CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

* 1. Giới thiệu

Trên thế giới hiện nay, tin học là một ngành phát triển không ngừng. Thời kỳ công nghiệp đòi hỏi thông tin nhanh chóng chính xác. Có thể nói tin học đã chiếm một vị trí quan trọng trong cuộc sống hàng ngày, trong các ngành khoa học kỹ thuật. Chúng đang được phát triển và áp dụng rộng rãi đặc biệt là những lĩnh vực kinh doanh, buôn bán.

Việc đưa tin học vào công việc quản lý, nhằm giảm bớt sức lao động của con  
người, tiết kiệm thời gian, độ chính xác cao và tiện lợi rất nhiều so việc quản lý giấy tờ bằng thủ công như trước đây. Tin học hóa nhằm thu hẹp không gian lưu trữ, tránh được thất lạc dữ liệu, tự động hóa và cụ thể hóa các thông tin theo yêu cầu của con người.

Hiện nay, nhu cầu buôn bán, trao đổi hàng hóa ngày càng tăng, hình thức mua và bán ngày càng được cải tiến. Ở trên một thị trường mà người mua và người bán không cần trực tiếp gặp nhau nhưng vẫn có thể thực hiện các giao dịch thì đòi hỏi cần phải có một môi trường ảo hỗ trợ những công việc đó và những website bán hang chính là môi trường không thể thuận lợi hơn để thực hiện những giao dịch đó, khiến chúng trở nên đơn giản, nhanh chóng và thuận tiện. Chính vì vậy mà em đã nghiên cứu, tìm hiểu về một công nghệ xây dựng website phổ biến nhất hiện nay là ASP.NET Core MVC. Với sự hướng dẫn rất nhiệt tình của thầy giáo **Mai Cường Thọ** em đã hoàn thành được đề tài nghiên cứu của mình. Trong quá làm bài chắc chắn  
em sẽ mắc phải những thiếu sót nên rất mong nhận được sự đóng góp ý kiến của thầy, cô để em có cơ hội sửa sai, phát triển trong thực tế và từng bước hoàn thiện mình.

* 1. Lý do chọn đề tài

**Về phần lý thuyết**: ASP.NET Core là gì? Nó là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (clound) hoặc chạy on-promise.

Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

Đồng thời nó đã trở thành một mã nguồn mở. Đây là một thay đổi rất lớn và theo mình là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.

**Về phần thực hành**: nhu cầu buôn bán, trao đổi hàng hóa ngày càng tăng, hình thức mua và bán ngày càng được cải tiến. Ở trên một thị trường mà người mua và người bán không cần trực tiếp gặp nhau nhưng vẫn có thể thực hiện các giao dịch thì đòi hỏi cần phải có một môi trường ảo hỗ trợ những công việc đó và những website bán hàng chính là môi trường không thể thuận lợi hơn để thực hiện những giao dịch đó, khiến chúng trở nên đơn giản, nhanh chóng và thuận tiện.

* 1. Mục tiêu đề tài
* Tìm hiểu lý thuyết về ASP.NET Core MVC
* Xác định cấu trúc, các thành phần chính trong ASP.NET Core MVC
* Xây dựng một website bán hải sản
* Quản lý sản phẩm, cập nhật, thêm, xóa, sửa sản phẩm
* Tùy chỉnh giao diện website
* Xem, sửa, quản lý các hóa đơn mà khách hàng đã đặt
  1. Phạm vi đề tài

Đề tài thiết kế website bán hải sản được giới hạn ở việc khách hàng lựa chọn sản phẩm, thêm vào giỏ và tiến hành điền thông tin khách hàng để gửi hóa đơn cho bên administration. Về phía admin sẽ tiếp nhận đơn hàng, thực hiện giao hàng đồng thời cập nhật, chỉnh sửa giao diện website bán hàng của mình.

* 1. Đối tượng nghiên cứu
* ASP.NET Core MVC
* Website bán hải sản
* HTML, Bootstrap, Javascript, jQuery, Razor Page
* MVC Model
* C#, MS SQL, Entity Framework

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

* 1. Giới thiệu về ASP.NET Core MVC

ASP.NET Core là gì? Nó là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (clound) hoặc chạy on-promise.

Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

Đồng thời nó đã trở thành một mã nguồn mở. Đây là một thay đổi rất lớn và theo mình là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.

* Tính năng của ASP.NET Core MVC bao gồm:
  + Routing
  + Model binding
  + Model validation
  + Dependency injection
  + Filters
  + Areas
  + Web APIs
  + Testability
  + Razor view engine
  + Strongly typed views
  + Tag Helpers
  + View Components
* Các đặc điểm nổi bật của .NET core
  + Open-source Framework  
    .NET Core là nền tảng nguồn mở được duy trì bởi Microsoft và có sẵn trên GitHub theo giấy phép MIT và Apache 2. Nó là một dự án .NET Foundation.
  + Cross-platform  
    Đa nền tảng. .NET Core chạy trên hệ điều hành Windows, macOS và Linux. Có thời gian chạy khác nhau cho mỗi hệ điều hành thực thi mã và tạo ra cùng một output.
  + Nhất quán giữa các kiến trúc  
    Bạn có thể thực thi mã giống nhau trong các kiến trúc tập lệnh khác nhau (bao gồm x64, x68 và RAM)
  + Hỗ trợ nhiều ngôn ngữ  
    Bạn có thể sử dụng ngôn ngữ lập trình C#, F# và Visual Basic để phát triển ứng dụng .NET Core.
  + Triển khai linh hoạt  
    Các ứng dụng .NET Core có thể được triển khai song song (cài đặt toàn bộ người dùng hoặc toàn hệ thống). No cũng có thể triển khai với các Container Docker.
  + Khả năng tương thích  
    .NET Core tương thích với .NET Framework và Mono API (thông qua .NET Standard).
  + Hỗ trợ bởi Microsoft  
    Vì .NET Core được phát triển bởi Microsoft nên tài liệu cũng như các cập nhật, hỗ trợ được update thường xuyên. Từ đó, giúp người sử dụng có thể giải quyết vấn đề nhanh chóng.
  + Công cụ CLI  
    .NET core bao gồm các công cụ CLI (Command – line –interface: Giao tiếp thông qua dòng lệnh) để phát triển và tích hợp liên tục.
  + Kiến trúc mô-đun  
    .NET core hỗ trợ cách tiếp cận kiến trúc mô-đun bằng cách sử dụng các gói NuGet. Có các gói NuGet khác nhau cho các tính năng khác nhau có thể được thêm vào dự án .NET core nếu cần. Nhờ đó bạn sẽ ít tiêu tốn dung lượng bộ nhớ, tăng hiệu suất và dễ bảo trì ứng dụng hơn.
  1. Một số khái niệm, kiến thức khi nghiên cứu
     1. Routing

Routing sẽ xác định URL và điều khiển thông tin tương ứng với URL mà người dùng web nhập vào. Tất cả các cấu hình Routing của một ứng dụng ASP.NET MVC được lưu trữ trong RouteTable, nó đóng vai trò định tuyến các URL để xác định các lớp xử lý tương ứng khi có request được gửi đến từ người dùng web.

