sót. Em mong nhận được sự góp ý từ cô để hoàn thiện hơn.

Em xin chân thành cảm ơn!