* + 1. Model binding

Trong ASP.NET Core MVC Model binding chuyển đổi dữ liệu yêu cầu từ phía client (form values, route data, query string parameters, HTTP headers) vào bên trong đối tượng để controller có thể xử lí. Kết quả là Controller của bạn không phải thực hiện công việc tìm ra dữ liệu từ đâu đến. Nó chỉ đơn giản là kiểm tra dữ liệu và tham số từ phương thức action.

* + 1. Model validation

ASP.NET Core MVC  hỗ trợ ràng buộc dữ liệu cho các thuộc tính trong model của bạn, các thuộc tính sẽ được kiểm tra ở client xem có hợp lệ không trước khi giá trị của thuộc tính đó được gửi về server. Cũng như trên server trước khi action của controller gọi.

* + 1. Dependency injection

Trong ASP.NET Core MVC controller có thể gửi yêu cầu cần thiết đến các service thông cấu trúc xây dựng của chúng.

* + 1. Filters

Filters giúp các lập trình viên đóng gói “cross-cutting concerns”. giống như là xử lý ngoại lệ và phân quyền. Filter được kích hoạt để chạy trước và sau các action của controller. Ví dụ về phân quyền [Authorize] được đặt trước action.

* + 1. Areas

Khi một website ASP.NET MVC trở nên quá lớn quá và phức tạp,số Controller chắc chắn sẽ tăng lên, với nhiều controller như vậy bạn sẽ thấy chúng có thể thuộc về một nhóm như phần Administrator, phần sản phẩm, phần nhóm sản phẩm… Areas cho phép bạn chia các Controllers,models và views tới các vị trí khác nhau trong solution với cùng một thư mục độc lập.

* + 1. Web APIs

Ngoài việc là một nền tảng tuyệt vời để xây dựng các trang web. ASP.NET Core MVC hỗ trợ rất nhiều cho việc xây dựng API Web. Bạn có thể xây dựng các dịch vụ tiếp cận nhiều khách hàng bao gồm trình duyệt web và thiết bị di động.

* + 1. Testability

The framework’s use of interfaces and dependency injection make it well-suited to unit testing

Việc xây dựng các framework’s sử dụng các interface và DI làm cho việc kiểm tra ứng dụng trở nên dễ dàng hơn và framework’s  bao gồm các tính năng (like a TestHost and InMemory provider for Entity Framework) chúng giúp ích cho lập trình viên rất nhiều.

* Razor view engine

ASP.NET Core MVC views sử dụng Razor view engine để render các code html đến view.Razor được sử dụng để tự động tạo nội dung web trên máy chủ.

* + 1. Tag Helpers

Tag Helpers cho phép mã phía máy chủ tham gia vào việc tạo và hiển thị các phần tử HTML trong các tệp razor.

Đây là một số khái niệm cơ bản trong ASP.NET Core MVC cho người mới học cần nắm rõ để có thể hiểu hết bản chất khi xây dựng ứng dụng.

* 1. JavaScript

Ngôn ngữ xây dựng nên React, yêu cầu mức độ trung bình: các khái niệm  
“object”, “prototype”, “callback” là bắt buộc, nếu bạn chưa biết gì, tôi nghĩ bạn nên  
đọc tài liệu tại đây và thực hành trong ít nhất 3 tuần trước khi quay lại để bắt đầu.

* 1. MVC Model

**MVC** là viết tắt của cụm từ “**Model-View-Controller**“. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. MVC là một **mẫu kiến trúc phần mềm** để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau như tên gọi: Model (dữ liệu), View (giao diện) và Controller (bộ điều khiển).

Đơn giản hơn, là mô hình này được chia thành 3 phần trong soure code. Và mỗi phần đảm nhận vai trò và nhiệm vụ riêng biệt nhau và độc lập.

**Mô hình MVC** (MVC pattern) thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

* 1. C#

C# (đọc là “C thăng” hay “C sharp” (“xi-sáp”)) là một ngôn ngữ lập trình **thuần hướng đối tượng** được phát triển bởi **Microsoft**.

C# ra đời năm 2000, được thiết kế chủ yếu bởi **Anders Hejlsberg** – kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, . . .

Được xây dựng dựa trên nền tảng của 2 ngôn ngữ lập trình mạnh nhất đó là [C++](https://howkteam.vn/redirect?Id=CDPvobsj8EgqSWOjQulkj8sNbtaZ%2f%2f%2bLu45dab%2bOgs79tBhcEL%2fXvarFqvp65mXDM5cj85Vgg25v91Y%2fii2Y3w%3d%3d) và [Java](https://howkteam.vn/redirect?Id=CDPvobsj8EgqSWOjQulkj8sNbtaZ%2f%2f%2bLu45dab%2bOgs79tBhcEL%2fXvarFqvp65mXDnCCwQnbi1wNnCX0zbZT1vw%3d%3d). Do đó C# được miêu tả là ngôn ngữ có sự cân bằng giữa C++, Visual Basic, Delphi và Java.

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng [Windows Forms](https://howkteam.vn/redirect?Id=CDPvobsj8EgqSWOjQulkjzRg1VzwfQLAW%2bO2Q5F6JjA2px1gbbm4NFOthD7wOJGuRtTpqeZp5YO5qEgrbfmgfg%3d%3d) hay [WPF (Windows Presentation Foundation)](https://howkteam.vn/redirect?Id=CDPvobsj8EgqSWOjQulkjzRg1VzwfQLAW%2bO2Q5F6JjCKai8dVle55AytfNzJYJPfgqeoRcFgXpijqlIYBjE1bg%3d%3d), . . . trở nên rất dễ dàng.

* 1. MS SQL và Entity Framework
     1. MS SQL
  + Phần mềm được Microsoft phát triển dựa trên RDBMS.
  + Cũng là một ORDBMS (Hệ quản trị cơ sở dữ liệu quan hệ đối tượng).
  + Một nền tảng độc lập.
  + Phần mềm sử dụng cả giao diện dòng lệnh và giao diện GUI.
  + Hỗ trợ ngôn ngữ [SQL](https://quantrimang.com/sql) (trước đây là SEQUEL - ngôn ngữ truy vấn tiếng Anh có cấu trúc) - vốn là sản phẩm của IBM.
    1. Entity Framework

Entity Framework ra đời nhằm hỗ trợ sự tương tác giữa các ứng dụng trên nền**tảng .NET** với các cơ sở dữ liệu quan hệ. Hay, Entity Frmework chính là công cụ giúp ánh xạ giữa các đối tượng trong ứng dựng, phần mềm của bạn với các bảng của một cơ sở dữ liệu quan hệ.

Emntity Framework giúp các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng với ít mã hơn so với các ứng dụng truyền thống. Lợi ích lớn nhất của nó là giúp lập trình viên giảm thiểu việc lập trình mã nguồn để thực hiện truy cập và tương tác với cơ sở dữ liệu.

Hiện nay, Entity framwork là 1 framework mạnh để phát triển ứng dụng Web với sự hỗ trợ đông đảo của cộng đồng.

CHƯƠNG 3: PHÂN TÍCH WEBSITE BÁN HẢI SẢN

* 1. Xác định yêu cầu

**Khách hàng:**

* Chọn lựa sản phẩm
* Xem giá
* Tìm kiếm sản phẩm
* Mua hàng
* Cung cấp thông tin
* Thanh toán sản phẩm

**Chủ của hàng:**

* Tùy chỉnh website
* Thêm xóa sửa sản phẩm
* Cập nhật sản phẩm nổi bật, sản phẩm mới
* Thay đổi banner, slideshow hình ảnh trang web
* Đăng thông tin về cửa hàng, thông báo…
* Quản lý hóa đơn khách hàng
* Xem thống kê các hóa đơn đã lập
  1. Phân tích yêu cầu

**Chủ cửa hàng:**

Cung cấp sản phẩm: Cung cấp các sản phẩm hải sản theo chủng loại. Các mặt  
hàng được chia theo từng chủng loại khác nhau. Mỗi mặt hàng tương ứng với một chủng loại, nếu mặt hàng nằm ngoài những chủng loại đã có thì chủ cửa hàng có thể thêm chủng loại mới. Tạo ra sự đa dạng trong mặt hàng.

Chỉnh sửa giao diện: Chủ cửa hàng có thể tùy chỉnh giao diện trang web, thay đổi banner, slide show hình ảnh. Thêm sản phẩm nổi bật cũng như post những bài thông báo, hướng dẫn mua hàng.

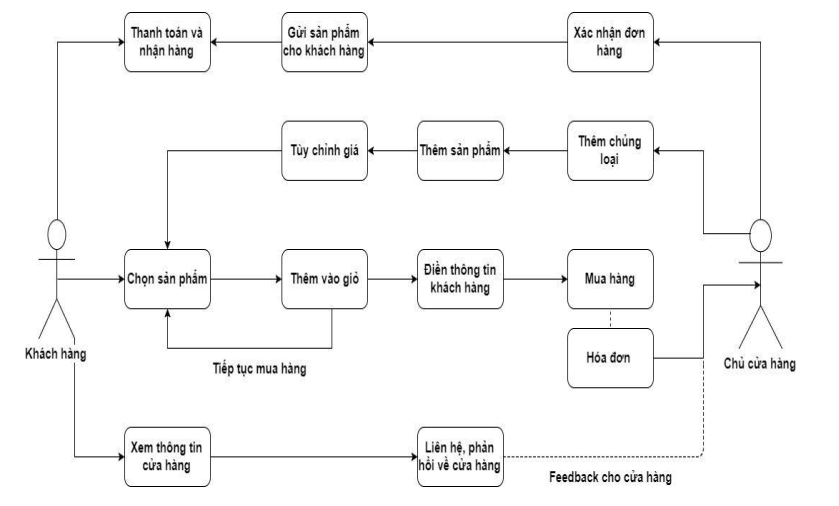
Quản lý hóa đơn: Xem những hóa đơn mà khách hàng đã gửi. Chỉnh sửa nếu có nhu cầu thay đổi từ khách hàng. Xóa nếu khách hàng hủy đơn hàng.

**Khách hàng:**

Tìm kiếm, lựa chọn sản phẩm từ cửa hàng cung cấp. Để tiến hành mua hàng,  
khách hàng cho sản phẩm vào giỏ, chỉnh số lượng mặt hàng. Cung cấp thông tin cho cửa hàng. Sau khi gửi đơn hàng, cửa hàng sẽ gọi điện xác nhận và giao tận nhà cho khách qua hình thức COD.

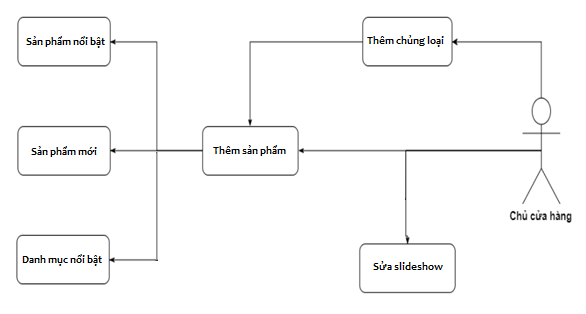
Ngoài ra khách hàng có thể xem thông tin về cửa hàng, liên hệ đóng góp ý kiến cũng như phản hồi về các sản phẩm đã mua.

* 1. Mô hình hóa yêu cầu
     1. Mô hình mua hàng



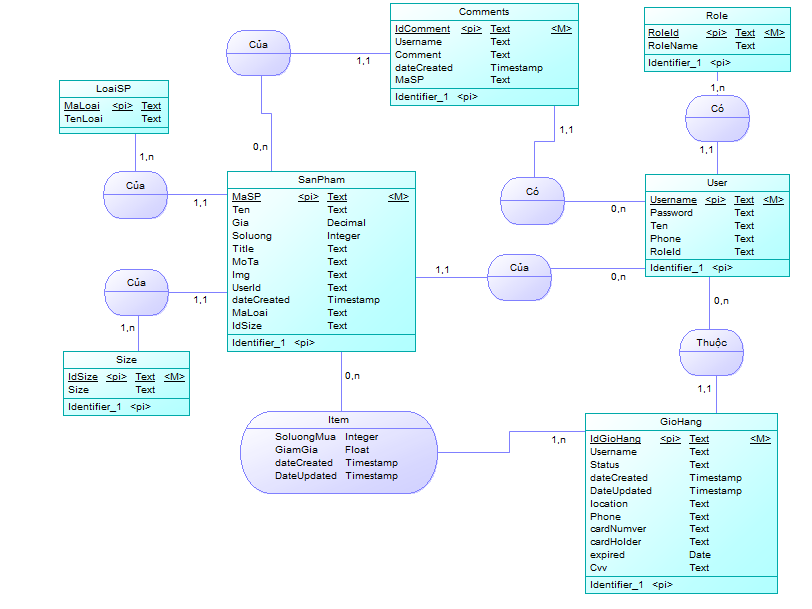
Hình 1Mô hình tổng quan việc mua hàng

* + 1. Mô hình tùy chỉnh website của chủ cửa hàng



Hình 2 Mô hình tùy chỉnh website

* 1. Phân tích cơ sở dữ liệu
     1. Mô hình quan niệm dữ liệu (Thực thể - Kết hợp)



Hình 3 Mô hình ERD

* + 1. Sơ đồ quan hệ

SANPHAM(**MaSP**, Ten, Gia, Soluong, Title, MoTa, Img, dateCreated, IdSize, MaLoai,Username)

LOAISP(**Maloai**, TenLoai)

SIZE(**IdSize**, Size)

GIOHANG(**IdGioHang**, dateCreated, dateUpdated, Status, loction, Phone, cardNumber, cardHolder, expired, Cvv, Username)

Item(**MaSP, IdGioHang**, SoLuongMua, GiamGia, dateCreated, dateUpdated)

Users(**Username**, Password, Ten, Phone, RoleId)

Comments(**IdComment**, Comment, dateCreated, Username, MaSP)

Roles(**RoleId**, RoleName)

* + 1. Kiểu dữ liệu của các thực thể

**SANPHAM**

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| MaSP | Varchar | Mã sản phẩm |
| Ten | Varchar | Tên sản phẩm |
| Gia | Decimal | Giá sản phẩm |
| Soluong | Interger | Số lượng sản phẩm còn |
| Title | Varchar | Tiêu đề sản phẩm |
| MoTa | Varchar | Mô tả sản phẩm |
| Img | Varchar | Đường dẫn hình ảnh sản phẩm |
| Username | Varchar | Người đăng |
| dateCreated | Date | Ngày tạo |
| MaLoai | Varchar | Loại sản phẩm |
| IdSize | Varchar | Size sản phẩm |

**LoaiSP**

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| MaLoai | Varchar | Mã loại sản phẩm |
| TenLoai | Varchar | Tên loại sản phẩm |

**Size**

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| IdSize | Varchar | Mã size sản phẩm |
| Size | Varchar | Tên size sản phẩm |

**GioHang**

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| IdGioHang | Varchar | Mã giỏ hàng |
| Username | Varchar | Tên tài khoản |
| CartItemId | Varchar | Mã item |
| Status | Varchar | Tình trạng đơn hàng |
| DateCreated | Date | Ngày tạo |
| DateUpdated | Date | Ngày sửa |
| Location | Varchar | Địa chỉ người mua |
| Phone | Varchar | SĐT người mua |
| cardNumber | Varchar | Số thẻ |
| cardHolder | Varchar | Chủ thẻ |
| Cvv | Varchar | Cvv |
| expired | date | Ngày hết hạn |

**Item**

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| IdGioHang | Varchar | Mã giỏ hàng |
| MaSp | Varchar | Mã sản phẩm |
| SoluongMua | Int | Số lượng mua |
| GiamGia | Decimal | Giảm giá |
| DateCreated | Date | Ngày tạo |
| DateUpdated | Date | Ngày cập nhật |

**Users**

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| Username | Varchar | Tên tài khoản |
| Password | Varchar | Mật khẩu |
| Ten | Varchar | Tên người dùng |
| Phone | Varchar | Số điện thoại |
| RoleId | Varchar | Vai trò |

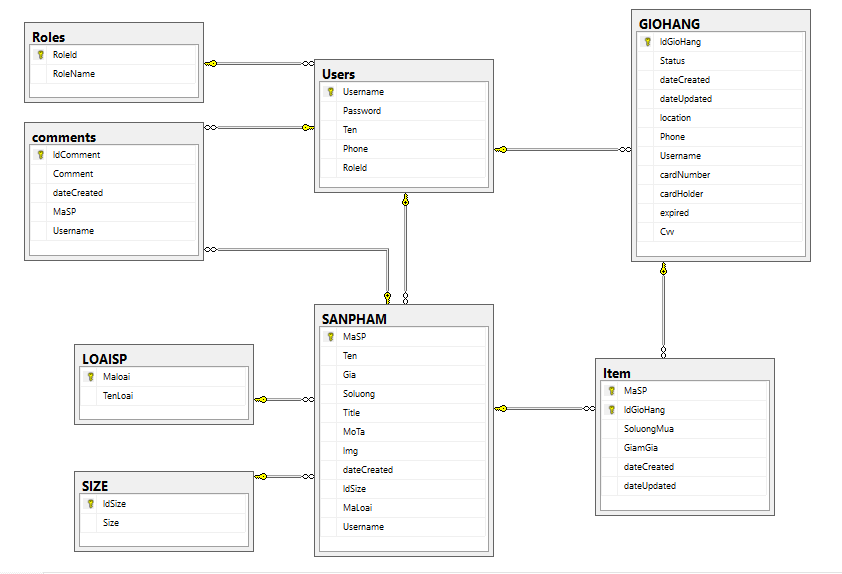
**Roles**

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| Id | Varchar | Mã role |
| RoleName | Varchar | Tên role |

**Comments**

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Mô tả |
| IdComment | Varchar | Mã comment |
| Username | Varchar | Mã tài khoản |
| Comment | Varchar | Nội dung bình luận |
| dateCreated | Date | Ngày tạo |
| MaSP | Varchar | Mã sản phẩm |

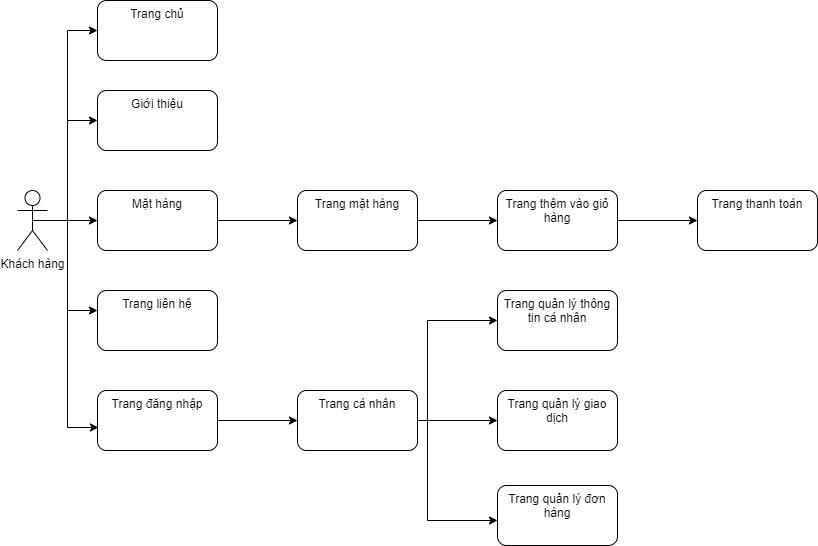
* + 1. Cơ sở dữ liệu



Hình 4 Cơ sở dư liệu quan hệ

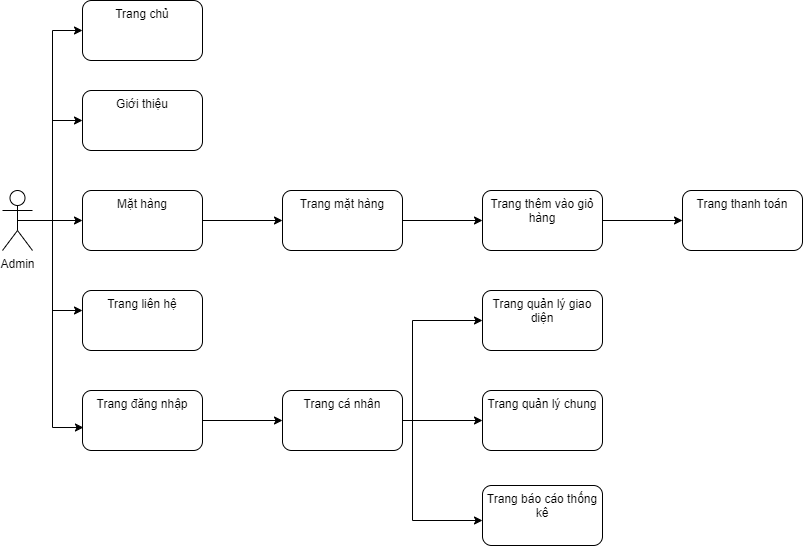
CHƯƠNG 4: THIẾT KẾ WEBSITE BÁN ĐỒ CHƠI

* 1. Sơ đồ giao diện
     1. Sơ đồ giao diện phía client



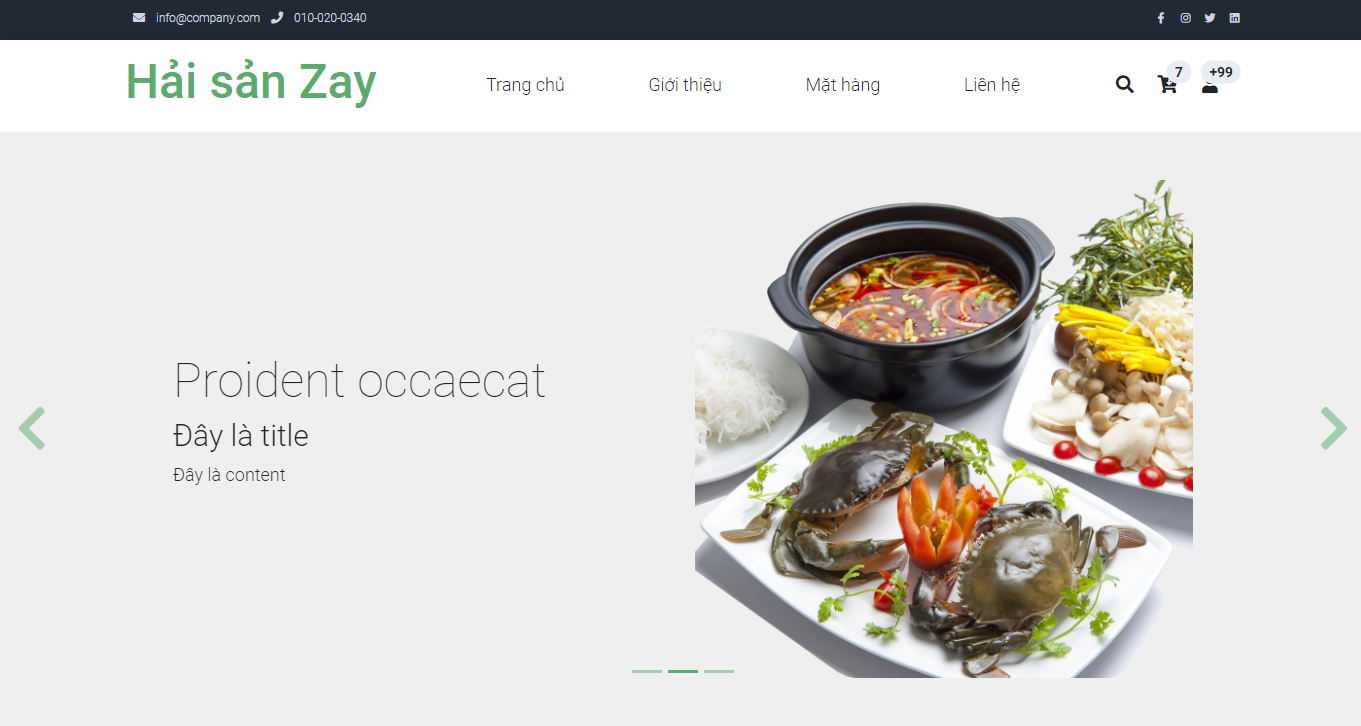
Hình 5 - Sơ đồ phía client

* + 1. Sơ đồ giao diện phía admin



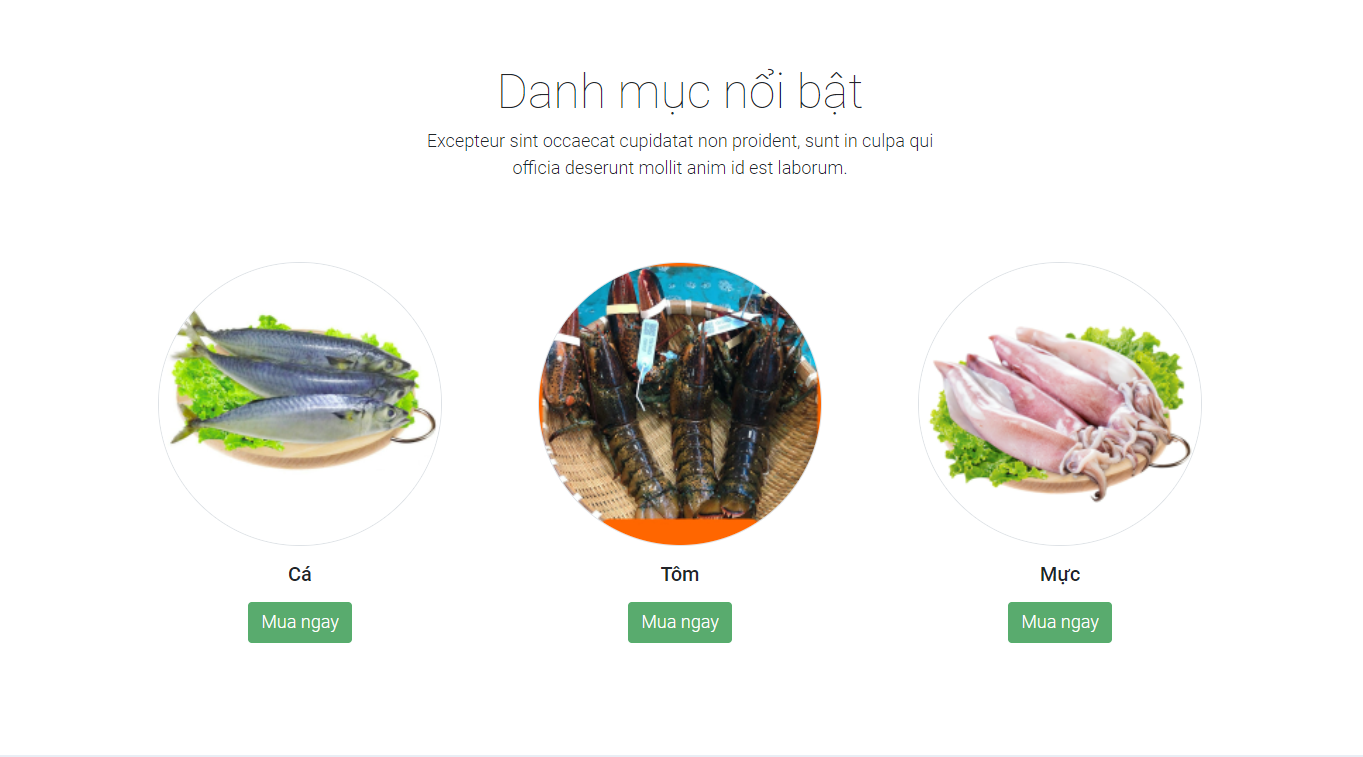
Hình 6 - Sơ đồ admin

* 1. Thiết kế giao diện
     1. Giao diện phía người dùng
        1. Trang chủ
           1. Phần header



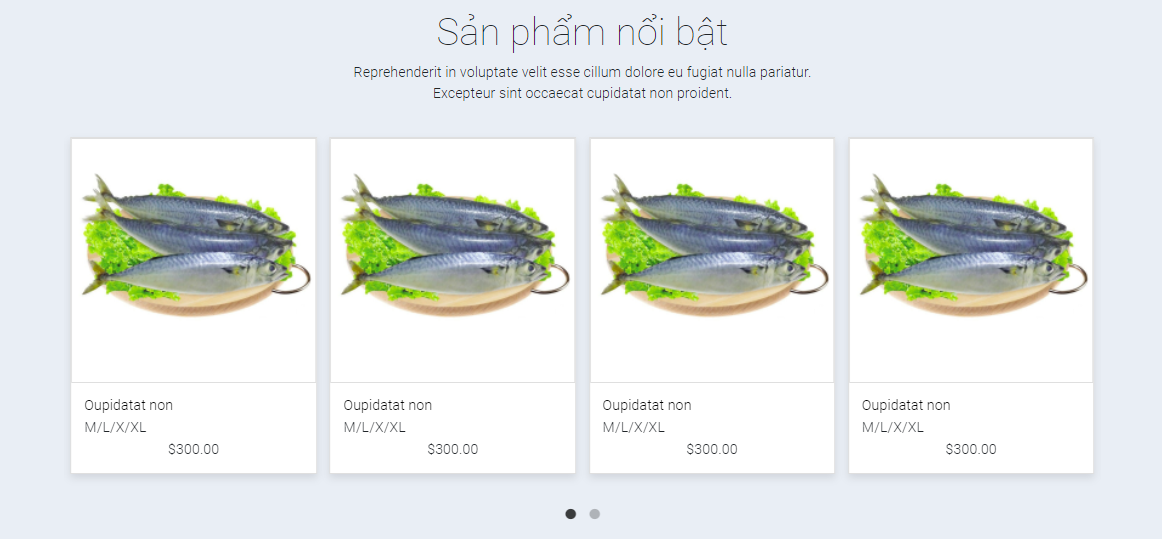
Hình 7 - Header

* + - * 1. Phần danh mục nổi bật



Hình 8 - Danh mục nổi bật

* + - * 1. Phần sản phẩm nổi bật



Hình 9 - Sản phẩm nổi bật

* + - * 1. Phần footer



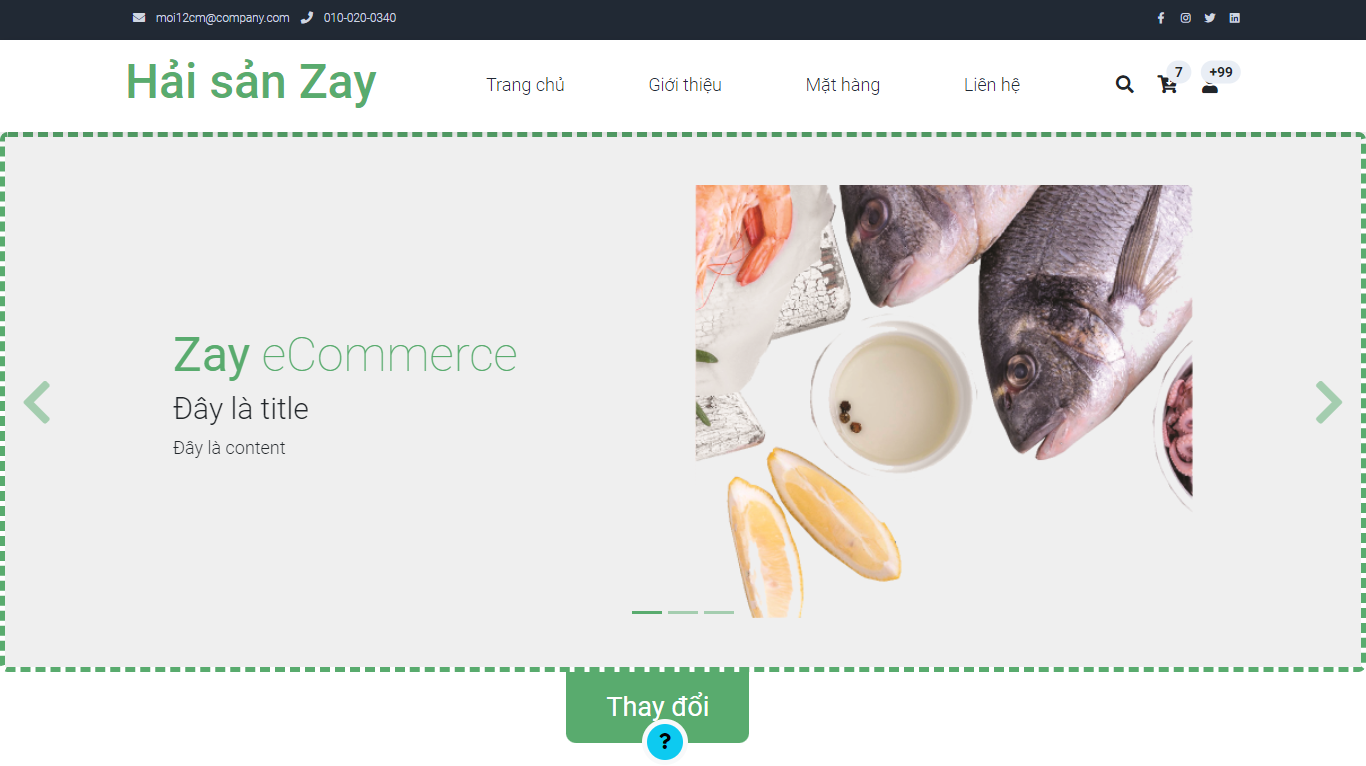
Hình 10 - Footer

* + - 1. Trang giới thiệu
      2. Trang mặt hàng
      3. Trang liên hệ
      4. Trang đăng nhập/đăng ký
      5. Trang cá nhân
    1. Giao diện admin
       1. Các trang chung

Các trang khách hàng có thì administration đều có

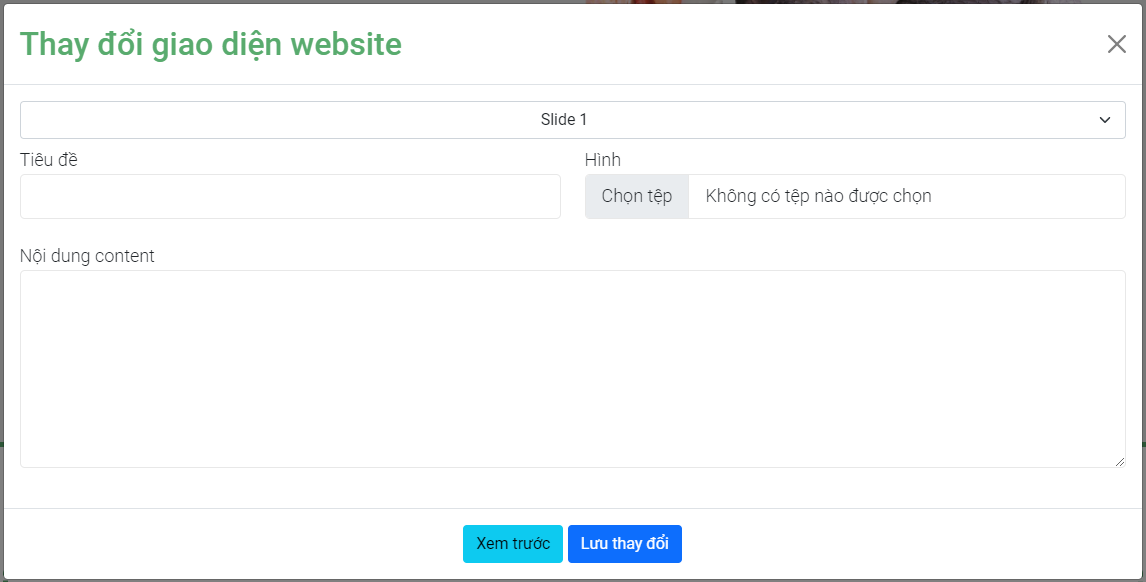
* + - 1. Trang quản lý giao diện

1. Header



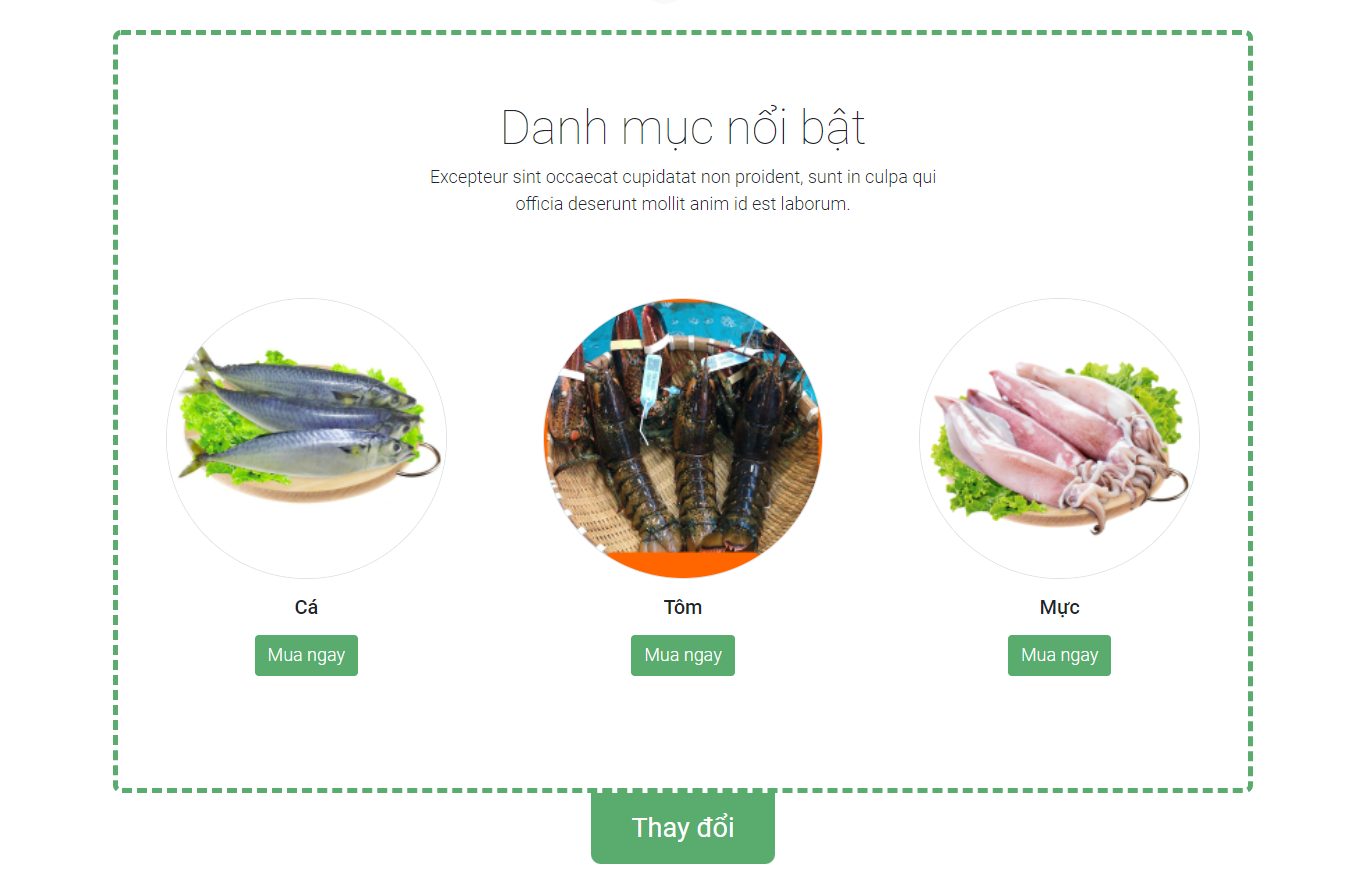
Hình 11 - Thay đổi giao diện slide

1. Thay đổi slide



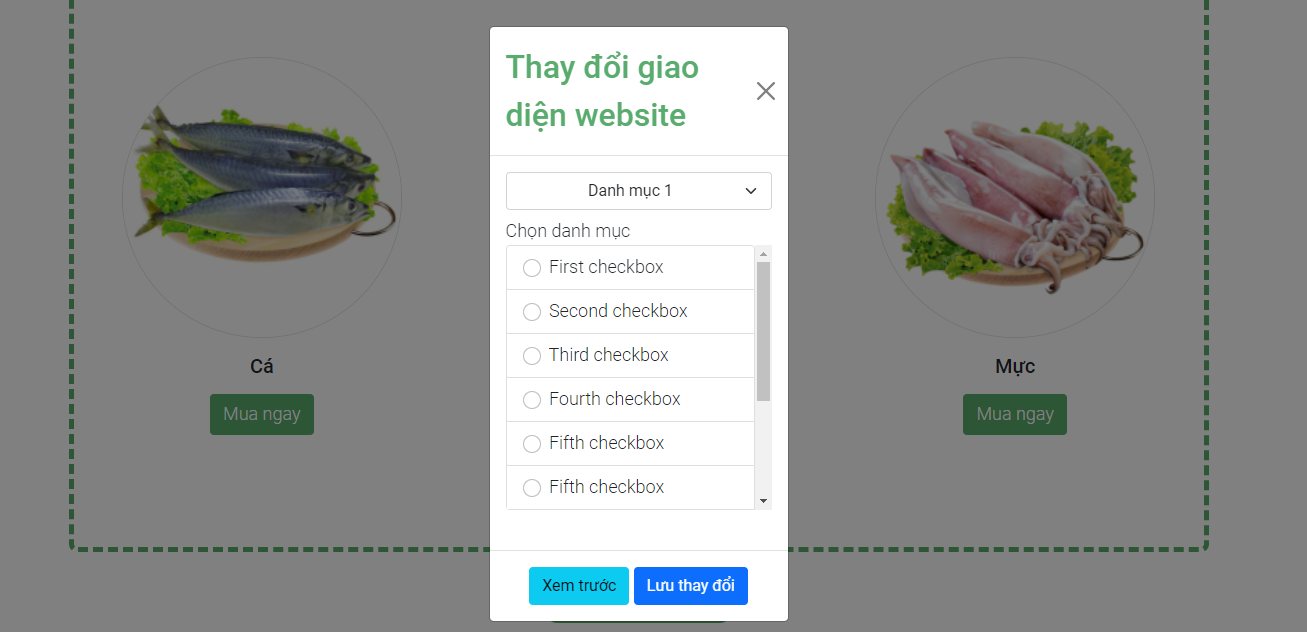
Hình 12 - Form thay đổi giao diện slide

1. Giao diện danh mục nổi bật



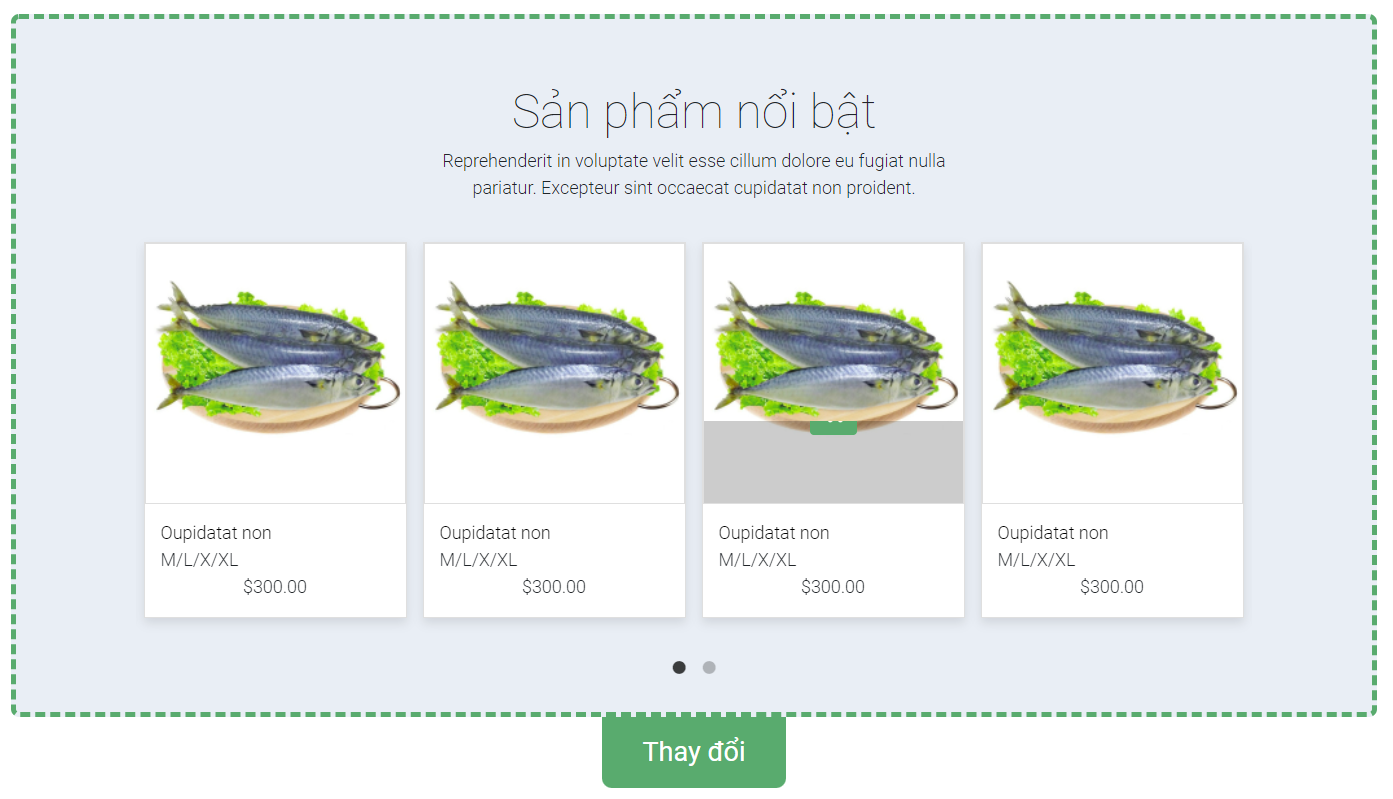
Hình 13 - Thay đổi danh mục nổi bật

1. Giao diện thay đổi danh mục nổi bật



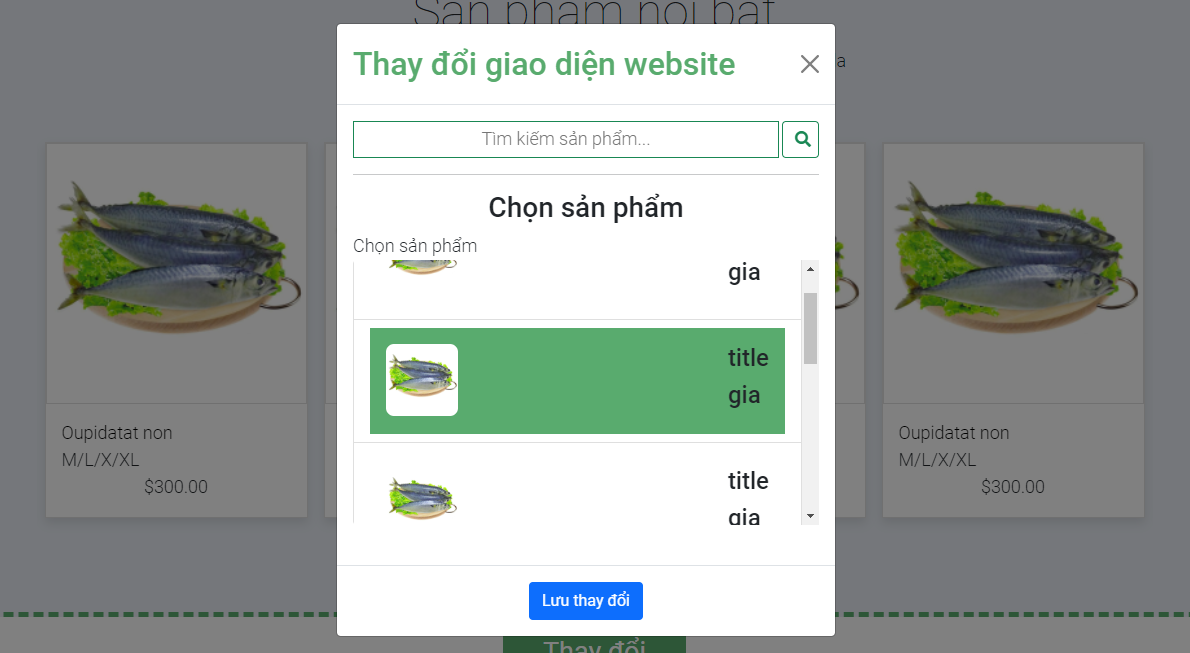
Hình 14 - Form thay đổi danh mục nổi bật

1. Giao diện sản phẩm nổi bật



Hình 15 - Thay đổi sản phẩm nổi bật

1. Giao diện thay đổi sản phẩm nổi bật



Hình 16 - Form thay đổi sản phẩm nổi bật